
GAZİ ÜNİVERSİTESİ
Gazi
1926

UÇTAN UCA TÜRKÇE KONUŞMA TANIMA İÇİN ÇIKTI DÜZELTME METODU ÖNERİSİ VE TEKRARLAYAN SİNİR AĞI TASARIMI

Recep Sinan ARSLAN

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

OCAK 2020

Recep Sinan ARSLAN tarafından hazırlanan "UÇTAN UCA TÜRKÇE KONUŞMA TANIMA İÇİN ÇIKTI DÜZELTME METODU ÖNERİSİ VE TEKRARLAYAN SİNİR AĞI TASARIMI" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Bilgisayar Mühendisliği Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Necaattin BARIŞÇI

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

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

Başkan: Prof. Dr. Sabri KOÇER

Bilgisayar Mühendisliği Ana Bilim Dalı, Necmettin Erbakan Üniversitesi

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

Üye: Doç. Dr. Nursal ARICI

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

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

Üye: Doç. Dr. Aydın ÇETİN

Bilgisayar Mühendisliği Ana Bilim Dalı, Gazi Üniversitesi

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

Üye: Doç. Dr. Halil Murat ÜNVER

Bilgisayar Mühendisliği Ana Bilim Dalı, Kırıkkale Üniversitesi

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

Tez Savunma Tarihi: 22/01/2020

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

……

Prof. Dr. Sena YAŞYERLİ

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.

Recep Sinan ARSLAN
22/01/2020

UÇTAN UCA TÜRKÇE KONUŞMA TANIMA İÇİN ÇIKTI DÜZELTME METODU ÖNERİSİ VE TEKRARLAYAN SİNİR AĞI TASARIMI

(Doktora Tezi)

Recep Sinan ARSLAN

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Ocak 2020

ÖZET

Otomatik konuşma tanıma (OKT), konuşma sinyallerinin girdi olarak alınması ve bilgisayarlar tarafından işlenebilmesi için metne dönüştürülmesi işlemidir. OKT uygulamaları çok yönlü ve gerçek hayatta yaygın olarak kullanılmasına rağmen, gürültülü ortamlarda, kelime dağarcığı büyümesi veya konuşma sinyalinin kalitesiz olması durumlarında yazımsal hatalar üretme eğilimindedirler. Bu çalışmada, OKT sistemlerinin üretmiş olduğu çıktılardaki hataların tespit edilmesi ve düzeltilmesi için alternatif hipotez önerisi yaklaşımına dayalı özgün bir model önerilmiştir. Hatalı kelimeleri belirleme, düzeltilebilir olanları seçme ve bu kelimelerin düzeltileceği aday sözcüklerin belirlenmesi gibi bir dizi işlem adımı içermektedir. Aday sözcüklerin belirlenmesinde "Levensthein" algoritması ve bu çalışma için hazırlanmış olan "Türkçe şablon kelimeler veritabanı" kullanılmaktadır. Önerilen modelin etkinliği, verimliliği ve sunduğu katkı düzeyi Uzun Kısa Süreli Bellek (UKSB) ve Geçitli Tekrarlayan Birim (GTB) bellek yapısının kullanıldığı uçtan uca Türkçe OKT sistemi ile test edilmiştir. Yapılan testler sonucunda, konuşma tanıma sisteminin performansı %4,60 oranında artış göstermiştir. 100 ve 500 kelime içeren sözcük dağarcığı ile yapılan testlerde sırasıyla %99,2 ve %80,3 oranında doğru tanıma performansı yakalanmıştır.

Bilim Kodu	:	92427
Anahtar Kelimeler	:	Konuşma Tanıma Sistemi, Uçtan-Uca, Çıktı Düzeltme, Türkçe, Bağlantıcı Zamansal Sınıflandırma, Tekrarlayan Sinir Ağı, Uzun Kısa Süreli Bellek
Sayfa Adedi	:	102
Danışman	:	Doç. Dr. Necaattin BARIŞÇI

DEVELOPMENT OF OUTPUT CORRECTION METHODOLOGY FOR TURKISH SPEECH RECOGNITION AND DESIGN OF A RECURRENT NEURAL NETWORK

(Ph. D. Thesis)

Recep Sinan ARSLAN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

January 2020

ABSTRACT

Automatic speech recognition (ASR) is the process of receiving speech signals as input and converting them into text for processing by computers. Although ASR applications are versatile and widely used in real life, they tend to produce spelling errors in noisy environments, increase of vocabulary size, or poor speech signals. In this study, an original model based on alternative hypothesis suggestion approach is proposed to detect and correct erroneous outputs produced by ASR systems. The method involves a series of processing steps, such as identifying the erroneous words, selecting the ones that can be corrected, and selecting candidate words to correction. “Levenshtein” algorithm and Turkish template words database prepared for this study are used in determining candidate words. The effectiveness, efficiency and contribution level of this proposed model has been tested with an end-to-end Turkish ASR system using Long short term memory and Gated recurrent unit memory structure. As a result of the tests, the performance of the speech recognition system has increased by 4,60%. In tests performed with vocabulary containing 100 and 500 words, 99,2% and 80,3% correct recognition performance were obtained, respectively.

Science Code	:	92427
Key Words	:	Speech Recognition, End-to-End, Output Correction, Turkish, Connectionist Temporal Classification, Recurrent Neural Network, Long Short Term Memory
Page Number	:	102
Supervisor	:	Assoc. Prof. Dr. Necaattin BARIŞÇI

TEŞEKKÜR

Tez çalışmam süresince engin bilgi birikiminin yanında akademik tecrübelerini benimle paylaşarak hoşgörü ve sabırla destek olan doktora tez danışmanım “Sayın Doç. Dr. Necaattin BARIŞÇI” hocama, tez izleme komitemde olmayı kabul eden, destekleyen ve tez çalışmama değerli katkılarını sunan “Sayın Prof. Dr. Sabri KOÇER” ve “Doç. Dr. Nursal ARICI” hocalarıma, tez çalışmam süresince, sunucularını kullanmış olduğum “Türk Ulusal Bilim e-Altyapısı” hizmetini sunan TUBİTAK ULAKBİM Yüksek Başarımlı ve Grid Hesaplama Merkezine, bu süreçte hep yanımda olan, beni destekleyen, büyük sevgi ve aşkını benden hiçbir zaman esirgemeyen hayat arkadaşım, zorlu günlerimin dostu eşim Gamze’ye, hayata gözlerimi açtığım ilk andan itibaren sevgi ve desteklerini esirgemeyen sevgili aileme sonsuz teşekkür, sevgi ve saygılarımı sunarım.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

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

Simgeler	**Açıklamalar**
gb	Gigabyte
khz	Kilohertz
ms	Milisaniye
sn	Saniye

Kısaltmalar	**Açıklamalar**
BZS	Bağlantıcı zamansal sınıflandırma
DÖK	Doğrusal öngörülü kodlama
GKM	Gaussian karışım modeli
GPU	Graphics processing unit
GTB	Geçitli tekrarlayan birim
HFD	Hızlı fourier dönüşümü
HTK	Hidden markov model toolkit
LCS	Longest common subsequence
LDC	Linguistic data consortium
MFKK	Mel frekansı kepstral katsayıları
SMM	Saklı markov model
TBA	Temel bileşen analizi
TSA	Tekrarlayan sinir ağı
UKSB	Uzun kısa süreli bellek
YSA	Yapay sinir ağı

1. GİRİŞ

İnsanların, bilgisayarlar ile iletişim kurabilmeleri için fare, dokunmatik ekranlar, klavye, mikrofon gibi birçok girdi aracı bulunmaktadır. Bu araçlar kullanılarak bilgisayara görevler gönderilir ve cevapları beklenir. Örnek olarak, bir araştırma için internet kullanılmak isteniyor ise fare veya klavye yardımıyla internet tarayıcısı açılır. Klavye ile araştırmak istenen bilginin girişi yapılır ve ihtiyaç duyulan bilgiye erişim sağlanır. Yaygın kullanılan bu aygıtlar kullanıcı ile bilgisayar arasındaki bilgi ve komut akışını kısıtlarlar. Bu nedenle, erişimin farklı şekillerde sağlanmasına yönelik olarak dokunmatik ekranlar, sesli komut ile telefon yönetimi gibi farklı çalışmalar yapılmaktadır. Bilgisayarlar ile iletişimin daha hızlı kurulması ve ihtiyaç duyulan bilgiye en kısa yoldan ulaşılması ile ilgili talepler gün geçtikçe artmaktadır. Böylece daha kısa zamanda daha fazla veriye erişim ile hem zamandan tasarruf sağlanması hem de hayatın kolaylaşması mümkün olacaktır [1].

Konuşma insanların en etkin ve doğal iletişim yöntemidir. Diğer aktivitelerin aksine çok fazla konsantrasyon gerektirmeden yapılabilmektedir [2]. Bu nedenle insanların mikrofon vasıtası ile bilgisayarlar ile sesli olarak iletişim kurabilmelerine yönelik talepler artış eğilimi göstermektedir. Böylece bilginin yazarak aktarılması gibi zor ve zaman alıcı bir yöntemden vazgeçerek, insanların doğal alışkanlıklarına tam uyan bir girdi aracı kullanılmış olacaktır. Bu sebeple, teknoloji firmaları ses ile bilgisayarların yönetilebilmesi sürecini devrim olarak nitelendirmektedirler. Bilgisayarların sesli kontrolü 1950'li yıllardan itibaren gelişen ve üzerinde çalışmalar yürütülen bir alan olmuştur.

İşaret işleme alanı içerisinde bir disiplin olan konuşma tanıma, gelişen teknoloji sürecinde kendine yer edinmeye çalışan önemli bir problemdir. İnsan sesinin mikrofon vasıtasıyla bilgisayarlar tarafından algılanarak tanınması işlemidir [3]. Bu işlem tüm insanların hayatında farkında olmadan defalarca yaptığı bir davranıştır. İnsanlar bunun için öncelikle kulağı vasıtasıyla konuşmayı tespit eder. Daha sonra sinyal seviyesini yükseltir ve beyne gönderir. Beyinde daha önceden öğrenilmiş dil ve kavram ilişkisi ile anlamlı bir hale getirir. Bilgisayarların benzer şekilde öğrenme sürecini tamamlayabilmesi için insanlara benzer bir model kullanması gereklidir. İnsan ile bilgisayar arasındaki en önemli fark ses tanıma sürecinde dil ve kavram ilişkisi yerine istatistiksel ve olasılıksal hesaplar kullanılmasıdır [4].

Günümüzde, otomatik konuşma tanıma (OKT) sistemleri derin sinir ağlarının ve yüksek kapasiteli bilgi işlem sistemlerinin kullanımı ile oldukça iyi sonuçlar üretmektedir. Bu sebeple, OKT sistemleri çağrı merkezleri, sesli dikte, insan-makine ara yüzü gerektiren işlemler, ticari faaliyetler (seyahat acentesi, hisse senedi piyasası yönetimi vb.), hava durumu raporları gibi birçok alanda kullanılmaya başlanılmıştır [5]. Sesli dikte işlemi OKT sistemlerinin kullanıldığı alanlardan birisidir. Teknolojinin gelişmesi ile birlikte, görüntü, video veya ses gibi multimedya verilerinin insanların günlük yaşamlarına etkisi gün geçtikçe artmaktadır. Bu etkilerden ilki sosyal medya mecrasında görülmektedir. Multimedya kullanımının artması ile birlikte bu verilerin incelenmesi ve analiz edilmesine ihtiyaç duyulmaktadır. Konuşma tanıma alanındaki yeni çalışmalar ve hata oranlarındaki düşüşler bu sistemlerin kullanıldığı alanlarda iyi sonuçların elde edilmesini ve daha yaygın kullanılmasını sağlamaktadır. Ancak, dünyada ve özellikle de Türkiye'de ticari olarak bulunan konuşma tanıma programlarının performansları, insanların doğal tanıma seviyeleri ile karşılaştırıldığında, hala oldukça düşük seviyededir. [6]. Bu durum OKT sistemleri ile ilgili daha fazla çalışılmasını ve daha karmaşık modeller üretilmesini gerektirmektedir. Böylece daha başarılı sonuçlar alınacak ve daha geniş bir alanda kullanılacaktır.

Sinir ağlarının 1990'lı yıllarda konuşma tanıma uygulamalarında kullanımı artmıştır. Ancak bu yıllarda otomatik konuşma tanıma sistemleri için daha yaygın olarak Saklı Markov Model (SMM) yaklaşımı kullanılmaktaydı. Sinir ağlarının performans değeri de bu yaklaşımın oldukça gerisinde kalmaktaydı. Bunun en önemli sebebi sinir ağlarında bulunan sıfıra yakınsama, eğitim için yeterli verinin elde edilememesi, büyük işlem gücü gereksinimleri ve korelasyon yapısındaki zayıflıklar gibi problemlerdir. Bu sorunlar sebebiyle performans kaybı yaşanmaktaydı [6]. 2000'li yılların başlarında yalıtık sözcük tanıma yerine sürekli konuşma tanıma çalışmaları yapılmıştır. Geniş sözcük dağarcığı ile doğaçlama konuşmalar incelenmiştir. Bu çalışmalarda %80 seviyelerinde doğruluk yakalanmıştır. GPU mimarisinin henüz yaygınlaşmaması ve bilgi işlem gücü yüksek bilgisayarların hala oldukça yüksek fiyatlı olması sebebiyle karmaşık sinir ağlarının eğitilebilmesi mümkün olmamıştır. Bu sebeple performans değeri daha fazla artış göstermemiştir [7]. 2010 yıllarında bu kısıtların ortadan kalkması ile sinir ağları ile OKT sistemleri geliştirme fikri üzerine çalışmalar hızlanmıştır. Geleneksel sinir ağlarının daha fazla katman barındıran türü olan derin sinir ağları önerisi getirilmiştir. Karmaşık ve güçlü modellerin kullanıldığı yapı ile ses tanıma sistemlerinde çok daha başarılı sonuçlar elde edilebilmiştir. Karmaşık modellerin eğitilebilmesi ile başarılı sonuçların alınması daha hızlı ve güçlü grafik işlemcilere ve

belleklere olan ihtiyacı artırmıştır. Günümüzde, son kullanıcı bilgisayarı ile derin sinir ağlarının eğitilmesi birkaç gün veya hafta almaktadır [8]. Konuşma tanıma teknolojilerinin geliştirilmesinde derin sinir ağlarının kullanılması ve daha başarılı modeller üretilmesi sebebiyle OKT sistemleri insanlar tarafından daha yaygın olarak kullanılmaktadır.

Konuşma tanıma sistemlerinde gelişmiş yapıları kurabilmek için, akustik model üretimi, dil modelinin oluşturulması, gerekli hallerde hizalama işlemleri, metinlerin optimize edilmesi gibi birçok farklı işlem adımının birlikte hazırlanması gereklidir. Ses verisi ile metin çıktısının bağlantıcı zamansal sınıflandırma gibi algoritmalar ile hizalanamadığı veya sistem çıktısındaki hatalı karakterlerin düzeltilmesi gerektiği durumlarda doğrudan insan müdahalesine ihtiyaç duyulabilmektedir [8]. Akustik model üretimlerinin dil modelinden bağımsız olarak gerçekleştirildiği yapılarda, bu modellerin birbiri ile bağlantılı olabileceği hususlar göz ardı edilmektedir. Ses verisinden bağımsız olarak metin veri setleri hazırlanmakta ve sonuca etkisi deney ve gözlem ile denetlenmektedir. Birbirinden bağımsız olarak çalışılan bu modellerin birlikte eğitime dâhil edildiği ve tanıma performanslarında ciddi iyileşmelerin kaydedilebildiği yapılara uçtan uca konuşma tanıma sistemleri denilmektedir. Bu yapılarda hizalama için bağlantıcı zamansal sınıflandırma algoritması kullanılmaktadır [9]. Saklı Markov model, uzman sistemler, sinir ağları gibi algoritmalar ile konuşma tanıma yapıldığında dil modeli için N-gram yapısı yaygın olarak tercih edilmektedir. Bu modelin dile bağımlı olarak tanıma sisteminden bağımsız olarak oluşturulması gerekmektedir. Ancak uçtan uca OKT sistemlerinde doğrudan konuşmadaki fonemler arasındaki ilişkiler gözetilerek eğitim yapılmakta ve akustik veriden metin çıktıları elde edilmektedir. Böylece model karmaşası ortadan kaldırılmakta ve daha performanslı sistemlerin geliştirilmesi mümkün olmaktadır.

Bu tez çalışmasında uçtan uça Türkçe konuşma tanıma sistemi geliştirilmiştir. Derin sinir ağı modelinde Uzun kısa süreli bellek (UKSB) ve geçitli tekrarlayan bellek yapısı kullanılmıştır. Akustik set ile metin arasında hizalama Bağlantıcı zamansal sınıflandırma algoritması ile otomatik olarak yapılmıştır. Oluşturulan model farklı deney setleri ile test edilmiştir. Yapılan testlerde kelime dağarcığı rastgele seçilmiş 100 kelime olarak belirlendiğinde, %99,2 tanıma performansı yakalanmıştır. Hata bulma ve düzeltme algoritma önerisinin kullanıldığı sistemlerde %4,60 performans artışı sağlanmıştır.

Tez çalışmasının literatür katkısı

Uçtan uca konuşma tanıma yapısında modellenmiş, akustik model ve dil modeli yapısının birlikte öğrenilebildiği ve Türkçe için çalışabilen bir yapı oluşturulmuştur. Bu yapı üzerinde gerekli inceleme ve çalışmalar yapılarak, genel sistem performansına katkı sunabilecek özgün bir hata bulma ve düzeltme yöntemi önerilmiştir. Alternatif kelime önerisi yaklaşımı kullanılmaktadır. Önerilen bu yaklaşım farklı veri setleri ile test edilmiş ve %4,60 oranında performans artışı yakalanmıştır. Bu çalışma ile Türkçe için geliştirilmiş olan OKT sistem çıktılarında hataların bulunması ve düzeltilmesi amaçlanmıştır. Böylece genel tanıma performansı artırılmıştır.

Modelin eğitimi ve testlerinin gerçekleştirilmesi için hem ses hem de metin dosyaları oluşturulmuştur. Akustik eğitim için Metu 1.0 veri seti kullanılmıştır. Metin veri seti birden fazla metin kütüphanesinin homojen bir şekilde bir araya getirilmesi ile oluşturulmuştur. Uçtan uca model eğitiminde kullanılabilecek ses ve metin veri setleri bu tez kapsamında oluşturulmuştur ve önerilen yeni modelin testlerinde kullanılmıştır. Metin veri setinde yaklaşık olarak 1.6 milyon benzersiz Türkçe kelime barınmakta olup, kullanılan akustik set içerisindeki tüm kelimeleri içermektedir.

Bu tez çalışmasının ikinci bölümünde, Türkçe dil yapısı ve bu alanda yapılan konuşma tanıma uygulamaları verilmiştir. Konuşma tanıma sistemlerinin teorik altyapısı, sınıflandırması, ihtiyaç duyulan farklı model yapıları ve kullanım alanları üçüncü bölümde anlatılmıştır. Dördüncü bölümde, bu tez çalışmasında da kullanılan derin sinir ağlarının yapısı, derin sinir ağlarını kullanarak ses tanıma uygulamaları geliştirmek için ihtiyaç duyulan algoritmalar ve altyapılara ilişkin detaylı bilgi verilmiştir. Önerilen özgün modelin yapısı, modelin test edilmesi için gerekli araçlar, veri setleri ve test ortamına ilişkin detaylar beşinci bölümde anlatılmıştır. Bu tez çalışmasında önerilen model için yapılan testler, OKT sisteminin tanıma performansı sonuçları ve hata düzeltme algoritmasının katkı düzeyine ilişkin sonuçlar ile modelin kısıtları ve gelecekte yapılabilecek çalışmalara ilişkin tüm detaylar altıncı bölümde bahsedilmiştir. Son olarak yedinci bölümde, bu çalışmada önerilen model ve sonuçları genel olarak değerlendirilmiştir.

2. TÜRKÇE DİL YAPISI VE LİTERATÜR ÖZETİ

Türkiye Türkçesi, Ural-Altay dil ailesine bağlı olarak Türk dillerinden ve Oğuz grubuna mensup bir dildir. Türkiye, Kıbrıs, Irak, Balkanlar, Orta Asya ve Orta Avrupa ülkeleri başta olmak üzere geniş bir coğrafyada konuşulmaktadır. Sondan eklemeli bir dildir. Alfabesinde 29 harf bulunmaktadır [10].

Günümüzde ses, haberleşme ve işaret işleme teknolojilerinin gelişmesi, ipek yolunun dili olarak kabul edilen ve yaklaşık 200 milyon insanın konuştuğu Türk dilleri için gürbüz ve güvenilir konuşma tanıma motorlarının geliştirilmesini zorunlu kılmaktadır. Bu sistemlerin Türkçe dilinin kelime haznesini, akustik parametrelerini, iletişim yapılarını ve dil modellerini dikkate alması beklenmektedir [11].

Türkçe özgür kelime dizilime imkân veren bir dildir. Kurucu unsurlarından kök ve türetici ekler ile birlikte sondan eklemeli bir dil yapısına sahip olması, cümle içerisindeki kelimelerin serbest hareket edebilmesine imkân tanımaktadır. Bu durumda Türkçe dili için üretilebilecek sözcük formlarının sayısı oldukça fazla olmaktadır [12]. Bu şekilde çok sayıda kelime türetilebilir olması daha geniş dağarcık ile çalışan konuşma tanıma sistemlerinin geliştirilmesini zorunlu kılmaktadır. Ayrıca, dil modeli oluşturulmasında tahmin edilmesi gereken parametrelerin sayısını artırmaktadır. Bu sebeple diğer dillerin aksine daha karmaşık modellerin oluşturulması gerekmektedir.

Türkçe sondan eklemeli diller kategorisinde yer almaktadır. İngilizce, Almanca gibi dillerden farklı dil yapısına sahiptir. Bu sebeple, daha yaygın kullanım alanına sahip ve daha iyi performansa sahip otomatik konuşma tanıma sistemlerinin geliştirildiği bu diller için önerilen modeller Türkçe için doğrudan uygulanabilir olmamaktadır. Her bir dil için farklı konuşma tanıma sistemleri geliştirilmeli ve farklı veri setleri ile eğitim ve testler gerçekleştirilmelidir. Geniş sözcük dağarcığına sahip bir Türkçe konuşma tanıma sistemi için bu çalışmada uçtan uca bir model önerilmiştir. Bu modelin dile özgü kaynakları kullanması ve başarılı performans seviyesine sahip olmasına dikkat edilmiştir.

Son 20 yıl içerisinde Türkçe dili ile ilgili yapılmış bildiri, makale, yüksek lisans ve doktora tezlerine ilişkin sayısal sonuçlar Çizelge 2.1'de verilmiştir. Çizelgeye göre sayılar

incelendiğinde yıllara göre ortalama çalışma sayısında son yıllarda düşüş olduğu gözlemlenmiştir. Bunun sebebinin günümüzde artık daha başarılı sonuçlar alınabilen konuşma tanıma sistemlerinin geliştirilmiş olmasından kaynaklandığı düşünülmektedir.

Çizelge 2.1. Türkçe dili ile ilgili çalışmaların tür ve yıllara göre dağılımı

Yayın Türü	2000-2005	2005-2010	2010-2016	2017-2020
Bildiri	10	19	17	9
Makale	6	10	6	4
Tez	25(2000 öncesi), 15(Yüksek Lisans), 3(Doktora)	19(Yüksek Lisans), 4(Doktora)	26(Yüksek Lisans),2(Doktora)	6(Yüksek Lisans), 2(Doktora)

Çalışmasın yıllara göre dağılımında günümüze daha yakın olan 2017-2020 yılları arasında, güncel yaklaşımların benimsendiği 9 bildiri, 4 makale ve 8 yüksek lisans ve doktora tezinin [1, 13-19] yayınlandığı tespit edilmiştir.

Yüksel Arslan tarafından 2018 yılında yapılan doktora tezinde [19] silah sesi, çığlık, araç kazası seslerinin tespiti ve tanınmasının yapılmasına yönelik bir sistem geliştirilmiştir. Çevresel ses tanıma türünde bir çalışmadır. Verilmiş olan bu üç ses için farklı modeller tasarlanmıştır. Silah sesinin tespitinde %99,9, çığlık sesinin ve araç kazası sesinin tespitinde %98,4 performans ile sınıflandırma yapılmıştır.

Behnam Asefisaray tarafından 2018 yılında yapılan doktora tezinde [1] derin öğrenme kullanılan uçtan uca Türkçe konuşma tanıma sistemi geliştirilmiştir. Bağlantıcı zamansal sınıflandırma kullanılarak hizalama yapılmıştır. Bu sistem kullanılarak metindeki noktalama işaretlerinin tahmin edilmesi ve düzeltilmesi amaçlanmıştır. Sonuçta virgül için %63,4, nokta için %68,9, soru işaretinde %82,5 ve noktalı virgül işaretinde %52,8 oranı ile doğru tahmin yapılmıştır.

Cihan Akın tarafından 2019 yılında yapılan yüksek lisans tezinde [13] işitsel ve görsel olarak ayırt edilmesi güç olan ikiz sesleri ayırt etme üzerine biyometrik tanıma temelli bir modelin geliştirilmesi amaçlanmıştır. İkiz seslerin eşleşmesi sağlandığında güvenlik kontrolünün sağlandığı düşünülmüştür. 78 kişinin ses ikizlerinin bulunduğu test seti ile yapılan çalışmalarda, %91 oranı ile doğru tespit yapılmıştır.

Yaseminhan Arpacı tarafından 2019 yılında yapılan yüksek lisans tezinde [14] çevresel ses tanımaya yönelik çeşitli öznitelik çıkarma ve sınıflandırma teknikleri kullanılan bir yaklaşım önerilmiştir. Amaç en başarılı öznitelik çıkarma ve sınıflandırma tekniğini bulmaktır. Bu amaçla farklı teknikler karşılaştırmalı olarak test edilmiş ve sonuçları değerlendirilmiştir. Sonuçta %95,2 ile en başarılı sonuçların MFKK ve karar destek sistemleri tekniğinin birlikte kullanıldığı yapı ile elde edilebildiği tespit edilmiştir.

Roaya Abdalrahman tarafından 2018 yılında yapılan yüksek lisans tezinde [15] biyometrik tanıma için konuşmacı tanımaya yönelik bir yazılım önerisi yapılmıştır. Metin bağımsız konuşmacı tanıma modeli tasarlanması sonrasında kullanıcıya ait şifre kelimenin tanınması ile bu modelin birleştirilmesi aşamasını içermektedir. Yapılan testler sonucunda %91,2 seviyesinde başarı yakalanmıştır.

Ferdi Koç tarafından 2019 yılında yapılan yüksek lisans tezinde [16] daha önceden kaydedilmiş komutların tanınmasında derin öğrenme yöntemleri kullanılmıştır. Ses verileri spektrogram ile görüntüye çevrilmiştir. Bu görüntüler kullanılarak komutlar sınıflandırılmıştır. Sonuçta %95 oranında sınıflandırma doğruluğuna ulaşılmıştır.

İlhan Sofuoğlu tarafından 2017 yılında yapılan yüksek lisans tezinde [17] yalıtık Türkçe ses komutları kullanılarak insansız hava araçlarını yönlendiren bir sistemin geliştirilmesi amaçlanmıştır. Komut temelli bir model tasarlanmış olup Google Speech API kullanılarak yazılım altyapısı geliştirilmiştir. Daha önce yapılmış benzer yaklaşımlara göre daha başarı sonuçlar alınmıştır.

Cansu Akyürek Anacur tarafından 2019 yılında yapılan yüksek lisans tezinde [18] engelli bireylerin hayata katılımını sağlamak üzere bir çalışma yapılmıştır. Buna göre, ambulans, itfaiye ve polis aracı gibi geçiş önceliği olan araçların siren ve uyarılarının tespit edilmesi ile işitme engelli bireylerin araç kullanılmalarında kolaylık sağlanması amaçlanmıştır. Farklı yöntemler karşılaştırmalı olarak denenmiş olup tanıma performansı %48,93 ile %80,85 aralığında elde edilmiştir.

Türkçe dili ile ilgili yapılmış olup farklı yaklaşımların kullanıldığı güncel 61 adet makale [20, 21, 22-81] gruplara ayrılarak detaylı olarak incelenmiş olup bu bölümde ortak özellikleri dikkate alınarak karşılaştırmalı olarak değerlendirilmiştir.

Konuşma moduna göre çalışmalar değerlendirildiğinde, akustik model eğitimi için farklı kelime dağarcıkları kullanıldığı anlaşılmıştır. [47, 49, 59, 62] numaralı çalışmalarda kullanılan kelime sayısı 20 ile sınırlı tutulmuş olup komut temelli problemlere çözüm üretilmiştir. Bunun yanında çok daha büyük kelime dağarcığı ile çalışan model önerileri de getirilmiştir [21, 53, 76]. Bu noktada kelime dağarcığı seçimi genelde probleme bağlı olmaktadır. Sesten metne dönüşüm yapan bir program için çok daha büyük bir veri seti ile çalışılması beklenirken, robot kontrolü gibi komut temelli sistemlerde daha küçük bir dağarcık ile çalışılması yeterli olmaktadır. Türkçe dilinde yapılan çalışmalarda farklı büyüklükte veri setleri kullanılmıştır.

Konuşma veri tabanların kullanımına bakıldığında, çalışmaların %90'lık kısmında Metu 1.0 akustik veri setini kullanıldığı tespit edilmiştir.

Akustik öznitelik çıkarımı için çalışmaların birçoğunda Mel Frekansı Kepstrum Katsayıları (MFKK) temelli bir özellik çıkarımı yapıldığı anlaşılmıştır. 13 farklı özellik içeren ve bunların delta değerleri ile birlikte 39 katsayının elde edilebildiği bir sistemdir. 39 adet katsayının problemin çözümünde yeterli olmadığı durumlarda Doğrusal Öngörülü Kodlama (DÖK) ve Temel Bileşen Analizi (TBA) yöntemleri MFKK yaklaşımı ile birlikte kullanılmıştır. Yapılacak çalışmalarda özniteliklerin belirlenmesi sonrasında ihtiyaç olması halinde birden fazla yöntem birlikte kullanılabilmektedir.

Tanıma birimine göre çalışmalar değerlendirildiğinde, Türkçenin sondan eklemeli dil olması ve dağarcık boyutunun çok büyük olması sebebiyle hiçbir çalışmada kelime tabanlı bir yaklaşımın kullanılmadığı görülmüştür. Bunun yerine kelime altı (morf, hece, kök+ek) dil modelleri ile çalışmalar yapılmıştır. Türkçe ile ilgili yapılan çalışmalarda morf tabanlı sistemler ile daha başarılı sonuçlar elde edilmiştir. Bunun en önemli sebebi, morf tabanlı sistemlerin kısıtlı bir kelime dağarcığı ile çalışmayı mümkün kılması nedeniyle tercih edildiği düşünülmektedir. Örneğin Türkçe otuz morf ile temsil edilebilmektedir. Böylece hem karmaşıklıktan hem işlem gücü gereksiniminden kaçınılmıştır. Ancak kelime altı dil modeli ile çalışma yapılan durumlarda performans düzeyi %70'in altında kalmaktadır. Bu sebeple, sonuçların N-gram dil modeli veya herhangi bir hata bulma ve düzeltme algoritması ile desteklenmesi gerekli olmaktadır.

Dil modelinin oluşturulması için kullanılan sözcük dağarcığına göre çalışmalar değerlendirildiğinde komut temelli sistemlerde 20 ve altında kelimenin bulunduğu tespit edilmiştir. Ancak geniş dağarcık otomatik Türkçe konuşma tanıma sistemi üzerine yapılan çalışmalarda 1 milyon ve üzerinde kelime ile çalışılmıştır [37]. Ayrıca, konu bağımlı çözümlerde kelime seçiminin konu bazlı olduğu, metin dikte işlemlerinde rastgele kelime seçimi yapıldığı anlaşılmıştır.

Türkçe konuşma tanıma yapılan çalışmalarda, sınıflandırıcı olarak Saklı Markov modeli tercih edilmiştir [21, 67, 69, 71, 72, 81]. Bu modellerin oluşturulmasında "Hidden Markov Toolkit (HTK)" kullanılmaktadır. Dil modeli oluşturmak için N-gram [46, 54] yapısından yararlanılmıştır. N-gram kelime yapısını oluşturabilmek için SRI Language Model Toolkit (SRILM) altyapısı kullanılmaktadır. Akustik modelleme yaparak eğitim yapmak için KALDI [42] yazılımının kullanıldığı tespit edilmiştir.

Akustik veri setleri incelendiğinde, ses örnekleme frekansı tercihinin 16 Khz olarak belirlendiği ve çözünürlüğünü 16 bit olarak seçildiği anlaşılmıştır. Ancak bazı çalışmalarda [27] 8 Khz örnekleme frekansı ve 8 bit çözünürlüğü değeri tercih edilmiştir. Farklı çözünürlük ve örnekleme frekansı seçiminin problem için sesin kalitesinin ne kadar önemli olduğuna göre belirlendiği anlaşılmıştır.

Konuşma tanıma için kullanılan modelleme yapılarına göre çalışmalar sınıflandırıldığında olasılıksal modeller ile çalışmaların daha yaygın olduğu tespit edilmiştir. Bu çalışmalarda kullanılmış olan algoritmalar incelendiğinde karar destek vektörü (KDV) yönteminin yaygın olarak kullanıldığı anlaşılmıştır [26, 28, 36, 38, 54, 56, 78]. Bu çalışmalar komut temelli konuşma tanıma sistemleri olup, küçük dağarcık ile modelleme yapılmıştır. KDV yönetimi dışında dinamik zaman bükmesi [23], yapay sinir ağı [32], Bayes algoritması [42, 56], bulanık mantık [2], genetik algoritma [79] gibi farklı yaklaşımlar ile sınıflandırma yapıldığı ve probleme bağlı olarak başarılı sonuçlar alınabildiği görülmüştür. Bu noktada farklı seçimlerin yapılmasının akustik ve dil setinin hacmi, problemin karmaşıklığı, programlama bilgisi gibi farklı sebeplere bağlı olduğu düşünülmektedir. Farklı sınıflandırma yaklaşımlarının yanında, bu tez çalışmasında da kullanılan derin sinir ağı yapısının daha çok güncel çalışmalarda [6, 76, 77] tercih edildiği ve oldukça başarılı sonuçların elde edildiği anlaşılmıştır.

Kimanuka ve Büyük tarafından 2018 yılında yapılan ve derin öğrenme ile Türkçe konuşma tanıma sistemi geliştirilmesinin amaçlandığı çalışmada [6], derin öğrenme ile Gauss Karışım Modeli (GKM)-SMM aynı ses ve metin veri tabanı kullanılarak karşılaştırılmıştır. Geçmişte ve günümüzde seslerin zamansal değişimlerini modellemek için SMM yaygın olarak kullanılmaktadır. Bu çalışmada hizalama işlemini yapmak için ileri beslemeli sinir ağları tercih edilmiştir. İngilizce, Almanca gibi dillerde daha başarılı sonuçlar üreten derin sinir ağı Türkçe dili için aynen uygulanmıştır. Bunun sonucunda Türkçe için kabul edilebilir seviyelerde gelişme kaydedilebilmiştir. Sonuçta sistem performansında %2,5 seviyesinde bir artış yakalanmıştır.

Arslan ve Barışçı tarafından gerçekleştirilen çalışmada [76] benzer şekilde derin sinir ağları ile Türkçe konuşma tanıma sistemlerinin performansının artırılmasına yönelik çalışmalar yapılmıştır. Bu çalışmada uçtan uca Türkçe konuşma tanıma sistemi tasarlanmıştır. Derin sinir ağı ile model eğitilmiştir. Hatalı çıktıların bulunması ve düzeltilmesine yönelik olarak özgün bir algoritma önerilmiştir. Bu model ile yapılan çıktı düzeltme işlemleri sayesinde sistem tanıma performansı %2,25 düzeyinde artmıştır. Arslan ve Barışçı tarafından gerçekleştirilen bir diğer çalışmada [77] derin öğrenme modeline uyumlaştırma tekniklerinin etkisi detaylı olarak incelenmiş ve karşılaştırmalı sonuçları gösterilmiştir. Karşılaştırmalı olarak verilen sonuçlara göre en başarılı sonuçların “Gradient Descent” tekniği ile alınabildiği gösterilmiştir. 11 farklı optimizasyon tekniğinin karşılaştırıldığı bu çalışmaya göre bu seçim oldukça önemli olmakta ve sistem tanıma performansını doğrudan etkilemektedir.

Bu bölümde farklı veri setleri ve yaklaşımlar kullanılarak Türkçe konuşma tanıma sistemi geliştirmeyi amaçlayan birçok çalışmadan bahsedilmiştir. Sonuç olarak, bu çalışmalarında temel amacı daha yüksek tanıma performansına sahip bir model geliştirmektir. Yapılan çalışmalar incelendiğinde, 3-20 arasında kelime dağarcığı ile komut temelli olarak yapılan çalışmalarda %100 sınıflandırma oranının yakalandığı görülmüştür [47, 57]. Kelime dağarcığının büyümesi veya kişi bağımsız modellerin tasarlandığı problemler için başarı seviyeleri %40 düzeyine inmektedir. Başarı düzeyinin veri seti seçimi, model tasarımı, kişi bağımlılığı ve probleminin karmaşıklığına göre değişkenlik gösterdiği anlaşılmıştır. Bu sebeple, OKT problemi için tanıma performansına doğrudan etki eden tüm farklı unsurların birlikte değerlendirilmesi gerektiği anlaşılmıştır.

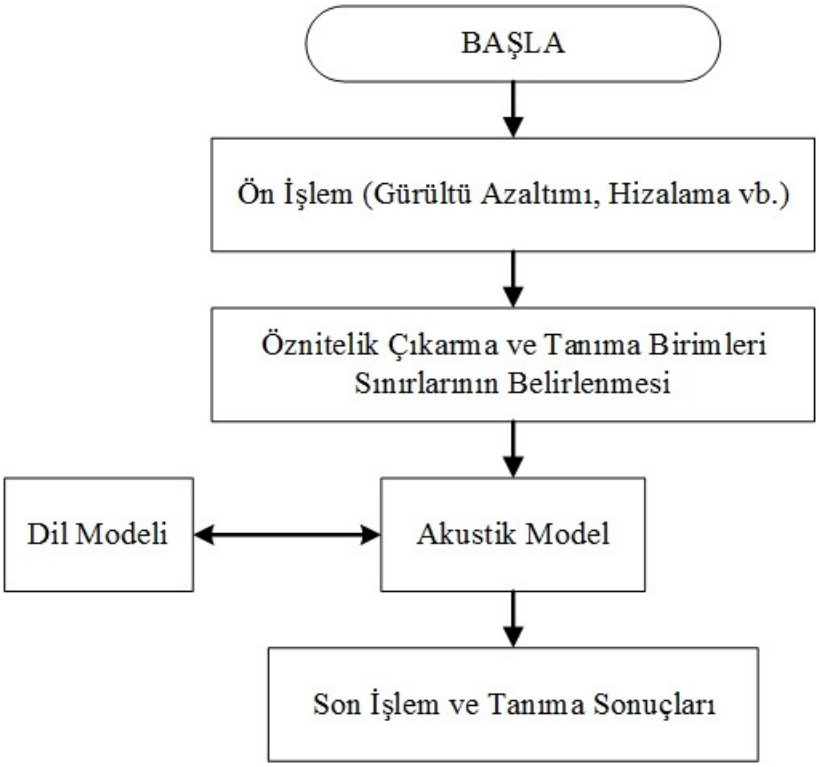
3. KONUŞMA TANIMA

Bu bölümde, konuşma tanıma sistemlerinin teorisi ve işlem süreçleri ile akustik ve dil model yapısında kullanılabilir yöntemler, modeller ve güncel yaklaşımlardan bahsedilecektir. Bu sistemlerin geliştirilmesinde kullanılabilir modeller hakkında kısa bilgi verilerek, bu tez çalışmasında kullanılan derin sinir ağı yapıları ayrı bir bölüm halinde anlatılacaktır. Geliştirilen yapıların performans ölçüm yöntemi, BZS tabanlı uçtan uca konuşma tanıma yapıları özetlenecektir. Böylece bu çalışmada önerilen modelin kullanıldığı temel yapının teorisi anlatılmış olacaktır.

