DERİN ÖĞRENME ALGORİTMALARI KULLANARAK YAZAR, TÜR VE CİNSİYET TANIMA

Melike BEKTAŞ

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Pınar TÜFEKCİ

2020

T.C.

TEKİRDAĞ NAMIK KEMAL ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

DERİN ÖĞRENME ALGORİTMALARI KULLANARAK YAZAR, TÜR VE CİNSİYET TANIMA

Melike BEKTAŞ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

DANIŞMAN: Doç. Dr. Pınar TÜFEKCİ

TEKİRDAĞ-2020

Bu tezde görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, tez içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri tezde eksiksiz biçimde kaynak göstererek belirttiğimi beyan ederim.

Melike BEKTAŞ

İMZA

Doç. Dr. Pınar TÜFEKCİ danışmanlığında, Melike BEKTAŞ tarafından hazırlanan “Derin Öğrenme Algoritmaları Kullanarak Yazar, Tür ve Cinsiyet Tanıma” başlıklı bu çalışma aşağıdaki jüri tarafından 26.11.2020 tarihinde Bilgisayar Mühendisliği Anabilim Dalı’nda Yüksek Lisans tezi olarak oy birliği ile kabul edilmiştir.

Jüri Başkanı : Doç. Dr. Pınar TÜFEKCİ *İmza:*

Üye : Doç. Dr. Erdinç UZUN *İmza:*

Üye : Dr. Öğr. Üyesi Faruk BULUT *İmza:*

Fen Bilimleri Enstitüsü Yönetim Kurulu adına

Doç. Dr. Bahar UYMAZ
Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

DERİN ÖĞRENME ALGORİTMALARI KULLANARAK YAZAR, TÜR VE CİNSİYET TANIMA

Melike BEKTAŞ

Tekirdağ Namık Kemal Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Pınar TÜFEKCİ

Günümüzde artan veri miktarı, bu verilerin sınıflandırılma ihtiyacını beraberinde getirmiştir. Sınıflandırma, benzer özellikte olan verilerin kategorize edilmesi işlemidir. Bu çalışmada, veri olarak Türkçe haber metinlerinin seçildiği ve bu verilerin yazar, tür ve cinsiyete göre sınıflandırılabilmelerini sağlayan, makine öğrenmesi ve derin öğrenme algoritmalarının sınıflandırıcı olarak kullanıldığı geniş kapsamlı bir modelleme çalışması yapılması amaçlanmıştır. Bu amaçla ilk olarak, bir gazetenin köşe yazarlarına ait köşe yazılarını içeren, yazar tanıma, tür tanıma ve cinsiyet tanıma işlemlerinde kullanılabilecek, büyük ölçekli ve çoklu sınıflara sahip, toplam 14 adet yeni veri seti oluşturulmuştur. Yazar tanıma için 7, tür tanıma için 6 ve cinsiyet tanıma için de 1 adet olan bu veri setleri, Türkçe diline özel, doğal dil işleme adımlarından geçirilerek, sınıflandırma işlemlerinin yapılacağı sınıflandırıcıların uygulandığı ve en yüksek doğruluk başarılarının araştırıldığı, modelleme aşaması için hazır hale getirilmiştir. Modelleme aşamasında, Türkçe metinlerde yazar tanıma, tür tanıma ve cinsiyet tanıma problemlerinin çözümüne yönelik makine öğrenmesi algoritmalarından Multinominal Naive Bayes (MNB) ve Random Forest (RF) algoritmaları, derin öğrenme algoritmalarından da Convolutional Neural Networks (CNN) ve Long Short Term Memory (LSTM) algoritmaları, sınıflandırıcı olarak veri setlerine uygulanmıştır. Ayrıca, bu sınıflandırıcılardan en yüksek performansın alındığı hiperparametre değerleri, uzun deneysel çalışmalar sonucunda bulunmaya çalışılmıştır. Modelleme sonucunda, her bir veri seti için en iyi modellere ait, doğruluk, kesinlik ve duyarlılık değerleri kullanılarak her modelin performansı bulunmuştur. Modelleme aşamasının sonucunda, yazar tanıma için, genel olarak tüm veri setleri arasında, en yüksek başarının alındığı en iyi model, % 95,81 doğruluk başarı değeriyle, AI-TNKU-7 veri seti için, CNN algoritmasının sınıflandırıcı olarak kullanıldığı model olarak bulunmuştur. Tür tanıma içinse, en yüksek başarının alındığı en iyi model, GI-TNKU-6 veri seti için LSTM algoritmasının sınıflandırıcı olarak kullanıldığı ve %96,73 doğruluk başarı değerinin alındığı model olmuştur. Cinsiyet tanıma için de, en yüksek başarının alındığı en iyi model, %88,68 doğruluk başarı değeriyle LSTM algoritmasının sınıflandırıcı olarak kullanıldığı model olarak bulunmuştur.

Anahtar kelimeler: Metin Sınıflandırma, Yazar Tanıma, Tür Tanıma, Cinsiyet Tanıma, Makine Öğrenmesi, Derin Öğrenme.

2020, 64 sayfa

ABSTRACT

MSc. Thesis

AUTHOR, GENRE AND GENDER IDENTIFICATION USING DEEP LEARNING ALGORITHMS

Melike BEKTAŞ

Tekirdağ Namik Kemal University

Graduate School of Natural and Applied Sciences

Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Pınar TÜFEKCİ

Nowadays, the increasing amount of data has brought the need to classify these data. Classification is the process of categorizing similar data. In this study, it is aimed to make a modeling study in which Turkish news texts are selected as data and that these data can be classified according to author, genre and gender, machine learning and deep learning algorithms are used as classifiers. For this purpose, firstly, a total of 14 new data sets with large-scale and multiple classes, which can be used in author identification, genre identification and gender identification processes, containing columnists of a newspaper, were created. These data sets, which are 7 for author identification, 6 for genre identification and 1 for gender identification, have been made ready for the modeling phase, where the classifiers for identification are applied and the highest accuracy successes are investigated by passing through natural language processing steps specific to Turkish language. In the modeling phase, Multinominal Naive Bayes (MNB) and Random Forest (RF) algorithms, which are machine learning algorithms for the solution of author identification, genre identification and gender identification problems in Turkish texts, and Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) from deep learning algorithms have been applied to data sets as classifiers. In addition, hyperparameter values with the highest performance from these classifiers have been tried to be found as a result of long experimental studies. As a result of modeling, using the accuracy, precision and recall values of the best models for each data set, the performance of each model was found. As a result of the modeling stage for author identification, it was seen that the CNN algorithm achieved the highest % 95.81accuracy in the AI-TNKU-7 data set compared to other algorithms used. As a result of the modeling for genre identification, an accuracy of % 96.73 was achieved with the LSTM algorithm in the GI-TNKU-6 data set. It has been observed that the success of deep learning algorithms is higher than machine learning algorithms in other data sets used in genre identification. As a result of the modeling phase for gender identification, the LSTM algorithm performed better than other classifiers and an accuracy success of % 88.68 was achieved.

Key words: Text Classification, Author Identification, Genre Identification, Gender Identification, Machine Learning, Deep Learning.

2020, 64 pages

İÇİNDEKİLER

ÇİZELGE DİZİNİ

ŞEKİL DİZİNİ

SİMGELER VE KISALTMALAR

AIS	: Artifical Immune System
BNB	: Bernoulli Naive Bayes
BOW	: Bag of Words
CNN	: Convolutional Neural Network
DT	: Decision Tree
GB	: Gradient Boosting
IDF	: Inverse Document Frequency
KNN	: K-Nearest Neighbors
LR	: Logistic Regression
LSTM	: Long Short Term Memory
ME	: Maximum Entropy
MNB	: Multinomial Naive Bayes
NB	: Naive Bayes
NLTK	: Natural Language Toolkit
ReLu	: Rectified Linear Unit
RF	: Random Forest
RNN	: Recurrent Neural Network
SGD	: Stochastic Gradient Descent
SMO	: Sequential Minimal Optimization
SVD	: Singular Value Decomposition
SVM	: Support Vector Machines
TF	: Term Frequency
TF-IDF	: Term Frequency-Inverse Document Frequency
WEKA	: Waikato Environment for Knowledge Analysis

TEŞEKKÜR

Yüksek lisans eğitimimde ve bu tez çalışmamda gösterdiği her türlü destek, ilgi ve yardımlarından dolayı çok kıymetli danışman hocam Doç. Dr. Pınar TÜFEKCİ'ye en kalbi duygularımla teşekkürlerimi sunarım.

Yüksek lisans ders aşamasında dersini aldığım Tekirdağ Namık Kemal Üniversitesi'ndeki tüm hocalarıma teşekkür ederim.

Eğitim hayatım boyunca her zaman bana destek olan, maddi ve manevi yardımlarını esirgemeyen anneme, babama, kardeşime ve bugünlere gelmemde üzerimde emeği bulunan tüm öğretmenlerime çok teşekkür ederim.

Kasım, 2020

Melike BEKTAŞ
Bilgisayar Mühendisi

1. GİRİŞ

Metin tabanlı veriler her geçen gün artmaktadır. Bu veriler gazetelerde, dergilerde, kitaplarda, kısa mesajlarda, maillerde, sosyal medya platformlarında, tabelalarda, reklam panolarında vb. yerlerde karşımıza çıkmaktadır. Artan veri miktarı ile birlikte bu verilerin benzer olanlarının bir yerde toplanması ve sınıflandırılması ihtiyacı ortaya çıkmıştır. Metin sınıflandırma, bir metnin daha önceden belirlenmiş olan sınıflardan hangisine ait olduğunu tespit etme işlemidir.

Metin sınıflandırmanın temel amacı, tasnif edilmemiş metin yığınlarının bir bilgisayar programı aracılığı ile sınıflandırılmasıdır (Aytekin vd., 2018). Bu sayede metin tabanlı verilerin, daha kolay bir şekilde yönetilmesi mümkün olacaktır.

Metin sınıflandırmanın günlük hayattaki uygulamaları: Bir metnin kim tarafından yazıldığının tespit edilmesi; bir metni yazan kişinin cinsiyetinin, yaşının veya mutlu, üzgün, kızgın vb. duygulardan hangi duyguyla o metni yazdığının belirlenmesi; bir metnin hangi türe ait olduğunun tahmin edilmesi; bir e-ticaret sitesinde bir ürün için yapılan olumlu ve olumsuz yorumların ayrıştırılması veya en beğenilen ürünlerin tespit edilmesi şeklinde olmaktadır.

Bilgisayar kullanılarak yapılan metin sınıflandırma uygulamalarının tarihçesi 1960'lı yıllarda başlamakla beraber çalışmaların yoğunluk kazanması ancak 80'lerin sonu ve 90'ların başında olmuştur (Tantuğ, 2014). 1970'li yıllarda yapılan çalışmalara bakıldığında, metin sınıflandırma işleminin dokümanların otomatik olarak dizinlemesi olarak karşımıza çıktığı görülmektedir. Çalışmalarda belirli bir konu için özel sözlükler oluşturulmuş, bu sözlüklerin içerisindeki kelimeler sınıf etiketi gibi düşünülerek dokümanların sınıflandırma işlemleri gerçekleştirilmiştir (Levent ve Diri, 2014). Aynı zamanda ilk yıllardaki çalışmalarda uzman sistemler yaklaşımının kullanıldığı ve sınıflandırmanın nasıl yapılacağını bilen bir uzmanın aldığı kararların benzetiminin yapıldığı kural tabanlı sistemlerin tasarlandığı görülmüştür (Tantuğ, 2014). Günümüzde ise metin sınıflandırma işlemleri, genellikle makine öğrenmesi ve derin öğrenme algoritmaları kullanılarak gerçekleştirilmektedir.

Makine öğrenmesinin bir alt dalı olan derin öğrenme, yapay sinir ağlarından, çok katmanlı sinir ağlarına yani derin yapay sinir ağlarına geçiş ile birlikte literatüre girmiştir (Şeker vd., 2017). Derin öğrenmenin gelişmesinde, günümüzde artan veri miktarı, kullanılan GPU'lar ile birlikte bilgisayarlardaki işlem gücünün artması, algoritmalardaki iyileştirmeler ve derin öğrenme kütüphanelerinin ve frameworklerinin etkisi olmuştur (Chollet, 2019). Derin öğrenme, nesne tanıma uygulamalarında, otonom araçlarda, sesli asistanlarda ve sentetik veri üretimi gibi birçok alanda kullanılmaktadır.

Bu çalışmada, metin sınıflandırma işlemlerinden, yazar, tür ve cinsiyet tanıma işlemleri gerçekleştirilecektir. Yazar tanıma, bir metnin kim tarafından yazıldığının tespit edilmesi işlemidir. Tür tanıma, bir metnin belirlenen türlerden hangisine ait olduğunun tespit edilmesi işlemidir. Cinsiyet tanıma ise bir metnin yazarının cinsiyetinin kadın mı ya da erkek mi olduğunun tahmin edilmesi işlemidir.

Yazar tanıma işlemi, bir metnin yazarının belli olmadığı durumlarda, bir metin üzerinde birden fazla kişinin hak talep ettiği ve kendisinin yazdığını iddia ettiği durumlarda, adli uygulamalarda, kimlik tespitinde, kütüphane uygulamalarında, terör ile mücadele soruşturmalarına yardımcı olmak amacı ile farklı alanlarda kullanılabilir (Stamatatos, 2008).

Tür tanıma işlemi, birçok metin tabanlı uygulama için faydalı olabilir. Örneğin, bir belgenin türü önceden biliniyorsa, bilgi erişim sonuçları kullanıcıya daha doğru bir şekilde sunulabilir (Lee ve Myaeng, 2004). Bir web ortamında aranılan bir dokümanın, türüne göre doğru ve kolay bir şekilde bulunabilmesi veya dokuman sınıflandırma işlemi yapan çevrimiçi bir kütüphane otomasyonunda, hem bir dokümanın hızlı bir şekilde bulunabilmesine hem de yeni gelen dokümanların doğru bir şekilde tasnif edilebilmesine olanak sağlayabilir.