3.1. Konuşma Tanıma Teorisi

Sinyal işleme araştırmalarında insanların iletişim kurma becerilerini taklit eden mekanik modeller yaratılması düşüncesi hâkimdir. Konuşma tanıma teknolojileri bilgisayarlara insanların sesli komutlarını takip etme veya konuşmalarını anlama imkânı tanır [80]. Konuşma tanıma, ses sinyalinin bir bilgisayar programı tarafından uygulanan algoritma vasıtasıyla kelime dizilerine dönüştürülmesi işlemidir.

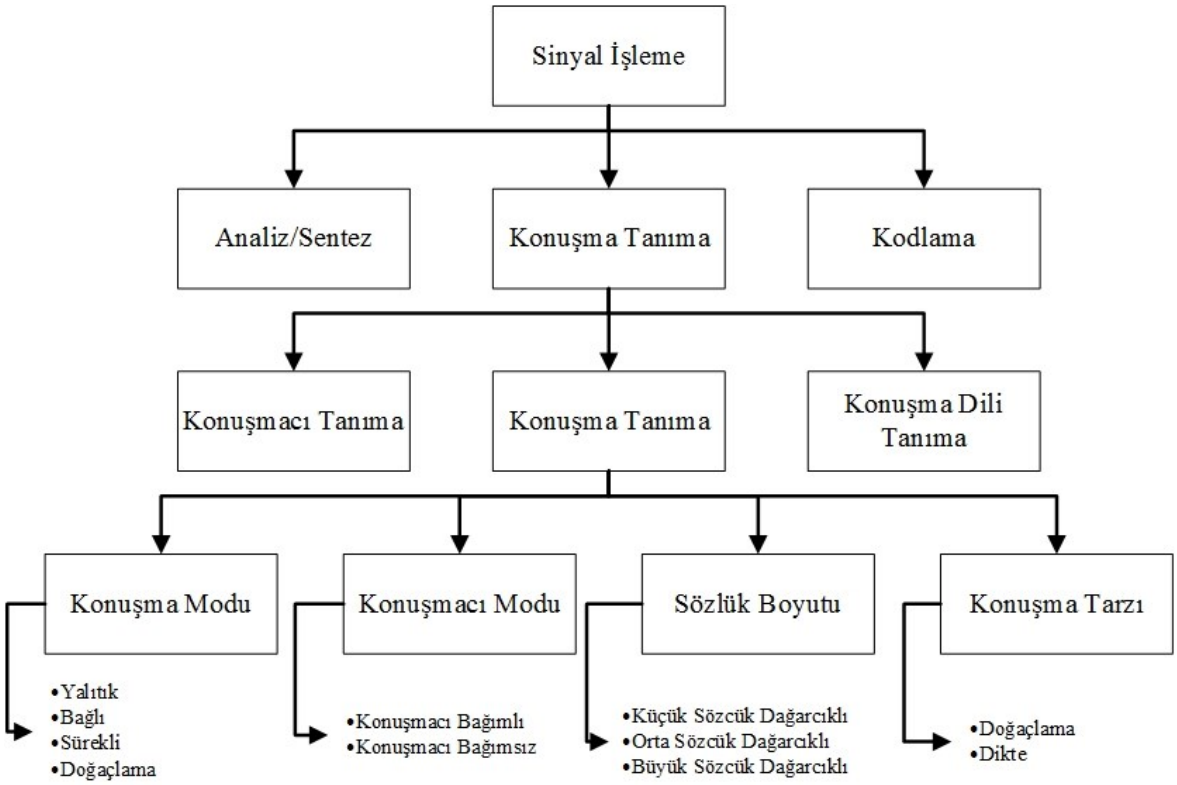
Ses tanıma sistemlerinin modellemesi, temel olarak Şekil 3.1'de verildiği gibidir. Ön işlem aşamasında giriş sinyali mikrofon aracılığı ile sisteme alındıktan sonra, ses örneklerinin çıkarılması, sayısallaştırılması, çeşitli filtrelerin yerleştirilmesi, bazı durumlarda etiketleme ve modellenebilecek bir formata dönüştürme gibi bir dizi işlem gerçekleştirilmektedir. Bu aşamada yapılan işlemler ile daha basit ve kolay işlenebilir, konuşma değişkenliği ve gürültüden arındırılmış bir ses sinyali üretilmektedir. Bir sonraki adımda, elde edilen örneklenmiş ses sinyalinin parametreleri yakalanır ve sesin belirli zaman aralıklarındaki öz niteliklerinin çıkarılması için hesaplamalar yapılır. Sesin öz niteliklerinin çıkarıldığı bu aşama oldukça kritiktir ve sese dair önemli ipuçları sağlar. Bu işlemler sonrasında sistem çıktısı olarak kelimeler üretilir. Bu üretimde hatalı olarak tespit edilmesi ve düzeltilmesi için dil modeli yapısından yararlanılır. Bu yapıda akustik model çıktısı olarak tahmin edilen kelimenin en olası hali dil modeli ile karşılaştırılarak bulunur. Böylece hatalı kelimenin düzeltilmesi ve sistem tanıma performansının artırılması amaçlanır. Arama aşamasında bir tür tahmin süreci işletilmektedir [5]. Bu işlem sonucunda en başarı sonuçlar sistem çıktısı olarak üretilir.



Şekil 3.1. Konuşma tanıma sistemleri temel mimarisi

3.2. Konuşma Tanıma Sistemlerinin Sınıflandırılması

Konuşma tanıma teorisinin temelini oluşturan konuşma işleme uygulamalarının farklı kategorilere ayrılmış hali Şekil 3.2'de gösterilmiştir. Konuşma analizi/sentezi ve kodlaması problemleri ile birlikte konuşma tanıma bu alanlar içerisinde daha yaygın çalışılan alanlardan birisidir. Her bir problem türü için farklı yaklaşımların kullanılması gereklidir. Farklı özgün problemleri ve çözüm yöntemleri bulunmaktadır. Konuşma işleme problemi içerisinde 3 farklı alt problem barınmaktadır. Bunlar konuşma tanıma, konuşmacı tanıma ve konuşmacı dili tanımadır. Ayrıca, konuşma tanıma uygulamalarının kapsamının geniş olması sebebiyle konuşma modu, konuşmacı modu, sözlük boyutu ve konuşma tarzı gibi sınıflandırmalar yapılmaktadır. Bir konuşma tanıma sisteminin tüm farklı türlere göre başarılı sonuçlara üretebilmesi mümkün değildir. Bu sebeple sınıflandırmalardan seçimler yapılarak problem oluşturulmalı ve sonuçlar üretilmelidir. Yalıtık, konuşmacı bağımlı, küçük sözcük dağarcıklı ve dikte sisteminde bir komut temelli sistem geliştirilebileceği gibi, doğaçlama, konuşmacı bağımsız, büyük sözcük dağarcık kullanılan dikte tarzında konuşmaların tanındığı otomatik konuşma tanıma sistemleri de geliştirilebilmektedir.



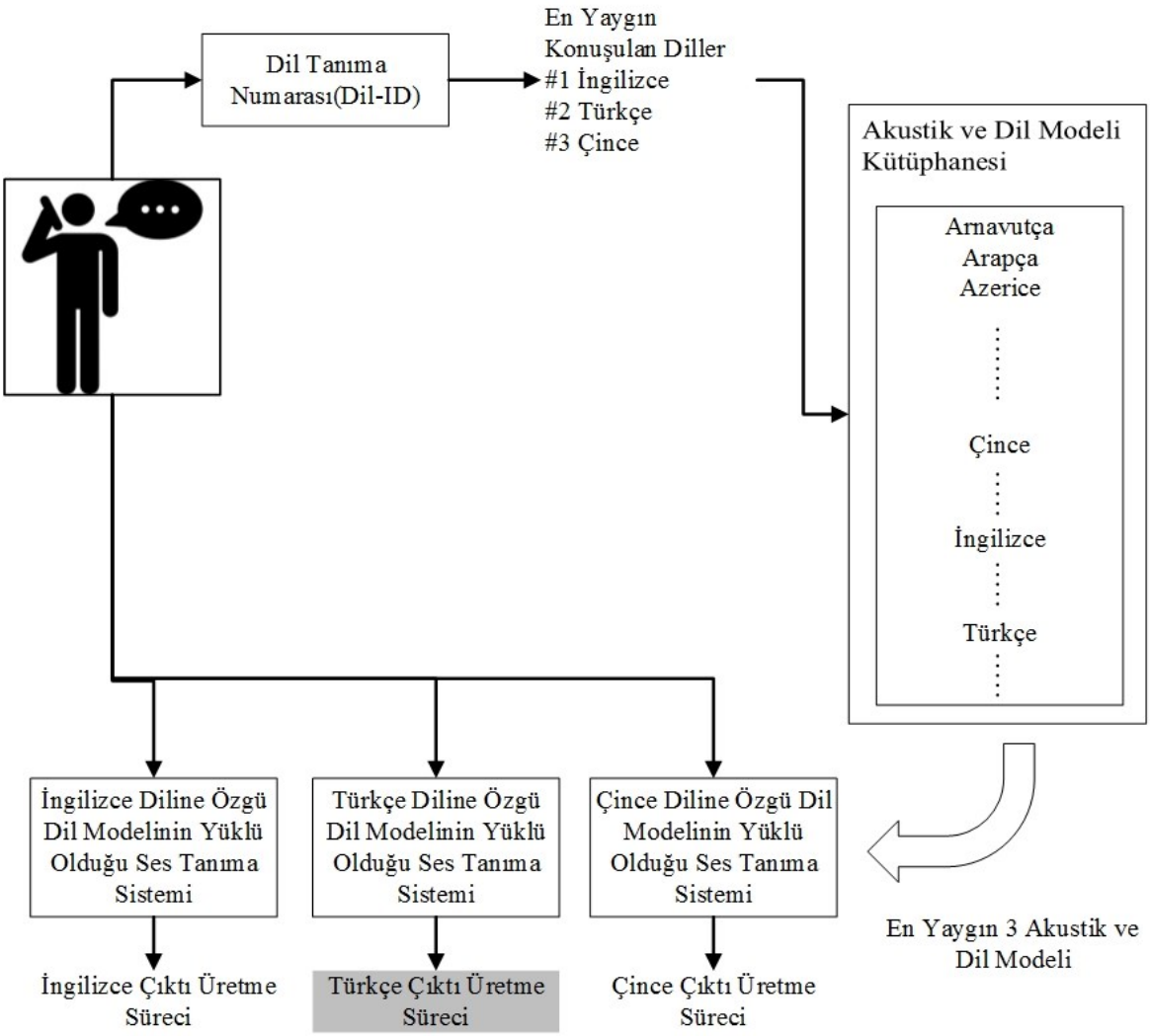
Şekil 3.2. Konuşma işleme sistemleri sınıflandırma şeması

3.2.1. Konuşmacı tanıma

Otomatik konuşmacı tanıma, kişinin kimliğini sesiyle doğrulamak için bilgisayarların kullanılmasıdır. Ses doğrulama, konuşmacı kimlik doğrulama, sesli kimlik doğrulama veya konuşmacı doğrulama gibi farklı isimler kullanılabilmektedir. Tanıma öncesinde herhangi bir kimlik iddiasında bulunulmaz ve sistem kişinin kim olduğunu, hangi gruba üye olduğunu veya daha önce tanınan biri olup olmadığına karar vermektedir. Konuşmacı doğrulamasında bir kimlik iddiasında bulunulur. Metne bağımlı türünde, sabit girdi sistem tarafından bilinir ve sabitlenir. Kişi mikrofona bu sabit metni okur ve ses sinyali kullanıcıların kimlik iddiasını kabul, ret veya yetersiz veri olarak değerlendirir. İkiz seslerin eşleştirilmesi temeline dayanır. Bunun yanında metin bağımsız olup kişinin ses frekansına bağlı olarak tanınmasına dayalı sistemlerde de geliştirilmektedir. Bu sistemlerde sesin mikrofon ile alınması sırasında ortam gürültüsü, sesin farklı sürümleri, mikrofonun yansıtıcı özellikleri dikkate alınmalıdır. Bu özelliklerin temsil edildiği ortamların yaratıldığı ve sisteme önceden aktarıldığı çözümlerde yüksek doğrulukla sonuçlar alınabilmektedir [82].

3.2.2. Konuşmacı dili tanıma

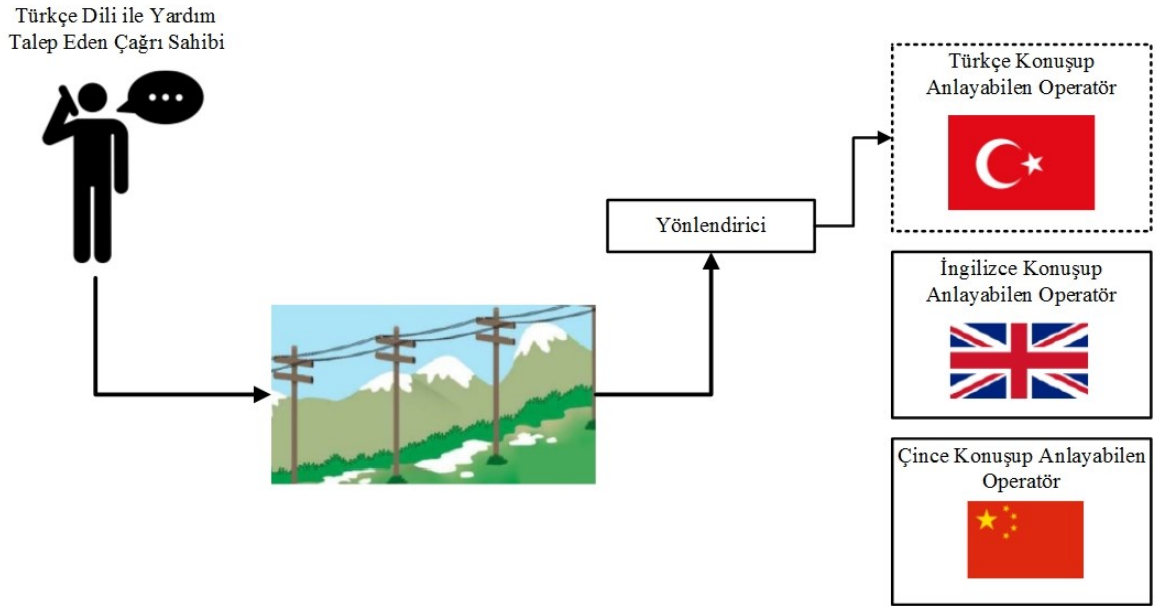
Otomatik dil tanımlama, konuşmacının konuştuğu dilin otomatik olarak tanımlanması süreci olarak kabul edilmektedir [83]. Çok dil kullanılan çeviri sistemlerinde veya acil arama yönlendirme yapan yerel operatörlerin müdahale süresinin kısaltılması gibi kritik uygulamalarda kullanılmaktadır. Konuşmacı dili tanımlama sistemlerinde üst düzey yaklaşımlarla karar verme yoluna gidilse de akustik modellemeye dayalı sistemler de daha başarılı sonuçlar alınabilmektedir [84]. Yeterli eğitim verisi ile çalışılan modellerde yüksek sınıflandırma performansları yakalanmıştır [85]. Eğitim sırasında konuşma dalga formları analiz edilir. Dil bağımlı modeller üretilir. Tanıma aşamasında, yeni bir ses işlenir ve eğitim sırasında üretilen modeller ile karşılaştırılır. Böylece problemlere çözüm üretilmiş olmaktadır.



Şekil 3.3. Gerçek zamanlı konuşma tanıma için dil tanımlama sistemi örneği [86]

Şekil 3.3'te örnek bir otel giriş veya uluslararası havaalanında çok dilli ses kontrol sistemine sahip bilgi alım sistemi örneği gösterilmektedir. Sisteme girdi olarak ses haricinde bir veri verilmiyor ise, sistem hangi dil ile konuşulduğuna karar verebilmelidir ve verilen komutları anlayabilmelidir. Bu sistemler aynı anda birkaç dili birlikte tanıyabilen ve farklı diller için hazırlanmış tanıma sistemlerinin paralel olarak çalıştığı bir mimaride tasarlanmalıdır. Kullanılacak olan farklı dil sayısının artışı beraberinde daha fazla donanım gerektirecektir. Alternatif olarak, konuşma tanıma işlemi öncesinde dil kimliği belirleme sistemi çalıştırılabilir. Bu sayede sesin öncelikle hangi dil olduğunun tespiti yapılır ve ses tanıma cihazına sadece o dilin modelleri yüklenerek tanıma işlemi gerçekleştirilir. Donanımsal ve işlemsel olarak avantajlı olmaktadır. Bu işlemler bittikten sonra, dil kimliğinin belirlenmesi işlemi yapılır [86].

Şekil 3.4'te insanları dinleyip çağrı sahibinin konuştuğu dile göre ilgili operatöre yönlendirme yapılan sistemler için ön işleme yapısı gösterilmiştir. Bu modele sahip sistemler hali hazırda polis şubelerinde, acil servis hizmetlerinde kullanılmaktadır. Dil hattında çalışan tercüman gelen çağrıyı dinleyerek hangi dil ile konuşulduğunu tespit etmeye çalışmaktadır. Tespit sonrasında çağrı o dili konuşabilen tercümana bir yönlendirici üzerinden iletilmektedir. Bu tür sistemler kurulurken, çalışacak kişilerin doğru seçilmesi mecburi hallerde değişikliğe gidilmesi gerekli olmaktadır. Gelen çağrıda konuşulan dilin kısa sürede tespit edilmesi ihtiyacı acil çağrılar için hayati bir konudur.

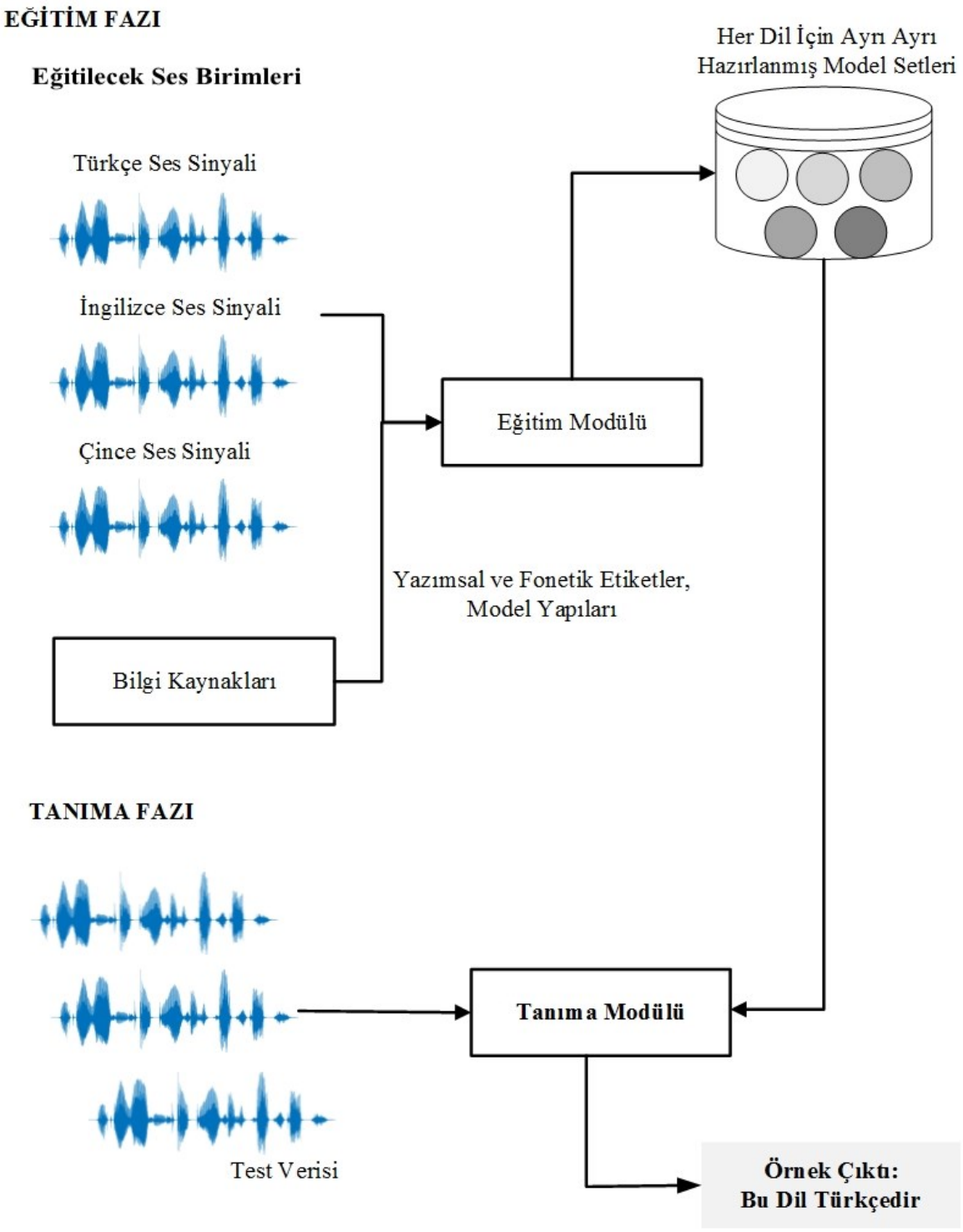


Şekil 3.4 Çok dilli asistanlık hizmetinin veya acil yardım sisteminin ön uç yapısı [86]

İnsanlar ve makineler bir dili diğerinden ayırmak için çeşitli ipuçlarından yararlanırlar. Dillerin farklı durumları dil bilim literatürlerinde tanımlanır ve araştırmacılar bu tanımlara atıfta bulunarak çalışmalarını yürütürler. Bir dili diğerinden ayırt etmek için temelde dil bilim çalışmalarına temel oluşturan aşağıdaki parametreler kullanılmaktadır [86]:

- Fonoloji: Bir fonem bir dildeki fonolojik birimin altta yatan zihinsel temsilidir ve akustik-fonetik birimlerin gerçekleştiği durumdur. Konuşmacının, konuşmayı düşündüğü anda üretilen gerçek sestir. Dil bilimciler tarafından ortaya konulan araştırmalarda, telefon ve ses birim setleri birçok dil için ortak olmakla birlikte bu durum için farklı ses birimleri ile karşılaşılmaktadır. Özellikle kullanım sıklıkları dilden dile farklılık gösterir. Fonoloji, bir dil için izin verilen fonem dizilerini düzenleyen kuralları ifade eder [86].
- Morfoloji: Kelime kökleri ve sözlükler dilden dile farklılık gösterir. Her dilin kendi sözcük dağarcığı ve kelimeleri oluşturma biçimi vardır. Bu biçimlerin değerlendirilmesi yoluyla dillerin birbirinden ayrılması mümkün olabilmektedir [86].
- Söz dizimi: Cümle kalıpları dilden dile farklılık gösterir. Aynı şekilde yazılan iki kelime farklı dillerde farklı şekillerde kullanılabilirler. Bu farklı dizilim dikkate alınarak hangi dil olduğunun belirlemesi yapılabilmektedir [86].
- Aruz(ölçü): Süre özellikleri, sesler arası adım aralıkları ve stres kalıpları dilden dile farklılık gösterir. Bu aruz özellikleri öğrenilerek dil belirlemesi yapılmaktadır.

Şekil 3.5'te örnek bir problem ve dil tanıma çözümünün şeması gösterilmiştir. Öncelikle, farklı dillere ait modellerin oluşturulabilmesi için o dile ait ses verileri toplanmaktadır. Bu verilerden, o dile ait yazımsal ve fonetik etiketler, model yapıları çıkarılmakta ve eğitim modülüne girdi olarak verilmektedir. Bu işlem sonrasında eğitim modülü çıktısı model olarak elde edilmektedir. Bu model test aşamasında kullanılmaktadır. Test işlemlerine geçildiğinde, öncelikte tanınması beklenen ses sinyali alınmaktadır. Bu sinyalin özellikleri çıkarılarak model üretilmektedir. Bu model daha önce oluşturulmuş model kümesi içerisinde aranmaktadır. Arama sonucunda eğer elde bulunan dil modelleri içerisinde eşleşme yakalanır ise girdinin dili belirlenmiş olur. Bu şekilde tasarlanmış otomatik dil belirleme yapıları oldukça kritik olup, acil çağrılar, güvenlik hizmetleri gibi süre problemi olan durumlarda faydalı olmaktadır.



Şekil 3.5. Kullanıcı dil tanıma sistemi [86]

Kapsamlı bir dil tanıma uygulaması tasarımında çok sayıda dile ait modelin olması beklenmektedir. Bu şemada en zorlu süreç her bir dil için modellerin oluşturulması aşamasıdır. Modeller oluşturulduktan sonra, test aşamasında pratik ve hızlı bir şekilde dil belirlenebilecektir.

3.2.3. Konuşma tanıma

Konuşma tanıma sistemleri girdi olarak aldıkları birimlere göre farklı sınıflara ayrılmaktadırlar. Farklı araştırma alanlarında daha özel sınıflandırmalar yapılmakla birlikte konuşma moduna göre 4 farklı gruba ayrılmaktadırlar.

Konuşma modu

Yalıtık sözcük tanıma: Ses sinyali içerisindeki kelime söyleyişleri arasında belirli bir miktarda sessiz bölümün olmasını zorunlu kılan konuşma tanıma türdür. Bu sistemlerde bir zaman periyodunda tek bir kelime veya sesin sisteme girdi olarak verilmesi istenir. Konuşmacı her bir ses birimi sonrasında duraksamalı ve sonrasında ikinci ses birimini seslendirilmelidir. Böylece ses içerisinde kelime sınırlarının belirlenmesi gibi zor ve tanıma performansını doğrudan etkileyen problemin yaşanmaması sağlanmış olur. Bu sistemler genelde komut temelli problemlerin çözümünde, ileri-geri, aç-kapa gibi görevlerin yerine getirilmesi gereken modellerde kullanılırlar.

Bağlı sözcük tanıma: Sesler arasında belirgin duraksamaların daha az olmasının kabul edilebilir olduğu sistemlerdir. Kısa duraksamalar ile konuşmaya devam edilebilmektedir. Bu durum kelime sınırlarının belirlenmesinde avantaj sağlamaktadır.

Sürekli konuşma tanıma: Kullanıcıların hemen hemen devamlı şekilde konuşarak, bilgisayar ile iletişimde bulunabildiği sistemlerdir. Bilgisayar diktesi olarak adlandırılmaktadır. Bu sistemlerde yalıtık ve bağlı sözcük tanıma sistemlerine göre daha ciddi problemler bulunmaktadır. Bunların en önemlisi sürekli ses sinyalindeki kelime sınırlarının belirlenmesidir. Çoğu halde sınırların belirlenmesi mümkün olamamakta veya belirlemek için ciddi hesaplamalara ihtiyaç duyulmaktadır. Günümüzde daha çok bu tipteki sistemler üzerinde çalışmalar yapılmaktadır.

Spontane konuşma tanıma: İnsanların doğal konuşma halleriyle ve herhangi bir hazırlık olmaksızın yaptıkları seslendirmelerin bilgisayarlar tarafından işlendiği sistemlerdir. Bu tipteki tanıma yapısında sürekli konuşma özelliklerinin yanında, konuşmacıların doğal hallerinde kullandıkları "ımm, aaa, uuu" gibi seslerin ve gürültü gibi durumlarında tespit edilmesi ve çözümlenmesi gerekmektedir.

Konuşmacı modu

Ses tanıma sistemlerinde konuşmacının değişmesi tanıma performansını doğrudan etkileyen önemli bir unsurdur. Seslendirmeyi yapan kişinin değişkenlik durumuna göre konuşmacı bağımlı ve bağımsız olarak iki farklı türe ayrılmaktadır [3].

Konuşmacı bağımlı: Tek kişinin referans şablonlarının oluşturulması ile probleme çözüm üretmeye çalışılmaktadır. Yeni bir kişinin tanınması için referans şablonlarının güncellenmesi gerekir. Farklı kişi ile test edilmesi halinde tanıma performansında ciddi düşüşler olabilmektedir.

Konuşmacı bağımsız: Birden fazla kişiye ait şablonların oluşturulduğu ve farklı kişilerin seslendirmelerinin de tanınabildiği yapılardır. Bu yapıdaki en önemli problem, kişinin değişmesi halinde ses sinyalinden üretilen özelliklerde değişme olmasıdır. Bu değişim, sistemin tanıma performansı oldukça düşürmektedir. Ancak geniş dağarcıklı otomatik konuşma tanıma sistemlerinde konuşmacı bağımsız modelleme yapılması beklenmektedir.

Sözlük boyutu

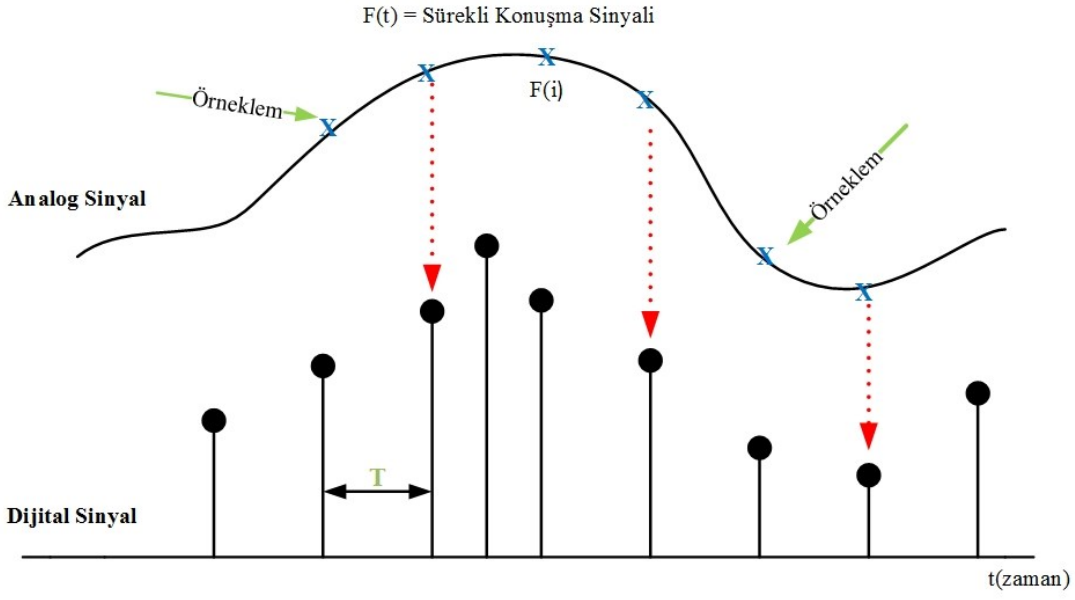
Bir konuşma tanıma sisteminde tanınabilen sözcük kümesine sistem sözlüğü denilmektedir. Kelime altı birimler ile çalışıldığı durumlarda ise ses kümesi olarak adlandırılmaktadır. Bir sözcüğün sistem tarafından tanınabilmesi için öncelikle sözlükte bulunması gereklidir. Ancak, sözcük boyutunun çok büyümesi tanıma başarısını olumsuz yönde etkilemektedir. Dağarcığın artırılması daha fazla işlem gücü ve daha fazla tanıma performansında düşüş demektir. Konuşma tanıma sistemlerinde kullanılan sözlük boyutuna göre dağarcıklar sınıflandırılmaktadır. Küçük dağarcıklı sistemlerde 100 kelime bulunurken, orta dağarcıklı yapılarda 100 ile 1000 kelime ve büyük dağarcıklı sistemlerde 1000'den fazla kelime bulunmaktadır. Komut temelli sistemlerde küçük dağarcıklı bir sözlük yeterli olabilmektedir. Bu sebeple başarı seviyeleri oldukça yüksektir. Ancak konuşma diktesi gibi sistemlerde çok daha fazla kelimeye ihtiyaç duyulmaktadır. Büyük dağarcıklı bir sözlük oluşturulması ve modelin bu sözlük ile test edilmesi halinde de sistem performansı hızlı bir şekilde düşmektedir. Bu noktada sözlük boyutunun büyüklüğü ve sistem performans düzeyi arasında bir denge yakalanmalıdır. Sözlük dışı kelimelerin varlığına ilişkin çözüm önerileri ortaya koyulmalıdır.

3.3. Akustik Ön-Uç

Konuşma tanıma sistemlerinin sinyali aldıktan sonra ilk noktası akustik ön uç bölümüdür. Bu adımda konuşma sinyalinin öz nitelikleri çıkarılır ve eğitim aşamasında kullanılabilir daha özet bilgisi üretilir. Gereksiz bilgilerden arındırma, işlem gücünde kazanç ve tanıma performansında artışa imkân tanır. Bu adımda konuşma sesinden gürültü gibi fazlalıkların temizlenmesi, sessiz kısımların etiketlenmesi, sesin vurgulanması gibi işlem öncesi hazırlık süreçleri yürütülür.

3.4. Öznitelik Çıkarımı

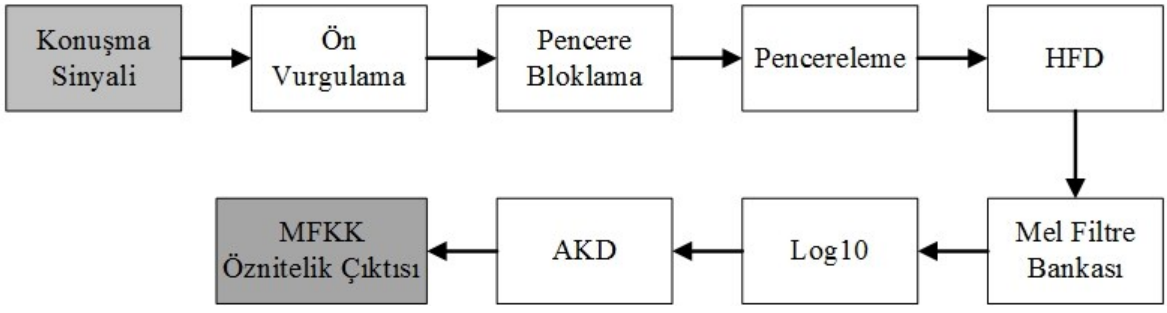
Konuşma sinyalinin konuşma tanıma süreçlerinde kullanılabilmesi için analog sinyalin dijital örneklere dönüştürülmesi gereklidir. Bu işlem mikrofon ve ses kartı vasıtasıyla gerçekleştirilir. Dijital örnekler şeklinde aktarılması sonucunda sürekli bir konuşma sinyali ayrık sinyal örneklerine dönüştürülür. Örnekleme yaparken belirlenmesi gereken iki parametre bulunmaktadır. Birincisi örneklem sıklığı, ikincisi ise örneklerin tutulduğu bit sayısıdır. F(t) konuşma sinyali için T zaman aralığında örneklem alınmaktadır. Örneklem sıklığı bir saniyede alınan örnek sayısı ile ifade edilir. Konuşma tanıma uygulamalarında kullanılan veri setleri incelendiğinde seslerin 8 Khz ve 16 Khz örneklem sıklığında kaydedildiği anlaşılmıştır.



Şekil 3.6. Analog sinyalin dijitalleştirilmesi

Şekil 3.6'da analog sinyalinin dijital hale dönüştürülmesi şeması gösterilmektedir. Bu noktada dikkat edilmesi gereken husus, 1 sn. için 16 Khz örneklem sıklığında yaklaşık 16 bin örnek alınması durumudur. 1 saniye de ortalama olarak bir harfin seslendirildiği varsayılırsa, 16 bin örnekten bu harfi tespit etmek mümkün değildir. Bu sebeple konuşma tanıma uygulamalarında zaman ve genlik boyutu yerine frekans boyutunda çalışılmaktadır. Sesin öznitelikleri frekans boyutundaki verilerden elde edilmektedir.

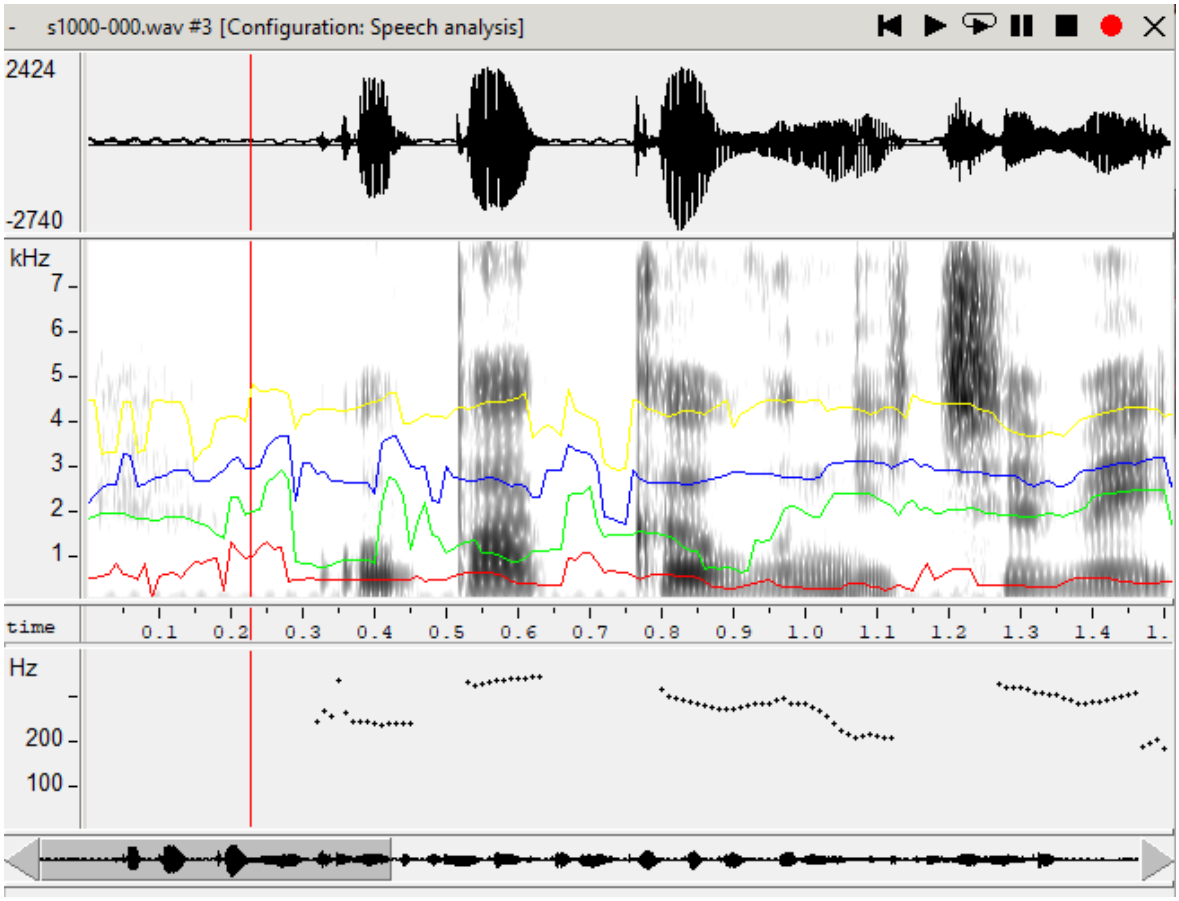
Sese ait özniteliklerin çıkarılması için Mel Frekans Kepstral Katsayıları(MFKK), Doğrusal Öngörülü Kodlama(DÖK), Temel Bileşen Analizi(TBA) gibi bir çok algoritma kullanılmaktadır. Bu çalışmada bu algoritmalardan en yaygın kullanım alanına sahip olan MFKK ile öznitelik çıkarımı yapılmıştır [87]. Enerji ile birlikte toplam 13 farklı özellik üretilebilmektedir. MFKK kullanılarak üretilen vektörlerde, sinyal üzerindeki örnek sayısı azaltılarak harflerin tanınması için gerekli bilgileri barındıran bilgiler elde edilir. İnsan algısının 1 Khz üzerindeki frekanslara duyarlı olması sebebiyle Mel skalası bu frekans üzerinde kullanılır [88]. MFKK öznitelik çıkarımı için Şekil 3.7'de verilmiş olan adımlar takip edilir [89]. Ön vurgulama aşaması, girdi işaretinin yüksek frekanslarda enerjisini yükseltmek için kullanılır. Giriş sinyalinin ayrık zaman dilimlerine ayrılması pencereleme aşamasında gerçekleştirilir. X ms genişlikte ve Y ms uzunlukta birbirini takip eden pencere zinciri oluşturulur. Hamming pencereleme yöntemi bu işlem için yaygın olarak kullanılmaktadır. Ses sinyali Hızlı Fourier Dönüşümü (HFD) ile zaman boyutundan frekans boyutuna taşınmaktadır. Böylece tanıma için gerekli bilgilere ulaşılmış olur. Frekans boyutunda elde edilen spektrum bir logaritmik Mel ölçeği ile kırpılır. Kırpılan değerler Mel filtre bankası ile işlenir ve keptral değerleri elde edilir. Kepstral katsayılarının filtre bankasından elde edilmesi için Ayrık Kosinüs Dönüşümü (AKD) kullanılır. Böylece bir ses sinyalinden tanıma yapabilmek için ihtiyaç duyulan özellikler elde edilmiş olur.



Şekil 3.7. MFKK yöntemi özellik çıkarım adımları [89]

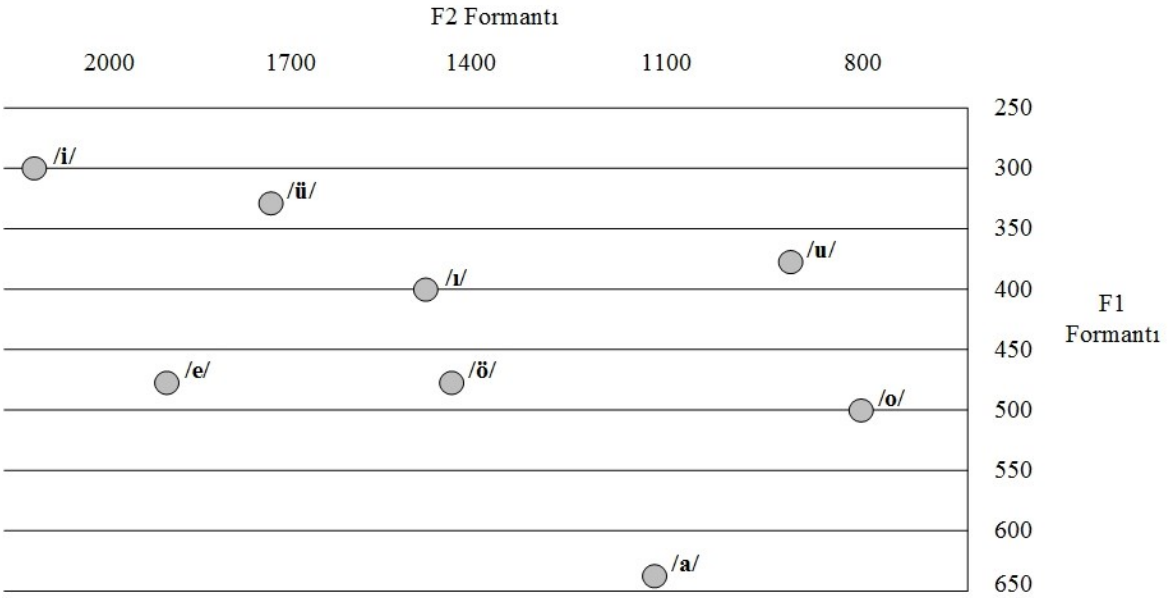
Şekil 3.8'de bu çalışmada kullanılan test veri setinden alınmış örnek bir "wav" dosyasının frekans boyutundaki spektrogram ve zaman boyutundaki "waveform" gösterimi verilmiştir. Zaman boyutundaki gösterim bir sinyalin içerdiği harf bilgisine ilişkin herhangi bir bilgi sunmazken, frekans boyutundaki spektrogram çiziminde koyu renkli kısımlar enerji seviyesinin daha yüksek olduğu bölümleri göstermektedir. Enerji yükselmesi ses içerisindeki fonem veya harflerin bulunduğu bölümü işaret eder.

Sarı, mavi, yeşil ve kırmızı olarak gösterilen çizgiler frekans spektrumundaki maksimum formant değerlerini ifade eder. F1-F4 formant aralığı ses yolunun en düşük tınlaşım frekansını göstermektedir. En düşük frekansta formant F1, ikinci seviye F2 ve üçüncü seviyede F3 olarak gösterilmektedir. Formant değerleri birçok problemin çözümünde oldukça faydalı bilgiler vermektedir. F1 ve F2 değerleri sesli harflerin tanınmasında kullanılırlar. F3 formant değeri ise daha çok sessiz harflerin tanınmasında gerekli bilgileri bulundurmaktadır.



Şekil 3.8. Örnek wav dosyası wavefrom ve spektrogram bilgileri gösterimi

Türkçe sesli harflerin F1 ve F2 seviyesindeki formant frekansları Şekil 3.9'da gösterilmiştir. Farklı harflerin farklı formant değerine sahip olması konuşma tanıma uygulamaları için oldukça kritiktir. Formant değerleri konuşmadan metne dönüşüm uygulamaları ile birlikte konuşmacı tanıma, konuşmacı doğrulama gibi problemlerde de kullanılmaktadırlar.



Şekil 3.9. Türkçe sesli harflerin f1 ve f2 formant değerleri [90]

3.5. Sistem Modellemesi

Bölüm 3.1'de genel sistem mimarisinde gösterildiği gibi özniteliklerin çıkarılması sonrasında akustik ve dil modelinin oluşturulması gerekmektedir. Akustik model ile konuşma sinyalinin metne dönüşüm işlemi gerçekleştirilir iken, dil modeli ile sonuçlar üzerinde düzeltme ve genel sistem performansını artırma amaçlanır. Dil modelinin büyüklüğü ve metin veri setinin model yapısı sonuçlara doğrudan katkı sunan bir durumdur.

3.6. Akustik Model Tasarımı

Otomatik konuşma tanıma teorisi, verilen bir akustik X dizisi için, W kelime dizilerini bulmayı amaçlamaktadır. Konuşma cümleleri $W = (W_1, W_2, \ldots, W_n)$ şeklinde belirtilen kelimelerin dizisi olarak gösterilir. "W_t" ayrık bir "t" zamanında söylenmiş belirli bir kelimedir. Kelimelerin dizisi söylenen sesli ifade ile bağlantılıdır ve bu sesli ifade "X" olarak

gösterilen akustik seslerin dizisidir [3]. Geniş sözcük dağarcıklı sürekli konuşma tanıma için standart yaklaşıma göre Eşitlik 3.1 kullanılmaktadır

$$P(W|A) = \arg\max P(W/A) \tag{3.1}$$

Eşitlik 3.1'e göre olası P(W|A) olasılıklı A sözcük dizisinin W akustik gözlem dizisini ürettiği varsayılmaktadır. Sonrasında, akustik gözlem sırasına bağlı olarak söylenmiş olan kelime dizisinin çözümlenmesi ve maksimum olasılıklı dizinin ortaya çıkarılması yapılmaktadır. Bu şekilde en yüksek olasılıklı sistem çıktısının üretilmesi sağlanmış olmaktadır. Eşiltlik 3.1, Bayes kuralına göre Eşitlik 3.2'teki gibi düzenlenmektedir.

$$P(W|A) = P(A/W)\,P(W) / P(A) \tag{3.2}$$

Eşitlik 3.2'ye göre P(A) olasılık değeri, W'den bağımsız olarak alınırsa, maksimum olasılık dağılımı tahminine göre Eşitlik 3.3 elde edilmiş olur.

$$W = \arg\max P(A/W)\,P(W) \tag{3.3}$$

Eşitlik 3.3 ses analizinin teorik alt yapısını açıklamada kullanılan en yaygın matematiksel modeldir. Bu eşitlik incelendiğinde, P(A/W) olarak tanımlanan denklemin ilk modeli, akustik model olarak isimlendirilir. Sözcük dizesine göre düzenlenmiş akustik model dizisinin olasılığını ifade eder. Büyük ölçekli konuşma tanıma sistemlerinde alt sözcük konuşma birimleri ile istatistiksel modeller oluşturulmaktadır. Sonrasında bu alt birimler bir araya getirilerek kelimeler ve kelimelerin bir araya getirilmesi ile de sözcük dizileri elde edilmektedir. Bu dizilerin üretilmesinde maksimum olasılığı dağılımı kullanılmaktadır. P(W) olarak tanımlanan ikincil terim dil modeli olarak isimlendirilir. Sözlü ifadeler dizisiyle ilgili olasılığı tanımlar. Bu tür dil modeli dilin söz dizimsel ve semantik kısıtlamalarını içerebilir [5]. Akustik model olasılık değeri ile bu modelin ürettiği çıktıya en yakın dil modeli dizisinin maksimum olasılık değerleri eşleştirilerek hatalı karakter sayısı en az olan sistem çıktılarının üretilmesi sağlanmış olmaktadır. Her iki model içinde en iyi olasılık değerleri seçilmektedir.

3.7. Dil Model Tasarımı

P(W) dildeki kelimelerin akustik modellemesinden bağımsız olarak, N-gram yöntemi ile modellenir [91]. Bu modelleme yapısında Markov varsayımı kullanıldığı durumlar için geçmişe dönük 3 ile 5 arasında kelimeyi olasılık hesaplamalarına dâhil etmek mümkün olabilmektedir. Daha fazla geçmişe yönelik bağımlılık varsa ve bu verinin de modele eklenmesi gerekiyor ise tekrarlayan sinir ağı yapıları kullanılmaktadır. Dil modeli tasarımlarında o dile ait kelime setleri ile çalışılmaktadır. Bu veri setlerinde mümkün olduğunca farklı ve çok sayıda kelimenin bulunması beklenmektedir. Böylece sistemin ürettiği çıktılar ile eşleşen kelimenin veri setinde olma olasılığı artırılmaktadır.

Dil modeli geniş bir kelime kataloğu ve bu kelimelerin cümle ile ilgili olma olasılıklarını vermektedir. Bir dili modellemek için kullanılan yöntemlerden en yaygın kullanılanı N-gram yaklaşımıdır. Bu yöntem belirli bir W_i kelimesinin, kendisinden önceki n-1 sayıdaki kelimeye bakarak olasılığının (P) hesaplanmasına dayalıdır. Dil modeli matematiksel olarak Eşitlik 3.4'te gösterildiği şekilde temsil edilir.

$$P(W_i \mid W_1, W_2, \ldots, W_{i-1}) = P(W_i \mid W_{i-(n-1)}, W_{i-(n-2)}, \ldots, W_{i-1}) \quad (3.4)$$

Tanıma yapılacak ses verisinin konusuna ve içeriğine bağlı olarak dil modelinin oluşturulması oldukça önemlidir. Sistemin çalışma performansına doğrudan etki eden bir unsurdur. Modelin oluşturulduğu sözlüğün genişliği ne kadar çok olursa olsun, her halde bazı kelimelerin derlem içerisinde bulunmaması durumu oluşacaktır. Bu durumda düzleme işlemi ile olasılık değerlerinde dengeleme yapılması gereklidir [92].

Dil modeli tasarımlarının tanıma birimine bağlı olarak farklı türleri bulunmaktadır. Kelime tabanlı dil modelinde tanıma birimi olarak doğrudan sözcükler kullanılır. Morfolojik veya istatistiksel yöntemlerin aksine doğrudan dil modeli oluşturulur. N-gram dil modelleri kelimeler kullanılarak hesaplanır [34]. Dil modeli tasarımlarında tanıma birimi değişikliği, modeli doğrudan etkileyen hususlardan birisidir. Bu yöntemlerden kök-ekler tabanlı dil modelinde tanıma birimleri kök ve kökten sonraki kısmını içerir. Olasılıklar kök ve ekler üzerinden oluşturulur. Morf tabanlı dil modelinde tanıma birimi olarak morflar kullanılır. Kelimelerin morflarına ayrılması işlemi otomatik olarak gerçekleştirilmelidir. Büyük derlem ile çalışıldığı durumlarda dilin yapısını bilen gözetimsiz algoritmalar kullanılmaktadır.

Hibrit dil modelinde ise, kelime ve kök-ekler tabanlı dil modeli yapısının birlikte kullanılmasına dayanmaktadır. Sık geçen kelimelerin kök ve eklerine ayrılarak modellenmesi, geriye kalan kelimelerin ise ayrıştırmadan doğrudan kelime olarak kullanılması temel temel alan modeller oluşturulabilmektedir. Sonuçta, birden fazla dil birimi ile modellerin oluşturulabilmesi ve konuşma tanıma sistemlerinde kullanılabilmesi mümkündür. Bu sebeple çalışılacak birimin seçimi probleme özel olarak yapılmalıdır.

3.8. Performans Analizi

Otomatik konuşma tanıma sistemi özellikle derin öğrenme ve güçlü bilgi işlem platformlarındaki ilerlemeler sayesinde son yıllarda hızlı bir ilerleme kaydetmiştir. Bunun sonucunda, metin tercümesi, kişisel asistanlar, medya izleme, sesten metne dönüşüm gibi birçok farklı uygulama geliştirilmiştir. Bu ilerlemeye rağmen, OKT sistemlerinin performansları hala akustik model ve dil modeli için kullanılan eğitim verilerinin, test koşullarına ne kadar uygun olduğuna bağlıdır. Bu nedenle hedef ortamlarda bir OKT sisteminin doğru sınıflandırma oranını tahmin etmek oldukça önemlidir.

Kelime hata oranı (KHO), büyük sözcük dağarcığı ile sürekli konuşma tanıma yapan sistemlerin performansını değerlendirmek için kullanılan standart bir yaklaşımdır [93]. OKT sistemi tarafından üretilen kelime dizisi, referans metin dizisi ile hizalanır. Yer değiştirme(Y), ekleme(E) ve silme(S) hatalarının toplamı hesaplanır. Referans metin dizisinde toplam N kelime olduğu varsayılırsa, kelime hata oranı Eşiklik 3.5'e verilmiş olan formüle göre hesaplanmaktadır.

$$\text{Kelime Hata Oranı(KHO)} = \frac{\text{Y+E+S}}{\text{N}}\text{X100} \tag{3.5}$$

KHO hesaplaması "Levenshtein" mesafesi ölçümüne dayanmaktadır. İki metin dizesi arasındaki farkın hesaplaması yapılır. Karakter başına hatalı işlem sayısının toplamının, toplam karakter sayına oranlanması ile bulunur. Farklı sistemleri karşılaştırma ve tek bir sistemdeki iyileştirmeleri değerlendirmek için iyi bir araçtır [93]. Bununla birlikte çıktılarda bulunan hatalara ilişkin hiçbir ayrıntı vermez. Bu sebeple hataların ana kaynaklarının tanımlanması ve araştırılması için ayrıca çalışmalar yapmak gereklidir.

3.9. Otomatik Konuşma Tanıma Sistemlerinin Tasarımında Karşılaşılan Zorluklar

Konuşma tanıma sistemlerinin geliştirilmesi oldukça zaman alıcı çalışmaları ve karmaşık matematiksel problemlerin çözülmesini gerektirmektedir. İleri seviyeli konuşma tanıma uygulamaları içinse uzun süreli ekip çalışmalarına ve büyük veri setlerine ihtiyaç duyulmaktadır. Bunun en önemli sebebi OKT sistemlerinde dikkat edilmesi gereken birçok unsurun bulunmasıdır.

Bu alanda yapılan çalışmalarda aşağıda verilmiş olan zorluklarla karşılaşılması olasıdır [94, 95]:

Konuşmanın insanlar tarafından anlaşılması

İnsanlar bir sesi dinlerken konuşma hakkındaki daha önceki bilgilerini kullanırlar. Standart bir konuşmada, kelimeler keyfi diziler halinde kullanılmazlar. Her dile ait bir takım dil bilgisi kuralları vardır ve insanlar bir konuşmayı dinlerken henüz söylenmemiş kelimeleri çoğu halde tahmin edebilirler. Ancak konuşma tanıma sistemlerinde sadece ses sinyalleri kullanılır ve bu sinyaller yardımıyla kelimelerin doğru tahmin edilmesi beklenir. Bu soruna çözüm üretmek için dil bilgisi yapısını modelleyebilen yapılar veya bazı istatistiksel modellemeler kullanılır. Ancak konuşmacıların ve dünyadaki bilgi birikiminin tamamının nasıl modellenebileceğine dair ciddi sorunlar vardır. Konuşma tanıma uygulamalarının geliştirilmesinde, probleme en yakın modellemenin yapılması daha iyi sonuçlar alınabilmesini sağlayacaktır.

Konuşulan dil ile yazı dilinin birbirine birebir eşit olmaması durumu

Konuşma tanıma sistemleri ile ilgili yapılan ilk çalışmalarda konuşulan dil ile yazılı dili arasındaki farkın önemsenmeyecek derece az olduğu düşünülmekteydi. Ancak, insanlar konuşurken, yazılı metin oluşturmaya göre, daha az dikkat ederler ve ciddi telaffuz hataları yapabilirler. Bu durum konuşmadan metne dönüştürme problemlerinde sorunlara sebep olmaktadır. Ayrıca diyalog şeklide konuşma yapılması durumunda farklı seslerin üst üste binmesinden dolayı kayıplar yaşanabilmektedir. Doğaçlama bir konuşmada, tekrarlar, dil sürçmesi, konuşma ortasında konu değişiklikleri olabilmektedir. Tüm bu durumlar konuşma

dili ile yazı dili arasında farkın oluşmasına neden olmaktadır. Bu sebeple OKT sistemlerinde bu eşitsizlik durumunun sebep olacağı problemlere dikkat edilmelidir.

Gürültü durumu

Konuşma sesleri, arka planda başka bir konuşmacı sesi, bir odada televizyon sesi gibi dış seslerle karışık olabilmektedir. Konuşma sinyali dışında olup, ses sinyali içerisinde bulunan tüm bu harici sesler gürültü olarak adlandırılır. Gürültü, konuşma sinyalinin tanınmasında sorunlara neden olabilmektedir. Bu sebeple konuşma sinyalinin filtrelenerek temizlenmesi gerekmektedir. Bu sorun dış ortamlardan mikrofon vasıtası ile seslerin alınarak işlendiği durumlarda daha çok olabilmektedir. OKT tasarımlarında bu problem ile karşılaşmamak için stüdyo ortamında kaydedilmiş ve gürültüden arındırılmış akustik veri setleri kullanılmaktadır.

Beden dili

İnsanlar birbirleri ile iletişim kurarken, konuşma harici el-kol hareketlerini, hareketli gözlerini ve duruş şekillerini de kullanırlar. Böylece söylemin anlaşılmasını kolaylaştırırlar. Geleneksel otomatik konuşma tanıma sistemlerinde eğer görüntü işleme ile birlikte hibrit bir model kullanılmıyor ise beden dilinin değerlendirilmesi mümkün olamamaktadır. Bu nedenle, beden dili kullanımı gerektiren problemler için, işaret işleme ile birlikte görüntü işleme yapıları kullanılmalıdır.

Kanal değişkenliği

Farklı mikrofon tipleri, zaman içinde gürültü ve konuşmacı değişkenliğinden kaynaklı bilgisayarda ayrık temsil gibi, akustik ses içeriğini etkileyen her durum kanal değişkenliği olarak adlandırılır. Kanal değişkenliğine sebep olan parametreler konuşma tanıma performansını etkilemektedirler ve model tasarımlarında gözetilmelidirler.

Konuşmacı değişkenliği

İnsanların sesleri birbirinden farklı olduğu gibi, tek bir kişinin dahi farklı zamanlarda ses sinyalleri birbirinden farklı olabilmektedir. Konuşmacıların konuşma stili bu farklılığın en

önemli unsurudur. İnsanlar aynı kelimeleri farklı şekilde telaffuz ederler. Bir kişinin evde veya kamusal alandaki konuşması farklı sinyal şekillerine sahip olabilmektedir. Ayrıca üzgün olma, korku, öfke gibi duygu durumlarında değişiklikler de konuşma stilini değiştirebilmektedir. Konuşma stili değişikliğinin yanında, bir kişinin aynı kelimeyi tekrar tekrar söylemesinde konuşma sinyalinin aynı olmaması durumu oluşmaktadır. Üretilen akustik sinyalde farklılıklar olacaktır. Konuşmanın gerçeklemesi zaman içinde değişkenlik gösterecektir. Konuşmacı değişkenliğinin bir diğer unsuru da söyleyiş yapısıdır. Ağız, bir dil içerisinde farklı gruplarda birbirinden farklı unsurları barındıran söyleyiş yapısıdır. Bölgesel lehçe, konuşmacının yaşamış olduğu bölgeye göre kelime dağarcığındaki, telaffuzundaki ve gramer özelliklerinde farklılığı gösterir. Sosyal lehçe ise, konuşmacının dâhil olduğu sosyal çevre sebebiyle aynı özelliklerde farklılıklar olabilmektedir. Aynı dil içerisindeki tüm bu söyleyiş farklılıklarının geliştirilen konuşma tanıma uygulamalarında gözetilmesi gereklidir.

3.10. Konuşma Tanıma Uygulamaları Kullanım Alanları

Otomatik konuşma tanıma sistemlerinde derin sinir ağı yapısının kullanılmaya başlaması ve bilgi işlem kapasitelerinin artması sayesinde oldukça başarılı sonuçlar alınmaktadır. Bu durum konuşma tanıma sistemlerinin gerçek uygulamalarda yaygın bir şekilde kullanılmaya başlamasını sağlamıştır. Çizelge 3.1'de bu alanda yapılan örnek uygulamalar, kullanım sektörleri ile birlikte detaylı olarak gösterilmiştir [5].

Çizelge 3.1. Konuşma tanıma uygulama örnekleri

Problem	Uygulama	Girdi Parametresi	Üretilen Çıktı
Metin Tercümesi	Bir dilden başka bir dile tercüme işlemlerinin otomatik olarak gerçekleştirilmesi	Konuşma Sinyali	Tercüme Metni
Metin Diktesi	Sürekli konuşma sinyalini alıp, metin çıktısını üreten sistemler	Konuşma Sinyali	Dikte Metni
Fiziksel olarak kısıtlı durumlar	Fiziksel olarak kısıtlı insanların engellerini ortadan kaldıracak çözümler	Konuşma Sinyali	Konuşma Metni, Görevi yerine getirme vb.
Sağlık Sektörü	Sağlık metinlerinin oluşturulmasında hız avantajı sunacak uygulamalar	Konuşma Sinyali	Dikte, görevi yerine getirme, çağrı işlemleri vb.

Problem	Uygulama	Girdi Parametresi	Üretilen Çıktı
Askeri Sistemler	Yüksek performanslı savaş uçakları, savunma sistemlerinin yönetimi, savaş yönetiminde yardımcı sistemler	Konuşma Sinyali	Dikte, görev yerine getirme, komutları uygulama vb.
Konuşma, Telefon Alt yapısı ve İletişimi	Çağrı merkezlerinde danışma ve sorun çözme süreçlerinin gerçek danışmanlar kullanılmaksızın yönetilmesi	Konuşma Sinyali	Konuşma Metni, Veri Değerlendirme Sonuçları

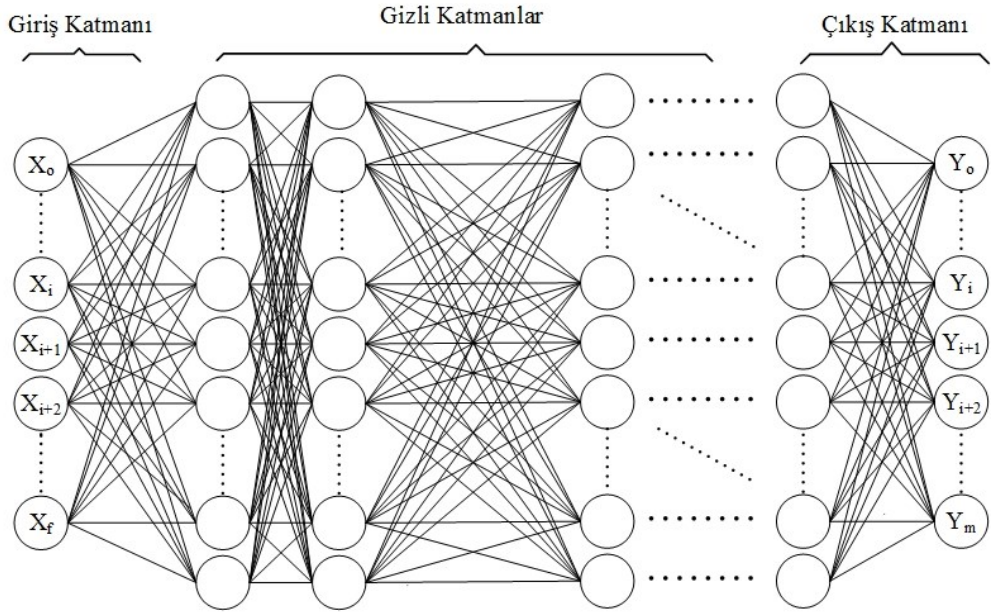
OKT sistemleri Çizelgede gösterildiği gibi günlük hayat içerisinde metin tercümesinden, sağlık sektörüne, eğitim sisteminden askeri sistemlere kadar birçok alanda kullanılmaktadır. Tüm modellerde girdi olarak sürekli konuşma sinyali alınmakta ve problemlere özel çıktılar üretilmektedir. Konuşma tanıma sistemlerinden tanıma performanslarının daha yükselmesi halinde çok daha yaygın kullanım alanlarına ulaşılacaktır.

4. DERİN SİNİR AĞLARI

Yapay sinir ağları doğrusal yapıda olmayan problemlerin çözümünde kullanılırlar. “X” girdisi ile “Y” çıktısı arasındaki ilişkiyi modellemeye çalışır. Sinir ağları tasarımında insan beyninin çalışma yapısında esinlenilmiştir [96]. Sinir ağlarının gelişmesi ve doğrusal olmayan sınıflandırıcıların kullanımı ile gerçek hayatta karşılaşılan problemlere çözüm üretilmiştir [97]. Sinir ağı yapısının model tasarımı ve kullandığı bellek yapısına göre farklı türleri bulunmaktadır. Bu çalışmada konuşma tanıma problemlerinde daha başarılı sonuçların alınabildiği derin sinir ağı kullanılmıştır.

4.1. Derin Sinir Ağlarının Yapısı

Derin sinir ağları, hiyerarşik mimarideki birçok katmandan oluşan bilgi işlem aşamalarının yapı sınıflandırması ve öznitelik veya temsil öğrenimi için kullandığı bir makine öğrenmesi tekniğidir [98]. Çok katmanlı makine öğrenme teknikleri kullanılarak verilerden gözetimli ve gözetimsiz anlam çıkarma işlemidir [99]. Bu ağ yapısı, yapay sinir ağlarının gelişiminde önemli bir adım olmuştur. Yapay zekâ, grafik modelleme, uyumlaştırma, model tanıma, işaret işleme gibi alanlarda yaygın olarak kullanılmaktadır. Şekil 3.1’de giriş katmanı, gizli katmanlar ve çıkış katmanından oluşan derin sinir ağı gösterilmiştir.



Şekil 4.1. Derin sinir ağı temel şeması

4.2. Derin Sinir Ağlarının Konuşma Tanımada Kullanımı

Konuşma tanıma ile ilgili olarak günümüze kadar yapılan birçok çalışmada SMM ve Gauss karışım modeli (GKM) kullanılmaktadır. SMM içinde her bir durumun olasılık dağılım fonksiyonu olarak GKM tercih edilir. Her bir ses örneği penceresinde hedeflenen duruma ne kadar benzediğinin gösterimi Gauss karışımları ile ifade edilir. Doğrusal olmayan verilerin modellemesinde olasılık tabanlı bu yaklaşımlar uygun değildir. Bir konuşma penceresi GKM ile modellenmek istenildiğinde yüzlerce parametreli bir modele ihtiyaç duyulmaktadır. Ancak MFKK tabanlı öznitelik çıkarma fonksiyonları ile elde edilen özniteliklerin derin sinir ağlarında kullanımı ile ses tanıma uygulamalarında başarılı sonuçlar elde edilebilmiştir. OKT sistemlerini performansı modelleme kapasitesinin artışı ve işlem gücü yüksek makinelerin yaygın kullanımı ile artırılabilmektedir. Ayrıca, SMM'in aksine derin sinir ağlarında MFKK için 39 adet özellik birçok problem için yeterli olabilmektedir.

Son yıllarda, bilgisayar sistemlerinin ucuzlaması, grafik işlemci ünite mimarilerinin gelişmesine bağlı olarak paralel işlem yapabilme kapasitelerinin artışı ile derin sinir ağlarının eğitimi daha kolay ve hızlı gerçekleştirilebilir olmuştur. Bu sayede çok hızlı matris işlemleri yapılabilmekte ve çok katmanlı yapıların eğitimi mümkün olmaktadır. Bu sayede başarılı sonuçların üretilebilir olması nedeniyle OKT sistemleri üzerine birçok çalışma yapılmaktadır [100-103]. Derin sinir ağlarının kullanıldığı çalışmalarda geleneksel sinir ağı ve SMM yapılarına göre çok daha başarılı sınıflandırma performansları yakalanmıştır.

4.3. Derin Sinir Ağlarının Eğitimi

Sinir ağlarında her bir katman ağırlık matrisi ve Bayes vektörü ile ifade edilir. Katmanlarda bulunan her bir düğüm kendine bir önceki katmandan gelen değerleri ağırlık matrisi ile çarparak ağırlıklandırır ve Bayes vektörü ile toplayarak bir sonraki katmana iletir. Böylece eğitim parametrelerinin güncellenmesi ve doğru çıktı elde edilene kadar bu işlemin devam etmesi sağlanır. Sonuçta model eğitimi tamamlandığında, girdiler çoğu için en iyi sonucu üreten çıktıların alınabileceği ağırlıkların elde edilmesi sağlanmış olur [104].

Derin sinir ağları, girişleri ve çıkışları arasında birden fazla gizli katmana sahip bir tür ileri beslemeli yapay sinir ağıdır. Her bir gizli birim (j)'in görevi kendisinden bir önceki katmandan gelen "x_j" değerini lojistik fonksiyondan geçirmek ve "y_j" sonucunu bir sonraki

katmana iletmektir. Bu işleme ilişkin matematiksel model Eşitlik 4.1'de verilmiştir. Lojistik fonksiyon, hiperbolik tanjant olarak sıklıkla kullanılmakla birlikte, türevi alınabilir tüm fonksiyonlar bu işlem için kullanılabilmektedir.

$$y_j = logistic(x_j) = \frac{1}{1+e^{-x}}, \quad x_j = b_j + \sum_i y_i w_{i,j} \tag{4.1}$$

"b_j", gizli katmanda bulunan "j" ünitesinin "bias" değeri olup, "w_{ij}" bir önceki (i) katmanından gelen ağırlığı ifade eder. Birden fazla sınıf bulunan problemlerde son katmandaki düğümün değeri Eşitlik 4.2'de gösterilen "Softmax" fonksiyonu ile normalize edilir ve sonuçlar anlamlı hale getirilir.

$$p_j = \frac{\exp(x_j)}{\sum_k \exp(x_k)} \tag{4.2}$$

Derin sinir ağları, hedef çıktılar ve her bir eğitim durumu için üretilen gerçek çıktılar arasındaki tutarsızlığı ölçen bir maliyet fonksiyonunun geri yayılan türevleri ile ayırt edici bir şekilde eğitilebilirler. Maliyet, olması beklenen/gereken çıktı ile ağın tahmin ettiği çıktı arasındaki farkı ifade eder. Eşitlik 4.3'de verilmiş olan maliyet fonksiyonunda, "C", hedef olasılıklar ile "softmax" fonksiyonunun ürettiği çıktı arasındaki çapraz entropi değeridir. Buradaki hedef olasılıklar, tipik olarak bir veya sıfırdır ve derin sinir ağını eğitmek için kullanılan bilgilerdir.

$$C = \sum_j d_j \log p_j \tag{4.3}$$

Büyük eğitim setleri için, ağırlıkları gradyan ile orantılı olarak güncellemeden önce, türev hesaplamalarında tüm veri yerine rastgele seçilmiş küçük parçaları kullanmak daha etkin bir yoldur. Eşitlik 4.4'de bu işleme ilişkin matematiksel model verilmiştir. "w_{ij}" değeri ağırlık matrisini, "ε" öğrenme katsayısını temsil etmektedir. Bu durumda maliyet fonksiyonu küçük bir veri üzerinden hesaplanır ve bu işlemin sonucuna göre girdi ağırlık değerlerinde güncelleme yapılır. Bu güncelleme değerinin adım büyüklüğüne öğrenme katsayısı ile karar verilir. Bu katsayı kullanılan optimizasyon tekniğine göre değişkenlik göstermektedir [104].

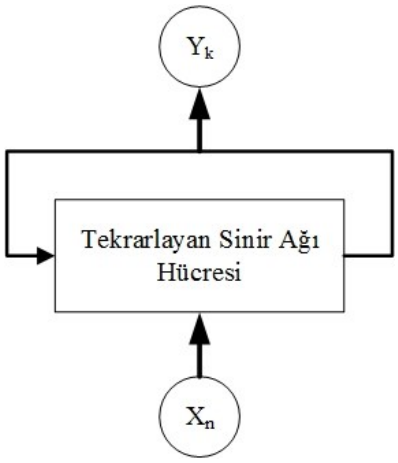
$$\Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) - \varepsilon \frac{\partial C}{\partial w_{ij}(t)} \tag{4.4}$$

Eğitim sürecinde yerel minimum veya eyer noktalarına takılmadan mutlak minimum değerlerine hızlı ve ezberleme olmaksızın ulaşmak oldukça önemlidir. Bu süreçte en önemli noktalardan birisi ağırlık matrisi ve standart sapma parametreleri için belirlenen başlangıç değerleridir. Bu değerlerin başlangıçta rastgele olarak seçilmesi ve eğitim sürecinde en iyi değerlere ulaşması mümkündür. Ancak bu alanda yapılan çalışmalar ile önerilen bazı yaklaşımlar sayesinde başlangıç değerlerin belirli sayılar olarak belirlenmesi mümkün olabilmektedir. Örnek olarak, Keras kütüphanesinin [105] ağırlıkların başlangıç değerlerine ilişkin olarak, sıfır değeri ile başlatma (Zeros), sabit bir değer ile başlatma (Constant), uniform dağılımı (RandomUniform) ile belirleme veya ortagonal matris kullanımı gibi farklı yaklaşımlarını bulunmaktadır [104]. Bu yöntemlerin kullanımı sayesinde ağırlık değerleri daha doğru bir noktadan başlayarak hızlı bir şekilde mutlak minimum değerine ulaşmış olacaktır.

4.4. Tekrarlayan Sinir Ağları

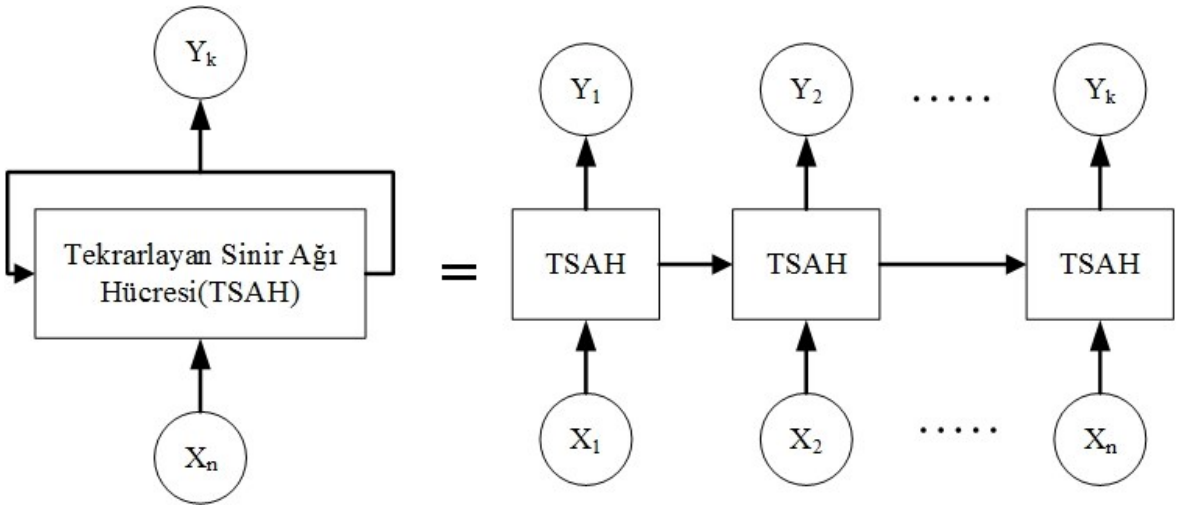
Sinir ağı yapılarında insanların gerçek hayattaki öğrenme sistematiği takip edilmektedir. İnsanlar, bir kavramı öğrendikten sonra, onu daha sonra öğreneceği yeni kavramlar için kullanırlar. Sürekli eski bilginin üst üste konulması şeklinde bir düşünme yapısı vardır. Geleneksel sinir ağlarında bu birikimli öğrenme düzeni gerçekleştirilemez. Özellikle konuşma tanıma gibi problemlerin çözümünde geleneksel sinir ağlarının kullanılamamasının en önemli sebebi bu eksikliktir. Örnek olarak, konuşma tanıma uygulamalarında bir sonraki sesin tanınmasında, bir önceki ses oldukça önemlidir. Her bir sesin bağımsız olarak değerlendirilmesi hatalı sonuçlara sebep olacaktır. Geleneksel sinir ağlarında bu şekilde bir bağımlılık modellenemediği için, bu soruna çözüm üretmek üzere Tekrarlayan sinir ağları (TSA) yapısı önerilmiştir. Bu ağ yapısı öğrenme aşamasında, eski bilginin sonraki adımlarda kullanılabilmesine imkân vermektedir [1].