Bir metnin yazarının demografik özelliklerinin otomatik olarak belirlenmesi ticari, adli tıp ve pazarlama alanlarındaki uygulamalarda giderek önem kazanmaya başlamıştır. Kullanıcıların yaş, cinsiyet, meslek, eğitim durumu gibi bilgileri demografik özelliklerinden bazılarıdır. Örneğin, bir bireyin saldırgan bir kişiliğe sahip olması metin belgesindeki mesajdan tespit edilebilir. Şirketler ürünlerini beğenen veya beğenmeyen kullanıcıları blog yazılarından ve kullanıcıların çevrimiçi ürün incelemelerini analiz kaynağı olarak kullanarak demografik özelliklerini öğrenmek isteyebilirler (Sboev vd., 2016). En önemli demografik özelliklerden olan cinsiyet bilgisinin, bir metin üzerinden giderek belirlenmesi, cinsiyet tanıma olarak tanımlanır.

Bu tez çalışması kapsamında amacımız, yeni oluşturduğumuz büyük ölçekli ve çoklu sınıflı veri setlerini kullanarak, Türkçe metinlerde yazar tanıma, tür tanıma ve cinsiyet tanıma problemlerinin çözümüne yönelik derin öğrenme ve makine öğrenmesi algoritmalarının kullanıldığı modeller oluşturmaktır. Bu amaçla, makine öğrenmesi algoritmalarından

Multinominal Naive Bayes (MNB) ve Random Forest (RF) algoritmaları; derin öğrenme algoritmalarından da Convolutional Neural Networks (CNN) ve Long Short Term Memory (LSTM) algoritmaları kullanılmıştır.

Bu çalışmanın ikinci bölümünde yazar, tür ve cinsiyet sınıflandırma ile ilgili kaynak özetleri, üçüncü bölümünde çalışmada kullanılan veri setleri, bu veri setlerinin oluşturulması, metin ön işlemleri, sınıflandırma algoritmaları, değerlendirme ölçütleri, çalışmada kullanılan kütüphaneler, dördüncü bölümünde araştırma bulguları ve beşinci bölümünde ise tartışma ve sonuçlar ele alınmıştır.

2. KAYNAK ÖZETLERİ

Bu bölümde, Türkçe ve farklı diller için yazar, tür ve cinsiyet tanıma ile ilgili yapılmış olan literatür çalışmalarına yer verilmiştir.

Amasyalı ve Diri (2006), çalışmalarında Türkçe haber metinlerini yazar, tür ve cinsiyet bakımından sınıflandırmışlardır. Çalışmada kullanılan veri setleri toplamda 630 metinden ve yazar tanıma için 18, tür tanıma için 3, cinsiyet tanıma için de 2 sınıftan oluşmaktadır. N-gram yöntemini ve sınıflandırıcı olarak Naive Bayes (NB), Support Vector Machines (SVM), RF ve C4.5 karar ağacı algoritmalarını kullanmışlardır. Yazar tanıma işlemi için, NB algoritması ile %83,3 doğruluk, tür tanıma işlemi için SVM algoritması ile %93,6 doğruluk, cinsiyet tanıma işlemi için de SVM algoritması ile %96,3 doğruluk başarı değerlerini elde etmişlerdir.

Kaban ve Diri (2008), çalışmalarında yapay bağışıklık sistemleri ile Türkçe metinlerde tür ve yazar tanıma işlemlerini gerçekleştirmişlerdir. Veri seti olarak haber web sitelerinden oluşturdukları 18 sınıftan oluşan 630 haber metnini yazar tanıma işlemi için, 5 sınıftan oluşan 250 haber metnini ise tür tanıma işlemi için kullanmışlardır. Yazar tanıma için, yapay bağışıklık sistemleri ile %99,6 doğruluk, tür tanıma için de SVM algoritması ile %98,2 doğruluk başarı değerleri elde etmişlerdir.

Yasdi ve Diri (2012), çalışmalarında farklı soyut özellik çıkarımı yöntemlerini kullanarak, Türkçe ve İngilizce metinleri, yazar, tür ve cinsiyet bakımından sınıflandırmışlardır. Yazar tanıma işlemi için 10 sınıftan ve her sınıfa ait 10 metinden oluşan toplam 100, tür tanıma işlemi için 4 sınıftan ve her sınıfa ait 50 metinden oluşan toplam 200 ve cinsiyet tanıma işlemi için de 2 sınıftan ve her sınıfa ait 100 metinden oluşan toplam 200 metin kullanmışlardır. Çalışma sonucunda, yazar tanıma için k-Nearest Neighbor (KNN) algoritması ile %99 doğruluk, tür tanıma için KNN algoritması ile %97,5 doğruluk ve cinsiyet tanıma için de SVM algoritması ile %94,5 doğruluk başarı değerlerini elde etmişlerdir.

Tüfekci ve Uzun (2013), çalışmalarında farklı terim ağırlıklandırma yöntemlerini kullanarak yazar tanıma problemini çözmeyi amaçlamışlardır. Çalışmalarında 14 farklı terim ağırlıklandırma yönteminin, sınıflandırma başarısına etkilerini incelemişlerdir. VK-1, VK-2 ve VK-3 olmak üzere sırasıyla 10 sınıftan oluşan 430 haber metinli, 69 sınıftan oluşan 910 haber metinli ve 18 sınıftan oluşan 630 haber metinli veri setlerini kullanmışlardır.

Sınıflandırma işlemi için SVM, MNB, RF ve C4.5 karar ağacı algoritmalarını kullanmışlardır. Çalışma sonucunda, en iyi sonucu SVM algoritmasından, ortalama %98,75 doğruluk başarısı ile elde etmişlerdir.

Şahin vd. (2017), çalışmalarında 3 sınıftan oluşan 1.255 şiir metnini yazar bakımından sınıflandırmışlardır. Çalışma sonucunda, Sequential Minimal Optimization (SMO) algoritması ile %70 doğruluk başarı değeri elde etmişlerdir.

Stamatos (2008), çalışmasında 10 sınıftan ve toplamda 1.000 metinden oluşan İngilizce ve Arapça metinlerde yazar tanıma işlemini gerçekleştirmiştir. Arapça metinlerden oluşan veri setinde %93,6 doğruluk, İngilizce metinlerden oluşan veri setinde ise %79,4 doğruluk başarısı değerleri elde etmiştir.

Diri ve Doğan (2010), çalışmalarında N-gram yöntemini kullanarak Türkçe metinleri yazar, tür ve cinsiyet bakımından sınıflandırmışlardır. Yazar tanıma işlemi için 20 yazardan oluşan toplam 800 metin, tür tanıma işlemi için 6 sınıftan oluşan toplam 480 metin, cinsiyet tanıma işlemi için ise 2 sınıftan oluşan toplam 800 metin kullanmışlardır. SVM algoritması ile yazar tanıma işleminde %89,5 doğruluk, tür tanıma işleminde %92,1 doğruluk, cinsiyet tanıma işleminde ise %91,1 doğruluk başarı değerleri elde etmişlerdir.

Solar-Company ve Wanner (2018), çalışmalarında hem yazar tanıma hem de cinsiyet tanıma işlemini gerçekleştirmişlerdir. Kullanmış oldukları ilk veri setini 23 köşe yazarının toplamda 4.284 İngilizce haber metininden, ikinci veri setini ise 16 yazarın toplamda 1.570 İngilizce kitap bölümünden oluşturmuşlardır. Her iki veri setini de cinsiyet ve yazar tanıma işleminde kullanmak için etiketlemişlerdir. Farklı özellik seçimlerinin başarısını libSVM algoritmasını kullanarak kıyaslamışlardır. Çalışma sonucunda ilk veri setinde yazar tanıma işlemi için %78,16 doğruluk, cinsiyet tanıma işlemi için %89,97 doğruluk, ikinci veri setinde yazar tanıma işlemi için %91,78 doğruluk, cinsiyet tanıma işlemi için %95,03 doğruluk başarısına ulaştıklarını bildirmişlerdir.

Acı ve Çırak (2019), çalışmalarında kültür sanat, spor, teknoloji, ekonomi, sağlık ve siyaset sınıflarını içeren Turkish Text Classification 3600 (TTC-3600) veri setini kullanarak Türkçe haber metinlerini sınıflandırmışlardır. Çalışmalarında iki farklı CNN modeli kullanmışlardır. Her iki modeli de hem ham veri ile hem de Zemberek yazılımını kullanarak kelime kökleri bulunmuş hali ile eğitip test etmişlerdir. Eğitilmiş olan bu iki modelin başarısını aynı veri seti üzerinde yapılmış olan diğer bir çalışma ile kıyaslamışlardır. Çalışma sonucunda, CNN algoritmasının %93,3 doğruluk ile makine öğrenmesi algoritmalarına kıyasla daha başarılı bir şekilde sınıflandırma yaptığı ifade edilmiştir.

Nergiz vd. (2019), tarafından yapılan çalışmada, Türkçe haber metinleri, LSTM algoritması kullanılarak sınıflandırılmıştır. Çalışmada spor, teknoloji, politika, kültür sanat, bilim teknoloji, sağlık ve magazin olmak üzere 7 sınıf ve her bir sınıfta 360.000 haber metni olan, toplamda 2.520.000 haber metninden oluşan bir veri seti kullanılmıştır. Veri seti üzerinde sınıflandırma aşamasına geçmeden önce, Fasttext, word2vec ve doc2vec modelleri kullanılarak 3 farklı kelime vektörü elde edilmiş ve bu modellerin başarı oranları kıyaslanmıştır. Çalışma sonucunda, Fasttext kelime vektörü yönteminin kullanıldığı model ile %96,18 doğruluk başarısı elde edildiği görülmüştür.

Vijayakumar vd. (2019), çalışmalarında İngilizce çevrimiçi restoran ve otel yorumlarından oluşan Yelp veri setini kullanarak yazar tanıma işlemini gerçekleştirmişlerdir. Sınıflandırma işlemi için MNB, Maximum Entropy (ME) ve SVM algoritmalarının başarısını doğal dil işleme tekniklerini de uygulayarak test etmişlerdir. Çalışma sonucunda, SVM algoritmasından en yüksek %90,5 doğruluk başarısı elde etiklerini bildirmişlerdir.

Tüfekci vd. (2012), çalışmalarında Türkçe dil bilgisi özelliklerini kullanarak, haber metinlerini ekonomi, sağlık, magazin, siyaset ve spor olmak üzere 5 kategoriye göre sınıflandırmışlardır. Kullanılan iki veri setinin, ilkinde 500 eğitim, 250 test verisi, ikinci veri setinde ise 750 eğitim, 400 test verisi yer almıştır. Çalışmada, özellik vektörünün boyutunun başarıdan ödün vermeden nasıl azaltılabileceği üzerinde durulmuştur. Sınıflandırıcı olarak kullanılan SVM, NB, RF, C4.5 karar ağacı algoritmalarından, NB algoritmasının %92,73 doğruluk başarısı ile sınıflandırma işlemini en yüksek doğrulukla yaptığı tespit edilmiştir.

Wongso vd. (2017), çalışmalarında Endonezya dilinden oluşturulmuş veri setini kullanarak haber metinlerini tür bakımından sınıflandırmışlardır. Çalışmada 5 sınıftan oluşan ve her bir sınıfta 1.000 metin olmak üzere toplam 5.000 adet haber metninden oluşan bir veri seti kullanmışlardır. Veri setindeki haber metinlerini, ekonomi, sağlık, spor, politika ve teknoloji olarak kategorize etmişlerdir. Veri seti oluşturulduktan sonra sınıflandırma işlemine geçmeden önce metinler, metin ön işleme aşamalarından geçirilmiştir. Sınıflandırma işlemi için makine öğrenmesi algoritmalarından NB ve SVM algoritmaları kullanılarak bu algoritmaların başarıları kıyaslanmıştır. Çalışma sonucunda, en iyi algoritma olarak NB algoritmasının %98,4 doğruluk başarısı ile sınıflandırma işlemini gerçekleştirdiği görülmüştür.

Sboev vd. (2016), çalışmalarında metin verilerinden cinsiyet tanıma işlemini gerçekleştirmişlerdir. Çalışmada kullanılan veri seti Rusça diline ait 1.867 metinden oluşturulmuştur. Sınıflandırma işlemi için, hem makine öğrenmesi hem de derin öğrenme algoritmalarını kullanmışlardır. Derin öğrenme algoritmalarından CNN, LSTM ile birlikte tümleşik olarak kullanılmıştır. Makine öğrenmesi algoritmalarından ise SVM, RF ve Extra Trees algoritmaları kullanılmıştır. Çalışmanın sonucunda en başarılı sınıflandırma işlemini gerçekleştiren algoritmanın, CNN ile LSTM'nin birlikte kullanıldığı model olduğu tespit edilmiştir.

Sboev vd. (2018), çalışmalarında Rusça metinlerden cinsiyet tanıma işlemini gerçekleştirmişlerdir. Gradient Boosting (GB) algoritmasını kullanarak %64 sınıflandırma başarısına ulaştıklarını bildirmişlerdir.

Sboev vd. (2018), çalışmalarında Rusça metinlerde cinsiyet tanıma işlemini CNN algoritmasını kullanarak %86 başarı ile gerçekleştirmişlerdir.

Cheng vd. (2011), çalışmalarında İngilizce metinlerden oluşan Enron ve Reuters veri setlerini kullanarak cinsiyet tanıma işlemini gerçekleştirmişlerdir. Çalışmalarında SVM, AdaBoost ve Logistic Regression (LR) algoritmalarını kullanmışlardır. Çalışma sonucunda, SVM algoritmasının %85,13 doğruluk başarısı ile diğer algoritmalara göre daha başarılı bir şekilde sınıflandırma işlemini gerçekleştirdiği görülmüştür.