Şekil 4.2'de tekrarlayan sinir ağının döngüye izin veren yapısı gösterilmektedir. "X_n" değerini girdi olarak alırken, "Y_k" değeri çıktı olarak üretmektedir. TSA hücresinin çıkışında üretilen bilgi tekrar bir sonraki adımda girdi olarak kullanılmaktadır. Böylece önceki bilgilerin sonraki adımlara aktarılması sağlanmış olmaktadır.



Şekil 4.2. Tekrarlayan sinir ağlarında döngü yapısı [106]

Döngü yapısı tekrarlayan sinir ağlarının karmaşık görünmesini sağlar. Ancak Şekil 4.3'te açık halinde görüleceği üzere geleneksel sinir ağından farklı değildir. Geleneksel sinir ağlarının teorik olarak sonsuz sayıdaki kopyasından oluşur. Listeler ve diziler gibi düşünülebilir. Tekrarlayan sinir ağları konuşma tanıma, dil modelleme [107-110], makine tercümesi [111-114] veya görüntü işleme [115-119] gibi birçok alanda başarılı sonuçlar alınarak kullanılmaktadır.



Şekil 4.3. Tekrarlayan sinir ağı yapısının açık hali [106]

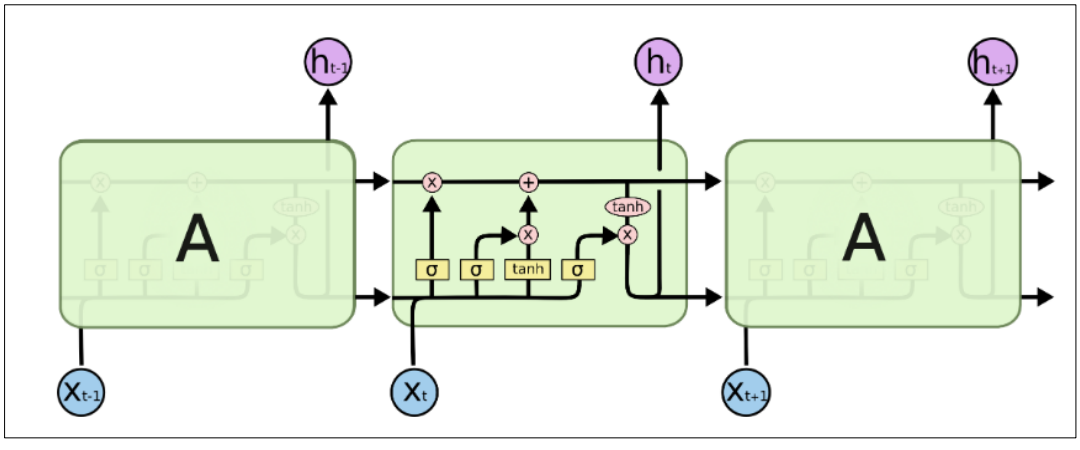
Tekrarlayan sinir ağları ile önceki bilgilerin yeni bilgilerin üretilmesinde kullanılmasına imkân tanımaktadır. Ancak bir hücreden önce elde edilmiş verilerden hepsinin değil de gerekli olanların alınmasına yönelik seçim yapılması gerektiğinde TSA yapıları bu işleme izin vermemektedir. Eğer yakın bir geçmişe ait bilgi ile yeni tahminler başarılı olarak

yapılabiliyor ise standart TSA'lar bu görev için uygundur. Ancak eğer çok daha önceki bilgilere ihtiyaç duyulan problemler ile çalışılıyor ise, standart TSA'lar bu aradaki farkın büyümesine bağlı olarak uzun süreli bağımlılığa çözüm üretememektedir [120]. TSA yapısındaki bu iki probleme çözüm üretmek üzere bir tür bellek yapısı olan Uzun kısa süreli bellek(UKSB) ve Geçitli Tekrarlayan Birim(GTB) yapıları önerilmiştir [106]. Bu bellek yapıları kullanılarak önceki bilgilerden seçimler yapılmakta ve gerekli olanlar bir sonraki aşamaya iletilmektedir. Bu yapı ile ses tanıma uygulamalarında oldukça başarılı sonuçlar alınabilmesi mümkün olmaktadır.

4.5. Uzun Kısa Süreli Bellek (UKSB) Yapısı

Uzun vadeli bağımlılıkları modelleyebilen tekrarlayan sinir ağı türüdür. Amacı bilgiyi uzun süre hafızada tutmaktır. Öğrenilecek bilgiye karar verip ağ eğitimini yapmak sinir ağının görevidir. Bu bellek yapısının kullanımı gün geçtikçe artmakta ve birçok problem için iyi sonuçlar alınmaktadır [121].

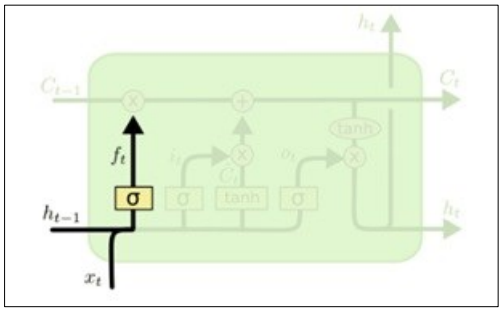
Standart bir TSA yapısında her bir katmanda bir "tanh" fonksiyonu bulunur. Yapısı oldukça basittir. Ancak UKSB yapısı zincir şekilde kurgulanmış ve tekrarlı bir modüller dizisidir. Şekil 4.4'te gösterildiği gibi tek bir katman yerine 4 kapı içeren özel bir yapısı vardır. Bu kapılar yardımıyla hücre durumunda gerekli değişiklikler yapılır.



Şekil 4.4. Dört etkin katman barındıran UKSB yapısı [122]

Her satır bir düğümün çıktısını diğerinin girişine kadar taşıyan vektörü taşır. Pembe durumlar vektörlerin matematiksel işlemlerini ifade eder. Sarı kutular öğrenen sinir ağı

katmanlarını ve çatal şeklindeki oklar verinin kopyalandığı ve farklı konumlara iletildiğini gösterir. UKSB hücresi bir taşıma bandı gibi çalışır ve döngü boyunca bu işlevini yerine getirir. Akış boyunca gerekli olan güncellemeler, veri ekleme ve çıkarma işlemleri kapılar vasıtası ile yapılır. Sigmoid fonksiyonu yardımıyla hangi bilginin ne kadarlık miktarının geçirilip geçirilmeyeceğine karar verilir. 0-1 aralığında üretilen değerlere göre 0 hiçbir verinin iletimine izin verme 1 ise tamamına izin ver anlamına gelmektedir. UKSB yapısında bu kapılardan üç adet bulunur.

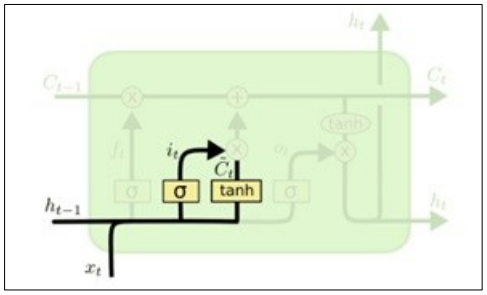


Şekil 4.5. Uzun kısa süreli bellek çalışma sistemi birinci işlem adımı [122]

$$f_t = \sigma(W_f \,.[h_{t-1}\,, x_t]) + \, b_f \tag{4.5}$$

Eşitlik 4.5'te "f_t" değişkeni, verinin unutulmasını yöneten kapının değerini, "h_{t-1}", t-1 zamanındaki UKSB bloğunun yani bir önceki bloğun çıktı değerini, "x_t", şimdiki zamandaki girdi değerini, "b_f" unut kapısının bias değeri, "W_f", f kapısının nöronları için gerekli ağırlık matrisini, "σ", sigmoid fonksiyonunu göstermektedir. UKSB çalışma sisteminde ilk adım Şekil 4.5'te gösterildiği gibi hangi bilgilerin bellekten atılacağına karar verilmesidir. Verinin unutulması gerektiğine karar verme işlemi sigmoid fonksiyonu kullanılarak yapılır "X_t" ve "h_{t-1}" değeri girdi olarak alınır ve her bir "C_{t-1}" durumu için 0 ile 1 arasında bir değer üretilir. 0 verinin tamamen atılacağını 1 ise verinin tamamen korunacağını ifade eder. Bu karar eski verinin bellekte tutulup tutulmayacağı kararını ifade eder.

Eşitlik 4.6 ve 4.7'de, i_t, girdi kapısını, W_i, i kapısının nöronları için ağırlık matrisini, h_{t-1} bir önceki katmanın çıkış değerini, x_t, şimdiki zamandaki girdi değerini, b_i i kapısının bias değerini, σ, sigmoid fonksiyonunu, C~, t zamanındaki hücre durumunu güncellemek üzere hazırlanan aday değeri ifade etmektedir.

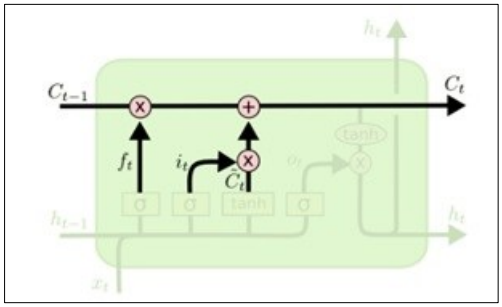


Şekil 4.6. Uzun kısa süreli bellek çalışma sistemi ikinci işlem adımı [122]

$$i_t = \sigma\,(W_i\,.[h_{t-1}, x_t]) + b_i \tag{4.6}$$

$$C\sim_t = \tanh(W_C\,.[h_{t-1}, x_t]) + b_C \tag{4.7}$$

UKSB yapısının ikinci adımı Şekil 4.6'da gösterilmiştir. Bu adımda hangi yeni bilginin belleğe kaydedileceğine karar verilmektedir. Sigmoid katmanı ile hangi değerlerin güncelleneceğine karar verilir. Sonrasında Eşitlik 4.7'de gösterildiği gibi "tanh" katmanı ile kaydedilebilecek yeni değerler vektörü oluşturulur(C~). Hangi değer yerine hangi yeni değerin koyulması gerektiğine karar verilmesi aşamasıdır. Bu aşama sonrasında artık ihtiyaç duyulmayan verilerin bellekten atılması gerekmektedir. Bu işlem Şekil 4.7'de gösterilen kapı yardımıyla yapılır.

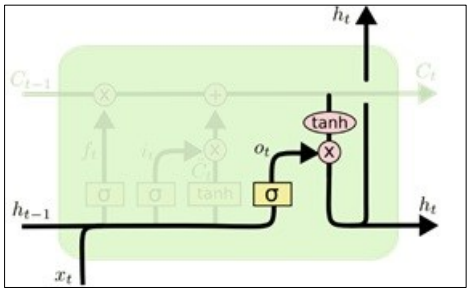


Şekil 4.7. Uzun kısa süreli bellek çalışma sistemi üçüncü işlem adımı [122]

$$C_t = f_t * C_{t-1} + i_t * C\sim_t \tag{4.8}$$

Eşitlik 4.8'de C_t, t zamanındaki hücre durumunu (belleğin durumunu) ifade eder. Unutulmak istenilen değerler Eşitlik 4.8'de gösterildiği gibi "ft" değeri ile çarpılır. Sonrasında yeni "C~" değeri eklenir. Her bir yeni değer için, eski değerinin ne kadar güncelleneceğine dair ölçeklemenin yapıldığı aşamadır [122].

Son olarak, Şekil 4.8 de gösterildiği gibi hangi değerin çıkış değeri olacağına karar vermek gereklidir. Bu değer elbette hücrede bulunan değere göre şekillenecektir ancak ihtiyaç duyulan kısmı seçilecektir. Bir sigmoid katmanı çalıştırılarak hücre durumunun hangi kısımlarının çıktı olarak üretileceğine karar verilir. Sonrasında bir "tanh" fonksiyonu ile değerler -1 ile 1 arasına normalize edilir. Bu değer sigmoid fonksiyonu ile çarpılarak durumun istenilen parçası çıktı olarak üretilir. Böylece unutulması gereken veri unutulmuş ve bellekte tutulması istenen değer bellekte kalmış olur.



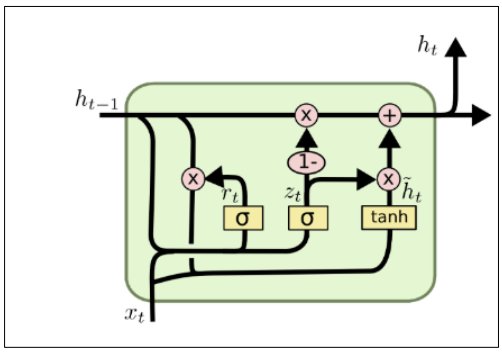
Şekil 4.8. Uzun kısa süreli bellek çalışma sistemi dördüncü işlem adımı [122]

$$o_t = \sigma\left(W_0\left[h_{t-1}, x_t\right] + b_0\right) \quad (4.9)$$

$$h_t = o_t * \tanh(C_t) \quad (4.10)$$

Eşitlik 4.9 ve 4.10'da o_t çıkış kapısını, h_{t-1} bir önceki bloğunun çıktısını, x_t, şimdiki zamandaki girdi değerini, b_o çıkış kapısının bias değerini, W_o, nöronlar için ağırlık matrisini, **σ**, sigmoid fonksiyonunu ifade eder.

Şekil 4.9'da gösterilmiş olan GTB [123] bellek yapısı, UKSB yapısına oldukça benzerdir. Bu yapının UKSB yapısına göre en önemli değişikliği hücre durumunun silinmesi işleminin ortadan kaldırılmasıdır. İkinci önemli değişiklikte, iki farklı kapı ile temsil edilen unut ve güncelle işleminin tek bir kapı ile birleştirilmesidir. Böylece daha hızlı işlem yapılması ve daha basit model ile çalışılması mümkün olmaktadır. Bu çalışmada hem UKSB yapısı hem de GTB bellek yapısına sahip yapılar ile testler gerçekleştirilmiştir.



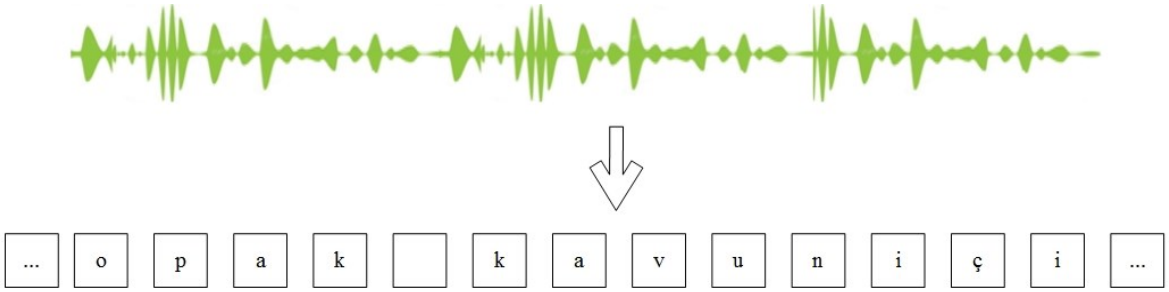
Şekil 4.9. Geçitli tekrarlayan birim (GTB) şeması [123]

UKSB kullanılan tekrarlayan sinir ağı modelleri ile daha iyi sonuçlar üretmek mümkündür. Bu bellek yapısının kullanıldığı durumlarda bu bölümde anlatıldığı gibi karmaşık görünen bir eşitlik dizisi kullanılmaktadır. Bu yapının anlaşılmasından sonra bellek içerisinde gün geçtikçe daha fazla bilginin tutulması sayesinde daha karmaşık problemlere daha başarılı çözümler üretilebilecektir.

4.6. Bağlantıcı Zamansal Sınıflandırma (BZS)

Bir konuşma tanıma uygulaması için ses dosyası ve buna karşılık gelen metin çıktısına sahip olunmalıdır. Ancak sesin hangi bölümünün metnin hangi bölümüne karşılık geldiğinin de belirlenmesi gereklidir. Bu işleme hizalama denilmektedir. Bu hizalama işlemine çözüm üretmek için kullanılabilecek ilk yöntem sabit bir değer kabul etmektir. Örnek olarak 1 karakter 10 adet ses girdisine karşılık gelir şeklindedir. Ancak insanların konuşma hızları ve akışkanlıkları değişkenlik gösterir ve böyle bir sabit değer kabul etmenin eşleşmeyi sağlayabilmesi mümkün görünmemektedir. İkinci yöntem ise, her bir karakterin sesteki konumunu insan yardımıyla hizalamaktır. Normal şartlarda bu iyi sonuçlar vermektedir ve küçük veri seti ile çalışılan birçok uygulamada tercih edilmektedir. Ancak veri setinin büyüdüğü durumlar bu yöntem oldukça zaman alıcıdır ve uygulanabilir değildir.

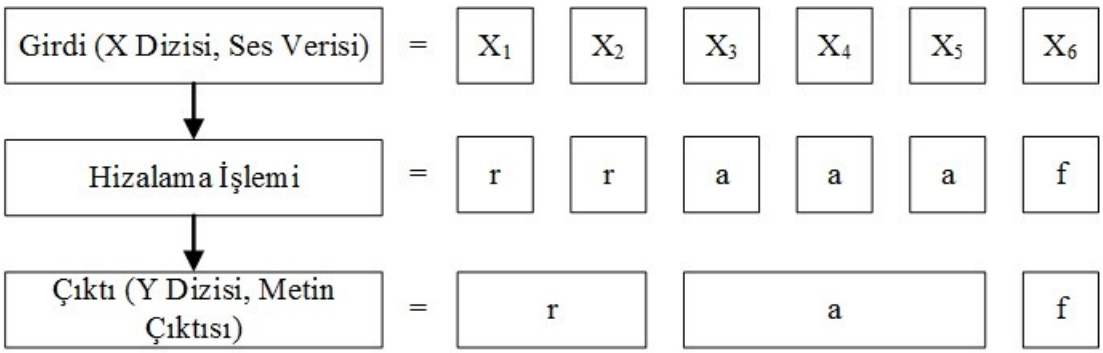
Herhangi bir ses sinyali ve buna karşılık gelen metin çıktısı Şekil 4.10'da örnek olarak gösterilmiştir. Buna göre bu iki verinin hizalanması ve sesin hangi bölümünün hangi karaktere karşılık geldiğinin belirlenmesi gereklidir.



Şekil 4.10. Konuşma tanıma: girdi değeri bir spektrogram veya frekans boyutunda herhangi bir öznitelik vektörü

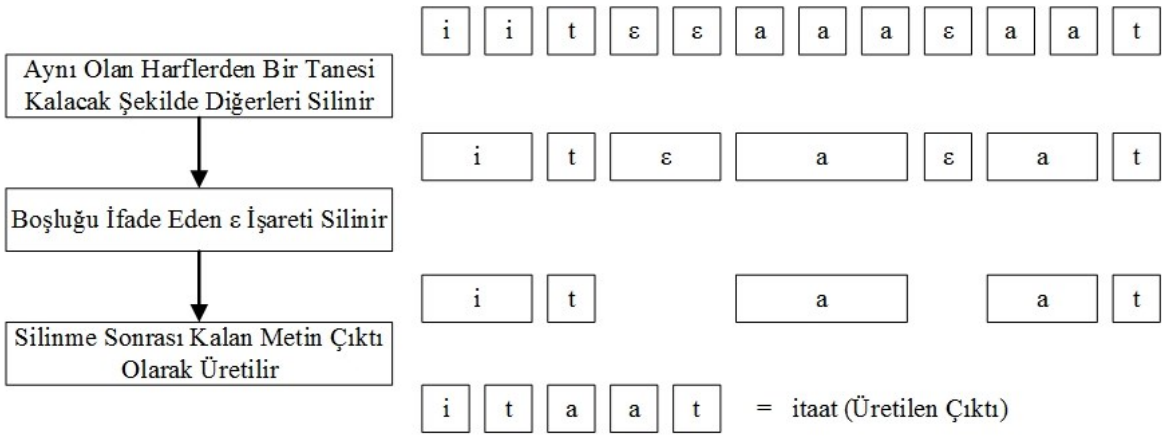
Bağlantıcı zamansal sınıflandırma (BZS) sabit değer kabul etme veya elle hizalama yöntemlerinden farklı olarak bu işlemi otomatik olarak gerçekleştirmeye imkan veren bir algoritmadır. Konuşma tanıma uygulamaları için oldukça uygundur [124]. $X = [X_1, X_2, \dots, X_T]$ ses işareti girdi olarak ve $Y = [Y_1, Y_2, \dots, Y_U]$ da bu sesin metne dönüştürülmüş hali olarak kullanıldığında X dizisinden Y dizisine bir haritalama işlemi yapmak gerekmektedir. Bu eşleşme sırasında, X ve Y dizilerinin uzunluklarının değişken olması, uzunluk oranlarının değişken olması ve en doğru hizalama diye bir kavramın olmaması en önemli sorunlardır. BZS algoritması bu sorunlara çözüm üreten bir yaklaşımdır. Bu algoritmaya göre belirli bir X ile tüm olası Y değerleri için dağılım yapılır. Bu dağılım ile en iyi çıktıyı belirlemek için olasılıksal değerlendirmeler ile tüm alternatifler belirlenir. Bu alternatifler içerisinde en başarılı olan seçilerek hizalama işlemi tamamlanır [9].

Altı birim uzunluğunda girdi dizisi X için çıktı dizisinin Y = [r, a, f] olduğunu varsayalım. Bu iki girdiyi hizalamanın en basit yöntemi Şekil 4.11'de gösterildiği gibi tekrar eden karakterlerin atılmasıdır [125].



Şekil 4.11. Tekrarlayan harfleri atarak hizalama yapma yöntemi örneği [9, 125]

Bu yaklaşım basit olmak ile birlikte, iki sorunu bulunmaktadır. Birincisi, her bir girdi adımının bir çıktıya hizalanması zorunluluğu yanlış olabilir. Çünkü sessizlik derecesine sahip bir girdi sebebiyle metin çıktısı üretilmiyor olabilmektedir. İkincisi ve en önemlisi de bu durumda arka arkaya aynı karaktere sahip hiçbir çıktı üretilemeyecektir. Kaan, hokka, cennet vb. hiçbir kelime düzgün olarak üretilemez. Bu soruna çözüm üretmek için çıktı olarak sadece harf yerine, boş bir karakteri ifade eden "ε" işareti kullanılabilir. Bu işarete sahip bölümler çıktıya dâhil edilmez ve Şekil 4.12'de gösterildiği gibi otomatik olarak silinir.

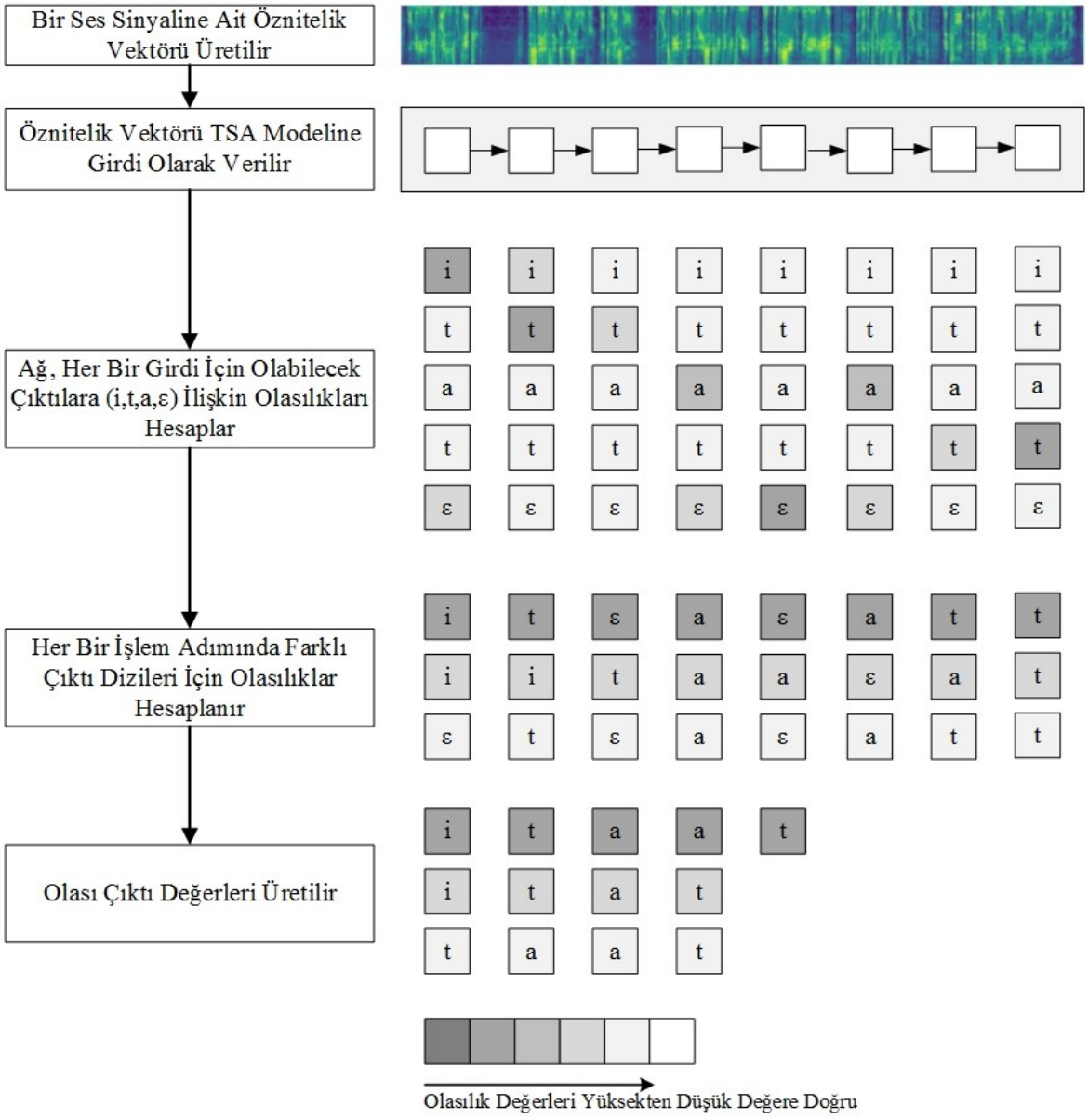


Şekil 4.12. "ε" işareti kullanılan hizalama yöntemi örneği [9, 125]

Şekil 4.12'de gösterildiği gibi eğer birbirini takip eden iki aynı karakter bulunuyor ise, aralarında bir adet "ε" işareti bulunması gereklidir. Böylece "itaat" ile "itat" kelimesi arasında fark oluşturulmuş olur. Şekil 4.13'te örnek bir girdi için çıktının üretimine dair olasılıksal hesaplamaları ve işlemler adımlarını gösterir şema verilmiştir [9, 125].

Şema incelendiğinde, öncelikle ses sinyalinden öznitelik parametreleri üretilir ve vektörler oluşturulur. Bu vektörlerdeki parametreler kullanılarak TSA modelleri eğitilir. Bu eğitim sonrasında ağ her bir girdi için olası çıktılara ilişkin (i,t,a,ε) olasılıkları hesaplar. Her bir işlem adımında bu süreç tekrarlanır. Şekilde daha koyu renkli olarak gösterilen kutularda sesin o bölümüne karşılık gelen karakter için en iyi olasılığa sahip değeri göstermektedir. Bu hesaplamalar tamamlandıktan sonra olasılık değerleri üzerinden sesin belirli bir bölümü için hangi karakterlerin çıktı olabileceğine dair değerlendirme yapılarak doğru kelimenin tahmin edilmesi sağlanır. Hizalama işleminin otomatik olarak yapılması için olası çıktılar üzerinde olasılık değerlendirmesi yapılır ve en yüksek olasılığa sahip satır çıktı olarak

üretilir. Elle hizalama yöntemine göre daha hızlı, tekrarlı karakterlerin atılması yöntemine göre daha doğru sonuçlar üretmektedir. Tekrarlayan sinir ağları ile birlikte kullanılmaktadır.



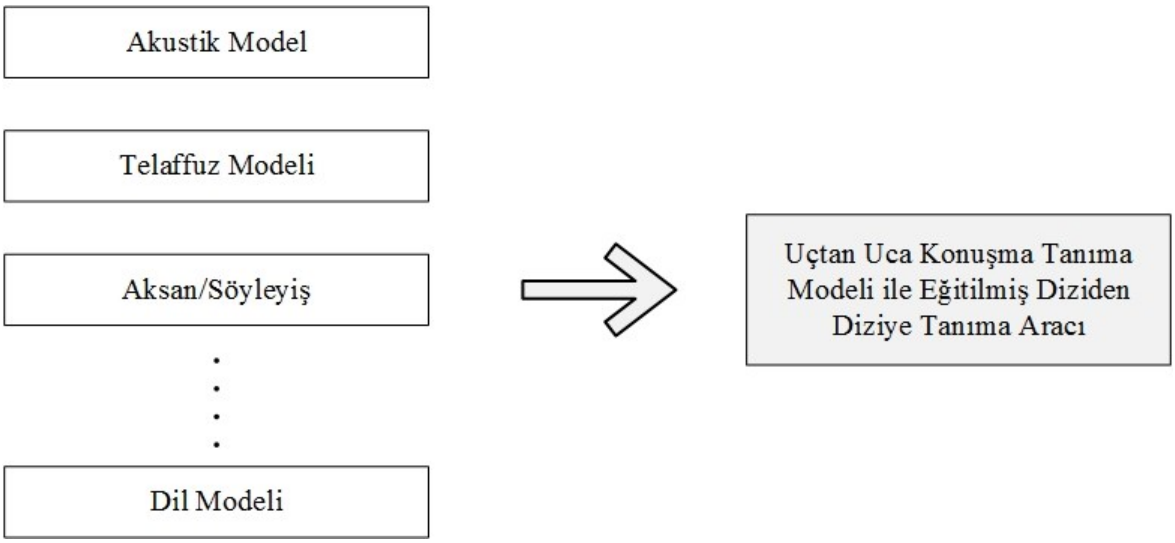
Şekil 4.13. "itaat" kelimesi için olası hizalamalar şeması [9, 125]

BZS fonksiyonun farklı şekillerinin kullanıldığı birçok uygulama örneği bulunmaktadır. Bu örneklerde genelde başarılı sonuçlar elde edilebilmiştir [77, 126-130]. Ancak, bu çalışmalarda akustik model tek başına eğitildiği için öğrenme tam olarak gerçekleşmemektedir ve güçlü bir dil modeline ihtiyaç duyulmaktadır. Bunun nedeni ise, BZS fonksiyonu girdiler arasındaki bağımsızlığı varsayımlar üzerinden değerlendirmekte ve uzun bağımlılıkları modelleyememektedir [131]. Buna çözüm olarak akustik verilerin dil

modeli ile birlikte eğitildiği uçtan uca sistemler önerilmektedir. Ayrı ayrı eğitimin yapılmadığı bu yapıda hem sistem karmaşıklığı azaltılmakta hem de BZS fonksiyonun dezavantajlı olduğu durumlar ortadan kaldırılmaktadır.

4.7. Uçtan Uca Konuşma Tanıma Modeli

Uçtan uca konuşma tanıma modellerinde ses doğrudan karakterlere dönüştürülür. Diziden diziye bir dönüşüm sistemi kullanılır [132]. Bu yapıda, dil modeli ve akustik model ayrı ayrı olmayıp tek bir model içerisinde öznitelikler üzerinden öğrenilmektedir. Çıktıdaki karakterler ile sinyal girdisi arasında bağımsızlık olabileceği değerlendirilmez. Girdi dizisini doğrudan çıktıya dönüştürebilen bir tür derin öğrenme yaklaşımıdır [133-136].



Şekil 4.14. Uçtan uca konuşma tanıma yaklaşımı [137]

Doğrudan kelimeleri veya grafikleri üretebilen Şekil 4.14'deki gibi bir uçtan uca eğitimli model, bir konuşma dizisini metin dizisine doğrudan çevirebilmekte ve bu sistemi oldukça basitleştirmektedir [137].

Uçtan uca konuşma tanıma modellerinin oluşturulmasına ilişkin olarak son yıllarda ciddi çalışmalar yapılmaktadır. Bu örnek çalışmalarda BZS metodu tercih edilmektedir. Girdi vektörü olarak MFKK benzeri özellik çıkarma metotları kullanılarak oluşturulmuş değişken uzunluktaki diziler kullanılmakta ve çıktı olarak yine değişken uzunlukta (metin, karakter,

fonem vb.) diziler üretilmektedir. Bu üretim sonrasında performans artışları yakalayabilmek için dil modellemesinden de yararlanılmaktadır.

Graves ve arkadaşlarının 2006 yılında yaptıkları çalışmada [124] ilk defa uçtan uca model önerisi yapılmıştır. Model 100 gizli katman bulunduran iki yönlü uzun kısa süreli bellek (UKSB) yapısına sahip bir derin sinir ağı içerir. Giriş katmanı boyutu 26 olup 12 MFKK özelliği içerir. Pencere boyutu 10 ms olup 5 ms kaydırma yapılmıştır. Sinyal, 26 adet filtre bankasından geçirilmiştir. Eğitim 0,0001 öğrenme katsayısı ile gerçekleştirilmiş olup optimizasyon tekniği olarak "Stochastic Gradient Descent" ve "Nesterov" birlikte kullanılmıştır. TIMIT [138] fonem veri seti ile yapılan testlerde %81'lik doğru tanıma oranı ile sınıflandırma yapılmıştır.

Maas modeli, ses işaretlerini fonemlere dönüştürmek yerine doğrudan karakterlere dönüştüren ilk başarılı çalışmalardan birisidir [139]. Switchboard [5] veri seti ile yapılan çalışmalarda başarılı sonuçlar elde edilmiştir. Akustik model üretimi için sinir ağlarının, dil modeli olarak karakter tabanlı bir yapının kullanıldığı ve "beam search" algoritması ile arama yapılan bir yapı önerilmiştir. Karakter tabanlı çalışılması sebebiyle bir kelime sözlüğüne ihtiyaç duyulmamıştır. Aktivasyon fonksiyonu olarak ReLU kullanılmıştır. MFKK özellik çıkarım metodu ile öznitelikler elde edilmiştir. Tam bağlı tekrarlayan sinir ağı kullanılmıştır. Çıkış karakter seti "-" veya kesme işareti gibi özel karakterler ile birlikte 33 karakterden oluşmaktadır. Modeller "Stokastic Gradient Descent" ve "Nesterov" optimizasyon teknikleri birlikte kullanılarak optimize edilmiş olup öğrenme katsayısı 0,00001, momentum değeri ise 0,95 olarak belirlenmiştir. Dil model üretimi için N-gram algoritmasından yararlanılmıştır. Model üretimi için yaklaşık 31 milyon kelime içeren bir kelime sözlüğünden kullanılmıştır. Bu işlemler sonucunda %73,3 performans oranı ile sınıflandırma yapılmıştır.

EESEN modeli [140] ile ses tanıma için ağırlıklı sonlu durum dönüştürücüleri kullanılmıştır. BZS ile kod çözümlemesi sürecine kelimelerin veya dil modellerinin dâhil edilmesi yoluna gidilmiştir. Kelime hata oranlarında, hibrit derin öğrenme sistemlerine benzer sonuçlar elde edilmiştir. Paket büyüklüğü 10 örnek olacak şekilde seçilmiştir. Öğrenme katsayısı $4*10^{-5}$ olarak belirlenmiştir. Diğer parametreler kelime hata oranlarını düşürecek şekilde kademeli olarak değiştirilmiştir. Çalışma sonucunda, dil modeli olmaksızın sistem tanıma performansı %52,9 seviyesinde kalmıştır. 7-gram karakter tabanlı dil modeli kullanıldığı durumlarda

performans seviyesi %64,1 oranına çıkmıştır. Aynı veri için 3 katmanlı tekrarlayan sinir ağı ile testler gerçekleştirildiğinde %69,2 tanıma performansı yakalanmıştır. Uçtan uca tasarlanmış olan bu model diğer çalışmalara yakın sonuçlar elde edilmiştir.

Baidu araştırmalarında ise karmaşık bir gürültü sinyali, konuşma sesine ilave edilmiş ve büyük veri kümesi kullanılarak model oluşturulmuştur [141]. Bu şekilde kelime dağarcığının oldukça büyük seçilmesi sebebiyle diğer çalışmalardan ayrılmaktadır. Bu araştırmaların ilki olan "Deep Speech-1", "Mass" modelinden esinlenerek oluşturulmuştur. Ağın üretiminde optimizasyon tekniği olarak "Stokastic Gradient Descent" ile "Nesterov" birlikte kullanılmıştır ve momentum değeri 0,99 olarak seçilmiştir. Pencere boyutu 20 ms olarak belirlenirken, 10 ms kaydırma yapılmıştır. Geliştirilen modelin üretmiş olduğu kelimelerdeki hataların fonetik yaklaşımlarla düzeltilebilmesinin mümkün olduğundan bahsedilmiştir. Model 495 bin farklı kelime içeren 220 milyon kelimeden oluşan N-gram dil modeli ile desteklenmiştir. Tekrarlayan sinir ağı kullanılarak modellenen bu sistemde çalışma zamanını kısaltmak için farklı yöntemler geliştirilmiştir. Çalışma sonucunda Switchboard veri seti ile çalışıldığı durumlarda %74,1, Switchboard ile Fisher veri setinin birlikte kullanıldığı durumlarda %84 tanıma performansına ulaşılmıştır. Geniş veri kümesi ile çalışılması sayesinde diğer çalışmalara göre daha başarılı sonuçlar elde edilmiştir.

4.8. Optimizasyon Teknikleri

Gradyan inişi (Gradient Descent) algoritması sinir ağlarında ağırlıkların güncellenmesinde kullanılan en yaygın optimizasyon yaklaşımlarından birisidir [142]. Bu nedenle caffe, tensorflow veya keras gibi yaygın kullanılan derin öğrenme kütüphanelerinde gradyan inişi yönteminin kullanılmasına yönelik altyapılar mevcuttur.

Gradyan inişi algoritması amaç fonksiyonunu maksimize ederek, her bir adımda üretilen çıktı parametrelerinin hedefe yaklaşabilmesi için gerekli güncellemelerin yapılması sürecinde kullanılır. Her bir adım için yapılacak güncelleme işleminin büyüklüğüne öğrenme katsayısı ile karar verilmektedir. Amaç üretilen çıktı değerinin hedeflenen değere yaklaşma yolculuğunda gidiş yolunun belirlenmesidir.

Gradyan inişi algoritması, maliyet fonksiyonu değerini mutlak minimum değerine indirebilmek için çeşitli yollar izleyebilmektedir. Bu ilerleme sürecinde bazı durumlarda

model yerel minimum veya eyer noktalarına takılabilmektedir. Bu sebeple, bu problemlere çözüm üreterek mutlak minimum değere ulaşabilmek için birçok optimizasyon tekniği kullanılabilmektedir. Bu bölümde gradyan inişi algoritmalarından bahsedilecektir.