Hussein vd. (2019), çalışmalarında kadın ve erkek olarak etiketlenmiş ve her sınıfta 70.000 adet Mısır lehçesine ait Arapça twitter verilerinden oluşturdukları EDGAD adlı veri setini kullanarak cinsiyet tanıma problemini çözmeye çalışmışlardır. Çalışmada kullandıkları veri setini her bir sınıf için de 70 açık hesaptan 1.000'er tweet olacak şekilde toplamışlardır. Çalışmada makine öğrenmesi algoritmalarından RF, Bernoulli Naive Bayes (BNB), Stochastic Gradient Descent (SGD) ve LR algoritmasını kullanarak sınıflandırma işlemini gerçekleştirmişlerdir. Çalışma sonucunda, LR algoritmasının %87,6 en yüksek doğruluk başarısı ile sınıflandırma işlemini gerçekleştirdiği görülmüştür.

Alsmearat vd. (2017), çalışmalarında Arapça haber metinlerini cinsiyet bakımından sınıflandırmışlardır. Veri setini 1.120 kadın, 1.057 erkek olmak üzere toplamda 2.177 haber metninden oluşturmuşlardır. Özellik çıkarımı için Stylometic Features ve Bag of Words (BOW) yaklaşımını kullanmışlar ve bu yaklaşımların makine öğrenmesi sınıflandırma başarısına etkilerini karşılaştırmışlardır. Çalışma sonucunda, Stylometic Features yaklaşımında JRip algoritmasıyla en yüksek doğruluk başarısı %80,4 olarak, BOW yaklaşımında ise SVM algoritmasıyla birlikte en yüksek doğruluk başarısı %73,9 olarak elde ettiklerini bildirmişlerdir.

Goenawan vd. (2019), çalışmalarında instagram yorumlarını kullanarak cinsiyet tahmini yapmışlardır. Kullandıkları veri setini 881 kadın, 488 erkek olmak üzere toplamda 1.369 açık hesaptan, 64.000 yorum toplayarak oluşturmuşlar ve bu yorumların 40.000 tanesini etiketlemişlerdir. Çalışmada NB, SVM ve karar ağacı tabanlı XGBoost, AdaBoost algoritmalarını kullanmışlardır. Çalışma sonucunda, NB algoritmasının %78,64 doğruluk ile kullanılan diğer algoritmalara kıyasla daha başarılı bir şekilde sınıflandırma işlemini gerçekleştirdiği görülmüştür.

Ritesh (2018), çalışmasında Wikipedia ve bebek isimleri web sitelerinden toplayarak oluşturduğu veri setini kullanarak isimden cinsiyet tahmini yapmıştır. Veri setini Hindistan, Japonya, Sri Lanka ve Batı ülkelerindeki 18.435 isimden oluşturmuştur. Çalışmasında LSTM, CNN ve CNN algoritmasının özel bir modeli olan LeNet modelini farklı kelime temsili yöntemlerini de kullanarak kıyaslamıştır. Çalışma sonucunda LSTM algoritmasının %84,30 doğruluk ile CNN ve LeNet algoritmasına göre daha başarılı bir şekilde cinsiyet tanıma problemini çözdüğü görülmüştür.

Abdallah vd. (2020), çalışmalarında WEKA programını kullanarak metin verilerinden, cinsiyet ve yaş tahmini yapmışlardır. Sınıflandırma işlemi için SVM, NB ve Decision Tree (DT) algoritmalarını kullanmışlardır. Çalışma sonucunda, cinsiyet tahmini için SVM algoritmasında %82,81 doğruluk başarısına, yaş tahmini problemi için de yine SVM algoritmasında %83,2 doğruluk başarısına ulaştıklarını bildirmişlerdir.

3. MATARYEL VE YÖNTEM

Bu bölümde, çalışmada yeni oluşturulan veri setlerinin tanıtımına, veri setlerine uygulanan ön işlemlere ve her bir veri seti için, uygulanan sınıflandırma algoritmaları ile oluşturulan modellere ve bu modellerin sonuçlarının değerlendirilmesine yer verilmiştir.

3.1. Veri Setlerinin Oluşturulması

Bu çalışmada, yazar, tür ve cinsiyet tanıma işlemleri için, bir gazetenin (https://www.hurriyet.com.tr/yazarlar/tum-yazarlar/) bazı köşe yazarlarına ait, 08.11.1997 tarihinden 24.04.2019 tarihine kadar olan arşivlerindeki köşe yazısı metinlerinden oluşan, büyük ölçekli ve çoklu sınıflı, yeni veri setleri oluşturulmuştur. Bu veri setlerindeki metinler, Python dilinde yazılmış olan bir web crawler kullanılarak çekilmiştir (Uzun, 2020). Veri çıkarma sürecinde, ilgili web sayfasında yer alan köşe yazarlarından bazıları seçilmiş ve her yazar için bir JSON kuralı oluşturulmuştur. Crawler programında yazarın JSON formatlı kuralı çalıştırıldığında yazarın adı ve soyadından oluşan bir dosya program tarafından otomatik olarak oluşturulmuş ve haberin bulunduğu sayfa bir html belgesi olarak haber metni ise .txt uzantılı bir metin belgesi olarak yazarın adı ile oluşturulan dosyaya kaydedilmiştir. Her bir .txt uzantılı dosyada haberin başlığı, haberin yayın tarihi ve haber metni verileri tutulmuştur.

3.2. Veri Setleri

Crawler ile çekilen ham haber metinleri, yazar tanıma, tür tanıma ve cinsiyet tanıma problemleri için uygun şekilde etiketlenerek, 7 tane yazar tanıma veri seti, 6 tane tür tanıma veri seti ve 1 tane de cinsiyet tanıma veri seti olmak üzere toplam 14 tane yeni veri seti oluşturulmuştur.

3.2.1. Yazar Tanıma Veri Setleri

Yazar tanıma için oluşturulan, AI-TNKU-1, 2, 3, 4, 5, 6 ve 7 olarak adlandırdığımız çoklu sınıf sayısına sahip, büyük ölçekli veri setlerine ait sınıf ve metin sayıları bilgileri Çizelge 3.1'de gösterilmiştir.

Çizelge 3.1. Yazar Tanıma Veri Setlerinin Ayrıntıları

Veri Setleri	Sınıf/Yazar Sayısı	Her Bir Sınıftaki Metin Sayısı	Toplam Metin Sayısı
AI-TNKU-1	68	100	6.800
AI-TNKU-2	50	200	10.000
AI-TNKU-3	38	300	11.400
AI-TNKU-4	33	400	13.200
AI-TNKU-5	27	500	13.500
AI-TNKU-6	16	1.000	16.000
AI-TNKU-7	9	2.000	18.000

3.2.2. Tür Tanıma Veri Setleri

Tür tanıma için oluşturulan, GI-TNKU-1, 2, 3, 4, 5 ve 6 olarak adlandırdığımız çoklu sınıf sayısına sahip, büyük ölçekli 6 tane yeni veri setine ait sınıf ve metin sayıları bilgileri Çizelge 3.2'de belirtilmiştir.

Çizelge 3.2. Tür Tanıma Veri Setlerinin Ayrıntıları

Veri Setleri	**Sınıf Sayısı**	**Sınıf (Tür) Adı**	**Her Bir Sınıftaki Metin Sayısı**	**Toplam Metin Sayısı**
GI-TNKU-1	7	Ekonomi / Genel / Genel & Siyaset / Magazin / Siyaset / Genel Sosyal Hayat / Günlük Sosyal Hayat	3.188	22.316
GI-TNKU-2	6	Ekonomi / Genel / Genel & Siyaset / Magazin / Siyaset / Günlük Sosyal Hayat	3.343	20.058
GI-TNKU-3	5	Ekonomi / Genel & Siyaset / Magazin / Siyaset / Günlük Sosyal Hayat	3.848	19.240
GI-TNKU-4	4	Ekonomi / Genel &Siyaset / Siyaset / Günlük Sosyal Hayat	4.064	16.256
GI-TNKU-5	3	Genel &Siyaset / Siyaset / Günlük Sosyal Hayat	4.760	14.280
GI-TNKU-6	2	Genel &Siyaset / Günlük Sosyal Hayat	5.831	11.662

3.2.3. Cinsiyet Tanıma Veri Setleri

Cinsiyet tanıma için ise IAG-TNK olarak adlandırdığımız yeni ve büyük ölçekli Türkçe bir veri seti oluşturulmuştur. Veri setine ait sınıf ve metin sayıları bilgileri Çizelge 3.3'de belirtilmiştir.

Çizelge 3.3. IAG-TNK Veri Seti Ayrıntıları

Sınıf	**Metin Sayısı**	**Kelime Sayısı**	**Yazar Sayısı**
Kadın	21.646	11.049.268	32
Erkek	21.646	11.146.652	38
Toplam	43.292	22.195.920	70

3.3. Veri Seti Ön İşlemleri

Türkçe, İngilizce gibi doğal dillerde yazılmış metinlerin, bilgisayarlar tarafından anlaşılır hale gelmesi için bu metinlerin doğal dil işleme adımlarından yani bazı ön işlemlerden geçirilmesi gerekmektedir. Bu ön işlemler, hem metin sınıflandırma başarısının artmasına hem de hesaplama işlemlerinin azalmasını sağlamaktadır. Bu çalışmada da, yazar, tür ve cinsiyet tanıma işlemlerinde kullanılan her bir veri setine, bu ön işlemler uygulanmış olup, sonrasında sınıflandırma işlemine geçilmiştir. Aşağıdaki alt başlıklarda, bu çalışmada uygulanan metin ön işlem adımlarından bahsedilmiştir.

3.3.1. Dizgelere Ayırma

Dizgelere ayırma işlemi, metinlerin anlamlı küçük parçalara ayrılması işlemidir. Metinleri, cümlelere ya da kelimelere ayırarak bu ön işlem aşaması gerçekleştirilebilir. Bu çalışmada, metinler, kelimelere ayrılmıştır ve diğer ön işlem aşamaları bu aşamadan sonra gerçekleştirilmiştir.

3.3.2. Metnin Küçük Harflere Dönüştürülmesi

Sınıflandırma işleminde daha başarılı bir performans elde etmek için, metinlerde geçen tüm büyük harfler küçük harflere çevrilir. Örneğin 'Türkiye' ve 'türkiye' kelimeleri bilgisayar tarafından iki farklı kelime olarak algılanmaktadır. Bu sorunu ortadan kaldırmak için bu işlem uygulanmaktadır. Çizelge 3.4'de küçük harfe çevirme işlemi görülmektedir.

Çizelge 3.4. Küçük Harfe Çevirme İşlemi

Orijinal Metin	**Küçük Harfe Çevrilmiş Hali**
Bilgisayar Mühendisliği	bilgisayar mühendisliği
BiLgiSayar MüHenDisLiği	bilgisayar mühendisliği
BİLGİSAYAR MÜHENDİSLİĞİ	bilgisayar mühendisliği

3.3.3. Durak Kelimelerinin Kaldırılması

Durak kelimeleri, dilde yaygın olarak kullanılan ve çoğu metinde geçen, anlam taşımayan kelimelerdir. Bu kelimeler, dilden dile farklılık göstermektedir. Bu kelimelerin, metin içerisinden kaldırılmaları, sınıflandırma başarısını olumlu yönde etkilediği için bu ön işlem uygulanmaktadır. Bu çalışmada, 124 tane Türkçe durak kelimesi metinlerin içerisinden çıkarılmıştır. NLTK Kütüphanesinde bulunmayan eksik durak kelimeleri ise bir web sitesinden alınmış ve durak kelimeleri listesine eklenmiştir (Türkçeöğretimi, 2009). Çizelge 3.5'de Türkçe ve İngilizce için bazı durak kelimeleri belirtilmiştir.

Çizelge 3.5. Türkçe Ve İngilizce İçin Bazı Durak Kelimeleri

Dil	**Durak Kelimeleri**
Türkçe	acaba, bazı, çok, daha, en, fakat gibi, hatta, için, kadar, mı, nasıl, oysa, önce, sen, şayet, tamam, üzere, veya, yani, zaten, …
İngilizce	about, at, because, that, were, here, from, do, under, over, what, and, are, before, so, such, it, more, other, could, to, very, you, into, no, …

3.3.4. Sayıların ve Noktalama İşaretlerinin Kaldırılması

Bu aşamada, metinlerde yer alan sınıflandırmaya etki etmeyen sayı ve rakamlar, noktalama işaretleri ve bazı ilgisiz karakterler (%, &, +, #, £, ~, >,$, @, €, =, *, } vb.) metinlerden çıkarılmıştır.

3.3.5. Kelime Köklerinin Bulunması

Kelime köklerinin bulunması, kelimelerin almış oldukları eklerden ayrılması işlemidir. Bu çalışmada, kelime köklerinin bulunması için açık kaynak kodlu Java dilinde yazılmış olan Zemberek Kütüphanesi kullanılmıştır (Akın ve Akın, 2007). Çizelge 3.6'da Zemberek Kütüphanesi kullanılarak elde edilmiş bazı kelime kökleri belirtilmiştir.

Çizelge 3.6. Zemberek Kütüphanesi Kullanılarak Elde Edilmiş Bazı Kelime Kökleri

Orijinal Kelime	**Kelimenin Kökü**
kahvenin	kahve
sektörünü	sektör
markasına	marka

3.3.6. Veri Setinin Bölünmesi

Yazar, tür ve cinsiyet tanıma için kullanılacak veri setlerine, 10 Fold Cross-Validation uygulanmıştır. 10 Fold Cross-Validation ile veri seti, rastgele olarak 10 eşit parçaya bölünür, daha sonra 10 farklı aşamada sırasıyla bu parçalardan biri test veri seti, diğer 9 parça da eğitim veri seti olarak kullanılır. Böylece, her bir alt küme en az bir kez test amacıyla kullanılmış olur (Kalaycı, 2018). Sınıflandırma işlemi de ayrılan her bir eğitim ve test verisi için 10 kez tekrarlanır. Şekil 3.1'de 10 Fold Cross-Validation kullanılarak veri setlerinin bölünmesi işlemi yer almaktadır.

1. Fold	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
2. Fold	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
3. Fold	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
4. Fold	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
5. Fold	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
6. Fold	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
7. Fold	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti
8. Fold	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti	Eğitim Seti
9. Fold	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti	Eğitim Seti
10. Fold	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Eğitim Seti	Test Seti

Şekil 3.1. 10 Fold Cross-Validation ile Test Aşamaları

3.3.7. Veri Setlerinin Sayısallaştırılması

Veri setlerinin, makine öğrenmesi ve derin öğrenme algoritmaları kullanılarak sınıflandırılabilmesi için metinlerin, sayısallaştırılması gereklidir. Bu amaçla, bu çalışmada makine öğrenmesi modellerinde TF-IDF terim ağırlıklandırma yöntemi, derin öğrenme modellerinde ise embedding katmanı ile bu işlem gerçekleştirilmiştir.

3.3.7.1. TF-IDF Ağırlıklandırma Yöntemi

TF-IDF değeri TF ve IDF ağırlıklandırma yöntemlerinin çarpılması ile bulunmaktadır. TF, yani terim frekansı değeri, bir kelimenin bir metin içerisinde geçme sıklığıdır. Terim ağırlıklandırma yöntemlerinin en sık kullanılan başlıca bileşenlerinden birisidir (Tüfekci ve Uzun, 2013). IDF, ters doküman frekansı değeri ise toplam doküman sayısının kelimenin geçtiği doküman sayısına bölümünün logaritmasıdır. TF, IDF ve TF-IDF değerleri sırası ile Denklem 3.1, Denklem 3.2 ve Denklem 3.3'deki formüller kullanılarak hesaplanmaktadır.

$$\mathrm{TF} = \mathrm{tf}_{i,j} \tag{3.1}$$

$$\mathrm{IDF} = \left(\log \frac{D}{\mathrm{df}_i}\right) \tag{3.2}$$

$$\mathrm{TF} - \mathrm{IDF} = \mathrm{TF} \times \mathrm{IDF} \tag{3.3}$$

$tf_{i,j}$ bir t_i teriminin d_j dökmanı içerisinde geçme sıklığını belirtir. D bir veri kümesindeki toplam dokümanların sayısı, df_i ise bir veri kümesindeki t_i terimini içeren dokümanların sayısıdır (Soucy ve Mineau, 2005). Bu çalışmada, makine öğrenmesi algoritmaları (NB ve RF) ile sınıflandırma işlemi yaptığımızda, metinler, TF-IDF ağırlıklandırma yöntemi kullanılarak sayısal değerlere dönüştürülmüştür.

3.3.7.2. Embedding Katmanı

Embedding katmanı, derin öğrenme modellerinde veri setinin sayısallaştırılıp vektör haline dönüştürülmesi amacıyla kullanılmıştır. Embedding işlemi, kelimeye vektör ilişkilendirmenin bir yöntemidir. Bir embedding katmanı oluşturduğumuzda, ağırlıkların değerleri diğer katmanlarda olduğu gibi rastgele belirlenir. Eğitim boyunca geri yayılım ile gradyan değerlerine göre güncellenir ve devamında gelen modelin yapısına uygun hale getirilir (Chollet, 2019).

3.4. Sınıflandırma Algoritmaları

3.4.1. Multinominal Naive Bayes

Naive Bayes algoritması, Bayes teoremi üzerine inşa edilmiş bir sınıflandırma algoritmasıdır. Bu teorem koşullu olasılıkların ilişkilerini istatistiksel olarak hesaplamaya dayanmaktadır. Bayes teoremi Denklem 3.4'de formül kullanılarak hesaplanmaktadır.

$$\mathrm{P}(\mathrm{L}\ |\mathrm{features}) = \frac{\mathrm{P}(\mathrm{features}\ |\ \mathrm{L})\mathrm{P}(\mathrm{L})}{\mathrm{P}(\mathrm{features})} \tag{3.4}$$

Klasik Naive Bayes, belirli bir belgenin kategorisini hesaplamak için kelimelerin ve kategorilerin ortak olasılıklarını kullanan olasılıksal bir sınıflandırıcıdır (Metsis vd., 2006). Naive Bayes sınıflandırıcılarının farklı türleri vardır: Gaussian Naive Bayes ve Multinominal Naive Bayes (MNB) (VanderPlas, 2016). Bu çalışmada, her bir veri seti için MNB sınıflandırıcısının kullanıldığı modeller oluşturulmuştur.

MNB genellikle, özelliklerin sınıflandırılacak belgelerdeki kelime sayıları veya sıklıkları ile ilgili olduğu metin sınıflandırmasında kullanılır. Veri seti, özelliklerin basit birçok terimli dağılımdan üretildiği varsayıldığı bu sınıflandırıcıya uygulanır. Çok terimli dağılım, bir dizi kategori arasında sayıları gözlemleme olasılığını tanımlar ve bu nedenle MNB, sayıları veya sayım oranlarını temsil eden özellikler için en uygun olanıdır (VanderPlas, 2016). En uygun multinom dağılımlı veri dağılımı, bu çalışmanın ilk modeli olarak modellenmiştir.

3.4.2. Random Forest

Random Forest algoritması, hem sınıflandırma hem de regresyon için kullanılabilen toplumsal öğrenme yöntemlerine ait gözetimli bir makine öğrenmesi algoritmasıdır. RF, adından da anlaşılacağı gibi, topluluk olarak çalışan birçok bireysel karar ağacından oluşur. RF içindeki her karar ağacı bir sınıf tahmini verir ve en yüksek puan alan sınıf, modelimizin tahminidir. RF algoritmasının işleyişi, aşağıdaki adımlarla anlatılmaktadır (Breiman, 2001):

Adım 1: İlk olarak, belirli bir veri kümesinden rastgele örnekler seçilerek başlanılır.

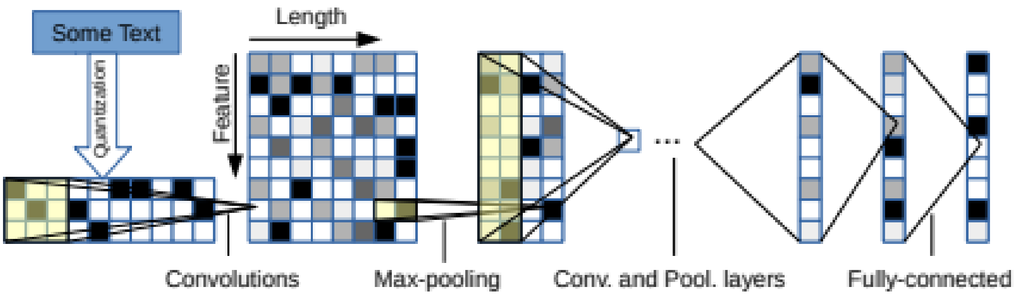
Adım 2: Daha sonra, bu algoritma her durum için bir karar ağacı oluşturur, her karar ağacından tahmin sonucunu alır.

Adım 3: Bu adımda, tahmin edilen her sonuç için oylama yapılır.

Adım 4: Son olarak, nihai tahmin sonucu olarak en çok oylanan tahmin sonucu seçilir.

3.4.3. Convolutional Neural Network

CNN algoritması, konvolüsyon ve havuzlama gibi işlemleri içerisinde barındıran bir yapay sinir ağı modelidir. Görüntü tabanlı veriler üzerinde başarısını kanıtlamış bir algoritmadır. Bununla birlikte 1D konvolüsyonel sinir ağları doğal dil işleme uygulamalarında da başarılı sonuçlar vermektedir. Şekil 3.2’de CNN algoritmasının temel yapısı yer almaktadır.



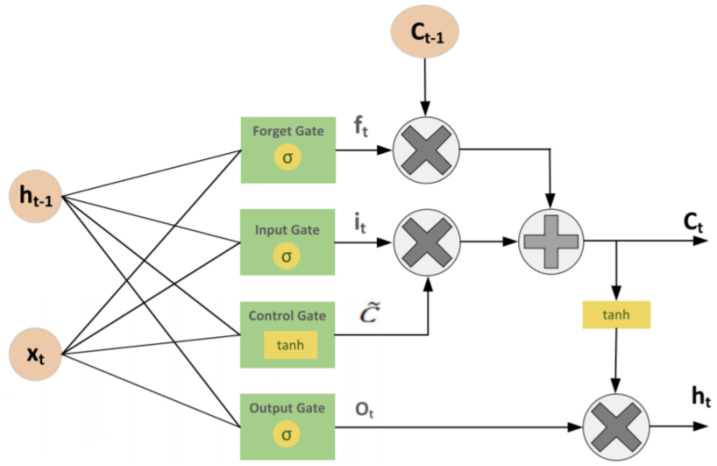
Şekil 3.2. CNN Algoritmasının Temel Yapısı (Zhang vd., 2015)

CNN algoritmasında, konvolüsyon işlemi özellikleri belirlemek amacıyla, pooling yani havuzlama işlemi ağırlık sayılarının düşürülmesi ve uygunluğun kontrol edilmesi, flatten katmanı ise klasik sinir ağı için verileri uygun hale getirmek amacıyla kullanılmaktadır (Çevik ve Kilimci, 2019).

3.4.4. Long Short Term Memory

LSTM, sıralı verilerdeki sınıflandırma ve regresyon sorunlarını çözmek için 1997 yılında Sepp Hochreiter ve Jürgen Schmidhuber tarafından tasarlanmış bir derin öğrenme algoritmasıdır. LSTM, bir tür tekrarlayan sinir ağıdır ve derin öğrenmede dizi yapısındaki verileri, öğrenmek için kullanılır. Bir LSTM ağının, bir paragraftaki cümlelerde olduğu gibi, geride kalan katmanlardan gelen bilgileri hatırlama yeteneği vardır (Elnagar vd., 2020).

Bir LSTM katmanı, bellek blokları olarak bilinen, tekrarlayan şekilde bağlanmış bir dizi bloktan oluşur. Bloklar, dijital bir bilgisayardaki bellek yongalarının farklılaştırılabilir bir versiyonu olarak düşünülebilir (Graves ve Schmidhuber, 2005). Bu bloklar Şekil 3.3’de gösterilen giriş kapısı, unutma kapısı, kontrol kapısı ve çıkış kapısı gibi dört kapıdan oluşmaktadır.



Şekil 3.3. LSTM'in Temel Yapısı (Sun vd., 2018)

Giriş kapısı şu şekilde tanımlanmaktadır:

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i) \tag{3.5}$$

Hangi bilgilerin bir sonraki hücreye aktarılacağına karar verir. İhmal edilecek olan önceki hafızanın girdisinden gelen bilgiye unutma kapısı tarafından karar verilir ve şu şekilde tanımlanmaktadır:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f) \tag{3.6}$$

Hücrenin güncellenmesi kontrol kapısı tarafından kontrol edilir ve aşağıdaki denklemlerle tanımlanmaktadır:

$$\begin{aligned} \tilde{C}_t &= \tanh(W_c \times [h_{t-1}, x_t] + b_c) \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \end{aligned} \tag{3.7}$$

Gizli katman (h_{t-1}) çıktı katmanı tarafından güncellenir ve bu da çıktıyı aşağıdaki gibi güncellemekten sorumludur:

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o) \tag{3.8}$$

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

Yukarıdaki denklemlerde tanh, değerleri -1 ila 1 aralığına ölçeklemek için kullanılır, σ sigmoid olarak alınan aktivasyon fonksiyonudur ve W, karşılık gelen ağırlık matrisleridir.

Bir LSTM'nin hücre durumu, bilginin bir hücreden diğerine kesintisiz olarak aktarıldığı kanaldır. LSTM hücreleri art arda sıralandığında hücreler arası bilgi akışı bu yollarla sağlanır (Tai vd., 2015).

3.5. Değerlendirme Ölçütleri

Bu çalışmada, her bir modelin başarısını değerlendirmek için doğruluk (accuracy), kesinlik (precision) ve duyarlılık (recall) metrikleri kullanılmıştır.

3.5.1. Doğruluk

Doğruluk, sınıflandırma algoritmalarının başarısını ölçmede en çok kullanılan değerlendirme ölçütlerinden birisidir. Genel olarak, değerlendirilen toplam örnek sayısına göre doğru tahminlerin oranını ölçer (Hossin vd., 2015). Doğruluk değeri Denklem 3.9 kullanılarak hesaplanır.

$$\text{Doğruluk} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{3.9}$$

3.5.2. Kesinlik

Kesinlik, sınıfı 1 olarak tahmin edilmiş (TP) örnek sayısının, sınıfı 1 olarak tahmin edilmiş tüm örnek sayısına (TP+FP) oranıdır (Nizam ve Akın, 2014). Kesinlik değeri Denklem 3.10 kullanılarak hesaplanır.

$$\text{Kesinlik} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.10}$$

3.5.3. Duyarlılık

Duyarlılık, doğru sınıflandırılmış pozitif örnek (TP) sayısının, toplam pozitif örnek sayısına (TP+FN) oranıdır (Nizam ve Akın, 2014). Duyarlılık değeri Denklem 3.11 kullanılarak hesaplanır.

$$\text{Duyarlılık} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.11}$$

3.6. Çalışmada Kullanılan Kütüphaneler

Çalışmadaki kodlar, Python programlama dilinin 3. versiyonu kullanılarak geliştirilmiştir. Metin ön işlemleri için açık kaynak kodlu NLTK Kütüphanesi, kelime köklerinin bulunması işlemi için Java programlama dili kullanılarak geliştirilmiş Zemberek Kütüphanesi, Zemberek Kütüphanesinin kodlarını Python programlama dilinde açabilmek için Jpype Kütüphanesi kullanılmıştır (Loper ve Bird, 2002). Sınıflandırma algoritmalarından MNB ve RF algoritmaları için Scikit-learn Kütüphanesi; CNN ve LSTM algoritmaları için ise Keras Kütüphanesi kullanılmıştır (Chollet, 2019). Derin öğrenme algoritmalarındaki matris işlemleri için Numpy Kütüphanesi, .txt formatlı veri setinin okunması için Glob Kütüphanesi projeye eklenmiştir (Oliphant, 2006). Modellerin sonuçlarının görselleştirilebilmesi için Matplotlib ve Seaborn Kütüphaneleri eklenmiştir (Hunter, 2007). Kodlar 25 gigabayta kadar ücretsiz GPU desteği sunan Google Colob platformu kullanılarak çalıştırılmıştır (Bisong, 2019). Çizelge 3.7'de çalışmada kullanılan kütüphaneler ve bu kütüphanelerin hangi aşamalarda kullanıldığı belirtilmiştir.