4.8.1. Gradyan inişi ve momentum

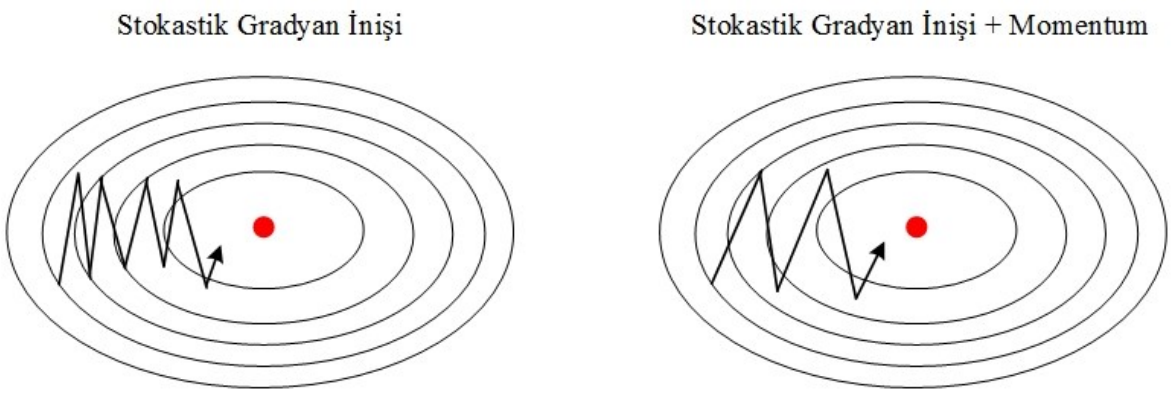
Stokastik Gradyan inişi (SGİ) algoritması, bir durumun mutlak minimum değerine ulaşabilmek için kullanılan yinelemeli bir optimizasyon tekniğidir. Momentum tekniği ise SGİ algoritmasının momentum değeri ile kontrol edilebildiği bir türüdür. Her iki algoritmanın matematiksel modelleri karşılaştırmalı olarak Eşitlik 4.11'de verilmiştir.

$$\begin{array}{cc} \text{SGİ} & \text{SGİ + Momentum} \\ \theta_j \leftarrow \theta_j - \in \nabla_{\theta_j} F(\theta) & v_{t+1} \leftarrow p v_t + \nabla_\theta F(\theta) \\ & \theta_j \leftarrow \theta_j - \in v_{t+1}) \end{array} \tag{4.11}$$

Eşitlik 4.11'de SGİ olarak verilmiş olan formülde , "θ" değeri problemde kullanılacak olan parametrelerden herhangi birisini ifade etmektedir. Ses tanıma için bu değer MFKK ile üretilen 13 özellikten birinin yerine kullanılır. Bu eşitlikte sadece bir parametre için gösterim yapılmıştır. Her bir parametre için bu hesaplamalar tekrarlı olarak yapılmalıdır. $\in$ öğrenme katsayısını, ∇ maliyet fonksiyonuna bakarak gradyan fark hesaplamasını, F maliyet/kayıp fonksiyonu ifade etmektedir. SGİ işleminde, her bir işlem adımında, orijinal θ değerinden, gradyan değerinin öğrenme katsayısı ile çarpımının çıkarılması ile hesaplama yapılır.

SGİ ve momentum değerinin birlikte kullanıldığı eşitlikte ise, p momentum değerini ve v_t son değişim değerini ifade etmektedir. Bu durumda, ağırlık güncellemesi momentum değeri kullanılarak yapılır. Güncellemelerde momentum değerinin kullanılması ile eğitim aşamasının daha hızlı gerçekleştirilmesi amaçlanmaktadır. Eğitim hızı p değeri ile ağırlıklandırılan gradyanların toplamıdır. Bu sebeple p değeri hızı kontrol eden bir parametre olarak düşünülmektedir. Bu sayede eyer noktaları ve yerel minimum değerleri daha az tehlikeli hale gelir. Çünkü mutlak minimum seviyesine ulaşım sürecinde sadece kayıp fonksiyonu değil, hız da etkili olmaktadır. Eğim yönünde ilerlemek yerine hız yönünde ilerleme tercih edilmektedir. Şekil 4.15'te görüleceği üzere momentum eklendiğinde zikzak

çizme sayısı azalmakta ve daha hızlı şekilde mutlak minimum değerine ulaşılmaktadır.



Şekil 4.15. Stokastik gradyan inişi ve momentum optimizasyon tekniğinin karşılaştırması [143]

4.8.2. Nesterov

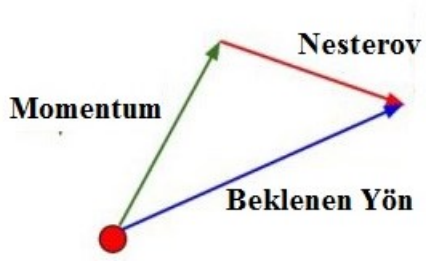
Nesterov momentum değerindeki bir değişikliği ifade etmektedir. Eşitlik 4.12'de gösterildiği gibi Gradyan hesaplamasında θ_j yerine $\theta_j + \gamma v_t$ kullanılmaktadır. γ değeri momentum değerini ifade etmektedir. Bu durum gradyan değerinin her zaman doğru yönde ilerlemesini sağlanmaktadır. Momentum değeri yanlış yönde ilerlerse, gradyan hala düzeltilebilir durumda bulunur. Standart momentumdan daha hızlı bir optimizasyon sağlamaktadır.

Nesterov

$$\theta = v_t$$

$$v_t = \gamma v_{t-1} + \nabla F(\theta - \gamma v_{t-1}) \quad (4.12)$$

Şekil 4.16'da gösterildiği gibi momentuma göre gradyan yönünden sapma olduğunda beklenen yönde ilerlemek için Nesterov ile düzeltme yapılabilmektedir. Bu düzeltici yapısı sebebiyle ses tanıma ile ilgili yapılmış çalışmalarda diğer optimizasyon teknikleriyle birlikte kullanılmıştır.



Şekil 4.16. Nesterov optimizasyon yönteminin momentuma katkısı

4.8.3. AdaGrad

AdaGrad yönteminde amaç optimizasyon sırasında gradyan değerlerinin karelerinin toplamını korumaktır. Bu durumda momentum yoktur ve gradyan kareleri toplamı ile ifade edilir. Eşitlik 4.13'te SGİ ve momentum ile AdaGrad yöntemi eşitlikleri karşılaştırmalı olarak gösterilmiştir.

SGİ + Momentum	AdaGrad	
$v_{t+1} = pv_t + \nabla_\theta F(\theta)$	$g_0 = 0$	
$\theta_j \leftarrow \theta_j - \in v_{t+1}$	$g_{t+1} \leftarrow g_t + \nabla_\theta F(\theta)^2$	(4.13)
	$\theta_j = \theta_j - \epsilon \frac{\nabla_\theta F}{\sqrt{g_{t+1} + 1e^{-5}}}$	

Eşitlik 4.13'te yeni eklenmiş olan "g" parametresi gradyan değerini, g_{t+1} gradyan değerlerinin üstel ortalamalarını ifade etmektedir. Ağırlık parametrelerini güncellemek için mevcut gradyan değerini "g" parametresinin karekköküne böleriz. İki yönlü bir uzayda bir kayıp fonksiyonunun gradyanının bir yönde çok yüksek diğer yönde ise çok küçük olduğu durumda, gradyanları doğrudan toplamak yerine karelerini toplamak daha uygun olacaktır. Çünkü güncelleme sırasında büyük olan değerler daha büyük kareler toplamına bölünür ve küçük olan değerlerde daha küçük olan kareler toplamına bölünür. Böylece küçük gradyan yönündeki yavaş hızla ilerleme sorunu çözülmüş olur. Ancak karelerini toplama işleminin sayısı arttığında değerlerin üstel şekilde hızlı yükselmesi sebebiyle gradyan değeri çok düşebilir ve öğrenme süreci çok yavaşlayabilir.

4.8.4. AdaDelta

Adadelta tekniği, AdaGrad'ın monoton olarak azalan öğrenme oranı sorununu çözmek üzere

önerilmiş bir optimizasyon algoritmasıdır. AdaGrad'da öğrenme oranı değerleri, gradyanların kareleri toplamının karekökü ne bölünmektedir. Bu işlem tüm öğrenme sürecinde baştan sona olacak şekilde yapılmaktadır. Ancak Adadelta yönteminde, tüm değerleri toplamak yerine kademeli olarak önceki değerleri düşüp, yeni değerlerin toplamının alınması şeklinde kaydırmalı bir yapı kullanılmaktadır. Bunun dışında algoritması AdaGrad ile aynı şekilde çalışmaktadır.

4.8.5. RMSProp

Adagrad yönteminin sahip olduğu kareler toplamının zamanla büyümesi problemine çözüm üreten bir diğer optimizasyon algoritmasıdır. Burada gerçekte toplamda bir azalma sağlanır. Eşitlik 4.14'te AdaGrad ile RMSProp optimizasyon teknikleri karşılaştırmalı olarak gösterilmiştir.

AdaGrad	RMSProp	
$g_0 = 0$	$g_0 = 0, \propto \cong 0{,}9$	
$g_{t+1} \leftarrow g_t + \nabla_\theta F(\theta)^2$	$g_{t+1} \leftarrow \propto . g_t + (1-\propto)\nabla_\theta F(\theta)^2$	(4.14)
$\theta_j = \theta_j - \epsilon \frac{\nabla_\theta F}{\sqrt{g_{t+1} + 1e^{-5}}}$	$\theta_j \leftarrow \in \frac{\nabla_\theta F}{\sqrt{g_{t+1}} + 1e^{-5}}$	

Eşitlik 4.14'te "$\propto$" başlangıçtaki öğrenme katsayısını g_{t+1} gradyan değerlerinin üstel ortalamasını, g_t gradyan değerini ifade etmektedir. RMSProp tekniğinde, gradyanların kareleri toplamı bir $\propto$ değeri ile çarpılır ve mevcut gradyan değerini de 1-$\propto$ ağırlığı ile çarparak ekler. Bunun dışında iki yönlü güncelleme yöntemi AdaGrad ile aynıdır. SGİ algoritması daha hızlı bir şekilde mutlak minimuma ulaşabilse de, oldukça uzun bir yol izler. Bu durumda da eyer noktalarına ya da yerel minimum değerlerine takılması olasıdır. Ancak RMSProp tekniğinde, en kısa yoldan mutlak minimum değerine ulaşılmaya çalışılır. Böylece eyer noktasına veya yerel minimum noktalarına takılma olasılığı azaltılmış olur. Bu sebeple SGİ yöntemine göre de avantajlı bir yöntemdir.

4.8.6. Adam

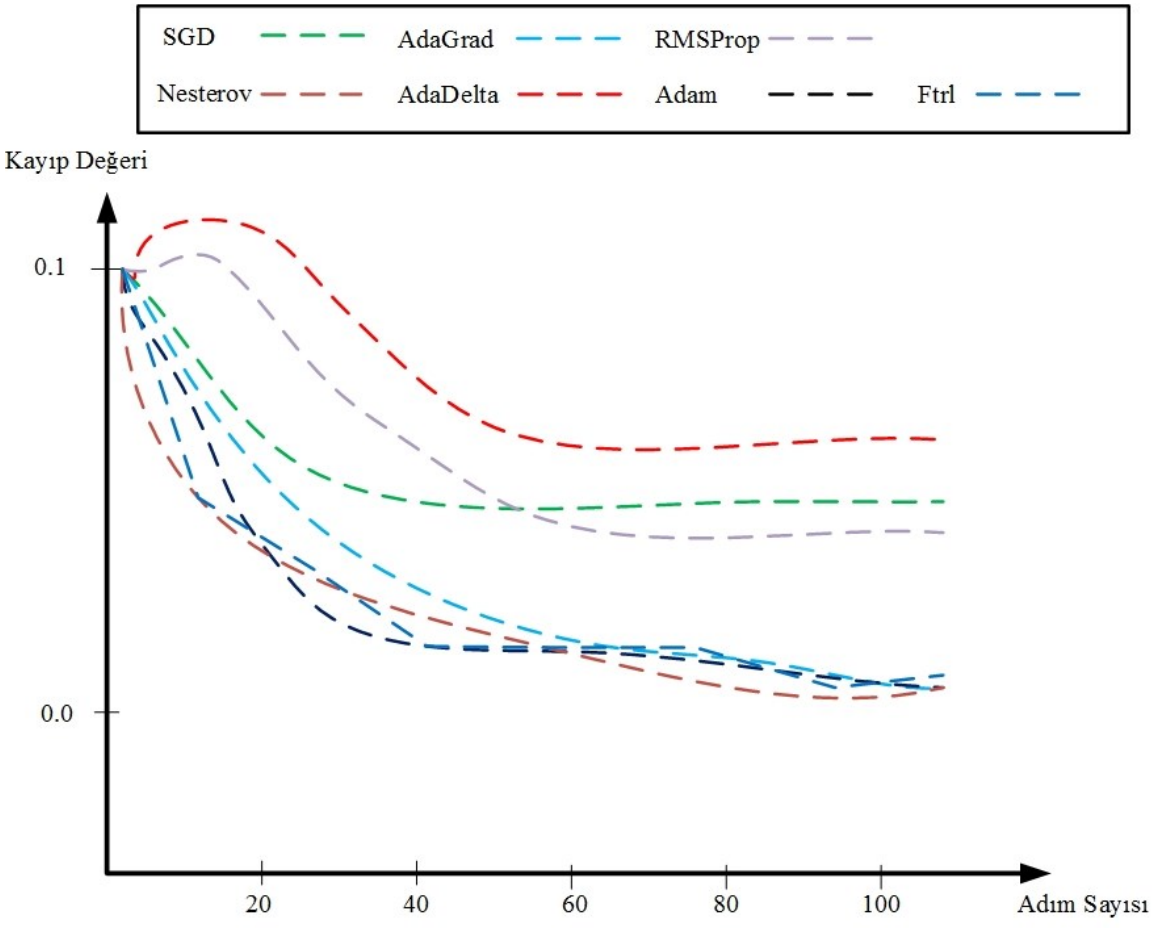
Her bir parametre için uyarlamalı öğrenme oranını belirleyen ve buna göre gradyanların birinci ve ikinci andaki tahminlerini hesaplayan yöntemdir. Adagrad ile RMSProp algoritmasının birleşimi şeklinde çalışır. Adam, öğrenme oranını Adagrad gibi basit bir

ortalama ile ölçeklendirmek yerine gradyanların üstel ortalamasını kullanır. Eşitlik 4.15'te m_t gradyanların ortalamasını, v_t gradyanların karaler ortalamasını, g_t gradyan değerini, β_1 birinci momentum değerini ve β_2 ikinci momentum değerini ifade etmektedir. Buna göre Adam algoritması, öncelikle üstel olarak artan gradyanların (m_t) ortalamasını günceller. Sonrasında, karesi alınmış gradyanların (v_t) güncellemesini yapar.

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) {g_t}^2 \quad (4.15)$$

4.8.7. Ftrl

Ftrl optimizasyon tekniği, öğrenme oranlarının farklı boyutları için farklı ayarlanabildiği bir algoritmadır. Eğitim verileri, hangi boyutta ne kadar büyük bir adım atılması gerektiğini değerlendirmekte ve buna göre öğrenme katsayısını güncellemektedir [130].



Şekil 4.17. Optimizasyon tekniklerinin adım sayısına göre kayıp değer değişim grafiği

Şekil 4.17'de farklı optimizasyon tekniklerinin adım sayısına göre kayıp değer değişimi gösterilmiştir. Şekil'den anlaşılacağı üzere her bir teknik için mutlak minimum değere ulaşma yolu farklı olmaktadır. SGD, Adadelta ve RMSProp tekniklerinde 100 adım sayısında kayıp fonksiyon değeri daha yüksek çıkmaktadır. Bu optimizasyon teknikleri seçildiğinde daha fazla adım sayıları ile model eğitimleri yapılmalıdır. Diğer taraftan Nesterov, AdaGrad, Adam ve Ftrl için daha hızlı bir şekilde mutlak minimum değere ulaşılmaktadır. Kayıp fonksiyon değeri de daha düşük olmaktadır. Sonuçta, bu bölümde de detaylı olarak verildiği gibi optimizasyon tekniklerinin mutlak minimum değere ulaşma hızı, izlediği yol, bu yol için ihtiyaç duyulan adım sayısı ve öğrenme katsayısı gibi farklı avantajlı ve dezavantajlı olduğu noktalar bulunmaktadır. Bu çalışmada örnek test verileri ile grafikte verilmiş olan tüm optimizasyon teknikleri test edilmiş olup, en iyi sonuçları alındığı SGİ algoritması model tasarımında tercih edilmiştir.

4.9. Tensorflow

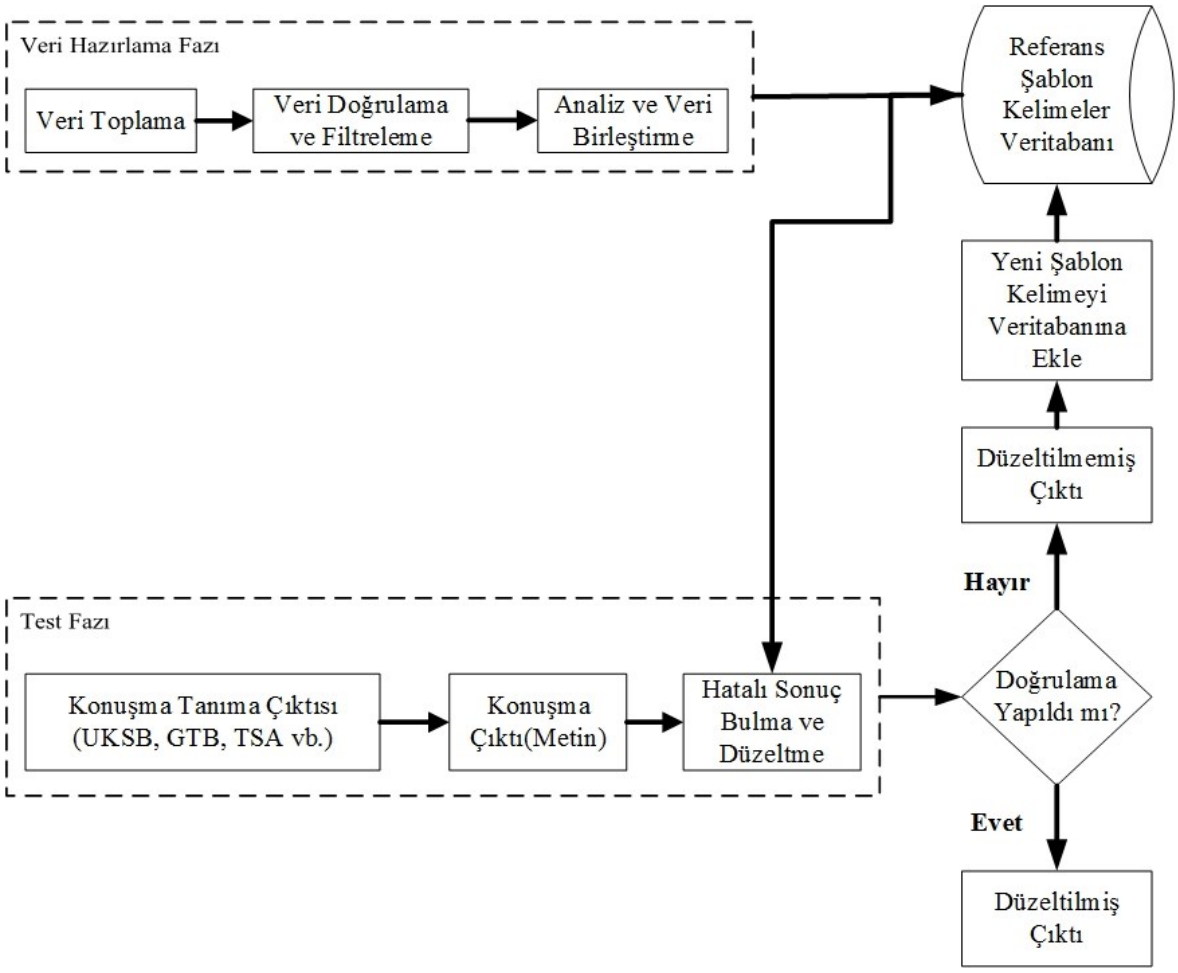
Google tarafından derin inanç ağları ve derin sinir ağlarının eğitimi ve kullanımına yönelik olarak arzu edilen sistem özelliklerini ve gereksinimlerini kapsayan Tensorflow sistemi önerilmiştir [144]. Büyük ölçekli makine öğrenme modellerinin kodlanması ve uygulanmasına yönelik olarak geliştirilmiştir. Veri akışı benzeri bir model kullanarak hesaplamaları yapar ve bunları farklı platformlar ile eşleştirir. Böylece binlerce GPU içeren özel makineler ile çalışılan eğitim sistemlerinden daha mütevazı boyutlu sistemlere kadar eğitim ve çıkarım yapılabilmesine imkân tanır. Tensorflow yapısında, düğümler matematiksel işlemleri temsil ederken, bağlantılar çok boyutlu dizileri temsil eden tensörlerden oluşmaktadır [89]. Tensorflow kütüphanesi, mobil cihazlarda, bilgisayarlarda ve gömülü sistemlerde çalışabilecek şekilde tasarlanmıştır. Bu sebeple farklı ortamlarda bulunan problemlerin çözülmesinde kullanılabilir. Konuşma tanıma, görüntü işleme, doğal dil işleme, robotik gibi birçok araştırma alanında kullanılmaktadır. 2017 yılında bir çalışmada [145], AlexNet modeli farklı derin öğrenme kütüphaneleri ile test edilmiştir. Her bir küme için hesaplama süreleri hesaplanmıştır. Tensorflow kütüphanesinin bu karşılaştırmada en kısa sürede tanıma gerçekleştirebildiği gösterilmiştir. Bu tez çalışmasında UKSB ve GTB bellek yapısında derin sinir ağı kullanılarak hazırlanmış modelin kodlanması ve sonuçlarının alınması için açık kaynak kodlu ve serbest geliştirme yapmaya imkân tanıması nedeniyle Tensorflow kütüphanesinden yararlanılmıştır.

5. OTOMATİK KONUŞMA TANIMA SİSTEMLERİNDE HATA BULMA VE DÜZELTME METODU ÖNERİSİ

Bu tez çalışmasında uçtan uca konuşma tanıma sistemi için kullanılabilir bir hata bulma ve düzeltme modeli önerisi yapılmıştır. Bu model kullanılarak üretilen metin çıktıları üzerinde hatalı karakterlerin doğruları ile değiştirilmesi ve böylece genel sistem performansında bir artış yakalanması mümkün olmuştur. Bu bölümde bu yeni modelin detayları yapılan deney sonuçları ile birlikte verilecektir.

Bağlantıcı zamansal sınıflandırma kullanılan uzun kısa süreli bellek yapısını içeren tekrarlayan sinir ağının kullanıldığı birçok çalışma yapılmıştır. Bu çalışmaların sınıflandırma performansları değişkenlik göstermektedir. Bu performans değerine etki eden veri setinin değişkenliği, eğitim modelinin karmaşıklığı ve eğitim süresi gibi birçok ölçüt bulunmaktadır. Bu sebeple otomatik konuşma tanıma sistemleri için tüm durumlarda tatmin edici performans seviyesinin yakalanabilmesi mümkün olamamaktadır. Bu sebeple geleneksel konuşma tanıma sistemlerin sonuçlarındaki hatalı durumların tespit edilmesi ve düzeltilmesine yönelik farklı yaklaşımlar bulunmaktadır [146-148]. Bunlar elle sonuçların düzeltilmesi, alternatif hipotez önerisi, desen tanıma ve sonuç düzeltme yaklaşımlarıdır. Önerilen model alternatif hipotez önerisi yaklaşıma karşılık gelmektedir. Önerilen model ile hatalı kelimeler için alternatif kelimeler önerilmekte ve bu şekilde hataların düzeltilmesi yapılmaktadır.

Bu çalışmada önerilen hata bulma ve düzeltme algoritmasında Şekil 5.1'de gösterildiği gibi veri hazırlama ve test fazı bulunmaktadır. Veri hazırlama fazı, test için gerekli veri derleminin oluşturulduğu aşamadır. Veri toplama, veri doğrulama ve filtreleme, analiz ve veri birleştirme adımlarını içerir. Böylece "referans şablon kelimeler" veritabanı oluşturulmaktadır. Ön hazırlık aşaması olarak planlanmıştır. İkinci aşamada ise veri hazırlık fazında oluşturulan veritabanı kullanılarak testler gerçekleştirilmektedir. Bu aşamada UKSB, GTB veya TSA tabanlı bir model hazırlama, sistem çıktılarını şablon veritabanı ile karşılaştırma ve hatalı karakterleri düzeltme adımlarını içermektedir. Bu düzeltme sonucunda sistem tanıma performansının yükseltilmesi sağlanmış olmaktadır.



Şekil 5.1. Hatalı sonuçları bulma ve düzeltme metodu önerisi şeması

5.1. Veri Hazırlama Fazı

Veri hazırlama fazında, Türkçe kelimeleri içeren veri setinin bir araya getirilmesi işlemleri gerçekleştirilmiştir. Bunun için Metu 1.0 [70, 149], Boun Corpus [150] ve Zemberek [151] kelime veri setleri kullanılmıştır. Bu veri kümeleri farklı amaçlar için oluşturulmuştur. Bu nedenle, verilerin doğrulanması ve filtrelenmesi gerekmektedir. Gerekli olmayan fazla verilerden arındırılmalı ve sadece kelimeleri içeren bir veri kümesi oluşturulmalıdır. Veri setinde bulunan kelime kök ve eklerini gösterir şemaların silinmesi, el ile hizalama yapılan setlerde hizalamaya ilişkin zaman verileri veya Türkçe olmayan karakterlerin ve noktalama işaretlerin düzeltilmesi gibi birçok farklı işlemin uygulanmasına ihtiyaç duyulmaktadır. Kullanılan 3 adet veri seti için gerekli düzenleme işlemleri sonrasında birleştirme işlemi gerçekleştirilmiştir. Birleştirme sonrası elde edilen kelime seti üzerinde tekrarlı verilerden arındırma ve ilişkisel veri yapısında tutmak için gerekli olan veri tipi dönüşümleri

yapılmıştır. Sonuç olarak yaklaşık 1.6 milyon eşsiz Türkçe kelime barındıran referans şablon kelimeler veritabanı oluşturulmuştur. Böylece sistem sonuçlarında düzeltme yapabilmek için gerekli olan Türkçe referans kelimeleri elde edilmiştir.

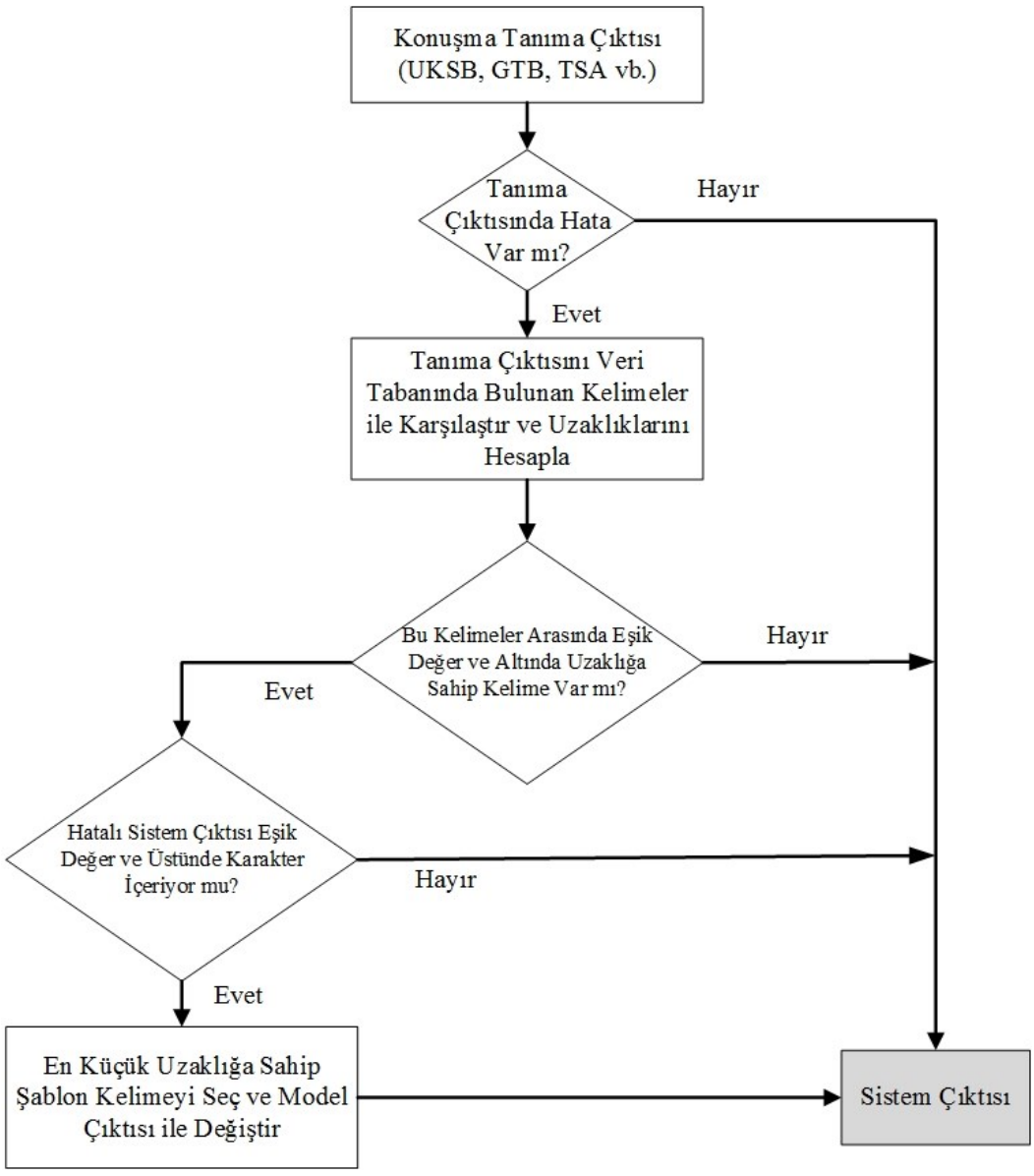
Metin veri seti oluşturulduktan sonra, akustik model eğitimi için ses verisinin elde edilmesi ve eğitim aşamasında kullanılmak üzere düzenlenmesi gereklidir. Bu çalışmada akustik veri seti olarak "Linguistic Data Consortium (LDC)" üzerinden dağıtımı yapılan ve birçok Türkçe konuşma tanıma geliştirme uygulamalarında kullanılan "Middle East Technical University Microphone Speech v 1.0" [70, 149] ses verisi kullanılmıştır. Veri setinin sessiz bir ortamda kaydedilmiş olması, gerekli ses filtreleme işlemlerinden geçmesi ve daha önce birçok uygulama da kullanılarak iyi sonuçlar üretmiş olması sebebiyle tercih edilmiştir.

5.2. Test Fazı

Önerilen modelin testlerini gerçekleştirebilmek için BZS kullanılan UKSB ve GTB yapısına sahip TSA modelleri oluşturulmuştur. Oluşturulan bu modeller daha önce oluşturulmuş olan ses verisi ile eğitilmiştir. Bu eğitim sonrasında testler gerçekleştirilerek metin çıktıları elde edilmiştir. Bu elde edilen çıktılar üzerinde hata bulma ve düzeltme algoritması çalıştırılmıştır. Bu şekilde farklı aşamaları bulunan özgün model önerisinin detayları Şekil 5.2'de gösterilmiştir.

Şekil 5.2'de verilmiş olan şema incelendiğinde, ilk olarak geleneksel UKSB veya GTB tabanlı otomatik konuşma tanıma sisteminin üretmiş olduğu çıktılarda herhangi bir hatalı karakter bulunup bulunmadığı kontrol edilir. Bu kontrol adımı, sistemin doğru kelimeleri tekrar düzeltmeye çalışarak zaman kaybetmesini önlemek amacıyla yönteme eklenmiştir. Eğer çıktı hatalı değilse herhangi bir düzeltme yapılmadan doğrudan sistem çıktısına iletilir. Ancak, üretilen kelimelerde karakter hataları var ise bu durumda hatalı sözcüğün yerine önerilecek alternatif önerinin bulunmasına yönelik olarak veri tabanındaki tüm şablonlar ile hatalı kelime arasında metin uzaklık değerleri hesaplanır. Bu hesaplamada elde edilen değerlerin optimum eşik değer ve altında olup olmadığı kontrol edilir. Eşik değerler yapılan testler sonucunda elde edilerek önerilen modele entegre edilmiştir. Eğer eşik değer ve altında uzaklık değerine sahip bir şablon sözcük bulunamaz ise hatalı çıktı üzerinde herhangi bir düzeltme işlemi yapılmamaktadır. Doğrudan sistem çıktısına iletilmektedir. Aksi durumda, eşik değer ve altında kelime varlığı tespit edilmiş olur ve bir sonraki kontrol aşamasına

geçilir. Bu aşamada, geleneksel modelin üretmiş olduğu çıktının belirli sayı ve üstünde karaktere sahip olması beklenir. Eğer hatalı çıktılarda yeterli karakter bulunmuyor ise düzeltme işlemi yapılmadan sistem çıktısına iletilir. Bu iki kontrol mekanizmasından geçen tüm hatalı kelimeler, veri tabanında bulunan şablon kelimeler ile değiştirilerek sistem çıktısı üretilir. Böylece alternatif bir kelime önerisi ile hataların azaltılması ve sistem tanıma performansının yükseltilmesi sağlanmış olur.



Şekil 5.2. Önerilen çıktı düzeltme algoritması şematik gösterimi

Düzeltme işlemi sürecinde hatalı kelime ile şablon kelimeler arasındaki uzaklıkların hesaplanmasına yönelik olarak farklı algoritmalar bulunmaktadır. Bu yöntemde, en yakın kelimenin bulunması halinde doğru kelimeye de ulaşılmış olması beklenmektedir. Bu

sebeple oldukça önemli olan uzaklık yönteminin seçimine yönelik olarak 11 farklı uzaklık hesaplama algoritması ile testler gerçekleştirilmiştir. Böylece en doğru alternatif kelimenin üretilmesi ve hatalı çıktının düzeltilmesi mümkün olmuştur.

```
Hata_Bulma_ve_Duzeltme_Metodu()
        Tüm Değişkenleri Varsayılan Değerlere Eşitle;

        /* Geleneksel UKSB, GTB, TSA vb. tabanlı bir model ile Ses Tanıma işlemini
           gerçekleştir. */

        Uçtan uca model sürecini işlet;

        /*Tanıma çıktısının düzeltme ihtiyaç duyup duymadığını kontrol et*/
        if(model çıktısı hatalı mı)
                /*Referans kelimeler ile model çıktısını karşılaştır*/
                for each kelime in templatedb
                        kelime ile sistem çıktısı arasındaki mesafeyi hesapla;
                end for

                if(eşik değer ve altında uzaklığa sahip kelime var mı)

                        if(sistem çıktısı eşik değer ve üstünde karaktere sahip mi)
                                // sistem çıktısını şablon kelime ile değiştirerek düzelt
                                return kelime;
                        else
                                return sistem çıktısı; // çıktı da herhangi bir düzeltme yapma
                else
                        return sistem çıktısı; // çıktı da herhangi bir düzeltme yapma
        else
                return sistem çıktısı; // çıktı da herhangi bir düzeltme yapma
```

Şekil 5.3. Yalancı kod: en yakın kelime bulma metodu

Hatalı kelimeye alternatif olacak şablon kelimenin bulunmasına ilişkin özgün yaklaşımın yalancı kod hali Şekil 5.3'te verilmiştir. Şekilde gösterilmiş olan yöntem her bir kelime için tekrar tekrar çağrılmaktadır. Yöntemin ilk aşamasında, tekrarlı çağrılma sebebiyle değişkenlerde bir veri bulunmayacağına garanti etmek üzere tüm değerler varsayılan durumlarına eşitlenmektedir. Daha sonra, geliştirilmiş olan UKSB veya GTB tabanlı

otomatik konuşma tanıma sistemi ile çıktılar üretilmektedir. Bu çıktıların hatalı olup olmadığı kontrol edilerek, döngü yardımıyla veri tabanında gerekli arama işlemleri yürütülür. Koşulların sağlanmadığı her durumda hatalı çıktı doğrudan sistem çıktısına iletilir ve herhangi bir düzeltme işlemi gerçekleştirilmez. Tüm koşulları sağlayan bir alternatif kelimenin bulunması halinde sistem çıktısı üretilmektedir.

5.3. Testler için Kullanılan Veri Setleri, Araçlar ve Algoritmalar

Bu çalışmada önerilen modelin farklı açılardan test edilebilmesi için ses verisi, metin verisi, veri tabanı araçları ve derin öğrenme modelinin geliştirilebilmesi için birçok farklı yazılım kütüphanesine ihtiyaç vardır.

UKSB tabanlı uçtan uca konuşma tanıma modellerinin eğitimi için akustik ses verisine ihtiyaç duyulmaktadır. Bu çalışmada, "Middle East Technical University Turkish Microphone Speech v.1." [70, 149] ses veri seti tercih edilmiştir. Türkçe konuşma tanıma uygulamalarının geliştirilmesinde kullanılmak üzere ses dosyaları barındırmaktadır. 60 erkek ve 60 kadın konuşmacının seslendirmesini içermektedir. Her birisinin ortalama olarak 300 kelime içerdiği 40 adet cümleyi seslendirmişlerdir. Toplam 500 dakikalık kayıt bulunmaktadır. Seslendirilen 40 adet cümle, 2462 farklı Türkçe kelimeyi barındırmaktadır ve rastgele seçilmiş kelimelerden oluşmakta olup üçlü ses formatında kaydedilmiştir. Konuşmacıların yaş aralığı 19 ile 50 arasında değişmekte olup ortalaması 23,9'dur. Oldukça sistematik olarak hazırlanmış bu Türkçe konuşma veri seti, birçok Türkçe konuşma tanıma uygulamasında da kullanılmıştır.

Geleneksel modelin tanıma çıktılarını düzeltebilmek için şablon veri tabanını oluşturmak üzere Türkçe farklı kelimeleri içerisinde barındıran metin veri setine ihtiyaç vardır. Mümkün olduğunca tüm Türkçe kelimeleri içerisinde bulundurması beklenmektedir. Aksi halde eşleşme yakalanamamakta ve sonuçlarda istenilen düzeltmeler gerçekleştirilememektedir. Bu çalışmada Türkçe konuşma tanıma uygulamalarında yaygın olarak kullanılan 3 Türkçe kelime grubu tercih edilmiştir. Bunlar Zemberek [151], Boun Corpus [150] ve Metu 1.0 [70, 149] akustik veri setidir. Zemberek, 2010 yılında yayınlanmış olup, açık kaynak kodlu olarak dağıtımı yapılan bir kütüphanedir. Türkçe dil bilgisi özelliklerini içermektedir. Yaklaşık olarak 1,15 milyon eşsiz Türkçe kelime barındırmaktadır. Doğal dil işleme, konuşma tanıma, bilgi güvenliği gibi Türkçe diline özel yapılan çalışmalarda kullanılmıştır

[152-156]. Boun Corpus, 2008 yılında yayınlanmış olup Türkçe dil kaynağı oluşturmaya yönelik bir çalışmadır. Yaklaşık olarak 1,4 milyon eşsiz Türkçe kelime barındırmaktadır. Metu v1.0 veri seti, 2002 yılında yayınlanmıştır. 2462 eşsiz Türkçe kelime barındırmaktadır. Elde edilen üç veri seti birleştirilerek referans şablon sözcük olarak kullanmak üzere içerisinde yaklaşık 1,6 milyon eşsiz kelime bulunan bir veritabanı oluşturulmuştur.