Çizelge 3.7. Çalışmada Kullanılan Kütüphaneler

Glob	**Veri setinin okunması**	**Aşama**
NLTK	Metin ön işlemleri	Doğal dil işleme aşamaları
re	Noktalama işaretleri ve sayıların atılması	
Zemberek	Kelime köklerinin bulunması	
Jpype	Zemberek Kütüphanesi kodlarının çalıştırılması	
Numpy	Matris işlemleri	
Scikit-learn	MNB, RF algoritmalarının çalıştırılması	Modelleme
Keras	CNN, LSTM algoritmalarının çalıştırılması	
Matplotlib	Görselleştirme	Modelin değerlendirilmesi
Seaborn	Görselleştirme	

4. ARAŞTIRMA BULGULARI

Bu bölümde, Türkçe için yazar, tür ve cinsiyet tanıma problemlerinin çözümüne yönelik, her bir veri seti için, ayrı olarak yapılan modelleme çalışmaları anlatılmıştır. Bu modelleme çalışmaları, makine öğrenmesi modelleri ve derin öğrenme modelleri olmak üzere 2 alt başlık altında detaylarıyla verilmiştir.

4.1. Makine Öğrenmesi Modelleri

Yazar, tür ve cinsiyet tanıma için oluşturulan veri setlerine, doğal dil işleme aşamalarından sonra, makine öğrenmesi algoritmalarından MNB ve RF sınıflandırıcıları uygulanmıştır. Makine öğrenmesi algoritmalarında, Sklearn Kütüphanesi içinde yer alan Pipeline, yani boru hattı metodu kullanılmıştır. Pipeline metodu içerisinde TfidfVectorizer ile metin verileri sayı vektörlerine dönüştürülmüş ve bu adım sonucunda elde edilmiş olan vektör çıktıları MultinomialNB ve RandomForestClassifier sınıflandırıcılarının girdisi olarak kullanılarak sınıflandırma işlemi gerçekleştirilmiştir. Çalışmada kullanılan tüm veri setlerinde bu işlem adımları gerçekleştirilmiştir. Şekil 4.1'de MNB ve RF algoritmalarının kod bloklarına yer verilmiştir.

```
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB

model = Pipeline([
    ('vect',TfidfVectorizer(lowercase=False, use_idf=True)),
    ('multinomial_bayes', MultinomialNB())
    ])
```

```
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier

random_forest = Pipeline([
        ('vect',TfidfVectorizer(lowercase=False, use_idf=True)),
        ('RandomForestClassifier', RandomForestClassifier(n_estimators=100))
        ])
```

Şekil 4.1. MNB ve RF Algoritmalarının Kod Blokları

Yazar, tür ve cinsiyet tanıma işlemleri için, tüm veri setlerinin her birinin MNB ve RF modelleri sonuçlarına Çizelge 4.1'de yer verilmiştir.

Çizelge 4.1. Tüm Veri Setleri için Makine Öğrenmesi Modellerinin Sonuçları

Metin Sınıflandırma	Veri Seti	Model	Doğruluk		Kesinlik		Duyarlılık	
			Eğitim	Test	Eğitim	Test	Eğitim	Test
Yazar Tanıma	AI-TNKU-1	MNB	87.00	68.47	90.59	76.32	87.00	68.47
		RF	100.0	**76.80**	100.0	**76.74**	100.0	**76.80**
	AI-TNKU-2	MNB	87.70	75.64	90.29	81.52	87.70	75.64
		RF	100.0	**80.60**	100.0	**80.62**	100.0	**80.60**
	AI-TNKU-3	MNB	89.42	80.09	91.28	84.98	89.42	80.09
		RF	100.0	**83.93**	100.0	**84.39**	100.0	**83.93**
	AI-TNKU-4	MNB	89.59	81.15	91.27	85.35	89.59	81.15
		RF	100.0	**85.17**	100.0	**85.47**	100.0	**85.17**
	AI-TNKU-5	MNB	91.92	85.47	92.98	88.11	91.92	85.47
		RF	100.0	88.88	100.0	89.16	100.0	88.88
	AI-TNKU-6	MNB	93.05	88.41	93.58	89.44	93.05	88.41
		RF	100.0	90.41	100.0	90.39	100.0	90.41
	AI-TNKU-7	MNB	95.65	92.87	95.93	93.59	95.65	92.87
		RF	100.0	95.72	100.0	95.85	100.0	95.72
Tür Tanıma	GI-TNKU-1	MNB	82.75	75.96	84.11	77.92	82.75	75.96
		RF	100.0	83.04	100.0	83.54	100.0	83.04
	GI-TNKU-2	MNB	85.05	79.02	85.87	80.42	85.05	79.01
		RF	100.0	84.52	100.0	84.96	100.0	84.52
	GI-TNKU-3	MNB	88.81	84.79	89.30	85.69	88.81	84.79
		RF	100.0	89.18	100.0	89.45	100.0	89.18
	GI-TNKU-4	MNB	88.52	84.97	89.11	85.99	88.52	84.96
		RF	100.0	88.34	100.0	88.87	100.0	88.33
	GI-TNKU-5	MNB	90.63	86.64	90.78	86.90	90.63	86.64
		RF	100.0	92.47	100.0	92.60	100.0	92.47
	GI-TNKU-6	MNB	93.45	90.72	93.56	90.60	93.36	90.12
		RF	100.0	95.93	100.0	95.95	100.0	95.93
Cinsiyet Tanıma	IAG-TNKU	MNB	85.09	82.93	85.10	82.98	85.09	82.93
		RF	99.99	84.95	99.99	85.25	99.99	85.18

4.2. Derin Öğrenme Modelleri

Yazar, tür ve cinsiyet tanıma için oluşturulan veri setlerine, doğal dil işleme aşamalarından geçtikten sonra, derin öğrenme algoritmalarından CNN ve LSTM algoritmalarına uygulanarak, en iyi modeller yapılan uzun deneysel çalışmalar sonucunda bulunmuştur. Burada, deneysel çalışmalar sonucunda bulunan en iyi modellere ait hiper parametre değerleri ve mimari bilgilerine yer verilmiştir. Derin öğrenme modelleri oluşturulurken Keras Kütüphanesi kullanılmıştır. Modeller temel olarak; katmanların eklenmesi, modelde kullanılacak hiper parametrelerin belirlenmesi ve modelin eğitilmesi şeklinde oluşturulmuştur.

Hiper parametreler, uygulanan modelin veriden öğrenerek ya da tahmin ederek değiştiremediği, modeli tasarlayan kişi tarafından belirlenen parametrelerdir (Tanyıldızı ve Demirtaş, 2019). Bu çalışmada embedding katmanında; max_len, max_words, embedding_dim hiper parametreleri, modeller oluşturulurken; nöron sayısı, kernel boyutu, aktivasyon fonksiyonu, katman sayısı, dropout katsayısı hiper parametreleri model çalıştırılırken ise; epoch, batch_size, optimizer ve loss fonksiyonu hiper parametrelerinin modelin başarısını arttıran uygun değerleri deneysel çalışmalar sonucunda elde edilmiştir.

Derin öğrenme modellerinde, hem CNN modelinde hem de LSTM modelinde, sırasıyla Sequential ve Embedding katmanları ile model oluşturulmaya başlanmıştır. Modellerin embedding katmanlarında, maksimum uzunluk boyutu (max_len) parametresi, maksimum kelime sayısı (max_words) parametresi ve embedding boyutu (embedding_dim) parametresi için, her bir veri setine özel yapılan deneysel modelleme çalışmaları sonucunda Çizelge 4.2'de belirtilen değerler bulunmuştur.

Çizelge 4.2. Derin Öğrenme Modellerinin (CNN ve LSTM) Embedding Katmanında Kullanılan Parametreler

<table>
<tr><th>Metin Sınıflandırma</th><th>Veri Seti</th><th>max_len</th><th>max_words</th><th>embedding_dim</th></tr>
<tr><td rowspan="7">Yazar Tanıma</td><td>AI-TNKU-1</td><td rowspan="4">2.000</td><td rowspan="4">100.000</td><td rowspan="4">300</td></tr>
<tr><td>AI-TNKU-2</td></tr>
<tr><td>AI-TNKU-3</td></tr>
<tr><td>AI-TNKU-4</td></tr>
<tr><td>AI-TNKU-5</td><td>2.000</td><td>150.000</td><td>400</td></tr>
<tr><td>AI-TNKU-6</td><td rowspan="2">1.000</td><td rowspan="2">100.000</td><td rowspan="2">400</td></tr>
<tr><td>AI-TNKU-7</td></tr>
<tr><td rowspan="6">Tür Tanıma</td><td>GI-TNKU-1</td><td rowspan="6">1.000</td><td rowspan="6">100.000</td><td rowspan="6">300</td></tr>
<tr><td>GI-TNKU-2</td></tr>
<tr><td>GI-TNKU-3</td></tr>
<tr><td>GI-TNKU-4</td></tr>
<tr><td>GI-TNKU-5</td></tr>
<tr><td>GI-TNKU-6</td></tr>
<tr><td>Cinsiyet Tanıma</td><td>IAG-TNKU</td><td>1.000</td><td>100.000</td><td>300</td></tr>
</table>

4.2.1. CNN Modelleri

Bu bölümde, oluşturulan yazar, tür ve cinsiyet tanıma veri setleri üzerinde, CNN algoritmasının kullanıldığı en iyi modellerden bahsedilmiştir. CNN modelinde, Sequential ve Embedding katmanlarının ardından, her veri setine özel Konvolüsyon, Pooling ve Dense katmanları gelmektedir. Aşağıda, her bir veri setine CNN algoritmasının uygulanması sonucu, en iyi hiper parametrelerin bulunmasıyla oluşturulan en iyi CNN modellerinin özet mimarileri verilmiştir:

1. **AI-TNKU-1 Veri Seti için CNN Modeli**

Yazar tanıma için oluşturulan AI-TNKU-1 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, metin verileri ile çalışıldığı için 16 nörondan oluşan conv1D katmanı ve aktivasyon fonksiyonu olarak ReLu eklenmiştir. Konvolüsyon katmanında kernel boyutu olarak 3×3 seçilmiştir. Sonrasında art arda 2 tane MaxPooling1D ve 2 tane 256 nörondan oluşan Dense katmanı modele eklenmiştir. Dense katmanlarında aktivasyon fonksiyonu olarak ReLu kullanılmıştır. Flatten katmanı eklenmiştir. Çıkış katmanı olan Dense katmanına 68 nöron eklenmiş ve aktivasyon fonksiyonu olarak softmax kullanılarak model tamamlanmıştır. Şekil 4.2'de, modelleme çalışmaları sonucunda, AI-TNKU-1 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiştir ve model 10 Fold Cross-Validation uygulanarak 6 epoch çalıştırılmıştır.

Layer (type)	Output Shape	Param #
embedding_10 (Embedding)	(None, 2000, 300)	45000000
conv1d_10 (Conv1D)	(None, 1998, 16)	14416
max_pooling1d_19 (MaxPooling	(None, 999, 16)	0
max_pooling1d_20 (MaxPooling	(None, 499, 16)	0
dense_28 (Dense)	(None, 499, 256)	4352
dense_29 (Dense)	(None, 499, 256)	65792
flatten_10 (Flatten)	(None, 127744)	0
dense_30 (Dense)	(None, 68)	8686660

Total params: 53,771,220
Trainable params: 53,771,220
Non-trainable params: 0

Şekil 4.2. AI-TNKU-1 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

2. AI-TNKU-2 Veri Seti için CNN Modeli

Yazar tanıma için oluşturulan AI-TNKU-2 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, 32 nöron, kernel boyutu olarak 3×3'lük matrisin ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Conv1D katmanı eklenmiştir. Daha sonra modele 2 tane MaxPooling1D katmanı, 256 nörondan oluşan Dense, Flatten ve 128 nörondan oluşan Dense katmanları eklenmiştir. Dense katmanlarında aktivasyon fonksiyonu olarak ReLu tercih edilmiştir. Çıkış katmanı olan Dense katmanında 50 nöron ve aktivasyon fonksiyonu olarak softmax kullanılmıştır. Şekil 4.3'de, modelleme çalışmaları sonucunda, AI-TNKU-2 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiştir ve model 10 Fold Cross-Validation uygulanarak 5 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 2000, 300)         45000000
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 1998, 32)          28832
_________________________________________________________________
max_pooling1d_19 (MaxPooling (None, 999, 32)           0
_________________________________________________________________
max_pooling1d_20 (MaxPooling (None, 499, 32)           0
_________________________________________________________________
dense_28 (Dense)             (None, 499, 256)          8448
_________________________________________________________________
flatten_10 (Flatten)         (None, 127744)            0
_________________________________________________________________
dense_29 (Dense)             (None, 128)               16351360
_________________________________________________________________
dense_30 (Dense)             (None, 50)                6450
=================================================================
Total params: 61,395,090
Trainable params: 61,395,090
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.3. AI-TNKU-2 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

3. AI-TNKU-3 Veri Seti için CNN Modeli

Yazar tanıma için oluşturulan AI-TNKU-3 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, 64 nöron, 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Conv1D katmanı, sonrasında 2 tane MaxPooling1D katmanları ve 256 nörondan oluşan ve yine bu katmanda aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı modele eklenmiştir. Sonrasında Flatten katmanı ve 38 nörondan oluşan ve aktivasyon fonksiyonu olarak softmax fonksiyonunun kullanıldığı çıkış katmanı olan Dense katmanı eklenerek model tamamlanmıştır. Şekil 4.4'de, modelleme çalışmaları sonucunda, AI-TNKU-3 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiştir ve model 10 Fold Cross-Validation uygulanarak 5 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 2000, 300)         45000000
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 1998, 64)          57664
_________________________________________________________________
max_pooling1d_19 (MaxPooling (None, 999, 64)           0
_________________________________________________________________
max_pooling1d_20 (MaxPooling (None, 499, 64)           0
_________________________________________________________________
dense_19 (Dense)             (None, 499, 256)          16640
_________________________________________________________________
flatten_10 (Flatten)         (None, 127744)            0
_________________________________________________________________
dense_20 (Dense)             (None, 38)                4854310
=================================================================
Total params: 49,928,614
Trainable params: 49,928,614
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.4. AI-TNKU-3 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

4. AI-TNKU-4 Veri Seti için CNN Modeli

Yazar tanıma için oluşturulan AI-TNKU-4 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, 16 nöron, 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Conv1D katmanı, sonrasında 2 tane MaxPooling1D katmanları ve 512 nörondan oluşan ve yine bu katmanda aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı modele eklenmiştir. Sonrasında Flatten katmanı ve 33 nörondan oluşan ve aktivasyon fonksiyonu olarak softmax fonksiyonunun kullanıldığı çıkış katmanı olan Dense katmanı eklenerek model tamamlanmıştır. Şekil 4.5'de, modelleme çalışmaları sonucunda, AI-TNKU-4 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiştir ve model 10 Fold Cross-Validation uygulanarak 5 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 2000, 300)         45000000
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 1998, 16)          14416
_________________________________________________________________
max_pooling1d_19 (MaxPooling (None, 999, 16)           0
_________________________________________________________________
max_pooling1d_20 (MaxPooling (None, 499, 16)           0
_________________________________________________________________
dense_19 (Dense)             (None, 499, 512)          8704
_________________________________________________________________
flatten_10 (Flatten)         (None, 255488)            0
_________________________________________________________________
dense_20 (Dense)             (None, 33)                8431137
=================================================================
Total params: 53,454,257
Trainable params: 53,454,257
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.5. AI-TNKU-4 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

5. AI-TNKU-5 Veri Seti için CNN Modeli

Yazar tanıma için oluşturulan AI-TNKU-5 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, 32 nöron, 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Conv1D katmanı, sonrasında 2 tane MaxPooling1D katmanları ve Flatten katmanı modele eklenmiştir. Modelin devamında art arda 2 tane Dense katmanı kullanılmıştır. Bu Dense katmanlarının ilkinde 256 nöron ve aktivasyon fonksiyonu olarak ReLu diğerinde ise veri seti 27 sınıftan oluştuğu için 27 nöron ve aktivasyon fonksiyonu olarak softmax kullanılarak model tamamlanmıştır. Şekil 4.6'da, modelleme çalışmaları sonucunda, AI-TNKU-5 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiştir ve model 10 Fold Cross-Validation uygulanarak 4 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 2000, 400)         60000000
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 1998, 32)          38432
_________________________________________________________________
max_pooling1d_19 (MaxPooling (None, 999, 32)           0
_________________________________________________________________
max_pooling1d_20 (MaxPooling (None, 499, 32)           0
_________________________________________________________________
flatten_10 (Flatten)         (None, 15968)             0
_________________________________________________________________
dense_19 (Dense)             (None, 256)               4088064
_________________________________________________________________
dense_20 (Dense)             (None, 27)                6939
=================================================================
Total params: 64,133,435
Trainable params: 64,133,435
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.6. AI-TNKU-5 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

6. AI-TNKU-6 Veri Seti için CNN Modeli

Yazar tanıma için oluşturulan AI-TNKU-6 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, 32 nöron, 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Conv1D katmanı, sonrasında 2 tane MaxPooling1D katmanları ve Flatten katmanı modele eklenmiştir. Modelin devamında art arda 2 tane Dense katmanı kullanılmıştır. Bu Dense katmanlarının ilkinde 512 nöron ve aktivasyon fonksiyonu olarak ReLu diğerinde ise veri seti 16 sınıftan oluştuğu için 16 nöron ve aktivasyon fonksiyonu olarak softmax kullanılarak model tamamlanmıştır. Şekil 4.7'de, modelleme çalışmaları sonucunda, AI-TNKU-6 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiştir ve model 10 Fold Cross-Validation uygulanarak 4 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 400)         40000000
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 998, 32)           38432
_________________________________________________________________
max_pooling1d_19 (MaxPooling (None, 499, 32)           0
_________________________________________________________________
max_pooling1d_20 (MaxPooling (None, 249, 32)           0
_________________________________________________________________
flatten_10 (Flatten)         (None, 7968)              0
_________________________________________________________________
dense_19 (Dense)             (None, 512)               4080128
_________________________________________________________________
dense_20 (Dense)             (None, 16)                8208
=================================================================
Total params: 44,126,768
Trainable params: 44,126,768
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.7. AI-TNKU-6 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

7. AI-TNKU-7 Veri Seti için CNN Modeli

Yazar tanıma için oluşturulan AI-TNKU-7 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, 64 nöron, 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Conv1D katmanı, sonrasında MaxPooling1D katmanı ve Flatten katmanı modele eklenmiştir. Modelin devamında art arda 2 tane Dense katmanı kullanılmıştır. Bu Dense katmanlarının ilkinde 512 nöron ve aktivasyon fonksiyonu olarak ReLu diğerinde ise veri seti 9 sınıftan oluştuğu için 9 nöron ve aktivasyon fonksiyonu olarak softmax kullanılarak model tamamlanmıştır. Şekil 4.8'de, modelleme çalışmaları sonucunda, AI-TNKU-7 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 3 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 400)         40000000
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 998, 64)           76864
_________________________________________________________________
max_pooling1d_10 (MaxPooling (None, 499, 64)           0
_________________________________________________________________
flatten_10 (Flatten)         (None, 31936)             0
_________________________________________________________________
dense_19 (Dense)             (None, 512)               16351744
_________________________________________________________________
dense_20 (Dense)             (None, 9)                 4617
=================================================================
Total params: 56,433,225
Trainable params: 56,433,225
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.8. AI-TNKU-7 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

8. GI-TNKU-1 Veri Seti için CNN Modeli

Tür tanıma için oluşturulan GI-TNKU-1 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 32 nöron, 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Conv1D katmanı, MaxPooling1D katmanı ve Flatten katmanı eklenmiştir. Modelin devamında 512 nöron ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Dense katmanı, 0.1 katsayısının kullanıldığı Dropout katmanı eklenmiştir. Çıkış katmanı olan Dense katmanında ise veri seti 7 sınıftan oluştuğu için 7 nöron ve aktivasyon fonksiyonu olarak softmax kullanılarak model tamamlanmıştır. Şekil 4.9'da, modelleme çalışmaları sonucunda, GI-TNKU-1 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 3 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 998, 32)           28832
_________________________________________________________________
max_pooling1d_10 (MaxPooling (None, 499, 32)           0
_________________________________________________________________
flatten_10 (Flatten)         (None, 15968)             0
_________________________________________________________________
dense_19 (Dense)             (None, 512)               8176128
_________________________________________________________________
dropout_10 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_20 (Dense)             (None, 7)                 3591
=================================================================
Total params: 38,208,551
Trainable params: 38,208,551
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.9. GI-TNKU-1 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

9. GI-TNKU-2 Veri Seti için CNN Modeli

Tür tanıma için oluşturulan GI-TNKU-2 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 32 nöron, 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu’nun kullanıldığı Conv1D katmanı ve MaxPooling1D katmanı eklenmiştir. Modelin devamında tekrardan 64 nöron ve 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu’nun kullanıldığı Conv1D katmanı ve MaxPooling1D katmanı modele eklenmiştir. Ardından Flatten ve 512 nöron ve aktivasyon fonksiyonu olarak ReLu’nun tercih edildiği Dense katmanı kullanılmıştır. Çıkış katmanı olan Dense katmanında ise veri seti 6 sınıftan oluştuğu için 6 nöron ve aktivasyon fonksiyonu olarak softmax kullanılarak model tamamlanmıştır. Şekil 4.10’da, GI-TNKU-2 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 3 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000
_________________________________________________________________
conv1d_19 (Conv1D)           (None, 998, 32)           28832
_________________________________________________________________
max_pooling1d_19 (MaxPooling (None, 499, 32)           0
_________________________________________________________________
conv1d_20 (Conv1D)           (None, 497, 64)           6208
_________________________________________________________________
max_pooling1d_20 (MaxPooling (None, 248, 64)           0
_________________________________________________________________
flatten_10 (Flatten)         (None, 15872)             0
_________________________________________________________________
dense_19 (Dense)             (None, 512)               8126976
_________________________________________________________________
dense_20 (Dense)             (None, 6)                 3078
=================================================================
Total params: 38,165,094
Trainable params: 38,165,094
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.10. GI-TNKU-2 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