1,6 milyon Türkçe kelimeyi barındırabilecek ve üzerinde hızlı bir şekilde arama yapmaya imkan verecek bir veri yönetim sistemine ihtiyaç duyulmaktadır. İlişkisel veritabanı yönetim araçları arasından bir seçim yapabilmek için testler gerçekleştirilmiştir. Bu testler sonucunda en hızlı sonuçların alınabildiği PostgreSql [157] veritabanı aracı, referans şablon kelimeleri barındırmak üzere seçilmiştir.

Test için oluşturulan modelin tasarımı ve eğitimi için Python [158] programlama dili kullanılmıştır. Test sonucunda elde edilen tanıma çıktılarındaki hataların bulunması ve düzeltilmesi algoritmasının kodlaması için Java kütüphanesinden yararlanılmıştır. Konuşma tanıma modelinin Python ile programlanması için Dill [159], Librosa [160], namedtupled [161], numpy [162], python_speech_features [163], tensorflow [164] kütüphaneleri kullanılmıştır.

İki kelimenin birbirine benzerlik durumlarını tespit etmek üzere uzaklıklarının hesaplanması için kullanılabilecek birçok algoritma bulunmaktadır. Bu algoritmalardan yaygın olarak kullanılanlarından Levenshtein [165], Damerau Levenshtein [166], Jaro Winkler [167, 168], Longest Common Subsequence [169], Metric LCS [169], Normalized Levenshtein [165], Optimal String Alignment [166], Precomputed Cosine [170], Qgram [171], Sift4 [172] ve Weighted Levenshtein [169] seçilerek testler gerçekleştirilmiştir. En başarı sonuçların alındığı yöntem seçilerek önerilen model içerisinde kullanılmıştır. Böylece daha iyi performans artışlarının yakalanabilmesi mümkün olmuştur.

5.3.1. Levenshtein algoritması

Bilgisayar bilimlerinde, iki dizi arasındaki mesafeyi ölçmek için kullanılan bir yöntemdir. Bir kelimeyi diğeri ile değiştirmek için gereken minimum tek karakterli düzenleme (ekleme, silme, yer değiştirme) sayısını ifade etmektedir. 1965 yılında “Vladimir Levenshtein” tarafından önerilmiştir [165].

“a ve b” metin dizileri arasındaki uzaklığı hesaplamak için “lev(a,b)” fonksiyonunun tanımlandığını varsayıldığında, Eşitlik 5.1’de gösterilen yaklaşım kullanılarak hesaplama yapılmaktadır.

$$lev_{a,b}(i,j) = \begin{cases} \text{maks}(i,j), & eğer\ \min(i,j) = 0 \\ min \begin{cases} lev_{a,b}(i-1,j) + 1 \\ lev_{a,b}(i,j-1) + 1 \\ lev_{a,b}(i-1,j-1) + 1_{(ai \neq bj)} \end{cases} \end{cases} \quad (5.1)$$

Eşitlik 5.1’de maksimum ve minimum değerleri almak için maks ve min fonksiyonları kullanılmaktadır. “i” ve “j” değerleri ise “a” ve “b” metin dizilerinin karakter indekslerini ifade etmektedir. Eşitliğe göre, “$a_i = b_j$” olduğunda gösterge fonksiyonu 0, diğer durumlarda 1 olarak alınmaktadır. “lev” fonksiyonu ile “a” karakter dizisinin “i.” indeksine kadar olan bölüm ile “b” karakter dizisinin “j.” indeksine kadar olan bölümün uzaklığını hesaplamaktadır. “i ve j” değerleri, 1’er artışlı değerleri ifade etmektedir.

5.3.2. Normalized-Levenshtein algoritması

Levenshtein algoritması kullanarak iki dize arasındaki mesafe hesaplanır. Sonrasında elde edilen değer [0,0 1,0] aralığında olacak şekilde normalize edilir. Ancak bu durumda bir metrik olmaktan çıkmaktadır. Benzerlik 1’e normalize edilmiş olarak kullanılır [165].

5.3.3. Weighted-Levenshtein algoritması

Farklı karakter işlemleri için farklı ağırlıkları tanımlamaya izin veren bir Levenshtein algoritması türüdür. Birbirine benzer harflerin hatalı tanınması ile birbirinden uzak harflerin hatalı tanınmasının maliyetini birbirinden ayıran bir yaklaşımdır. Karakter düzenine bağlı olarak birbirine yakın konumdaki harflerin hatalı yazılması daha düşük maliyetli olarak değerlendirilir [169].

5.3.4. Damerau-Levenshtein algoritması

Levenshtein algoritmasının daha gelişmiş bir yaklaşımı olarak önerilmiştir [166]. İki dizi arasındaki mesafeyi ölçmek için kullanılır. Bu ölçüm bir kelimeyi diğeri haline getirmek için

gereken minimum işlem sayısını (tek bir karakter silinmesi, eklenmesi, yerine koyma veya bitişik karakterin yer değiştirilmesi) ifade eder. Levenshtein algoritmasından en önemli farkı, 3 karakter düzenleme işlemine ek olarak, bitişik karakter yer değiştirilmesini işlemlerini de işlem sayısına dâhil etmesidir.

"a ve b" karakter dizileri için Eşitlik 5.2'deki gibi "$d_{a,b}$ (i, j)" tanımlandığında, "i" değeri, "a" dizisinin "i." indeksine kadar olan bölümünü, "j" ise "b" dizisinin "j." indeksine olan bölümünü ifade eder. Buna göre, özyinelemeli olarak aşağıda verilmiş olan eşitlik tanımlanmaktadır.

$$d_{a,b}(i,j) = min \begin{cases} 0 & ,eğer\ i = j = 0 \\ d_{a,b}(i-1,j) + 1 & ,eğer\ i > 0 \\ d_{a,b}(i\ ,j-1) + 1 & ,eğer\ j > 0 \\ d_{a,b}\ (i-1,j-1) +\ 1_{(ai \neq bj)} & ,eğer\ i,j > 0 \\ d_{a,b}(i-2,j-2) + 1\ \ ,eğer\ i,j > 1\ ve\ a[i] = n[j-1] ve\ a[i-1] = b[j] \end{cases} \quad (5.2)$$

Eşitlik 5.2'ye göre, birinci satırda bulunan eşitlik herhangi bir tanıma metninin bulunmadığı durumu ifade eder. İkinci satırda bulunan eşitlik a ve b karakter dizileri için i. ve j. indekse kadar olan silme işlemlerini, üçüncü satırda bulunan eşitlik, i. ve j. indekse kadar olan karakter ekleme işlemlerini ifade eder. 4. satırda bulunan eşitlik ise, a ve b karakter dizilerinin aynı olup olmadığına göre bir eşleşme veya uyumsuzluğa karşılık gelir. "ai = bj" olduğunda gösterge fonksiyon 0 olarak alınmaktadır, diğer durumlarda 1 alınır. Son olarak, 5. satırda bulunan eşitlik ise, birbirini izleyen iki karakterin yer değiştirmesi durumunu temsil eder. Böylece Damerau Levenshtein algoritmasındaki tüm durumları temsil eden bir eşitlik ile formüle edilmiş olmaktadır.

5.3.5. Optimal metin hizalama algoritması

Damerau-Levenshtein (DL) yaklaşımının bir varyantı olan bu algoritma, alt dizenin bir defadan fazla düzenlenmemesi koşuluyla dizeleri eşitlemek için gereken düzenleme işlemlerinin sayısını hesaplamaktadır. DL algoritmasından en önemli farklı alt dize oluşturulması işlemidir. Levenshtein algoritmasından farkı ise yer değiştirme işlemlerini de dikkate almasıdır [166].

5.3.6. Jaro-Winkler

İki dizi arasındaki uzaklığı ölçümlemekte kullanılan ve 1990 yılında önerilmiş olan yöntemdir [167, 168]. Bu yöntemde, dizinin belirli uzunluktaki ilk kısmı için eşleşme yakalandığında bunun değerini ifade etmek üzere belirli seviyede derecelendirme yapılır. Mesafe ne kadar küçük ise o kadar iyi bir eşleşme sağlanmış olduğu kabul edilir.

s1 ve s2 iki diziyi temsil etmek üzere, $sim_{i,j}$;

$$sim_j = \begin{cases} 0 & ,eğer\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|s1|} + \frac{m}{|s2|} + \frac{m-t}{|m|}\right) & ,diğer\ durumlar \end{cases} \quad (5.3)$$

Eşitlik 5.3'te "$|s1|$", "s_i" dizisinin uzunluğunu, "m" eşleşen karakter sayısını, "t" yeri tutmayan karakter sayısının yarısını ifade etmektedir. s1 dizisinin her bir karakteri, s2 dizisindeki tüm eşleşen karakterler ile karşılaştırılır. Eşleşme sağlanıp, yerde farklılık söz konusu ise 2 ye bölünerek kullanılır. Bu şekilde hesaplama yapılarak iki metin arasındaki mesafe ölçümlenmiş olmaktadır.

5.3.7. Longest common subsequence (LCS)

Bir metin dizisi ile birden fazla metin dizisi barındıran grup arasındaki karşılaştırmada kullanılan bir yaklaşımdır. En iyi eşleşmenin sağlandığı metin dizisi bulunmaya çalışılmaktadır [169]. Bu işlemin yapılması için, problem daha alt problemlere bölünerek çözümlenmeye çalışılır. Alt dizilere bölünmesindeki amaç daha basit problemlere bölerek hesaplamayı yapmaktır.

X = (X_1, X_2, X_3, …, X_m) ve Y = (Y_1, Y_2, Y_3, …, Y_n) şeklinde iki dizi arasındaki uzaklığın hesaplanması gerektiği düşünülmektedir. X dizisinin alt dizisi "$X_{1,2,3,\ldots M}$" ve Y dizisinin alt dizisi "$Y_{1,2,.M}$" olsun. Bu durumda LCS(X_i, Y_j) Eşitlik 5.4'te verilmiş olan formüle göre hesaplanmaktadır.

$$LCS(X_i, Y_j) = \begin{cases} 0 & ,eğer\ i = 0\ veya\ y = 0 \\ LCS(X_{i-1}, Y_{j-1})^{x_i} & ,eğer\ i,j > 0\ ve\ x_i = y_j \\ max\{LCS(X_i, Y_{j-1}), LCS(X_{i-1}, Y_j)\} & ,eğer\ i,j > 0\ ve\ x_i \neq y_j \end{cases} \quad (5.4)$$

"X_i ile Y_j" arasındaki LCS değerini bulmak için "x_i ve y_j" elemanları karşılaştırılmalıdır. Eğer eşitlik yakalanır ise LCS (X_{i-1}, Y_{j-1}) değeri "x_i" değeri ile genişletilir. Eğer eşitlik yakalanmaz ise, LCS (X_i, Y_{j-1}) ile LCS(X_{i-1}, Y_j) değerinden uzun olanı muhafaza edilmektedir. Eğer uzunluk eşit ise herhangi bir tanesi hafızada tutulmaktadır. Bu şekilde sıralı olarak işlemlere devam edilmekte ve en iyi eşleşme yakalanmaktadır.

5.3.8. Metric LCS

Bu yaklaşımda, en uzun ortak alt dizilere dayanan bir dizi benzerliği ölçüsü kullanılmaktadır. Bu metrik Şekil 5.5'te verilmiş olan formüle göre tanımlanmaktadır [156].

$$d(x, y) = 1 - f(x, y)/M(x, y) \quad (5.5)$$

Eşitlik 5.5'te x ve y iki sonlu uzunluktaki diziyi ifade ederken, f(x,y), x ve y arasındaki en uzun ortak alt dizinin uzunluğunu ifade etmektedir. M(x,y) ise x ve y arasında en uzun dizinin boyutunu temsil etmektedir. Metrik kısa dizi uzun olan diziden ne kadar farklıdır sorusunun cevabına göre üretilmektedir.

5.3.9. Qgram

Q-gram iki dizi arasındaki eşleştirmeyi yapmadan önce belirlenecek "q" uzunluktaki karakter dizilerini tüm dizi üzerinde kaydırmak suretiyle oluşturmaktadır. Sonrasında bu q uzunluktaki dizilerdeki eşleşme sayısı hesaplanır. Bu yöntemin arkasındaki en temel fikir eğer iki dizi birbirine çok yakınsa zaten oluşturulan q gram dizilerinde eşleşme sayısı oldukça fazla çıkacaktır. Ne kadar çok q dizisi eşleşmesi sağlanırsa en yakın iki dizi bulunmuş olmaktadır. DNA eşleşmesi gibi biyolojik problemlerin çözümünde tercih edilen bir yaklaşımdır [171].

5.3.10. Kosinüs benzerliği

Kosinüs benzerliği, Dice katsayısına benzer ortak vektör tabanlı bir benzerlik ölçüsüdür. Vektör tabanlı olması sebebiyle öncelikle giriş dizisi vektör uzayına dönüştürülür. Böylece Öklid kosinüs kuralı ile benzerlik belirlenebilir. Diğer yaklaşımlar için boyut sınırlaması getirmektedir. Bu sebeple genelde diğer yöntemler ile birlikte kullanılmaktadır. Bu yaklaşımın temel formülü Eşitlik 5.6'da gösterilmiştir. [170].

$$\cos(q,r) = \sum_{y} \frac{q(y)r(y)}{\sqrt{\sum_y q(y)^2 \sum_y r(y)^2}} \tag{5.6}$$

"q" ve "r" sırasıyla iki metin dizisine ait vektörleri ifade etmektedir. Elde edilen benzerlik "-1" ise iki metin birbirine en uzak ve "1" ise en yakın durumunu göstermektedir. Karşılaştırma işlemi öncesinde metinler üzerinde normalizasyon işlemi yapılmaktadır. Metin özelliklerinden türetilen "q" ve "r" vektörleri kullanılarak benzerlik değeri ölçülmektedir.

5.4. Test için Oluşturulmuş Uçtan Uca Konuşma Tanıma Modeli Tasarımı

Bu tez çalışmasında önerilen yaklaşımın geleneksel modele verdiği katkı seviyesini ölçebilmek için örnek bir uçtan uca konuşma tanıma modeli tasarlanmıştır. Bu modelin tasarımında, sonuca doğrudan etki eden, ağın derinliği, optimizasyon tekniklerinin değişkenliği, kullanılan veri seti büyüklüğünün hata oranına etkisi gibi birçok husus birlikte değerlendirilmiştir. Bu bölümde, en iyi değerlerin bulunmasına yönelik yapılan testler ve sonuçları verilmiştir. Böylece daha iyi tanıma performansına sahip bir modelin geliştirilmesi amaçlanmıştır.

Bu testlerin ilk aşamasında optimizasyon tekniği seçiminin UKSB veya GTB tabanlı bir otomatik konuşma tanıma sisteminin sınıflandırma performansına etkisi gözlemlenmeye çalışılmıştır. Testler için oluşturulan ses verisi kelime dağarcığı rastgele seçilmiş 100 kelimeden oluşmaktadır. Testlerin gerçekleştirilmesi için tasarlanan modele ait temel parametreler Çizelge 5.1'de verilmiştir. Buna göre öğrenme katsayısı hariç diğer tüm değerleri farklı optimizasyon seçenekleri için sabit ve aynı olarak belirlenmiştir. Böylece karşılaştırma süreçlerini etkileyecek diğer faktörlerin ortadan kaldırılması mümkün

olmuştur. Öğrenme katsayının farklı olarak belirleme zorunluğu bulunmaktadır. Çünkü her bir tekniğin mutlak minimum değere ulaşma yolu ve hızı birbirinden farklıdır.

Çizelge 5.1. Farklı optimizasyon tekniklerinin karşılaştırılmasında kullanılacak modele ait temel parametreler

Optimizasyon Algoritması	ÖK	EDV	Standart Sapma	Adım	PB	ÖS	Ortalama
GradientDescent	0,01	0-10	0,1	5000	1	1	0
ProximalGradientdescent	0,01	0-10	0,1	5000	1	1	0
RMSPropOptimizer	0,01	0-10	0,1	5000	1	1	0
MomentumOptimizer	0,005	0-10	0,01	5000	1	1	0
AdadeltaOptimizer	10	0-10	0,1	5000	1	1	0
AdagradOptimizer	0,1	0-10	0,1	5000	1	1	0
AdamOptimizer	0,01	0-10	0,1	5000	1	1	0
FtrlOptimizer	0,01	0-10	0,1	5000	1	1	0
ProximalAdagradOptimizer	0,01	0-10	0,1	5000	1	1	0

Çizelge 5.1'de başlık olarak verilmiş olan "ÖK" öğrenme katsayısını, "EDV" eğitim ve doğrulama veri setlerini, "PB" paket boyutunu ve "ÖS" örneklem sayısını ifade etmektedir. Çizelgede verilmiş olan topolojiler kullanılarak her bir aşamada 100 rastgele kelimenin bulunduğu ses verisinin kullanıldığı test ortamlarında deneyler yapılmıştır. Bu deneylerin sonucunda elde edilen sonuçlar detaylı olarak Çizelge 5.2'de gösterilmiştir.

Çizelge 5.2. En iyi performans sonuçları

Optimizasyon Algoritması	EHO <%50	EHO = %0	DHO <%50	DHO=%0	OEHO	ODHO
GradientDescent	0,9458	0,5234	0,9368	0,3446	0,06911	0,09120
ProximalGradientdescent	0,9238	0,5186	0,9136	0,3644	0,08813	0,10271
RMSPropOptimizer	0,957	0,4692	0,9524	0,1248	0,06259	0,10279
MomentumOptimizer	0,8772	0,4686	0,8676	0,3512	0,13514	0,14953
AdadeltaOptimizer	0,002	0,3954	0,8914	0,1198	0,12255	0,16414
AdagradOptimizer	0,9134	0,3084	0,9016	0,0198	0,10911	0,18034
AdamOptimizer	0,8484	0,3958	0,8426	0,0444	0,16545	0,22592
FtrlOptimizer	0,1198	0	0,0766	0	0,73971	0,75667
ProximalAdagradOptimizer	0,0862	0	0,049	0	0,75492	0,77598

Çizelge 5.2'de başlık olarak verilmiş olan "EHO" eğitim hata oranını, "DHO" doğrulama hata oranını, "OEHO" ortalama eğitim hata oranını ve "ODHO" ortalama doğrulama hata oranını temsil etmektedir. Çizelge incelendiğinde hem eğitim hem de doğrulama hata değerleri için en iyi sonuçların "Gradient Descent" optimizasyon tekniği ile elde edildiği tespit edilmiştir. Ancak "Proximal Gradient Descent" ve "RMSPropOptimizer" tekniklerinde de yaklaşık olarak benzer sonuçların elde edildiği, aynı testlerin benzer şartlarda tekrarlanması halinde ilk üç sıradaki optimizasyon tekniği sonuçlarının sıralarının değişebildiği gözlemlenmiştir. Bu sebeple, önerilen hata bulma ve düzeltme algoritmasının test edilebilmesi için oluşturulacak modelde en iyi performansa sahip üç farklı optimizasyon tekniğinde herhangi birinin tercih edilmesi faydalı olacaktır. Bu testler sonucunda optimizasyon tekniğinin değişiminin sonuçlara doğrudan etki eden bir husus olduğu anlaşılmıştır. Çünkü tanıma performans seviyesi %23 ile %91 arasında değişkenlik göstermiştir. Bunun yanında tanıma performansının çok düşük olduğu tekniklerde öğrenmenin gerçekleşmediği anlaşılmıştır.

Çizelge 5.2'de gösterilen sonuçlara adım sayısı sabit olarak 5000 belirlendiği durumlarda ulaşılmıştır. Rastgele seçilmiş 100 kelime içeren dağarcık ile yapılan testlerde adım sayısı değişiminin sistem tanıma performansına etkisi tespit edilmek üzere deneyler yapılmıştır. Bu tespit için daha önce belirlenmiş olan ve en iyi tanıma performansına sahip 3 optimizasyon tekniği için farklı adım sayıları ile testler gerçekleştirilmiştir. Bu testlerde elde edilen en başarılı sonuçlar Çizelge 5.3'te gösterilmiştir

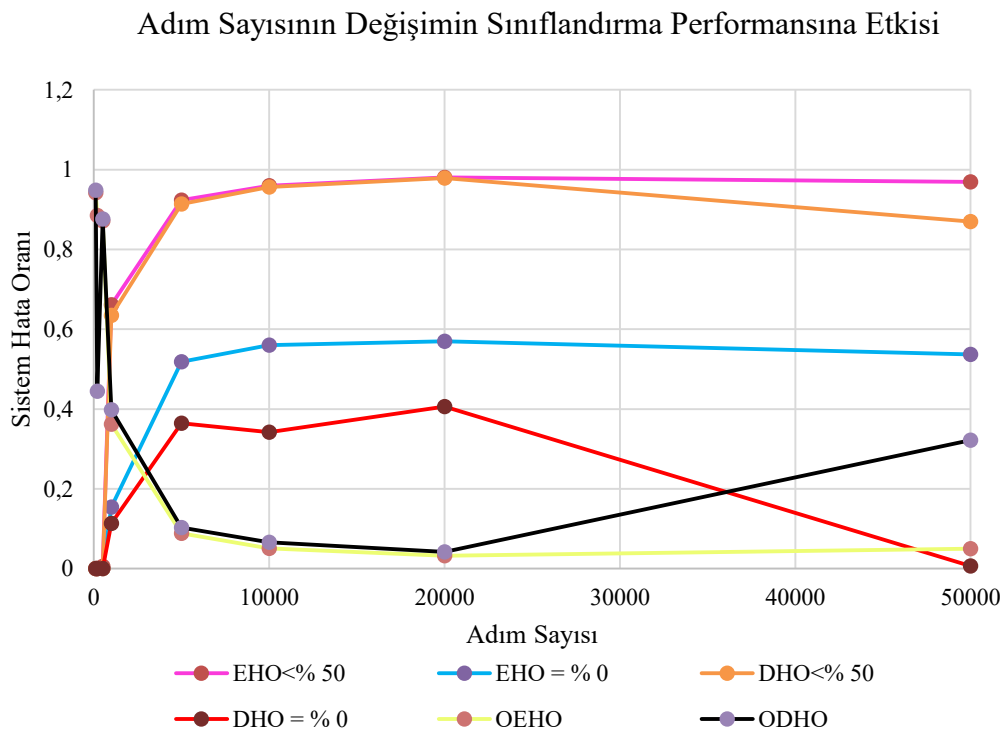
Çizelge 5.3. Adım sayısı değişiminin genel tanıma performansına etkisi

Optimizasyon Algoritması	Adım Sayısı	EHO < %50	EHO = %0	DHO < %50	DHO = %0	Ortalama EHO	Ortalama DHO
ProximalGradientDescent	20000	0,9807	0,5695	0,9789	0,4061	0,0320	0,0422
GradientDescent	10000	0,9648	0,5561	0,9613	0,4085	0,0475	0,0568
GradientDescent	20000	0,9771	0,5759	0,9754	0,1214	0,0346	0,0879
GradientDescent	5000	0,9458	0,5234	0,9368	0,3446	0,0691	0,0912
ProximalGradientdescent	5000	0,9238	0,5186	0,9136	0,3644	0,0881	0,1027

Çizelge 5.3'te başlık olarak verilmiş olan "EHO" eğitim hata oranını, "DHO" doğrulama hata oranını, "OEHO" ortalama eğtim hata oranını ve "ODHO" ortalama doğrulama hata oranını temsil etmektedir. Çizelge incelendiğinde, en başarılı tanıma performansına sahip

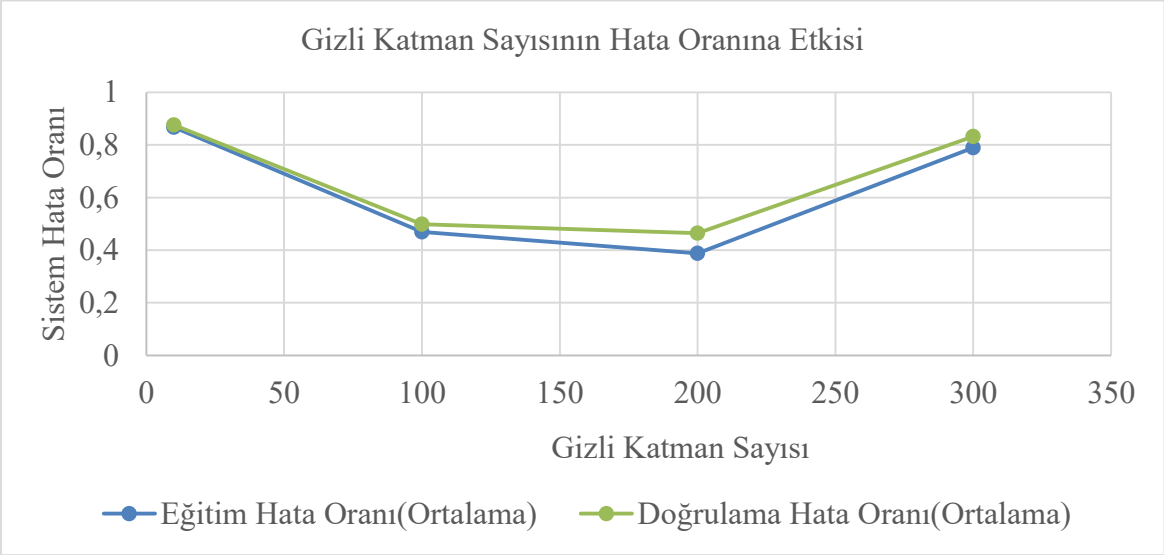
değerlerin “ProximalGradientDescent” ve “GradientDescent” algoritmaları ile alındığı tespit edilmiştir. %91 seviyesinde olan en başarılı tanıma sonucu bu aşamada adım sayısının artırılması ile %96 seviyesine yükselmiştir. Öğrenme katsayının düşürülerek daha fazla adımda öğrenme sürecini gerçekleştirmek daha iyi performansa sahip otomatik konuşma tanıma sistemlerini mümkün kılmaktadır. Bu aşama en kritik husus öğrenme katsayısı ile adım sayısını değişiminin birlikte değerlendirilmesi durumudur. Aksi halde öğrenme katsayısı sabit tutulup adım sayısı artırılırsa, eğitim durmakta ve tanıma performansı tekrar gerileme duruma geçmektedir. Tersi durumda da öğrenme katsayısı yüksek olduğu için mutlak minimum değerine ulaşmak çoğu halde mümkün olmamaktadır.

Şekil 5.4’te “EHO” eğitim hata oranını ve “DHO” doğrulama hata oranını ifade etmektedir. Şekilde, eğitim katsayısı sabit tutularak adım sayısı değişiminin sistem tanıma performansına etkisi gösterilmiştir. Buna göre eğitim katsayısının sabit tutulduğu durumlarda sistem hata oranı, eğitim ve doğrulama hata oranı adım sayısının 20000 ve üzerinde olduğu durumlarda yükselmektedir. Eğitim hata oranı ve doğrulama hata oranı sarı ve siyah çizgiler ile gösterilmiştir.



Şekil 5.4. Adım sayısının sınıflandırma performansına etkisi

Bu çalışmada UKSB veya GTB kullanılan bir tür derin öğrenme yapısı kullanılmaktadır. Bu sebeple gizli katman sayısının durumu modelin eğitim sürecini doğrudan etkilemektedir. Bu etkiyi gözlemleyebilmek için daha önce en iyi sonuçların alınabildiği optimizasyon tekniği ve adım sayısı sabit tutularak farklı gizli katman sayıları ile testler yapılmıştır. Şekil 5.5'te bu sonuçların değişimi gösterilmiştir. Buna göre hem eğitim hem de doğrulama hata oranları paralellik göstermektedir. Gizli katman sayının çok az olduğu ve geleneksel sinir ağlarına benzer yapılarda sistem tanıma performansları oldukça düşmektedir. Benzer şekilde gizli katman sayının çok arttığı ve modelin daha karmaşık hale geldiği durumlarda da sistem tanıma performansı düşmektedir. Bu sebeple 100 ile 200 aralığında ortalama bir değerin belirlenmesi halinde daha yüksek performans sonuçlarına ulaşılabilmesi mümkün olmaktadır.



Şekil 5.5. Gizli katman sayısının hata oranına etkisi

Bu bölümde verilmiş olan sonuçlar 100 rastgele kelime barındıran kelime dağarcığı ile test edilmiştir. Sonuçların daha geniş dağarcık ile benzer olup olmadığını gözlemlemek üzere 10000 rastgele kelime barındıran bir ses verisi hazırlanmıştır. Bu veri seti kullanılarak modelin tanıma performansı test edilmiş olup sonuçlar Çizelge 5.4'te gösterilmiştir. Çizelge incelendiğinde, en iyi sonuçların alındığı modellerde sistem tanıma performans seviyesi %96'dan %50 seviyelerine kadar inmektedir. Kelime dağarcığı otomatik konuşma tanıma sistemlerinde oldukça kritik bir parametredir ve sonuçları doğrudan etkilemiştir.

Çizelge 5.4. 10000 rastgele kelime barındıran kelime dağarcığı ile yapılan testlerin sonuçları

Optimizasyon Algoritması	Adım	EHO < %50	EHO = %0	DHO < %50	DHO = %0	Ortalama EHO	Ortalama DHO
GradientDescent	10000	0,2025	0	0,3911	0	0,4188	0,5036
AdagradOptimizer	10000	0,2875	0	0,4781	0	0,4702	0,5292
MomentumOptimizer	10000	0,2198	0,0002	0,4409	0	0,4358	0,5303
RMSProp	5000	0,5904	0	0,7731	0	0,4227	0,5445
AdamOptimizer	10000	0,5431	0	0,6041	0	0,5422	0,5551
AdadeltaOptimizer	5000	0,7198	0	0,8430	0	0,5310	0,5954
MomentumOptimizer	5000	0,6906	0	0,8204	0	0,5173	0,6028
AdagradOptimizer	5000	0,7518	0	0,8465	0	0,5644	0,6156
AdamOptimizer	5000	0,9500	0	0,9623	0	0,6548	0,6625
GradientDescent	10000	0,7337	0	0,9040	0	0,6309	0,6924
FtrlOptimizer	5000	1	0	1	0	0,9087	0,9096

Çizelge 5.4'te verilmiş olan "EHO" eğitim hata oranını ve "DHO" doğrulama hata oranını göstermektedir. Optimizasyon algoritmalarının kelime dağarcığın büyümesi duruma göstermiş olduğu etki benzerdir.

Bu bölümde gerçekleştirilen testler ile en iyi tanıma performansına sahip otomatik konuşma tanıma sistemi modelinin oluşturulması amaçlanmıştır. Böylece daha başarılı bir model ile hatalı sonuçları bulma ve düzeltme özgün modeli test edilerek sonuçları değerlendirilmiştir. Sonuçlar detaylı olarak incelendiğinde, optimizasyon tekniği değişiminin, adım sayısı artışının, gizli katman seviyesinin değişkenliğinin sonuçlara doğrudan etki eden bir husus olduğu anlaşılmıştır. Aynı veri seti ile yapılan deneylerde sistem tanıma performanslarında ciddi başarı farklılıklarına sebep olmaktadır. Bir diğer önemli husus ise kelime dağarcığı değişiminin ne kadar önemli ve etkin bir unsur olduğudur. Aynı modelin 100 kelime ile yapılan testlerinde %96 seviyesinde başarılı yakalanabilmekte iken 10000 kelime ile %50 seviyesine kadar düşüşler yaşanabilmektedir. Bu bölümde elde edilen model ile bu tez çalışmasında önerilen yaklaşım birlikte değerlendirilmiş olup sonuçları Bölüm 6 Bulgular ve Değerlendirme altında verilmiştir.

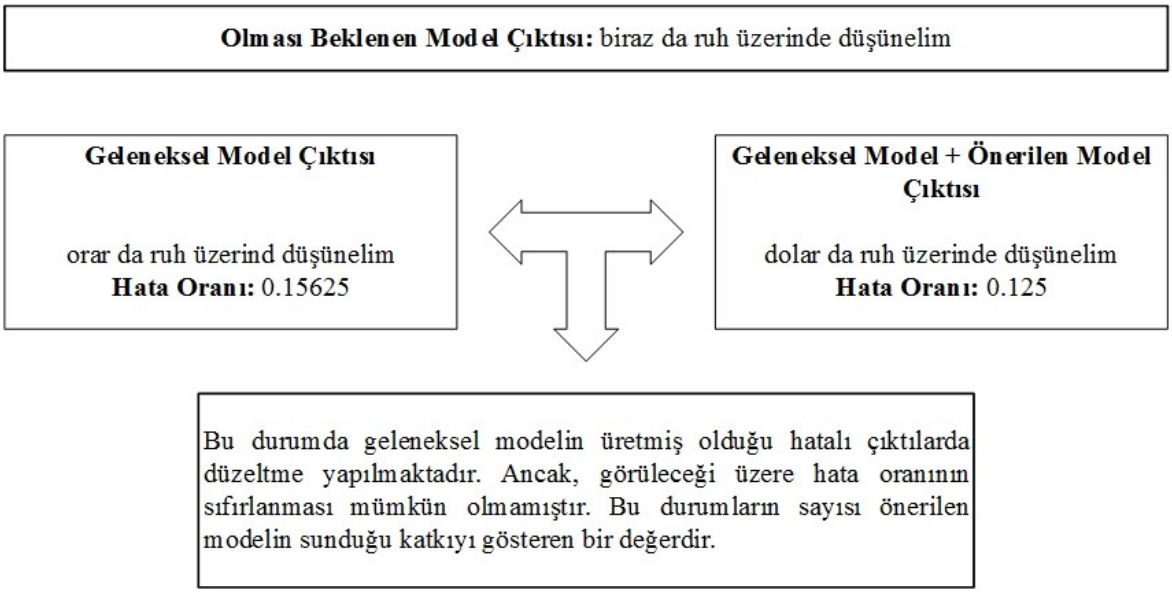
6. BULGULAR VE DEĞERLENDİRME

Bu tez çalışmasında otomatik konuşma tanıma sistemleri için hata bulma ve düzeltme özgün metodu önerisi yapılmıştır. Bu model ile tanıma performansının artırılması amaçlanmıştır. Bu amaç doğrultusunda, oluşturulan test ortamlarında sonuçlar değerlendirilmiş olup, sistemin sunduğu gerçek katkı ile birlikte, gelecekte yapılacak olan gelişmelere ışık tutacak bulgular elde edilmiştir.

Bu çalışmada önerilen modelin tanıma performansına katkı seviyesini ölçebilmek için Bölüm 5'te yapılan testler sonucunda elde edilen en iyi sonuçlara sahip bir model tasarımı yapılmıştır. Optimizasyon tekniği olarak "Gradient Descent", adım sayısı olarak 10000 ve gizli katman sayısı olarak 128 değeri belirlenmiştir. Model yapısı temel olarak tekrarlayan sinir ağı olmakla birlikte hem UKSB hem de GTB bellek yapısı kullanılmıştır ve testler ayrı ayrı gerçekleştirilmiştir. Akustik ses verisi olarak Metu 1.0 ve metin veri seti olarak bu çalışma için hazırlanmış 1,6 milyon farklı kelime barındıran kelime seti kullanılmıştır. Böylece önerilen modelin geniş kelime dağarcığına sahip konuşma tanıma sistemlerinin tanıma performansına ne seviyede katkı sunduğu tespit edilmiştir. Yapılan testlerde farklı veri setleri oluşturulmuştur. Ancak diğer tüm model parametreleri sabit tutulmuştur.

Yapılan testler ile önerilen modelin katkı seviyesini ölçebilmek için iki farklı ölçüt kullanılmıştır. Bu ölçütlerden birincisi 10000 adım olarak yapılan bir eğitim için ortalama konuşma tanıma performansının geleneksel model göre ne kadar yükseldiğinin tespit edilmesidir. Bu önemli bir kıstas olup önerilen yaklaşımın gerçek katkısını göstermektedir. Bir diğer ölçüt ise, her bir adımda testler gerçekleştirerek karşılaştırma yapılmakta ve önerilen model çıktısı ile geleneksel model çıktısı karşılaştırılmaktadır. Toplam adım sayısı üzerinden karşılaştırma da hangi durumun daha başarılı olduğu tespit edilmektedir. İkinci kıstas önerilen modelin doğrudan sistem genel performansına olan etkisini göstermese de gelecekte yapılacak geliştirmeleri için ciddi bir öngörü vermektedir. Karşılaştırma sonucunda farkın fazla olması düzeltme işleminin genel olarak başarılı olduğunu göstermektedir. Bu karşılaştırma ölçütlerine ilişkin örnek şema Şekil 6.1'de gösterilmiştir. Olması beklenen model çıktısına en yakın cümlenin üretilmesine çalışılmıştır. Örnekte görüleceği üzere geleneksel sistem çıktısında tanıma performansı %84,40 seviyesinde iken,

hata düzeltimi sonrasında %87,5 seviyesine çıkmıştır. Performans artışı %2,90 olmuştur. Bu şekildeki örneklerin sayısı artırılarak ortalama katkı seviyesi belirlenmiştir.



Şekil 6.1. Önerilen modelin örnek test gösterimi

Bu çalışmada önerilen model temel olarak hatalı sistem çıktılarının tespit edilmesi ve bu çıktıların veri tabanındaki şablon kelimeler iler karşılaştırılarak alternatif kelime önerilmesine dayanmaktadır. Bu sebeple en önemli parametrelerden birisi iki kelime arasındaki farkın hesaplanmasında kullanılacak olan algoritmanın belirlenmesidir. Çizelge 6.1'de uzaklık algoritması değişimine göre yapılan testlerin sonuçları verilmiştir. Çizelge detaylı olarak incelendiğinde uzaklık algoritması seçimine göre sistemin performans katkı seviyesi değişkenlik göstermektedir. Her durumda %0'ın üzerinde katkı sunulmuş olsa da "Normalized Levensthein" ve "Qgram" algoritmalarında daha başarılı sonuçlar alınmıştır. Ortalama performans katkı seviyeleri sırasıyla %3,32 ve %3,84 olmuştur. Ayrıca, başarı sayısındaki sayısal farkın oransal karşılığı %52,20 olmuştur. 10000 adımlık test için önerilen model 7610 adımda daha başarılı olmuştur. Ayrıca, sistem genel performansı %79,60 seviyesinde olmuştur. Bazı uzaklık algoritmalarında çok düşük katkı düzeylerinin yakalanmasının sebebi bazı uzaklık algoritmalarında parametre seçimlerinde en iyi değerlere ulaşılamamasından kaynaklanmaktadır. Test setinin değiştirilmesi uzaklık algoritmasının katkı düzeyine etkisini hiçbir şekilde değiştirmemektedir.

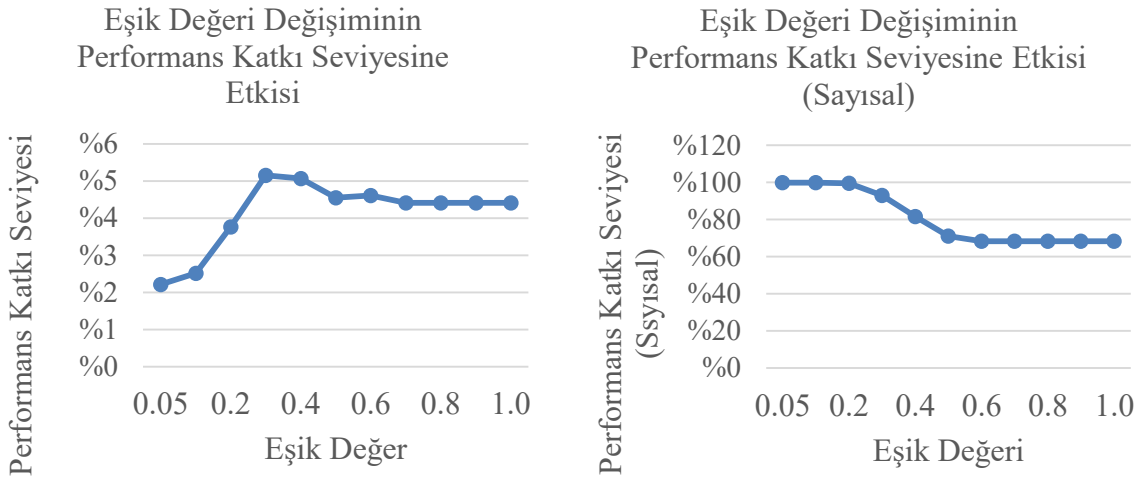
Çizelge 6.1. Farklı uzaklık algoritmalarının tanıma sonuçlarına sunduğu katkı durumu

Uzaklık Ölçüm Algoritması	Düzeltme Öncesi Hata Oranı	Düzeltme Sonrası Hata Oranı	Daha İyi Sonuç Sayısı (Düzeltilmemiş-Düzeltilmiş)	Başarı Oransal Farkı (%)	Genel Sistem Performans Katkısı (%)
NormalizedLevenstain	0,3062	0,2730	2390-7610	%52,20	%3,32
DamerauLevenstein	14,8955	14,8143	3831-6169	%23,38	%0,54
JaroWinkler	0,2315	0,2093	3499-6501	%30,02	%2,22
LongestCommonSubsequence	17,7432	17,1749	3476-6524	%30,48	%3,20
MetricLCS	0,2835	0,2558	2289-7711	%54,22	%2,76
OptimalStringAlignment	14,9061	14,8344	3831-6169	%23,38	%0,46
PrecomputedCosine	0,2860	0,2798	4397-5703	%13,06	%0,60
Qgram	27,5372	26,4720	3879-6121	%22,42	%3,84
Levensthein	14,9278	14,8536	3851-6149	%22,98	%0,49
Sift4	19,8155	19,8039	4402-5598	%11,96	%0,58
WeightedLevensthein	14,9278	14,8536	3851-6149	%22,98	%0,43

Çizelge 6.1'de gösterilen sonuçlara göre en iyi sonuçların alınabildiği "Normalized Levensthein (NL)" algoritmasının kullanıldığı ses tanıma modelinde Şekil 5.2'de verilen modelde gösterildiği gibi iki farklı eşik değerinin belirlenmesi ihtiyacı bulunmaktadır. Eşik değerine göre her iki kıstası da sağlayan durumlarda hatalı çıktılar düzeltilmekte aksi durumda herhangi bir düzeltme işlemi yapılmamaktadır.

"Referans şablon kelimeler" veri tabanında bulunan sözcükler ile sistem çıktısının karşılaştırılmakta ve hatalı kelimeye en yakın şablon kelime bulunmaktadır. Bu karşılaştırma sürecinde bazı sorunlu durumlar ile karşılaşılmıştır. Bu sorunlardan en önemlisi hatalı karakterler barındıran sistem çıktısının yine hatalı olan başka bir şablon kelime ile değiştirilmesi durumudur. Bu durumda sistemin hata oranı artmakta ve genel tanıma performansı olumsuz olarak etkilenmektedir. Bu sorunu aşmak için, hatalı kelime ile alternatif olarak önerilen şablon kelime arasındaki uzaklığın belirli bir eşik değerin altında kalması istenilmiştir. Böylece eğer doğru alternatif kelime bulunabiliyorsa düzeltmenin yapılması aksi durumda hatalı kelime önerisi yerine hatalı çıktının düzetilmemesi sağlanmıştır. Yapılan testlerde eşik değeri kullanımında daha başarılı sonuçlar alınmıştır.

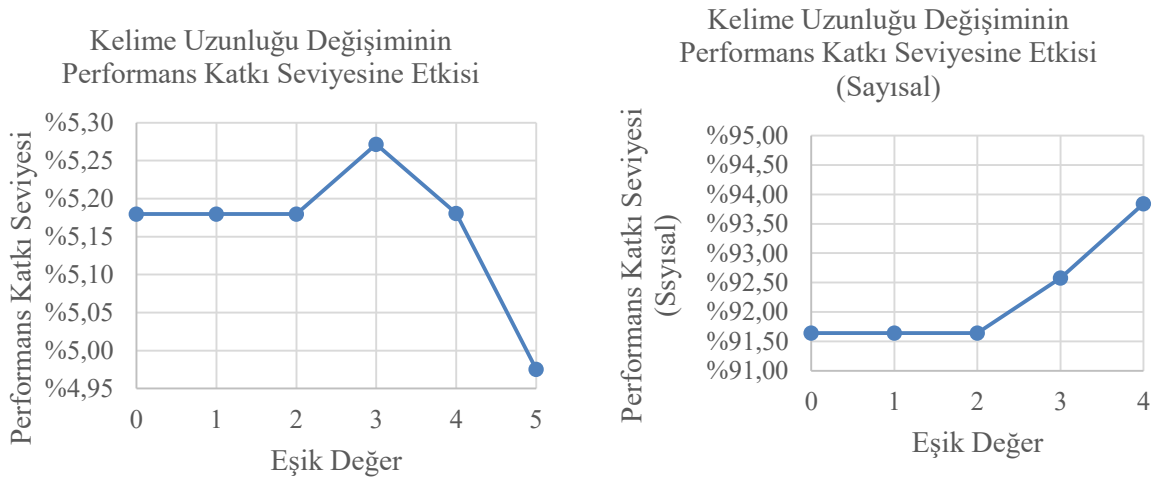
Şekil 6.2'de eşik değerinin değişmesinin önerilen modelin genel sistem performansına katkı seviyesine etkisi gösterilmektedir. Verilmiş olan grafikler incelendiğinde, performans katkısının eşik değer değişimine göre farklılık gösterdiği anlaşılmaktadır. Ortalama katkı seviyesi %2 ile başlayıp %4,2 seviyelerine kadar çıkmaktadır. Bunun yanında 0-1 aralığında değişken eşik değer için en başarılı değerin 0,33 ile yakalanabildiği belirlenmiştir. Bu durumda önerilen modelin performans katkısı %4,60 seviyelerine çıkmaktadır. Şekil 6.1'de verilmiş olan ikinci grafik (sağda) incelendiğinde de benzer şekilde sayısal performans katkı seviyesinin eşik değere göre değiştiği görülmektedir. Her durumda hatalı karakterler belirli seviyede düzeltilmektedir ancak oransal etki eşik değere bağlı olarak farklılık oluşturmaktadır. Eşik değeri değişiminin sayısal karşılaştırma katkısına göre değişimi ikinci grafikte verilmiştir. Sayısal katkı %70 ile %100 aralığında değişkenlik göstermektedir. Buna göre her iki grafik birlikte değerlendirildiğinde eşik değerinin "0,33" olarak belirlenmesi halinde önerilen modelin en yüksek hata düzeltme seviyesine ulaştığı belirlenmiştir. Böylece önerilen model ile daha fazla hatanın bulunması ve düzeltilmesi mümkün olmuştur.



Şekil 6.2. Önerilen model optimizasyonunda eşik değer değişim grafiği

Türkçe dilinin sondan eklemeli diller arasında yer alması sebebiyle bir kökten türetilebilecek kelime sayısı oldukça fazladır. Bu kelimeleri içerisinde barından veri setinde, bir kelime ile türevleri sayıca fazla bulunmaktadır. Belirli bir sayıda karakter barındırmayan('a','b','ac','ba' vb.) kelimelere alternatif kelime arayışında veri tabanında birçok farklı kelime önerisi bulunmaktadır. Bu kelimelerden hangisinin doğru alternatif kelime olacağı seçilememektedir. Bu soruna çözüm üretmek için önerilen yönteme ikinci bir

kontrol aşaması konulmuştur ve üretilen hatalı çıktıların belirli sayıda karaktere sahip olması beklenmiştir. Böylece alternatif kelime sayısının azalması ve doğru şablon kelimenin daha düşük hata ile bulunabilmesi mümkün olmuştur. Hatalı kelimelerde bulunması gereken minimum karakter sayısının belirlenmesine yönelik olarak 100 rastgele kelime barındıran veri seti ile yapılan testler ve sonuçları Şekil 6.3'te gösterilmiştir. Bu grafiklerde farklı karakter sayılarının eşik değer olarak belirlendiği durumlarda performans katkı seviyesi değişimi detaylı olarak gösterilmiştir.



Şekil 6.3. Kelime uzunluğu değişiminin model katkı seviyesine etkisi

Şekil 6.3'te verilmiş olan grafikler incelendiğinde, ortalama performans artış seviyesinin eşik değer değişimine göre farklılık gösterdiği anlaşılmıştır. Performans artış miktarı %5,17 ile başlayıp %5,27 seviyene kadar çıkmakta sonrasında %5'li seviyelerin altına kadar düşmektedir. Bu sebeple eşik değerinin doğru belirlenmesi oldukça önemli bir durumdur. Ayrıca, sayısal katkı seviyesi de benzer şekilde %91,5 ile %94,8 aralığında değişmektedir. Sayısal katkı seviyesinin her durumda belirli bir oranın üzerinde katkı sağlaması sebebiyle, genel performans katkı seviyesine göre eşik değer belirlenmiş olup en iyi sonuçlar 3 karakter ve üzerindeki hatalı çıktıların düzeltilmesi ile alınmıştır. Bu durumda, ortalama olarak %5,27 oranında performans artışı yakalanmıştır.

Bölüm 5 ve 6'da yapılan testler sonucunda elde edilen optimum parametreler kullanılarak bir model hazırlanmıştır. Bu modelin optimizasyon tekniği "Gradient Descent", öğrenme katsayısı "0,01", standart sapması "0,1", adım sayısı "10000", paket boyutu "1", örnek sayısı "1", gizli katman sayısı "128", uzaklık hesaplama algoritması "Normalized Levensthtein",

bellek yapısı “UKSB ve GTB” olarak belirlenmiştir. Bu parametrelere uygun 5 adet UKSB yapısında, 5 adet GTB bellek yapısında test verisi hazırlanmıştır. Uzaklık hesaplama için eşik değerin “0,33” ve hatalı kelime uzunluğu sayısının “3” ve daha büyük bir sayı olarak belirlendiği durumda yapılan deneyler sonucunda elde edilen sonuçlar Çizelge 6.2’de detaylı olarak verilmiştir.

Çizelge 6.2. Eşik Değer Kullanımında Performans Artış Test Sonuçları

Test Set	Önerilen Model Bellek Yapısı	Standart Model Ortalama Performans Artışı	Standart Model + Eşik Değer = 0,33 + Kelime Uzunluğu > 3	Standart Model Sayısal Değer Farkı	Standart Model Sayısal Değer Farkı + Eşik Değer = 0,33 + Kelime Uzunluğu >3
Set1	UKSB	%4,27	%5,18	%64,90	%91,64
Set2	UKSB	%1,85	%4,38	%38,58	%87,60
Set3	UKSB	%3,96	%4,64	%81,62	%93,04
Set4	UKSB	%3,30	%4,52	%61,12	%90,96
Set5	UKSB	%1,70	%4,30	%30,88	%87,96
Ortalama	UKSB	%3,01	%4,60	%55,42	%90,24
Set1	GTB	%3,37	%4,26	%59,83	%75,26
Set2	GTB	%3,85	%6,16	%45,28	%88,84
Set3	GTB	%4,01	%4,25	%62,57	%87,50
Set4	GTB	%2,27	%3,76	%80,79	%88,42
Set5	GTB	%2,18	%4,32	%40,63	%83,06
Ortalama	GTB	%3,13	%4,55	%57,82	%84,62

Çizelge 6.2’de UKSB ve GTB bellek yapısındaki 10 farklı test için standart modelin sunduğu performans artışı ile eşik değer kullanımında elde edilen performans artışı sonuçları karşılaştırmalı olarak gösterilmiştir. Buna göre tüm testlerde eşik değer kullanımı daha iyi sonuçlar elde edilmesini sağlamıştır. Ortalama olarak UKSB bellek yapısı için %3,01 olan performans artışı, %4,60 seviyesine çıkmıştır. Benzer şekilde sayısal değer farkı da %55,42’den %90,24’e çıkmıştır. Sayısal değer farkında ciddi bir artış kaydedilmektedir. GTB bellek yapısının kullanıldığı durumda da %3,13 olan artış %4,55 seviyesine ve %57,82 olan sayısal değer farkı, %84,62 seviyesine çıkmaktadır. Her iki bellek yapısı içinde eşik değer kullanımı genel sistem performans artış oranına katkı sunmaktadır.

Önerilen modelin hatalı kelimeleri düzeltme seviyesi genel tanıma performansına bağımlı şekilde değişkenlik göstermektedir. Bu noktada, sistem performansına göre etki seviyesinin değişimi test edilmiş olup Çizelge 6.3'te sonuçları verilmiştir.

Çizelge 6.3. Genel tanıma hata oranının önerilen model katkısına etkisi

Sistem Hata Oranı	Standart Model Hata Oranı	Standart Model + Düzeltme Metodu	Standart Model Ortalama Performans Artışı	Standart Model Sayısal Karşılaştırma Sonucu	Standart Model + Düzeltme Metodu Sayısal Karşılaştırma Sonucu	Standart Model + Düzeltme Metodu Sayısal Karşılaştırma Sonucu Oransal
<0,025	0,01815	0	%1,82	0	43	%100,00
<0,05	0,03508	0,00175	%3,33	0	236	%100,00
<0,1	0,06652	0,01485	%5,17	3	900	%99,34
<0,2	0,12601	0,06850	%5,75	17	2888	%98,83
<0,3	0,18610	0,12768	%5,84	81	5745	%97,22
<0,4	0,22481	0,16677	%5,80	181	7584	%95,34
<0,5	0,24317	0,18497	%5,82	238	8257	%94,40
<0,6	0,26021	0,20374	%5,65	319	8689	%92,92
<0,9	0,30344	0,25057	%5,29	371	9584	%92,55

Çizelge 6.3 incelendiğinde, daha iyi performansa sahip sistemler için hatalı kelime düzeltme algoritmasının daha fazla katkı sunduğu anlaşılmıştır. Hatalı kelime sayısının fazla olduğu, genel tanıma performansının düştüğü durumlarda ise düzeltme yeterince gerçekleştirilememekte, bu da önerilen modelin katkı düzeyini etkilemektedir. %98,2 üzerinde doğrulukla tanıma yapan sistemlerde tüm hatalı durumlar düzeltilmekte ve sistem tanıma performansı %100 seviyesine çıkarılmaktadır. Benzer şekilde %96,5 seviyesindeki bir sistemin de tanıma seviyesi %99,9 düzeyine yükselmektedir. En başarılı sonuçlar ise %80 ve üzerinde doğrulukla konuşma tanıma yapan sistemlerde alınmakta ve önerilen model %5,84 seviyesinde artış sağlamaktadır.

Bu tez çalışmasında verilmiş olan optimum değerlere sahip uçtan uca Türkçe otomatik konuşma tanıma sisteminin kelime dağarcığına bağlı olarak performans seviyesi ve önerilen modelin katkı durumu değerlendirilmiş olup Çizelge 6.4'te sonuçları gösterilmiştir.

Çizelge 6.4. Kelime dağarcık değişiminin genel tanıma performansına etkisi

Test Seti	Kelime Dağarcığı (Kelime Sayısı)	Uzaklık Hesaplama Algoritması	Sınıflandırma Yaklaşımı	Veritabanı Karşılaştırma Eşik Değeri	Kelime Uzunluğu Eşik Değeri	Uksb Sistemi Performans Seviyesi	Uksb Sistemi + Önerilen Çıktı Düzeltme Algoritması
set1	10	Normalized L.	UKSB	0,33	3	%100	%100
set2	20	Normalized L.	UKSB	0,33	3	%100	%100
set3	100	Normalized L.	UKSB	0,33	3	%99,2	%99,4
set4	200	Normalized L.	UKSB	0,33	3	%93,2	%99,1
set5	500	Normalized L.	UKSB	0,33	3	%75,3	%80,3
set6	1000	Normalized L.	UKSB	0,33	3	%66,8	%71,2
set1	10	Normalized L.	GTB	0,33	3	%100	%100
set2	20	Normalized L.	GTB	0,33	3	%100	%100
set3	100	Normalized L.	GTB	0,33	3	%98,7	%98,9
set4	200	Normalized L.	GTB	0,33	3	%91,3	%94,2
set5	500	Normalized L.	GTB	0,33	3	%74,1	%80,2
set6	1000	Normalized L.	GTB	0,33	3	%65,8	%70,3

Hem UKSB hem de GTB bellek yapısına sahip yöntemler ile testler gerçekleştirilmiştir. Tüm testlerde veritabanı karşılaştırma için eşik değer “0,33” olarak belirlenirken, kelime uzunluğunda eşik değer “3” olarak kullanılmıştır. Kelime dağarcığı ise 10 ile 1000 aralığında değiştirilmiş olup tanıma performans seviyesi %100 ile %66,8 aralığında değişmiştir. Bu çalışmada önerilmiş olan düzeltme algoritması kullanıldığı durumlarda ise en düşük tanıma seviyesi %71,2 olarak bulunmuştur. Kelime dağarcığının sınırlı olduğu problemler için geliştirilen sistem çok daha başarılı sonuçlar üretmişken, dağarcığın büyümesi ile tanıma performansı düşüş eğilimi göstermektedir. Bu olağan bir durum olup, dağarcık otomatik konuşma tanıma sistemleri için oldukça kritik bir parametredir.

Otomatik konuşma tanıma sistemleri için önerilen çeşitli hata bulma ve düzeltme algoritmaları bulunmaktadır. Bu algoritmaların tamamında çıktılar üzerinde farklı işlemler yapılmakta ve hatalı çıktıların düzeltilmesi süreci işletilmektedir. Farklı işlemlere tabi tutulması işlem süresinde artışı da beraberinde getirmektedir. Düzeltme için ihtiyaç duyulan işlem süresi önemli bir husus olup yapılan testlerde dikkat edilmelidir. Geliştirilen sistemler de düzeltme için çok yüksek süreler kullanılması, önerilen modelin kullanım alanının daralmasına, gerçek zamanlı uygulamalarda kullanılmamasına neden olacaktır. Bu noktada beklenen değer probleme özel olarak kabul edilebilir sürelerde kalmasıdır. İhtiyaç duyulan bu sürenin tespitine yönelik olarak yapılan testlerde işlem süreleri takip edilmiş olup sonuçları Çizelge 6.5’te verilmiştir.

Çizelge 6.5. Düzeltme algoritması ve optimizasyon işlemlerinin model işlem zamanına etkisi

Test Seti	Veritabanı Okuma Süresi (Milisaniye)	Düzeltme İçin İhtiyaç Duyulan Süre (Milisaniye)	Veri Setindeki Toplam Kelime Sayısı
Set1	228	102644	60853
Set2	218	109556	66016
Set3	288	97086	62839
Set4	396	83097	60632
Set5	235	92522	55382
Ortalama	273	96981	61144

Çizelge 6.5 incelendiğinde veri tabanında hali hazırda kayıtlı olan 1,6 milyon kelimenin okunabilmesi için ortalama olarak 273 ms'ye ihtiyaç duyulmaktadır. Türkçe gibi yeterli veri kaynağında sahip olmayan diller için farklı kelime sayısının artırılması halinde ihtiyaç duyulan bu süre artabilecektir, ancak bu önerilen modelin sunduğu katkının artışını da beraberinde getirecektir. Veritabanı okuma süresindeki ufak değişikliklerin sebebi test anında çalışan başka süreçlerin etkisidir. Genel tanıma süresini uzatan ikinci önemli süreç ise, bu çalışmada önerilen hatalı kelime düzeltme yönteminin ihtiyaç duyduğu işlem süresidir. Bunun ile ilgili olarak oluşturulan 5 farklı test setinde ortalama olarak 61144 adet kelime bulunmaktadır ve bu kelimeleri düzeltilmesi içinde 96981 ms süreye ihtiyaç duyulmaktadır. Bu durumda her bir kelime için ortalama olarak 1,50 ms kullanılmaktadır. Bu durum 1 dk'lık bir konuşma için ortalama 0,2 sn'ye ihtiyaç duyulduğunu göstermektedir. Bu sürenin kabul edilebilir olduğu problemler için kullanılabilir bir yöntem geliştirilmiştir. Bu süreler i7 2.0 GHz işlemci ve 8 GB Ram belleğe sahip standart bir son kullanıcı bilgisayarında alınmış olup, daha güçlü makinelerde çok daha kısa sürelerde bu model işlevini yerine getirebilecektir.

6.1. Önerilen Modelin Benzer Çalışmalar ile Karşılaştırması

Günümüzde birçok farklı yöntem kullanılarak otomatik konuşma tanıma sistemleri geliştirilmekte ve kullanıcılara sunulmaktadır. Tanıma performansının seviyesi ile sinyal durumu, gürültü, veri seti büyüklüğü gibi durumlara göre değişkenlik gösterebilmektedir. Bu değişkenliği tolere edebilmek için hatalı sonuçları bulma ve düzeltme ile ilgili farklı özgün yaklaşımlar önerilmektedir. Her bir yaklaşımın kendine göre avantajlı ve dezavantajlı noktası olmak ile birlikte, genel tanıma performanslarına değişen oranlarda katkı

sunmaktadır. Bu alanda yapılmış çalışmalardan bazıları detayları ile birlikte Çizelge 6.6'da gösterilmiştir. Çizelge incelendiğinde, farklı diller için yapılan bu çalışmalarda o dile özgü veri setleri kullanılmıştır. Tümünde sürekli konuşma tanıma sistemi olarak geliştirme yapılmıştır. Kullanılan yöntemlerden birçoğu özgün yaklaşımlar olmak ile birlikte N-gram gibi geleneksel dil modeli yaklaşımlarda tercih edilmiştir. Bu çalışmaların ortalama performans katkı düzeyleri %2,25 ile %10,78 aralığında değişkenlik göstermektedir. Artış seviyesi modelin standart tanıma performansı, kullanılan veri setinin büyüklüğü ve önerilen modelin katkısına göre değişkenlik göstermektedir. Önerilen model çizelge de verilmiş olan benzer çalışmalar ile karşılaştırıldığında, %4,60 seviyesinde katkı sunmakta olup diğer çalışmalar ile benzer düzeyde performans artışı sağlamıştır.

Çizelge 6.6. Benzer hata düzeltme algoritması önerileri ve katkı seviyeleri

Yayın Bilgisi	Kullanılan/Önerilen Yöntem	Konuşma Tipi	Veriseti	Ortalama Performans Artış Seviyesi
Yohei ve ark. [173]	Ngram + normalize uzaklık mesafesi	Sürekli Konuşma Tanıma	Veriseti: CSJ konuşma veritabanı, MFKK, 2596 eğitim videosu, 520 bin kelime, Japonca	%38,82 ->%28,04 %10,78 performans artışı
Yusuke Nakashima ve ark. [174]	Özgün model önerisi	Sürekli Konuşma Tanıma	gazete ve internet verisi, Japonca	%8,6 -> %4,3 gazete verisi %4,3 performans artışı %9,8 -> %6,5 internet verisi %3,3 performans artışı
Yu ve ark. [175]	Sözlük uyarlama, dil modeli ayarlama, yeni kelimeler ekleme, yeni telaffuzları öğrenme ve dil modeli uyarlama	Sürekli Konuşma Tanıma	Microsoft ses verisi, İngilizce	sistem performansının %10'u aştığı durumlar için %11 performans artışı
Yongmei Shi ve Lina Zhou [176]	Gürültülü veri ile çalışabilen özgün model önerisi	Sürekli Konuşma Tanıma	32 ingilizce cümle, 71800 kelime, İngilizce	%61 -> %53 gürültülü veri %8 performans artışı

Çizelge 6.6. (Devam) Benzer hata düzeltme algoritması önerileri ve katkı seviyeleri

Yayın Bilgisi	Kullanılan/Önerilen Yöntem	Konuşma Tipi	Veriseti	Ortalama Performans Artış Seviyesi
Bassil Youssef ve Paul Semaan [177]	N-gram + Özgün model önerisi	Sürekli Konuşma Tanıma	farklı İngilizce makaleler, 5 farklı konuşmacıdan 100 farklı kelime , Microsoft N-gram veriseti, İngilizce	
Arslan, R.S. ve Barışçı, N. [37]	Özgün Model Önerisi	Sürekli Konuşma Tanıma	Metu 1.0 veriseti, Zemberek, Boun Kelime Seti, Türkçe	%2,25 performans artışı
Önerilen Model	Alternatif hipotez temelli özgün model önerisi	Sürekli Konuşma Tanıma	Metu 1.0 veriseti, Zemberek, Boun Kelime Seti, Türkçe	%4,60 performans artışı

Performans artışına yönelik olarak yapılan çalışmaların yanında genel tanıma sonuçlarına göre değerlendirmenin yapılması ve bu çalışmada önerilen model ile yakalanan %70 ve üzerindeki tanıma oranı ile diğer çalışmaların karşılaştırılması mümkündür. Bu karşılaştırmada Türkçe ile ilgili yapılmış çalışmalar değerlendirilmiştir ve sonuçları Çizelge 6.7'de verilmiştir. Çizelge incelendiğinde, Türkçe ile ilgili yapılmış çalışmalarda Saklı Markov Model, Gaussian Karışım Modeli, Karar Destek Vektörleri gibi farklı yöntemlerde kullanılmaktadır. Akustik veri seti olarak tümünde Metu 1.0 kullanılmaktadır. Ancak verisetinde probleme özel olarak alt kategorilere ayırma veya verisetinin belirli bir bölümünü kullanma gibi farklılıklar olmaktadır. Veriseti kullanım farklıklarına ve yapılan çalışmaların kapsamlarına göre değişkenlik göstermekle birlikte %64,19 ile %100 arasında değişen seviyelerde tanıma performansları yakalanmıştır. Çalışmaların tamamında sözcük altı birimler ile çalışılmış olup, Büyük ve ark. [178] tarafından yapılan çalışmada sözcük temelli yaklaşım ile %52,95 seviyelerinde bir tanıma yapılabilmiştir. Oran oldukça düşük kalmakta olup, Türkçe dili ile ilgili konuşma tanıma sistemlerinde sözcük yerine alt birimler ile çalışma daha yaygın olarak tercih edilmiştir.

Çizelge 6.7. Metu 1.0 veri setini kullanan farklı yaklaşımların hata oranı sonuçları

Çalışma Adı	Tanıma Yaklaşımı	Veriseti	Sistem Tanıma Performansı
Keser ve ark. [179]	Ortak Vektör Yaklaşımı (OVY)	Metu 1.0 Veriseti	%70
Aksoylar ve ark.[28]	SMM	Metu 1.0 Veriseti, SUVoice	Spor haberleri: %63
Salor ve ark. [70]	SMM	Metu 1.0 Veriseti	%70,8
Çiloğlu ve ark. [55]	SMM, N-gram	Metu 1.0 Veriseti	%64,19
Büyük ve ark. [178]	SMM	Metu 1.0 Veriseti -2151 Test Data	Sözcük: %52,95 Hece: %68,80 Yarı-sözcük:%45,11
Bu tez çalışmasında önerilen model	TSA, UKSB	Metu 1.0 Veriseti	10 kelime: 100 100 kelime: %99,2 1000 kelime: %71,2

Günümüzde kullanılmakta olan “Apple Siri”, “Whatsapp Speech Dictation” ve “Google Speech” programlarının Türkçe dil desteği bulunmaktadır. Bu programlardan “Apple Siri” ile metin diktesi işlemi yapılamamaktadır. Kısa seslendirmelerin metne dönüştürülmesi için uygundur. “Whatsapp Dictation” ile yapılan testlerde ortamdaki gürültü durumunun çıktıları doğrudan etkilediği ve tanıma performansını düşüren bir unsur olduğu tespit edilmiştir. Son olarak “Google Speech” bu programlar içerisinde en başarılı sonuçların alınabildiği Türkçe konuşma tanıma sistemi olmuştur. Bu tez çalışmasında kullanılan metu 1.0 veri seti içerisinden seçilmiş ve içerisinde rastgele 100 kelime barındıran bir akustik set kullanılarak bu uygulamalar değerlendirilmiş ve sonuçları karşılaştırmalı olarak Çizelge 6.8’de gösterilmiştir.

Çizelge 6.8. Önerilen modelin ticari uygulamalar ile karşılaştırma sonuçları

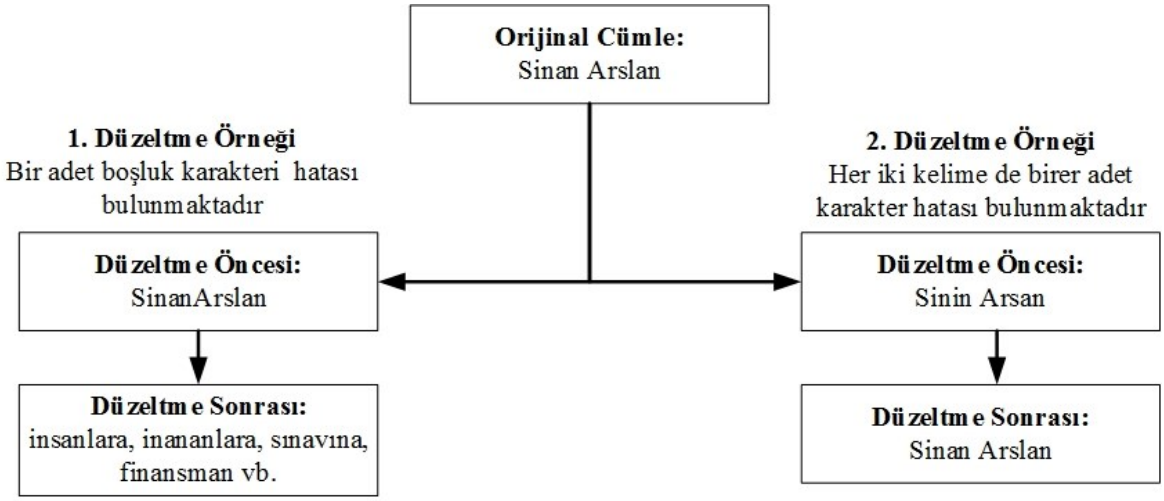
Uygulama Adı	Sınıflandırma Yöntemi	Konuşma Türü	Veri Seti	Sistem Tanıma Performansı
Apple Siri Dictation [180]	Derin Sinir Ağları	Kısa Cümleler	Metu 1.0 Veriseti	%83,7
Whatsapp Dictation [181]	Derin Sinir Ağları	Sürekli Konuşma Tanıma	Metu 1.0 Veriseti	%85,2
Google Speech [182]	Derin Sinir Ağları	Sürekli Konuşma Tanıma	Metu 1.0 Veriseti	%98,83
Bu tez çalışmasında önerilen model	TSA, UKSB, Derin sinir ağı	Sürekli Konuşma Tanıma	Metu 1.0 Veriseti	%99,2

Çizelge 6.8 incelendiğinde, en başarılı sonuçların %99,2 ile bu tez çalışmasında önerilen uygulama ile alındığı tespit edilmiştir. Bunun yanında "Google Speech" ile %98,83, "Apple Siri" ile %83,7 ve "Whatsapp Dictation" ile %85,2 başarı seviyesi yakalanmıştır. Tüm modellerde derin sinir ağı yapısı kullanılmış olup, sürekli konuşma tanıma sistemi şeklinde tasarlanmıştır. Çizelgede verilmiş olan tüm uygulamalarda gürültüsüz ve yavaş konuşmaların daha başarılı olarak tanındığı ancak tersi durumda tanıma başarı seviyelerinin hızlı bir şekilde düştüğü gözlemlenmiştir.

Bu tez çalışmasında önerilen modelin tüm detayları, farklı test sonuçları bu bölüm içerisinde verilmiş ve değerlendirilmiştir. Son durumda ortalama olarak %4,60 seviyesinde bir performans artışı yakalanmış ve dağarcık boyutuna göre %66,8 ile %100 aralığında değişen oranlarda genel tanıma performans değeri yakalanmıştır.

6.1. Önerilen Modelin Kısıtları ve Çözüm Önerileri

Bu tez çalışmasında önerilen özgün yaklaşım otomatik konuşma tanıma sistemleri için iyi sonuçlar üretmiş olsa da test aşamasında tespit edilmiş olan bazı kısıtlı durumları bulunmaktadır. Bu problemlerin çözülmesi için özgün yaklaşımlara ihtiyaç duyulmaktadır.



Şekil 6.4. Boşluk karakter problemi ve bir karakter hatası problemi

Bu çalışmada önerilen algoritmanın performans katkısının belirli seviyelerde kalmasının en önemli sebebi hatalı çıktılarda bulunan boşluk karakteri problemidir. Şekil 6.4'te gösterildiği gibi, "SinanArslan" şeklinde üretilen bir çıktı için alternatif kelime önerileri "insanlara,

inananlara, sinavina …” kelimeleri olmaktadır. Ancak düzeltme işleminde ihtiyaç duyulan tek işlem “Sinan” ve “Arslan” kelimeleri arasına boşluk karakteri eklemektir. Bu sorunun çözülememesi sebebiyle boşluk karakter hatası bulunduran tüm durumlarda hatalı şekilde düzeltme yapılmaktadır. Bu durum önerilen modelin katkı seviyesini düşürmektedir. Bu sorunun çözülmesi halinde %4,60 olan katkı düzeyi daha yüksek seviyelere çıkacaktır.

Önerilen modelin bir diğer problemi de hatalı kelime de çok sayıda hatalı karakterin bulunması durumunda alternatif kelime öneri sayısının oldukça fazla olabilmesi durumudur. Böylece hangi kelimenin doğru kelime olabileceği tahmin edilememekte ve düzeltme işleminde sorunlar yaşanmaktadır. Bu soruna kısmen çözüm üretmek üzere karakter sayısına eşik değer konulmuştur. Ancak bu öneri soruna kesin çözüm olmamıştır. Bu sorunlu durum katkı seviyesinin daha fazla artmasını engelleyen bir husustur.

6.2. Özgün Modelin Geliştirilmesine Yönelik Yeni Öneriler

Bu tez çalışmasında önerilen modelin testlerinde %4,60’lık performans artışı yakalanmıştır. Gelecekte önerilen bu modelin geliştirilmesi ve daha başarılı modellerin üretilmesine yönelik olarak birçok farklı çalışma yapılacaktır.

Önerilen model ile küçük dağarcıklı sistemlerde %100 başarı sağlanmıştır. Bu sebeple otonom araçlarda komut temelli ve konu bağımlı sistemler için önerilen modelin kullanılabilmesi mümkündür. Küçük dağarcıklı bir problem olan otonom araçlarda önerilen model ile başarılı sonuçlar üretilebilecektir. Bu sebeple bu modelin araçlarda kullanılma fikri test edilmeli ve sonuçları değerlendirilmelidir.

Bu çalışmada Akustik veri seti olarak Metu 1.0 kullanılmıştır. Bu set içerisinde duygu durumunun değişkenlik göstermesi durumu gözetilmemiştir. Ancak duygu durumuna göre ses frekansının değişimine bağlı olarak konuşma tanıma ve düzeltme modelinin nasıl değişkenlik göstereceği, hazırlanacak yeni bir akustik veri seti ile test edilecektir. Böylece önerilen modelin bu değişkenliğe göre sonuçlarının nasıl değişeceği gözlemlenecektir. Bu çalışmada önerilen modelde, akustik veri ile metin verisi arasındaki hizalamayı yapabilmek için bağlantıcı zamansal sınıflandırma kullanılmıştır.

7. SONUÇLAR

Otomatik konuşma tanıma sistemleri, konuşma sinyalinin alınması ve bilgisayarların işleyebileceği metin formatına dönüştürülmesi işlemlerini yapmaktadırlar. Bu işlemlerin yapılabilmesi için karmaşık bir algoritma ve modelleme, akustik ve metin veri seti ile çeşitli yazılım kütüphanelerine ihtiyaç duyulmaktadır. Günümüzde derin öğrenme mimarisi ile daha başarılı OKT sistemleri geliştirilmektir. Bunun en önemli sebebi de bilgi işlem kapasitelerinin artması ve GPU sistemlerinin yaygınlaşmasıdır. Daha başarılı modellerin geliştirilmesi sayesinde OKT sistemlerinin ticari uygulama olarak kullanımı yaygınlaşmaktadır.

Bu çalışmada derin öğrenme modellemesi kullanan uçtan uca Türkçe konuşma tanıma uygulaması geliştirilmiştir. Bu modellemenin yapılması için metu 1.0 akustik veri seti, özgün olarak hazırlanmış metin verisi, UKSB ve GTB bellek kullanılan tekrarlayan sinir ağı modeli kullanılmıştır. Bu tasarım farklı ortamlarda test edilmiştir. Bu testler sonucunda kelime dağarcığına göre değişmek ile birlikte 100 kelime dağarcığı ile çalışmalarda %100 performans seviyesi yakalanmıştır. Kelime dağarcığının büyümesi ile tanıma performansında düşüşler yaşanmış olmak ile birlikte en düşük seviye de %71,4 olmuştur.

Bu çalışma için tasarlanan modelin performansının %71,4 seviyesinde olması sebebiyle, hatalı çıktıların bulunması ve düzeltilmesine yönelik olarak alternatif hipotez önerisi yaklaşımına dayalı özgün bir model önerilmiştir. Bu model ile Türkçe OKT sistemlerinde performans artışlarının yakalanması amaçlanmıştır. Yöntem OKT çıktılarındaki hatalı durumları tespit edilerek referans şablon kelimeler veri tabanında bulunan kelimeler ile değiştirilmesi temeline dayanmaktadır. Bu karşılaştırma sonucunda hatalı karakterler düzeltilmektedir. Bu düzeltme sonucunda genel sistem performansında ortalama olarak %4,60 seviyesinde artış yakalanmıştır.

Bu artış seviyesi farklı diller için hazırlanmış hata düzeltme algoritmaları ile karşılaştırıldığında benzer seviye de katkı sunduğu tespit edilmiştir. Benzer şekilde Türkçe diline özgü olarak hazırlanmış çalışmalar ile karşılaştırma yapıldığında ise son durumda daha başarılı sonuçlar üretebildiği anlaşılmıştır.

Bu çalışmada önerilen özgün model ile Türkçe OKT sistemlerinin tanıma performanslarının artırılması ve daha başarılı sistemlerin üretilebilmesi amaçlanmıştır.

KAYNAKLAR

1. Asefisaray, B. (2018). U*çtan uca konuşma tanıma modeli: Türkçe'deki deneyler*, Doktora Tezi, Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 10-85.

2. Köklükaya, E. ve Coşkun, İ. (2003). Endüktif öğrenmeyi kullanarak konuşmayı tanıma. *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 7(1), 1-8.