10. GI-TNKU-3 Veri Seti için CNN Modeli

Tür tanıma için oluşturulan GI-TNKU-3 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 32 nöron, 5×5 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Conv1D katmanı, MaxPooling1D katmanı ve 0.2 katsayısının belirlendiği Dropout katmanı eklenmiştir. Modelin devamında tekrardan 32 nöron ve 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Conv1D katmanı ve MaxPooling1D katmanı modele eklenmiştir. Ardından Flatten ve 512 nöron ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Dense katmanı ve yine 0.2 katsayısının belirlendiği Dropout katmanı kullanılmıştır. Çıkış katmanı olan Dense katmanında ise veri seti 5 sınıftan oluştuğu için 5 nöron ve aktivasyon fonksiyonu olarak softmax kullanılarak model tamamlanmıştır. Şekil 4.11'de, GI-TNKU-3 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 4 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         3000000
_________________________________________________________________
conv1d_19 (Conv1D)           (None, 996, 32)           48032
_________________________________________________________________
max_pooling1d_19 (MaxPooling (None, 498, 32)           0
_________________________________________________________________
dropout_19 (Dropout)         (None, 498, 32)           0
_________________________________________________________________
conv1d_20 (Conv1D)           (None, 496, 32)           3104
_________________________________________________________________
max_pooling1d_20 (MaxPooling (None, 248, 32)           0
_________________________________________________________________
flatten_10 (Flatten)         (None, 7936)              0
_________________________________________________________________
dense_19 (Dense)             (None, 512)               4063744
_________________________________________________________________
dropout_20 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_20 (Dense)             (None, 5)                 2565
=================================================================
Total params: 7,117,445
Trainable params: 7,117,445
Non-trainable params: 0
```

Şekil 4.11. GI-TNKU-3 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

11. GI-TNKU-4 Veri Seti için CNN Modeli

Tür tanıma için oluşturulan GI-TNKU-4 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 64 nöron, 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Conv1D katmanı ve MaxPooling1D katmanı modele eklenmiştir. Modelin devamında tekrardan 64 nöron ve 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Conv1D katmanı ve MaxPooling1D katmanı modele eklenmiştir. Ardından Flatten ve 512 nöron ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Dense katmanı ve 0.2 katsayısının belirlendiği Dropout katmanı kullanılmıştır. Çıkış katmanı olan Dense katmanında ise veri seti 4 sınıftan oluştuğu için 4 nöron ve aktivasyon fonksiyonu olarak softmax kullanılarak model tamamlanmıştır. Şekil 4.12'de, GI-TNKU-4 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 2 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000  
_________________________________________________________________
conv1d_19 (Conv1D)           (None, 998, 64)           57664     
_________________________________________________________________
max_pooling1d_19 (MaxPooling (None, 499, 64)           0         
_________________________________________________________________
conv1d_20 (Conv1D)           (None, 497, 64)           12352     
_________________________________________________________________
max_pooling1d_20 (MaxPooling (None, 248, 64)           0         
_________________________________________________________________
flatten_10 (Flatten)         (None, 15872)             0         
_________________________________________________________________
dense_19 (Dense)             (None, 512)               8126976   
_________________________________________________________________
dropout_10 (Dropout)         (None, 512)               0         
_________________________________________________________________
dense_20 (Dense)             (None, 4)                 2052      
=================================================================
Total params: 38,199,044
Trainable params: 38,199,044
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.12. GI-TNKU-4 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

12. GI-TNKU-5 Veri Seti için CNN Modeli

Tür tanıma için oluşturulan GI-TNKU-5 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 32 nöron, 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Conv1D katmanı ve MaxPooling1D katmanı modele eklenmiştir. Modelin devamında tekrardan 32 nöron ve 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Conv1D katmanı ve MaxPooling1D katmanı modele eklenmiştir. Ardından Flatten ve 512 nöron ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Dense katmanı kullanılmıştır. Çıkış katmanı olan Dense katmanında ise veri seti 3 sınıftan oluştuğu için 3 nöron ve aktivasyon fonksiyonu olarak softmax kullanılarak model tamamlanmıştır. Şekil 4.13'de, GI-TNKU-5 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 3 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_13 (Embedding)     (None, 1000, 300)         30000000
_________________________________________________________________
conv1d_25 (Conv1D)           (None, 998, 32)           28832
_________________________________________________________________
max_pooling1d_25 (MaxPooling (None, 499, 32)           0
_________________________________________________________________
conv1d_26 (Conv1D)           (None, 497, 32)           3104
_________________________________________________________________
max_pooling1d_26 (MaxPooling (None, 248, 32)           0
_________________________________________________________________
flatten_13 (Flatten)         (None, 7936)              0
_________________________________________________________________
dense_25 (Dense)             (None, 512)               4063744
_________________________________________________________________
dense_26 (Dense)             (None, 3)                 1539
=================================================================
Total params: 34,097,219
Trainable params: 34,097,219
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.13. GI-TNKU-5 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

13. GI-TNKU-6 Veri Seti için CNN Modeli

Tür tanıma için oluşturulan GI-TNKU-6 adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 3 kez 64 nöron, 3×3 matris boyutunda kernel ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Conv1D katmanı ve MaxPooling1D katmanları modele eklenmiştir. Ardından tekrardan MaxPooling1D katmanı, Flatten ve 512 nöron ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Dense katmanı kullanılmıştır. Dropout katmanı 0.2 katsayısı kullanılarak eklenmiştir. Çıkış katmanı olan Dense katmanında ise veri seti 2 sınıftan oluştuğu için 1 nöron ve aktivasyon fonksiyonu olarak sigmoid kullanılarak model tamamlanmıştır. Şekil 4.14'de, GI-TNKU-6 veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak binary_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 3 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000
_________________________________________________________________
conv1d_28 (Conv1D)           (None, 998, 64)           57664
_________________________________________________________________
max_pooling1d_37 (MaxPooling (None, 499, 64)           0
_________________________________________________________________
conv1d_29 (Conv1D)           (None, 497, 64)           12352
_________________________________________________________________
max_pooling1d_38 (MaxPooling (None, 248, 64)           0
_________________________________________________________________
conv1d_30 (Conv1D)           (None, 246, 64)           12352
_________________________________________________________________
max_pooling1d_39 (MaxPooling (None, 123, 64)           0
_________________________________________________________________
max_pooling1d_40 (MaxPooling (None, 61, 64)            0
_________________________________________________________________
flatten_10 (Flatten)         (None, 3904)              0
_________________________________________________________________
dense_19 (Dense)             (None, 512)               1999360
_________________________________________________________________
dropout_10 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_20 (Dense)             (None, 1)                 513
=================================================================
Total params: 32,082,241
Trainable params: 32,082,241
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.14. GI-TNKU-6 Veri Seti için En İyi CNN Modelinin Özet Mimarisi

14. IAG-TNKU Veri Seti için CNN Modeli

Cinsiyet tanıma için oluşturulan IAG-TNKU adlı veri setinin, modelleme sonucunda bulunan en iyi CNN modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 64 nöron, 3×3 matris boyutundan oluşan kernel ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Conv1D katmanı, MaxPooling1D ve Flatten katmanları eklenmiştir. Modelin devamında 128 nöron, aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense, 0.2 katsayısının tercih edildiği Dropout katmanları kullanılmıştır. Çıkış katmanı olan Dense katmanında ise veri seti 2 sınıftan oluştuğu için 1 nöron ve aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılarak model tamamlanmıştır. Şekil 4.15'de, IAG-TNKU veri seti için bulunmuş olan en iyi CNN modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 128, loss fonksiyonu olarak binary_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 2 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000  
_________________________________________________________________
conv1d_10 (Conv1D)           (None, 998, 64)           57664     
_________________________________________________________________
max_pooling1d_10 (MaxPooling (None, 499, 64)           0         
_________________________________________________________________
flatten_10 (Flatten)         (None, 31936)             0         
_________________________________________________________________
dense_19 (Dense)             (None, 128)               4087936   
_________________________________________________________________
dropout_10 (Dropout)         (None, 128)               0         
_________________________________________________________________
dense_20 (Dense)             (None, 1)                 129       
=================================================================
Total params: 34,145,729
Trainable params: 34,145,729
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.15. IAG-TNKU Veri Seti için En İyi CNN Modelinin Özet Mimarisi

4.2.2. LSTM Modelleri

Bu bölümde, oluşturulan yazar, tür ve cinsiyet tanıma veri setleri üzerinde, LSTM algoritmasının kullanıldığı en iyi modellerden bahsedilmiştir. LSTM modelinde, Sequential ve Embedding katmanlarının ardından, her veri setine özel LSTM, Dense, Pooling ve Flatten katmanları gelmektedir. Aşağıda, her bir veri setine LSTM algoritmasının uygulanması sonucu, en iyi hiper parametrelerin bulunmasıyla oluşturulan en iyi LSTM modellerinin özet mimarileri verilmiştir:

1. **AI-TNKU-1 Veri Seti için LSTM Modeli**

Yazar tanıma için oluşturulan AI-TNKU-1 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırası ile 32 nörondan oluşan LSTM katmanı, 256 nörondan oluşan TimeDistributedDense katmanı eklenmiştir. Sonrasında art arda 2 kere AveragePooling1D katmanları, 256 nörondan oluşan Dense katmanı eklenmiş ve Dense katmanında aktivasyon fonksiyonu olarak ReLu kullanılmıştır. Dropout katmanında 0.2 katsayısı kullanılmış ve daha sonra Flatten katmanı eklenmiştir. Çıkış katmanı olan Dense katmanında veri seti 68 sınıftan oluştuğu için 68 nöron, aktivasyon fonksiyonu olarak softmax kullanılmıştır. Şekil 4.16'da AI-TNKU-1 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 6 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         45000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 1000, 32)          42624
_________________________________________________________________
time_distributed_10 (TimeDis (None, 1000, 256)         8448
_________________________________________________________________
average_pooling1d_19 (Averag (None, 500, 256)          0
_________________________________________________________________
average_pooling1d_20 (Averag (None, 250, 256)          0
_________________________________________________________________
dense_29 (Dense)             (None, 250, 256)          65792
_________________________________________________________________
dropout_10 (Dropout)         (None, 250, 256)          0
_________________________________________________________________
flatten_10 (Flatten)         (None, 64000)             0
_________________________________________________________________
dense_30 (Dense)             (None, 68)                4352068
=================================================================
Total params: 49,468,932
Trainable params: 49,468,932
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.16. AI-TNKU-1 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

2. AI-TNKU-2 Veri Seti için LSTM Modeli

Yazar tanıma için oluşturulan AI-TNKU-2 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 12 nörondan oluşan LSTM katmanı, 200 nörondan oluşan TimeDistributedDense katmanı ve 0.2 katsayısı kullanılarak Dropout katmanları eklenmiştir. Bu katmanların ardından 2 tane AveragePooling1D katmanı, 16 nörondan oluşan Dense katmanı ve Flatten katmanları eklenmiş ve Dense katmanında aktivasyon fonksiyonu olarak ReLu kullanılmıştır. Çıkış katmanı olan Dense katmanında veri seti 50 sınıftan oluştuğu için 50 nöron, aktivasyon fonksiyonu olarak da softmax kullanılmıştır. Şekil 4.17'de, AI-TNKU-2 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 5 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_10 (Embedding)     (None, 2000, 300)         45000000  
_________________________________________________________________
lstm_10 (LSTM)               (None, 2000, 12)          15024     
_________________________________________________________________
time_distributed_10 (TimeDis (None, 2000, 200)         2600      
_________________________________________________________________
dropout_10 (Dropout)         (None, 2000, 200)         0         
_________________________________________________________________
average_pooling1d_19 (Averag (None, 1000, 200)         0         
_________________________________________________________________
average_pooling1d_20 (Averag (None, 500, 200)          0         
_________________________________________________________________
dense_29 (Dense)             (None, 500, 16)           3216      
_________________________________________________________________
flatten_10 (Flatten)         (None, 8000)              0         
_________________________________________________________________
dense_30 (Dense)             (None, 50)                400050    
=================================================================
Total params: 45,420,890
Trainable params: 45,420,890
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.17. AI-TNKU-2 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

3. AI-TNKU-3 Veri Seti için LSTM Modeli

Yazar tanıma için oluşturulan AI-TNKU-3 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 200 nörondan oluşan TimeDistributedDense katmanı, 2 tane AveragePooling1D katmanları ve 0.3 katsayısının belirlendiği Dropout katmanı eklenmiştir. Modelin devamında 32 nöronun kullanıldığı ve aktivasyon fonksiyonu olarak ReLu'nun seçildiği Dense katmanı ve Flatten katmanı eklenmiştir. Çıkış katmanı olan Dense katmanında veri seti 38 sınıftan oluştuğu için 38 nöron ve aktivasyon fonksiyonu olarak softmax kullanılarak model tamamlanmıştır. Şekil 4.18'de, AI-TNKU-3 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 5 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 2000, 300)         45000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 2000, 16)          20288
_________________________________________________________________
time_distributed_10 (TimeDis (None, 2000, 200)         3400
_________________________________________________________________
average_pooling1d_19 (Averag (None, 1000, 200)         0
_________________________________________________________________
average_pooling1d_20 (Averag (None, 500, 200)          0
_________________________________________________________________
dropout_10 (Dropout)         (None, 500, 200)          0
_________________________________________________________________
dense_29 (Dense)             (None, 500, 32)           6432
_________________________________________________________________
flatten_10 (Flatten)         (None, 16000)             0
_________________________________________________________________
dense_30 (Dense)             (None, 38)                608038
=================================================================
Total params: 45,638,158
Trainable params: 45,638,158
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.18. AI-TNKU-3 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

4. AI-TNKU-4 Veri Seti için LSTM Modeli

Yazar tanıma için oluşturulan AI-TNKU-4 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 200 nörondan oluşan TimeDistributedDense katmanı ve ardından 2 tane AveragePooling1D katmanları modele eklenmiştir. Modelin devamında 64 nörondan oluşan ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı ve sonrasında Flatten katmanı eklenmiştir. Çıkış katmanı olan Dense katmanına 33 sınıftan oluşan bir veri seti kullanıldığı için 33 nöron ve aktivasyon fonksiyonu olarak da softmax fonksiyonu kullanılmıştır. Şekil 4.19'da, AI-TNKU-4 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 5 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 2000, 300)         45000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 2000, 16)          20288
_________________________________________________________________
time_distributed_10 (TimeDis (None, 2000, 200)         3400
_________________________________________________________________
average_pooling1d_19 (Averag (None, 1000, 200)         0
_________________________________________________________________
average_pooling1d_20 (Averag (None, 500, 200)          0
_________________________________________________________________
dense_29 (Dense)             (None, 500, 64)           12864
_________________________________________________________________
flatten_10 (Flatten)         (None, 32000)             0
_________________________________________________________________
dropout_10 (Dropout)         (None, 32000)             0
_________________________________________________________________
dense_30 (Dense)             (None, 33)                1056033
=================================================================
Total params: 46,092,585
Trainable params: 46,092,585
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.19. AI-TNKU-4 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

5. AI-TNKU-5 Veri Seti için LSTM Modeli

Yazar tanıma için oluşturulan AI-TNKU-5 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 250 nörondan oluşan TimeDistributedDense katmanı ve ardından 2 tane AveragePooling1D katmanları modele eklenmiştir. Modelin devamında Flatten katmanı, 512 nörondan ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı kullanılmıştır. Sonrasında 0.2 katsayısının belirlendiği Dropout katmanı, 27 sınıftan oluşan bir veri seti kullanıldığı için 27 nörondan oluşan ve aktivasyon fonksiyonu olarak softmax fonksiyonunun kullanıldığı Dense katmanı eklenmiştir. Şekil 4.20'de, AI-TNKU-5 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 4 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_10 (Embedding)     (None, 2000, 400)         60000000  
_________________________________________________________________
lstm_10 (LSTM)               (None, 2000, 16)          26688     
_________________________________________________________________
time_distributed_10 (TimeDis (None, 2000, 250)         4250      
_________________________________________________________________
average_pooling1d_19 (Averag (None, 1000, 250)         0         
_________________________________________________________________
average_pooling1d_20 (Averag (None, 500, 250)          0         
_________________________________________________________________
flatten_10 (Flatten)         (None, 125000)            0         
_________________________________________________________________
dense_29 (Dense)             (None, 512)               64000512  
_________________________________________________________________
dropout_10 (Dropout)         (None, 512)               0         
_________________________________________________________________
dense_30 (Dense)             (None, 27)                13851     
=================================================================
Total params: 124,045,301
Trainable params: 124,045,301
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.20. AI-TNKU-5 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

6. AI-TNK-6 Veri Seti için LSTM Modeli

Yazar tanıma için oluşturulan AI-TNKU-6 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 200 nörondan oluşan TimeDistributedDense katmanı ve ardından 2 tane AveragePooling1D katmanları modele eklenmiştir. Modelin devamında Flatten katmanı, 256 nörondan ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı kullanılmıştır. Sonrasında 0.2 katsayısının belirlendiği Dropout katmanı ve 16 sınıftan oluşan bir veri seti kullanıldığı için 16 nörondan oluşan aktivasyon fonksiyonu olarak softmax fonksiyonunun kullanıldığı Dense katmanı eklenmiştir. Şekil 4.21'de, AI-TNKU-6 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 4 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 400)         40000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 1000, 16)          26688
_________________________________________________________________
time_distributed_10 (TimeDis (None, 1000, 200)         3400
_________________________________________________________________
average_pooling1d_19 (Averag (None, 500, 200)          0
_________________________________________________________________
average_pooling1d_20 (Averag (None, 250, 200)          0
_________________________________________________________________
flatten_10 (Flatten)         (None, 50000)             0
_________________________________________________________________
dense_29 (Dense)             (None, 256)               12800256
_________________________________________________________________
dropout_10 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_30 (Dense)             (None, 16)                4112
=================================================================
Total params: 52,834,456
Trainable params: 52,834,456
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.21. AI-TNKU-6 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

7. AI-TNKU-7 Veri Seti için LSTM Modeli

Yazar tanıma için oluşturulan AI-TNKU-7 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 12 nörondan oluşan LSTM katmanı, 200 nörondan oluşan TimeDistributedDense katmanı ve ardından AveragePooling1D katmanı modele eklenmiştir. Modelin devamında Flatten katmanı, 256 nörondan ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı kullanılmıştır. Sonrasında 0.2 katsayısının belirlendiği Dropout katmanı ve 9 sınıftan oluşan bir veri seti kullanıldığı için 9 nörondan oluşan aktivasyon fonksiyonu olarak softmax fonksiyonunun kullanıldığı Dense katmanı eklenmiştir. Şekil 4.22'de, AI-TNKU-7 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 3 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 400)         40000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 1000, 12)          19824
_________________________________________________________________
time_distributed_10 (TimeDis (None, 1000, 200)         2600
_________________________________________________________________
average_pooling1d_10 (Averag (None, 500, 200)          0
_________________________________________________________________
flatten_10 (Flatten)         (None, 100000)            0
_________________________________________________________________
dense_29 (Dense)             (None, 256)               25600256
_________________________________________________________________
dropout_10 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_30 (Dense)             (None, 9)                 2313
=================================================================
Total params: 65,624,993
Trainable params: 65,624,993
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.22. AI-TNKU-7 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

8. GI-TNKU-1 Veri Seti için LSTM Modeli

Tür tanıma için oluşturulan GI-TNKU-1 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 256 nörondan oluşan ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı, 0.2 katsayısının belirlendiği Dropout katmanı ve 200 nöronun kullanıldığı TimeDistributedDense katmanı modele eklenmiştir. Devamında AveragePooling1D ve Flatten katmanı eklenmiştir. Çıkış katmanı olan Dense katmanında ise 7 nöron ve aktivasyon fonksiyonu olarak softmax kullanılmıştır. Şekil 4.23'de, GI-TNKU-1 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 3 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 1000, 16)          20288
_________________________________________________________________
dense_28 (Dense)             (None, 1000, 256)         4352
_________________________________________________________________
dropout_10 (Dropout)         (None, 1000, 256)         0
_________________________________________________________________
time_distributed_10 (TimeDis (None, 1000, 200)         51400
_________________________________________________________________
average_pooling1d_10 (Averag (None, 500, 200)          0
_________________________________________________________________
flatten_10 (Flatten)         (None, 100000)            0
_________________________________________________________________
dense_30 (Dense)             (None, 7)                 700007
=================================================================
Total params: 30,776,047
Trainable params: 30,776,047
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.23**.** GI-TNKU-1 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

9. GI-TNKU-2 Veri Seti için LSTM Modeli

Tür tanıma için oluşturulan GI-TNKU-2 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 256 nörondan oluşan ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı, 0.2 katsayısının belirlendiği Dropout katmanı ve 200 nöronun kullanıldığı TimeDistributedDense katmanı modele eklenmiştir. Devamında AveragePooling1D katmanı, Flatten katmanı ve 0.2 katsayısının kullanıldığı Dropout katmanı eklenmiştir. Çıkış katmanı olan Dense katmanında ise 6 nöron ve aktivasyon fonksiyonu olarak softmax kullanılmıştır. Şekil 4.24'de, GI-TNKU-2 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 3 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 1000, 16)          20288
_________________________________________________________________
dense_28 (Dense)             (None, 1000, 256)         4352
_________________________________________________________________
dropout_19 (Dropout)         (None, 1000, 256)         0
_________________________________________________________________
time_distributed_10 (TimeDis (None, 1000, 200)         51400
_________________________________________________________________
average_pooling1d_10 (Averag (None, 500, 200)          0
_________________________________________________________________
flatten_10 (Flatten)         (None, 100000)            0
_________________________________________________________________
dropout_20 (Dropout)         (None, 100000)            0
_________________________________________________________________
dense_30 (Dense)             (None, 6)                 600006
=================================================================
Total params: 30,676,046
Trainable params: 30,676,046
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.24. GI-TNKU-2 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

10. GI-TNKU-3 Veri Seti için LSTM Modeli

Tür tanıma için oluşturulan GI-TNKU-3 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 512 nörondan oluşan ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı, 0.2 katsayısının belirlendiği Dropout katmanı ve 250 nöronun kullanıldığı TimeDistributedDense katmanı modele eklenmiştir. Devamında AveragePooling1D katmanı ve Flatten katmanı eklenmiştir. Çıkış katmanı olan Dense katmanında ise veri seti 5 sınıftan oluştuğu için 5 nöron ve aktivasyon fonksiyonu olarak softmax kullanılmıştır. Şekil 4.25'de, GI-TNKU-3 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 4 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 1000, 16)          20288
_________________________________________________________________
dense_28 (Dense)             (None, 1000, 512)         8704
_________________________________________________________________
dropout_10 (Dropout)         (None, 1000, 512)         0
_________________________________________________________________
time_distributed_10 (TimeDis (None, 1000, 250)         128250
_________________________________________________________________
average_pooling1d_10 (Averag (None, 500, 250)          0
_________________________________________________________________
flatten_10 (Flatten)         (None, 125000)            0
_________________________________________________________________
dense_30 (Dense)             (None, 5)                 625005
=================================================================
Total params: 30,782,247
Trainable params: 30,782,247
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.25. GI-TNKU-3 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

11. GI-TNKU-4 Veri Seti için LSTM Modeli

Tür tanıma için oluşturulan GI-TNKU-4 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 256 nörondan oluşan ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı, 0.2 katsayısının belirlendiği Dropout katmanı ve 250 nöronun kullanıldığı TimeDistributedDense katmanı modele eklenmiştir. Devamında AveragePooling1D katmanı, Flatten katmanı ve 0.1 katsayısının belirlendiği Dropout katmanı modele eklenmiştir. Çıkış katmanı olan Dense katmanında ise veri seti 4 sınıftan oluştuğu için 4 nöron ve aktivasyon fonksiyonu olarak softmax kullanılmıştır. Şekil 4.26'da, GI-TNKU-4 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 2 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 1000, 16)          20288
_________________________________________________________________
dense_28 (Dense)             (None, 1000, 256)         4352
_________________________________________________________________
dropout_19 (Dropout)         (None, 1000, 256)         0
_________________________________________________________________
time_distributed_10 (TimeDis (None, 1000, 250)         64250
_________________________________________________________________
average_pooling1d_10 (Averag (None, 500, 250)          0
_________________________________________________________________
flatten_10 (Flatten)         (None, 125000)            0
_________________________________________________________________
dropout_20 (Dropout)         (None, 125000)            0
_________________________________________________________________
dense_30 (Dense)             (None, 4)                 500004
=================================================================
Total params: 30,588,894
Trainable params: 30,588,894
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.26. GI-TNKU-4 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

12. GI-TNKU-5 Veri Seti için LSTM Modeli

Tür tanıma için oluşturulan GI-TNKU-5 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 256 nörondan oluşan ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı, 0.2 katsayısının belirlendiği Dropout katmanı ve 200 nöronun kullanıldığı TimeDistributedDense katmanı modele eklenmiştir. Devamında AveragePooling1D katmanı, Flatten katmanı ve 0.2 katsayısının belirlendiği Dropout katmanı modele eklenmiştir. Çıkış katmanı olan Dense katmanında ise veri seti 3 sınıftan oluştuğu için 3 nöron ve aktivasyon fonksiyonu olarak softmax kullanılmıştır. Şekil 4.27'de, GI-TNKU-5 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyon olarak categorical_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 3 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 1000, 16)          20288
_________________________________________________________________
dense_37 (Dense)             (None, 1000, 256)         4352
_________________________________________________________________
dropout_19 (Dropout)         (None, 1000, 256)         0
_________________________________________________________________
time_distributed_19 (TimeDis (None, 1000, 200)         51400
_________________________________________________________________
average_pooling1d_10 (Averag (None, 500, 200)          0
_________________________________________________________________
time_distributed_20 (TimeDis (None, 500, 200)          40200
_________________________________________________________________
flatten_10 (Flatten)         (None, 100000)            0
_________________________________________________________________
dropout_20 (Dropout)         (None, 100000)            0
_________________________________________________________________
dense_40 (Dense)             (None, 3)                 300003
=================================================================
Total params: 30,416,243
Trainable params: 30,416,243
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.27. GI-TNKU-5 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

13. GI-TNKU-6 Veri Seti için LSTM Modeli

Tür tanıma için oluşturulan GI-TNKU-6 adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 512 nörondan oluşan ve aktivasyon fonksiyonu olarak ReLu'nun kullanıldığı Dense katmanı ve 200 nöronun kullanıldığı TimeDistributedDense katmanı modele eklenmiştir. Devamında AveragePooling1D katmanı ve Flatten katmanı kullanılmıştır. Çıkış katmanı olan Dense katmanında ise veri seti 2 sınıftan oluştuğu için 1 nöron ve aktivasyon fonksiyonu olarak sigmoid fonksiyonu kullanılmıştır. Şekil 4.28'de, GI-TNKU-6 veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak binary_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 3 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000  
_________________________________________________________________
lstm_10 (LSTM)               (None, 1000, 16)          20288     
_________________________________________________________________
dense_28 (Dense)             (None, 1000, 512)         8704      
_________________________________________________________________
time_distributed_10 (TimeDis (None, 1000, 200)         102600    
_________________________________________________________________
average_pooling1d_10 (Averag (None, 500, 200)          0         
_________________________________________________________________
flatten_10 (Flatten)         (None, 100000)            0         
_________________________________________________________________
dense_30 (Dense)             (None, 1)                 100001    
=================================================================
Total params: 30,231,593
Trainable params: 30,231,593
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.28**.** GI-TNKU-6 Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

14. IAG-TNKU Veri Seti için LSTM Modeli

Cinsiyet tanıma için oluşturulan IAG-TNKU adlı veri setinin, modelleme sonucunda bulunan en iyi LSTM modelinde, Sequential ve Embedding katmanlarının ardından, sırasıyla 16 nörondan oluşan LSTM katmanı, 256 nörondan oluşan ve aktivasyon fonksiyonu olarak ReLu'nun tercih edildiği Dense katmanı, 0.2 katsayısının kullanıldığı Dropout katmanı modele eklenmiştir. Bu katmanların ardından 200 nörondan oluşan TimeDistributedDense katmanı, AveragePooling1D, Flatten ve 0.2 katsayısının kullanıldığı Dropout katmanı kullanılmıştır. Çıkış katmanı olan Dense katmanında 1 nöron ve aktivasyon fonksiyonu olarak sigmoid kullanılarak model tamamlanmıştır. Şekil 4.29'da, IAG-TNKU veri seti için bulunmuş olan en iyi LSTM modelinin özet mimarisi yer almaktadır. Bu modelde optimizer olarak adam, batch_size parametresi olarak 64, loss fonksiyonu olarak binary_crossentropy seçilmiş ve model 10 Fold Cross-Validation uygulanarak 2 epoch çalıştırılmıştır.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_10 (Embedding)     (None, 1000, 300)         30000000
_________________________________________________________________
lstm_10 (LSTM)               (None, 1000, 16)          20288
_________________________________________________________________
dense_28 (Dense)             (None, 1000, 256)         4352
_________________________________________________________________
dropout_19 (Dropout)         (None, 1000, 256)         0
_________________________________________________________________
time_distributed_10 (TimeDis (None, 1000, 200)         51400
_________________________________________________________________
average_pooling1d_10 (Averag (None, 500, 200)          0
_________________________________________________________________
flatten_10 (Flatten)         (None, 100000)            0
_________________________________________________________________
dropout_20 (Dropout)         (None, 100000)            0
_________________________________________________________________
dense_30 (Dense)             (None, 1)                 100001
=================================================================
Total params: 30,176,041
Trainable params: 30,176,041
Non-trainable params: 0
_________________________________________________________________
```