3. Yalçın, N. (2008). Konuşma tanıma teorisi ve teknikleri. *Kastamonu Eğitim Dergisi*, 16(1), 249-266.

4. Gürel, A. ve Arslan, L. M. (2008). Konuşma tanıma için insan-makine karşılaştırması. *Dilbilim Araştırmaları Dergisi*, 77-90.

5. Anusuya, M. A. ve Katti, S. K. (2009). Speech recognition by machine: a review. *International Journal of Computer Science and Information Security*, 181-205.

6. Kimanuka, U. A. ve Büyük, O. (2018). Turkish speech recognition based on deep neural networks. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22, 319-329.

7. Juang, B.H. ve Rabiner, L. (2005). Automatic speech recognition – A brief history of the technology development. *Encylopedia of Language and Linguistics*, 1-24.

8. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y. ve Schwarz, P. (2011, Aralık). *The Kaldi speech recognition toolkit*. IEEE Workshop on Automatic Speech Recognition and Understanding, Hawaii, 1-4.

9. Graves, A. ve Jaitly, N. (2014, Haziran). *Towards end-to-end speech recognition with recurrent neural networks*. Proceedings of the 31st International Conference on Machine Learning, Beijing, 1-9.

10. İnternet: Türkçe'nin oluşumu ve Türkçe'nin dünya dilleri arasındaki yeri. Url: http://www.turkcede.org/turk-dili/729-turkcenin-olusumu-ve-turkcenin-dunya-dilleri-arasindaki-yeri.html, Son Erişim Tarihi: 20.12.2019.

11. Palaz, H., Kanak, A., Bicil, Y., Doğan, M. U. ve İslam, T. (2005, Mayıs). *TREN- Turkish speech recognition platform*. Proceedings of the IEEE 13th Signal Processing and Communications Applications Conference (SIU), Antalya, 1-4.

12. Hankamer, J. (1989). Lexical Representation and Process, Morphological Parsing and the Lexicon. Cambridge: MIT Press, 392-408.

13. Akın, C. (2019). *Voice recognition system with score level fusion methods and embedded system design*, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 1-25.

14. Arpacı, Y. (2019). *Environmental sound recognition with various feature extraction and classification techniques*, Yüksek Lisans Tezi, Ankara Yıldırım Beyazıt Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 1-26.

15. Abdalrahman, R. S. (2018). *Konuşmacı ve konuşma tabanlı kaskad biyometrik kontrol sistemi tasarımı*, Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 1-28.

16. Koç, F. (2019). *Spektrogram tekniği kullanılarak derin öğrenme yöntemleri ile ses tanım*a, Yüksek Lisans Tezi, Van Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü, Van, 1-17.

17. Sofuoğlu, İ. (2017). Speech recognition algorithm implementation for remote controlling unmanned aerial vehicles (UAVs), Yüksek Lisans Tezi, Yaşar Üniversitesi Fen Bilimleri Enstitüsü, 1-19.

18. Anacur, C. A. (2019). *Doğrusal tahmini kodlama yöntemi kullanılarak trafikteki uyarıcı seslerin tespiti*, Yüksek Lisans Tezi, Van Yüzüncü Yıl Üniversitesi, Fen Bilimleri Enstitüsü, 1-17.

19. Arslan, Y. (2018). *Detection and recognition of sounds from hazardous events for surveillance applications*, Doktora Tezi, Ankara Yıldırım Beyazıt Üniversitesi Fen Bilimleri Enstitüsü, 1-25.

20. Büyük, O. (2018). Mobil araçlarda Türkçe konuşma tanıma için yeni bir veritabanı ve bu veritabanı ile elde edilen ilk konuşma tanıma sonuçları. *Pamukkale Üniversitesi Mühendislik Bilim Dergisi*, 24(2), 180-184.

21. Arısoy, E., Dutağacı, H. ve Arslan, L. M. (2006). A unified language model for large vocabulary continuous speech recognition of Turkish. *Signal Processing*, 86, 2844-2862.

22. Oflazer, K., Hakkani-Tür, D. Z. ve Tür, G. (1999, Haziran). *Design for a Turkish treebank*. 37th Annual Meeting of the Association for Computational Linguistics, Maryland, 1-9.

23. Şen, K. Ö., Köse, C. ve Aras, S. (2007, Mayıs). *Bir konuşmacının konuştuğu dilin belirlenmesi*. 3. İletisim Teknolojileri Ulusal Sempozyumu, Adana, 1-7.

24. Akın, A. A., Demir, C. ve Doğan, M .U. (2012, Nisan). K*elime altı dil modelleme yöntemlerinin Türkçe konuşma tanıma için iyileştirilmesi*. Signal Processing and Communications Applications Conference (SIU), Fethiye, 1-4.

25. Eşref, Y. ve Can, B. (2019, Nisan). *Using morpheme-level attention mechanism for Turkish sequence labeling*. Signal Processing and Communications Applications Conference (SIU), Sivas, 1-4.

26. Oflazoğlu, Ç. ve Yıldırım, S. (2012, Nisan). *Türkçe konuşmada kızgın duygunun akustik bilgi kullanılarak tanınması*. Signal Processing and Communications Applications Conference (SIU), Muğla, 1-4.

27. Karagöz, G. ve Demiroğlu, C. (2012, Nisan). *Sesli yanıt çağrı akışında dilbilgisi tabanlı Türkçe konuşma tanıma sistemi tanıtımı*. Signal Processing and Communications Applications Conference (SIU), Muğla, 1-2.

28. Aksoylar, C., Mutluergil, S. O. ve Erdogan, H. (2009, Nisan). *The anatomy of a Turkish speech recognition system*. Proceedings of the IEEE Signal Processing and Communications Applications Conference (SIU), Antalya, 512-515.

29. Arısoy, E. ve Saraçlar, M. (2007, Haziran). *Türkçe haber programları için konuşma tanıma*. Signal Processing and Communications Applications Conference (SIU), Eskişehir, 1-4.

30. Oflazoğlu, Ç. ve Yıldırım, S. (2011, Nisan). *Türkçe duyusal konuşma veritabanı*. Signal Processing and Communications Applications Conference (SIU), Antalya, 1153-1156.

31. Uzun, İ., Arısoy, E., Edizkan, R. ve Saraçlar, M. (2008, Nisan). D*ağıtık yapıda Türkçe sürekli konuşma tanıma sisteminde seyrek paket kayıplarının analizi ve telafisi*. Signal Processing and Communications Applications Conference (SIU), Didim, 1-4.

32. Peker, O. (2016, Mayıs). *Türkçe kelime ile ses tanıma*. Signal Processing and Communications Applications Conference (SIU), Zonguldak, 1737- 1740.

33. Uzun, İ. ve Edizkan, R. (2008, Nisan). *Dağıtık yapıdaki Türkçe sürekli konuşma tanıma sisteminin patlatmalı paket kaybı durumda başarımının iyileştirilmesi*. Signal Processing and Communications Applications Conference (SIU), Didim, 1-4.

34. Susman, D., Köprü, S. ve Yazıcı, A. (2012, Nisan). *Kısıtlı ses külliyatı ile Türkçe geniş dağarcıklı sürekli konuşma tanıma*. Signal Processing and Communications Applications Conference (SIU), Muğla, 1-4.

35. Bayer, A. O., Çiloğlu, T. ve Yöndem, M. T. (2006, Nisan). *Türkçe konuşma tanıma için farklı dil modellerinin incelenmesi*. Signal Processing and Communications Applications Conference (SIU), Antalya, 1-4.

36. Büyük, O., Haznedaroğlu, A. ve Arslan, L. M. (2007, Haziran). *Dil modeli uyarlanabilir Türkçe ses tanıma yazılımı*. Signal Processing and Communications Applications Conference (SIU), Eskişehir, 1-4.

37. Arısoy, E. ve Saraçlar, M. (2006, Nisan). *Türkçe geniş dağarcıklı konuşma tanıma sistemleri için örünün yeniden değerlendirilmesi tabanlı dil modellemesi yaklaşımları*. Signal Processing and Communications Applications Conference (SIU), Antalya, 1-4.

38. Tombaloğlu, B. ve Erdem, H. (2016, Mayıs). *MFCC-SVM tabanlı Türkçe konuşma tanıma sisteminin geliştirilmesi*. Signal Processing and Communications Applications Conference (SIU), Zonguldak, 929-932.

39. Çeliktaş, H. ve Hanilçi, C. (2017, Mayıs). *Türkçe metne bağlı konuşmacı tanıma üzerine bir çalışma*. Signal Processing and Communications Applications Conference (SIU), Antalya, 1-4.

40. Keser, S. ve Edizkan, R. (2009, Nisan). A*ltuzay sınıflama ile sesbirim temelli Türkçe yalıtık Türkçe kelime tanıma*. Signal Processing and Communications Applications Conference (SIU), Antalya, 93-96.

41. Yılmaz, O. ve Saraçlar, M. (2011, Nisan). *Türkçe haber programlarının yazılandırılması için işitsel bölütleme*. Signal Processing and Communications Applications Conference (SIU), Antalya, 379-382.

42. Arısoy, E. (2017, Mayıs). *Türkçe sözlü ders anlatımları için otomatik yazılandırma ve geri getirim sistemi geliştirilmesi*. Signal Processing and Communications Applications Conference (SIU), Antalya, 1-4.

43. Aşlıyan, R., Günel, K. ve Yakhno, T. (2008, Nisan). *Dinamik zaman bükmesi ve çok katmanlı algılayıcı kullanarak hece tabanlı Türkçe konuşma tanıma*. Signal Processing and Communications Applications Conference (SIU), Çanakkale, 1-8.

44. Kuru, K., Arda, K. ve Baykal, N. (2007, Haziran). *Konuşma tanıma sistemi kullanılarak bilgisayar ile etkileşimli yapısal tıbbi rapor hazırlanması*. Signal Processing and Communications Applications Conference (SIU), Eskişehir, 1-7.

45. Büyük, O., Erdoğan, H. ve Oflazer, K. (2005, Mayıs). *Konuşma tanımada karma dil birimleri kullanımı ve dil kısıtlarının gerçeklenmesi*. Signal Processing and Communications Applications Conference (SIU), Kayseri, 111-114.

46. Can, D. ve Saraçlar, M. (2009, Nisan). *Türkçe haber bültenlerinin açık kaynak yazılımları ile yazılandırılması*. Signal Processing and Communications Applications Conference (SIU), Antalya, 325-328.

47. Edizkan, R., Tiryaki, B., Büyükcan, T. ve Uzun, İ. (2007, Şubat). *Sesli komut tanıma ile gezgin araç kontrolü*. 9. Akademik Bilişim Konferansı, Kütahya, 1-10.

48. Uzun, İ. ve Edizkan, R. (2008, Nisan). *Dağıtılmış sistemlerde internet protokolü üzerinden yalıtık ses tanıma*. Signal Processing and Communications Applications Conference (SIU), Didim, 1-4.

49. Bayğın, M. ve Karaköse, M. (2012, Nisan). G*erçek zamanlı ses tanıma tabanlı akıllı ev uygulaması*. Signal Processing and Communications Applications Conference (SIU), Muğla, 1-4.

50. Gülmezoğlu, M. B., Dzhafarov, V., Edizan, R. ve Barkana, A. (2007). The common vector approach and its comparison with other subspace methods in case of sufficient data. *Computer Speech and Language*, 21(2), 266-281.

51. Türk, O. ve Arslan, L. M. (2004, Nisan). *Konuşma terapisine yönelik konuşma tanıma yöntemleri*. Signal Processing and Communications Applications Conference (SIU), Kuşadası, 410-413.

52. Caner, M. ve Üstün, S. V. (2006). Yapay sinir ağları ile konuşmacı kimliğini tanıma uygulaması. *Journal of Engineering Sciences*, 12(2), 279-284.

53. Arısoy, E. ve Arslan, L. M. (2005, Mayıs). *Türkçe gazete haberleri dikte sistemi*. Signal Processing and Communications Applications Conference (SIU), Kayseri, 629-632.

54. Aksungurlu, T., Parlak, S., Sak, H. ve Saraçlar, M. (2008, Nisan) *Türkçe haber bültenleri için dil modelleme yaklaşımlarının karşılaştırılması*. Signal Processing and Communications Applications Conference (SIU), Zonguldak, 1-4.

55. Çiloğlu, T., Çömez, M. ve Sahin, S. (2004, Nisan). *Language modelling for Turkish as an agglutinative languages*. Signal Processing and Communications Applications Conference (SIU), Kusadası, 1-2.

56. Oflazoğlu, Ç. ve Yıldırım, S. (2015, Mayıs). *Türkçe duygusal konuşma için duygu sınıflarının ikili sınıflandırma performansları*. Signal Processing and Communications Applications Conference (SIU), Malatya, 2353-2356.

57. Yalçın, N. ve Bay, Ö. F. (2007). HTK ile konuşmacıdan bağımsız Türkçe konuşma tanıma sistem oluşturma. *Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi Dergisi*, 21, 79-97.

58. Liu, C., Zhang, Y., Zhang, P. ve Wang, Y. (2018, Kasım) *Evaluating modeling units and sub-word features in language models for Turkish ASR*. International Symposium on Chinese Spoken Language Processing, Taipei City, 1-5.

59. Gelegin, İ. ve Bolat, B. (2011, Ekim). *Ayrık kelime tabanlı bir konuşma tanıma sistemiyle bilgisayar kontrolü*. Elektrik-Elektronik ve Bilgisayar Sempozyumu, Elazığ, 1-4.

60. Asefisaray, B., Mengüşoğlu, E., Hacıömeroğlu, M. ve Sever, H. (2016). Türkçe ses tanıma sistemlerinde dil modeli boyutunun doğruluk oranına etkisi. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 100-105.

61. Özbey, C. ve Bayar, S. (2017, Şubat). Otomatik ses tanıma: Türkçe için genel dağarcıklı akustik model oluşturulması ve test edilmesi. 19. Akademik Bilişim Konferansı, Aksaray, 1-6.

62. Çakır, H. ve Okutan, B. (2011). Ses kontrollü web tarayıcı. *Bilişim Teknolojileri Dergisi*, 4(1), 1-6.

63. Sak, H., Saraçlar, M. ve Güngör, T. (2012). Morpholexical and discriminative language models for Turkish automatic speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 20(8):2341-2351.

64. Can, B. ve Artuner, H. (2013, Aralık). *A syllable-based Turkish speech recognition system by using time delay neural network (TDNN)*. International Conference on Soft Computing and Pattern Recognition (SoCPaR), Hanoi, 219-224.

65. Akyol, A. ve Erdoğan, H. (2004). Filler model based confidence measures for spoken dialogue systems: a case study for Turkish. *Acoustics, Speech and Signal Processing*, 1-5.

66. Uzun, İ. ve Edizkan, R. (2011, Haziran). Performance improvement in distributed Turkish continuous speech recognition system using packet loss concealment techniques. Innovations in Intelligent Systems and Applications (INISTA), İstanbul, 375-378.

67. Sak, H., Saraçlar, M. ve Güngör, T. (2010, Nisan). *Morphology-based and sub-word language modelling for Turkish speech recognition*. Signal Processing and Communications Applications Conference (SIU), Diyarbakır, 5402-5405.

68. Yalçın, N. ve Altun, Y. (2012, Haziran). A sample work developed using speech recognition technology about basic language education: speak and learn with speech recognition. Second World Conference on Educational Technology Researches, Nicosia, 342-346.

69. Arısoy, E. ve Arslan, L. M. (2004, Eylül). *Turkish radiology dictation system*. 9th Conference Speech and Computer, Saint-Petersburg, 1-5.

70. Salor, Ö., Pellom, B. L., Çiloğlu, T. ve Demirekler, M. (2007). Turkish speech corpora and recognition tools developed by porting sonic: towards multilingual speech recognition. *Computer Speech and Language*, 21(4), 580-593.

71. Erdoğan, H., Büyük, O. ve Oflazer, K. (2005, Mayıs). *Incorporating language constraints in sub-word-based speech recognition*. Signal Processing and Communications Applications Conference (SIU), Kayseri, 1-6.

72. Arısoy, E. ve Arslan, L. M. (2005, Mayıs). T*urkish dictation system for broadcast news applications*. Signal Processing and Communications Applications Conference (SIU), Kayseri, 629-632.

73. Çarkı, K., Geutner, P. ve Schultz, T. (2000, Haziran). *Turkish LVCSR: towards better speech recognition for agglutinative language*. Signal Processing and Communications Applications Conference (SIU), Antalya, 1563-1566.

74. Yakar, Ö. ve Aşlıyan, R. (2016, Şubat). *Turkish speech recognition using hidden markov model*. Akademik Bilişim Konferansı, Aydın, 1-7.

75. Hakkani-Tür, D. Z., Oflazer, K. ve Tür, G. (2002). Statistical morphological disambiguation for agglutinative languages. *Computers and the Humanities*, 381-410.

76. Arslan, R. S. ve Barışçı, N. (2019). Development of output correction methodology for long short-term memory-based speech recognition. *Sustainability*, 11, 4250.

77. Arslan, R. S. ve Barışçı, N. (2018, Eylül). Farklı optimizasyon tekniklerinin bağlantıcı zamansal sınıflandırma kullanılan uçtan uca Türkçe konuşma tanıma sistemlerine etkisi. 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, 1-6.

78. Eray, O., Tokat, S. ve İplikci, S. (2018, Mart). *An application of speech recognition with support vector machines*. International Symposium on Digital Forensic and Security (ISDFS), Antalya, 1-6.

79. Korkmaz, Y., Boyacı, A. ve Tuncer, T. (2019). Turkish vowel classification based on acoustical and decompositional features optimized by genetic algorithm. *Applied Acoustics*, 154, 28-35.

80. Tran, D. T. (2000). *Fuzzy approaches to speech and speaker recognition*, Doktora Tezi, University of Canberra, Canberra, 70-85.

81. Yalçın, N. (2006). Konuşma tanıma teknoloji yardımıyla ilköğretim birinci sınıf öğrencilerine ilkokuma yazma öğretimi için bir yazılım geliştirme, Doktora Tezi, Gazi Üniversitesi Fen Bilimler Enstitüsü, 1-10.

82. Campbell, J. (1997). Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9), 1437-1462.

83. Muthusamy, Y. K., Barnand, E. ve Cole, R. A. (1994). Reviewing automatic language identification. *IEEE Signal Processing Magazine*, 11(4), 33-41.

84. Pedro, T., Gleason, T., McCree, A., Reynolds, D., Singer, E., Sturim, D. ve Richardson, F. (2010, Mart). *The MİTLL NEST LRE 2009 language recognition system*. International Conference on Acoustics, Speech and Signal Processing, Dallas, 4994-4997.

85. Dominguez, J. G., Moreno, I. L., Pedrosa, J. F., Ramos, D., Toledano, D. T. ve Rodrigues, J. G. (2010). Multilevel and session variability compensated language recognition: TVS-UAM systems at NIST LRE 2009. *IEEE Journal of Selected Topics in Signal Processing*, 4(6), 1084-1093.

86. Zissman, M. A. ve Berkling, K. M. (2001). Automatic language identification. *Speech Communication*, 35, 115-124.

87. Davis, S. B. ve Mermelstein, P. (1990). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *In Readings in Speech Recognition*, 65-74.

88. Tiwari, V. (2010). MFCC and its applications in speaker recognition. *International Journal on Emerging Technologies*, 19-22.

89. Karpagavalli, S. ve Chandra, E. (2016). A review on automatic speech recognition architecture and approaches. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 393-404.

90. Williams, J., Hinton, G. ve Rumelhart, D. E. (1986). Learning representations by back-propagating errors, *Nature*, 323(6088), 533-536.

91. Manning, C. D. ve Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge: MIT Press, 100-125.

92. Chen, S. F. ve Goodman, J. (1999). An empirical study of smoothing techniques for language modeling. *Computer, Speech and Language*, 13(4), 359-393.

93. Ali, A., Renals, S. (2018, Ocak). *Word error rate estimation for speech recognition: e-WER*. Proceedings of the 56. Annual meeting of the association for computational linguistics, Melbourne, (2), 1-5.

94. Shipra J. A. ve Singh, R. P. (2012). Automatic speech recognition: a review. *International Journal of Computer Applications*, 60(9), 1-11.
95. Arslan, R. S. (2018). *Türkçede derin öğrenme ile konuşma tanıma uygulamaları*, Uzmanlık Tezi, Radyo ve Televizyon Üst Kurulu, Ankara, 52-120.

96. McCulloch, W. S. ve Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 15-133.

97. Cybenko, G. (1989). Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4), 303-314.

98. Wan, J., Wang, D., Hoi, S., Wu, P., Zhu, J., Li, J. ve Zhang, Y. (2014, Kasım). *Deep learning for content-based image retrieval: a comprehensive study*. Proceedings of the 22nd ACM International Conference on Multimedia, Orlando, 157-166.

99. Anwer, A. ve Mahmood, O. (2017). *Derin öğrenme yöntemleri ile göğüs kanseri teşhisi*, Yüksek Lisans Tezi, THK Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 25-37.

100. Deng, L., Jinyu L., Huang J. T., Yao, K., Yu, D., Seide, F., Seltzer, M., Zweig G., He, X., Williams, J., Gong, Y. ve Acero, A. (2013, Mayıs). *Recent advances in deep learning for speech research at Microsoft*. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, 8604–8608.

101. Agarwalla, S. ve Sarma, K. (2016). Machine learning based sample extraction for automatic speech recognition using dialectal Assamese speech. *Neural Network*, 78, 97-111.

102. Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H. G. ve Ogata, T. (2015). Audio-visual speech recognition using deep learning. *Applied Intelligence*, 42(4), 722-737.

103. Deng, L., Hinton, G. ve Kingsbury, B. (2013, Mayıs). *New types of deep neural network learning for speech recognition and related applications: an overview*. International Conference on Acoustics, Speech and Signal Processing, Vancouver, 5899-8603.

104. Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N. ve Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition - the shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6), 82-97.

105. İnternet: Keras: the python deep learning library. Url: https://keras.io, Son Erişim Tarihi: 21/12/2019.

106. Hochreiter, S. ve Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.

107. Yang, T., Tseng, T. ve Chen, C. (2016, Kasım). *Recurrent neural network-based language models with variation in net topology, language and granularity*. International Conference on Asian Language Processing (IALP), Tainan, 1-4.

108. Xiao, Y. ve Keung, J. (2018, Aralık). *Improving bug localization with character-level convolutional neural network and recurrent neural network.* 25th Asia-Pacific Software Engineering Conference (APSEC), Nara, 1-2.

109. Güngör, O., Üsküdarlı, S. ve Güngör, T. (2018, Mayıs). *Recurrent neural networks for Turkish named entity recognition*. Signal Processing and Communications Applications Conference (SIU), İzmir, 1-4.

110. Basnet, A. ve Timalsina, A. K. (2018, Ekim). *Improving Nepali news recommendation using classification based on LSTM recurrent neural networks*. International Conference on Computing, Communication and Security (ICCCS), Nepal, 1-5.

111. Yılmaz, E. ve El-Kahlout, İ. (2014, Nisan). *The use of recurrent neural networks language model in Turkish-English machine translation*. 22nd Signal Processing and Communications Applications Conference (SIU), Trabzon, 1-4.

112. Adlaon, K. M. ve Marcos, N. (2018, Kasım). *Neural machine translation for Cebuano to Tagalog with subword unit translation*. International Conference on Asian Language Processing (IALP), Bandung, 1-6.

113. Zhao, B. ve Tam Y. C. (2014). Bilingual recurrent neural networks for improved statistical machine translation. *Spoken Language Technology (SLT)*, 1-5.

114. Hermanto, A., Adji, T. B. ve Setiawan, N. A. (2015, Ekim). *Recurrent neural network language model for English-Indonesian machine translation: experimental study*. International Conference on Science in Information Technology (ICSITech), Yogyakarta, 1-5.

115. Safari, E. ve Zhang, S. (2003, Aralık). *Integrated recurrent neural network for image resolution enhancement from multiple image frames*. IEEE Proceedings Vision, Image and Signal Processing, Yogyakarta, 150(5), 1-5.

116. Castro, J. B., Feitosa, R. Q. ve Happ, N. P. (2018, Temmuz). *A hybrid recurrent convolutional neural network for crop type recognition based on multitemporal sar image sequences*. IGARSS-IEEE International Geoscience and Remote Sensing Symposium, Valencia, 1-4.

117. Qin, C., Schlemper, J., Caballero, J., Price, A. N. ve Rueckert D. (2018). Convolutional recurrent neural networks for dynamic MR image reconstruction. *IEEE Transactions on Medical Imaging*, 38(1), 280-290.

118. Li, Y., Liang, W. ve Tan, J. (2017, Aralık). *Bi-directional LSTM recurrent neural network for lumbar vertebrae identification in x-ray images.* International Conference on Computer Systems, Electronics and Control, Dalian, 1-5.

119. Mou, L. ve Zhu, X. X. (2018, Temmuz). *A recurrent convolution neural network for land cover change detection in multispectral images*. IGARSS International Geoscience and Remote Sensing Symposium, Valencia, 1-4.

120. Bengio, Y., Simard, P. ve Frasconi, P. (1994). Learning long-term dependencies with gradient descent in difficult. *IEEE Transactions on Neural Networks*, 5(2), 157-166.

121. Schmidhuber, J., Greff, K., Srivasava, R. K., Kutnik J. ve Steunebrink, B. R. (2017). LSTM: a search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222-2232.

122. İnternet: Understanding LSTM Networks. Url: https://colah.github.io/posts/2015-08-Understanding-LSTMs/, Son Erişim Tarihi: 21/12/2019.

123. Cho, K., Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H. ve Bengio, Y. (2014, Ekim). *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. In Proceedings of the Empiricial Methods in Natural Language Processing, Doha, 1-15.

124. Graves, A., Fernandez S., Gomez, F. ve Schmidhuber, J. (2006, Haziran). *Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks*. Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, 369-376.

125. İnternet: Sequence Modelling with CTC. Url: https://distill.pub/2017/ctc/, Son Erişim Tarihi: 21/12/2019.

126. Cernak, M. ve Tong, S. (2018, Nisan). *Nasal speech sounds detection using connectionist temporal classification*. International Conference on Acoustics, Speech and Signal Processing, Calgary, 1-5.

127. Wöllmer, M., Schuller, B. ve Rigoll, G. (2013, Mayıs). *Probabilistic ASR feature extraction applying context-sensitive connectionist temporal classification networks*. International Conference on Acoustics, Speech and Signal Processing, Vancouver, 7125-7129.

128. Bai, Y., Yi, J., Ni, H, Wen, Z., Liu, B., Li, Y. ve Tao, J. (2016, Ekim). *End to end keywords spotting based on connectionist temporal classification for mandarin*. International Symposium on Chinese Spoken Language Processing (ISCSLP), Tianjin, 1-5.

129. Salazar, J., Kirchhoff, K. ve Huang, Z. (2019, Mayıs). *Self-attention networks for connectionist temporal classification in speech recognition*. International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, 7115-7119.

130. Lee, D., Lim, M., Park, H., Kang, Y., Park, J., Jang, G. J. ve Kim, J. H. (2017). Long short-term memory recurrent neural network-based acoustic model using connectionist temporal classification on a large-scale training corpus. *China communications*, 14(9), 23-31.

131. Chan, W., Jaitly, N., Le, Q. ve Vinyals, O. (2016, Mart). *Listen, attend and spell: a neural network for large vocabulary conversational speech recognition*. International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shangai, 4960-1964.

132. Sutskever, I., Vinyals, O. ve Le, Q. V. (2014). Sequence to sequence learning with neural networks. *In Advances in Neural Information Processing Systems*, 1-9.

133. Wu, B., Li K., Ge F., Huang, Z., Yang, M., Siniscalchi, S. M. ve Lee, C. H. (2017). An end-to-end deep learning approach to simultaneous speech dereverberation and acoustic modelling for robust speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8), 1289-1300.

134. Xu, H., Ding, S. ve Watanabe, S. (2019, Mayıs). *Improving end-to-end speech recognition with pronunciation-assisted sub-word modelling*. International Conference on Acoustics, Speech and Signal Processing, Brighton, 1-5.

135. Sumit, S. H., Muntasir, T., Zaman, M. M. A, Nandi, R. N. ve Sourov, T. (2018, Eylül). *Noise robust end-to-end speech recognition for Bangla language*. International Conference on Bangla Speech and Language Processing (ICBSLP), Sylhet, 1-5.

136. Lu, L., Zhang, X. ve Renals, S. (2016, Mart). *On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition*. International Conference on Acoustics, Speech and Signal Processing, Shangai, 5060-5064.

137. Li, B., He, Y. ve Chang, S. (2018, Kasım). T*owards end-to-end speech recognition (Tutorial-4)*. International Symposium on Chinese Spoken Language Processing, Tapai, 1-17.

138. İnternet: LDC Team. TIMIT Acoustic-Phonetic Continuous Speech Corpus. Url: https:// https://catalog.ldc.upenn.edu/LDC93S1, Son Erişim Tarihi: 21/12/2019.

139. Maas, A., Xie, Z., Jurafsky, D. ve Ng, A. (2015). Lexicon-free conversational speech recognition with neural networks. *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 345 - 354.

140. Miao, Y., Gowayyed, M. ve Metze, F. (2015, Aralık). EESEN: *end-to-end speech recognition using deep RNN models and WSFT-based decoding*. Workshop on Automatic Speech Recognition and Understanding, Scottsdale, 167 - 174.

141. Krizhevskay, A., Sutskever, I. ve Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 1090 - 1098.

142. Rude, S. (2017). An overview of gradient descent optimization algorithms. *Insight Center for Data Analytics*, 1-14.

143. Agrawal, R. (2019). Optimization algorithms for Deep Learning. Url: https://medium.com/analytics-vidhya/optimization-algorithms-for-deep-learning-1f1a2bd4c46b, Son erişim Tarihi: 29/01/2020.

144. Tensorflow, T. (2016). Tensorflow: large-scale machine learning on heterogeneous distributed systems, *White Paper- Google Research*, Kaliforniya, 1-19.

145. Shi, S., Wang, Q., Xu, P. ve Chu, X. (2016, Temmuz). *Benchmarking state-of-the-art deep learning software tools*. International Conference on Cloud Computing and Big Data, Macau, 1-14.

146. Babu, C.G., Sampath, P., Hariharan, S., Balakumar, S. ve Noufal, M. (2017, Kasım). *Performance analysis of hybrid model of robust automatic continuous speech recognition system*. International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, 1-4.

147. Swietojanski, P., Ghoshal, A. ve Renals, S. (2013, Aralık). *Hybrid acoustic models for distant and multichannel large vocabulary speech recognition*. Workshop on Automatic Speech Recognition and Understanding, Olomouc, 285-290.

148. Jouvet, D. ve Vinuesa, N. (2012, Mart). *Classification margin for improved class-based speech recognition performance*. International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, 4285-4288.

149. Salor, Ö., Pellom, B., Çiloğlu, T., Hacıoğlu, K. ve Demirekler, M. (2002, Eylül). O*n developing new text and audio corpora and speech recognition tools for the Turkish language*. International Conference on Spoken Language Processing (ICSLP), Denver, 1-5.

150. Akın, A. A. ve Akın, M. D. (2007). Zemberek, an open source NLP framework for Turkish languages. *Structure*, 10, 1-5.

151. Sak, H., Güngör, T. ve Saraçlar, M. (2008). Turkish language resources: morphological parser, morphological disambiguator and web corpus. *In Advances in Natural Language Processing*, 417-427.

152. Canbay, Y., Yazıcı, H. ve Sağıroğlu, Ş. (2017, Haziran). *A Turkish language based data leakage prevention system*. International Symposium on Digital Forensic and Security (ISDFS), Tirgu Mures, 1-6.

153. Gemci, F. ve Peker, K. A. (2013, Kasım). *Extracting Turkish tweet topics using LDA*. International Conference on Electrical and Electronics Engineering (ELECO), Bursa, 1-4.

154. Keleş, M. K. ve Özel, S. A. (2017, Ekim). *Similarity detection between Turkish text documents with distance metrics*. International Conference on Computer Science and Engineering (UBMK), Antalya, 1-6.

155. Arslan, E. ve Orhan, U. (2016, Ağustos). *Graph-based lemmatization of Turkish words by using morphological similarity*. International Symposium on Innovations in Intelligent Systems and Applications (INISTA), Sinaia, 1-5.

156. Yıldız, A. ve Demirci, M. (2017, Ekim). *Corporate e-mail classification system*. International Conference on Computer Science and Engineering (UBMK), Antalya, 1-6.

157. İnternet: Koopman, J., Database Journal - The Knowledge Center for Database Professionals. Url: http://www.postgresql.org, Son Erişim Tarihi: 21/12/2019.

158. İnternet: Rossum, V., Python, G. Corporation for National Research Initiatives (CNRI). Url: https://www.python.org, Son Erişim Tarihi: 21/12/2019.

159. İnternet: McKerns, M. M., Strand, L., Sullivan, T., Fang, A. ve Aivazis, M. A. Building a Framework for Predictive Science. Url: https://pypi.org/project/dill/, Son Erişim Tarihi: 21/12/2019.

160. İnternet: McKerns, M.M., Aivazis ve M. Pathos: A Framework for Heterogeneous Computing. Url: https://librosa.github.io/librosa/, Son Erişim Tarihi: 21/12/2019.

161. İnternet: Hettinger, R. Namedtuple Factory Function for Tuples with Named Fields. Url: https://docs.python.org/2/library/collections.html, Son Erişim Tarihi: 21/12/2019.

162. İnternet: Oliphant, T. E. Guide to Numpy, Url: https://www.numpy.org/, Son Erişim Tarihi: 21/12/2019.

163. İnternet: Lyons, J. Python Speech Features. Url: https://python-speech-features.readthedocs.io/en/latest/, Son Erişim Tarihi: 21/12/2019.

164. İnternet: Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G. ve Isard, M. Tensorflow: A System for Large-Scale Machine Learning. Url: https://www.tensorflow.org/install/pip, Son Erişim Tarihi: 21/12/2019.

165. Levenshtein, V. I. (1996). Binary codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory*, 10(8), 707-710.

166. Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. *Communication of the ACM*, 7(3), 171-176.

167. Jaro, M. A. (1989). Advances in record linkage methodology as applied to the 1985 census of Tampa Florida. *Journal of the American Statistical Association*, 84(406), 414-420.

168. Winkler, W. E. (1990, Ocak). *String comparator metrics and enhanced decision rules in the Fellegi-Sunter model of record linkage.* Proceedings of the Section on Survey Research Methods, Washington, 354-359.

169. Aldous, D., Diaconis, P. (1999). Longest increasing subsequences: from patience sorting to the Baik–Deift–Johansson theorem. *Bulletin of the American Mathematical Society*, 36(4), 413-432.

170. Singhal, A. (2001). Modern information retrieval: a brief overview. *IEEE Data Engineering Bulletin*, 24(4) 35-43.

171. Lu, J., Lin, C., Wang, W., Li, C. ve Wang, H. (2013, Kasım). S*tring similarity measures and joins with synonyms*. Acm Sigmod International Conference on Management of Data, New York, 373-384.

172. İnternet: Siderite Z. Super-Fast and Accurate String Distance Algorithm: Sift4. Url: https://siderite.blogspot.com/2014/11/super-fast-and-accurate-string-distance.html, Son Erişim Tarihi: 21/12/2019.

173. Yohei, F., Katsuyuki, T., Tetsuya T. ve Yasua A. (2015, Temmuz). *Word-Error Correction of Continuous Speech Recognition Based on Normalized Relevance Distance*. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, 1-6.

174. Yusuke, N., Zhipeng, Z. ve Nobuhiko, N. (2011). Efficient Speech-recognition Error Correction for More Usable Speech-to-text input. *NTT DOCOMO Technical Journal*, 11(2), 1-8.

175. Dong Y., Mei-Yuh H., Mau, P. Alex, A., Deng, L. (2004, Ekim). *Unsupervised Learning from Users error correction in speech dictation*. In the proceedings of ICSLP, Jeju, 1969-1972.

176. Yongmei, S. ve Zhou, L. (2009). Supporting dictation speech recognition error correction: the impact of external information. *Behaviour and Information Technology*, 1-15.

177. Youssef, B., Semaan, P. (2012). ASR context-sensitive error correction based on Microsoft n-gram dataset. *Journal of Computing*, 4(1), 1-9.

178. Büyük, O., Erdoğan, H. ve Oflazer, K. (2005, Mayıs). *Using hybrid lexicon units and incorporating language constraints in speech recognition*. Signal Processing and Communications Applications Conference (SIU), Kayseri, 111-114.

179. Keser, S. ve Edizkan, R. (2009, Nisan). *Phoneme-based isolated Turkish word recognition with subspace classifier.* Signal Processing and Communications Applications Conference (SIU), Antalya, 93-96.

180. İnternet: Apple Siri – Speech to Text. Url: https://www.apple.com/tr/siri/, Son Erişim Tarihi: 03/02/2020

181. İnternet: Whatsapp Dictation – Speech to Text. Url: https://www.whatsapp.com/?lang=tr, Son Erişim Tarihi: 03/02/2020.

182. İnternet: Google Speech – Speech to Text. Url: https://cloud.google.com/speech-to-text/?hl=tr, Son Erişim Tarihi: 03/02/2020.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	: Arslan, Recep Sinan
Uyruğu	: T.C.
Doğum tarihi ve yeri	: 11.03.1987, Yozgat
Medeni hali	: Evli
Telefon	: 0 (541) 495 75 53
E-mail	: recep.sinan.arslan@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Doktora	Gazi Üniversitesi / Bilgisayar Mühendisliği	Devam Ediyor
Yüksek lisans	Yıldırım Beyazıt Üniversitesi / Bilgisayar Mühendisliği	2015
Lisans	Karadeniz Teknik Üniversitesi / Bilgisayar Mühendisliği	2010
Lise	Boğazlıyan Anadolu Lisesi	2005

İş Deneyimi

Yıl	Yer	Görev
2019 - Halen	Yozgat Bozok Üniversitesi	Öğretim Görevlisi
2015 - 2019	Radyo ve Televizyon Üst Kurulu	Üst Kurul Uzmanı
2011 - 2015	Çalışma ve Sosyal Güvenlik Bakanlığı	İş Müfettişi

Yabancı Dil

İngilizce

Yayınlar

Arslan, R. S. ve Barışçı, N. (2019). Development of output correction methodology for long short term memory-based speech recognition. *Sustainability*, 11, 4250.

Arslan, R. S. ve Barışçı, N. (2018, Eylül). *Farklı optimizasyon tekniklerinin bağlantıcı zamansal sınıflandırma kullanılan uçtan uca Türkçe konuşma tanıma sistemlerine etkisi.* 2nd

International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, 1-6.

Arslan, R. S., Doğru, İ. A. ve Barışçı, N. (2019). Permission-based malware detection system for android using machine learning techniques. *International Journal of Software Engineering and Knowledge Engineering*, 29(01), 43-61.

Arslan, R. S., Doğru, İ. A. ve Barışçı, N. (2017). Android mobil uygulamalar için izin karşılaştırma tabanlı kötücül yazılım tespiti. *Politeknik Dergisi*, 20(1), 175-189.

Arslan, R. S. (2018). *Türkçede derin öğrenme ile konuşma tanıma uygulamaları*, Uzmanlık Tezi, Radyo ve Televizyon Üst Kurulu, Ankara, 52-120.

Hobiler

Yüzme, Dağcılık

GAZİ GELECEKTİR..