Şekil 4.29. IAG-TNKU Veri Seti için En İyi LSTM Modelinin Özet Mimarisi

4.2.3. Derin Öğrenme Modellerinin Sonuçları

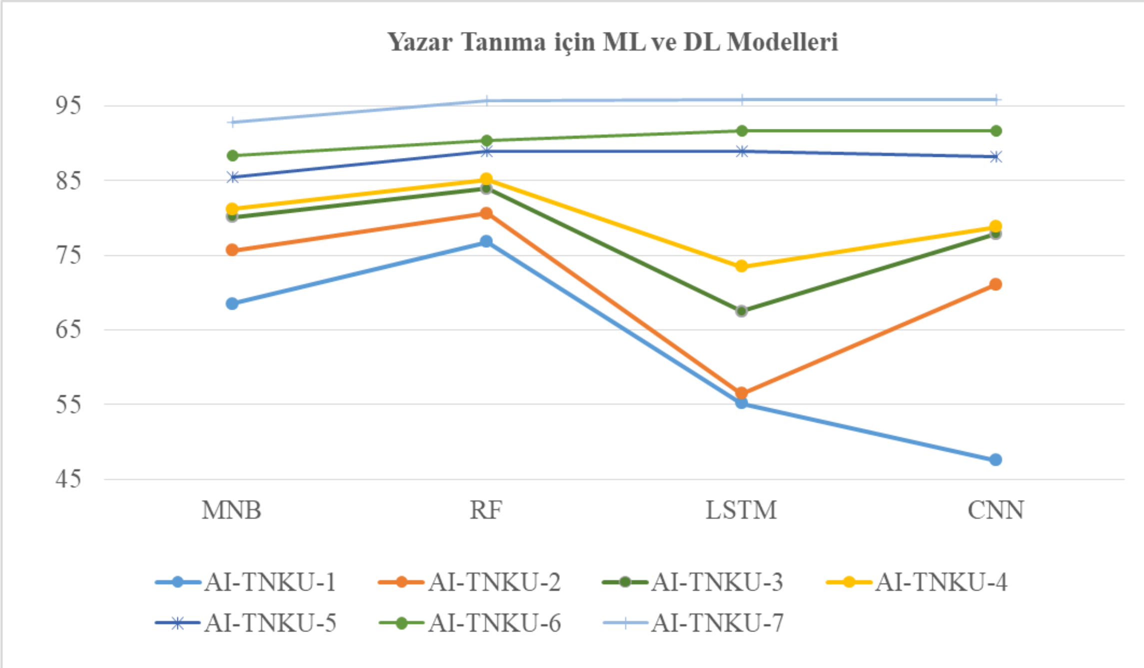
Yazar, tür ve cinsiyet tanıma işlemleri için, tüm veri setlerinin her birinin CNN ve LSTM modelleri sonuçlarına Çizelge 4.3'de yer verilmiştir. Makine öğrenmesi algoritmaları ile kıyaslandığında yüksek performans gösteren algoritmaların doğruluk, kesinlik ve duyarlılık değerleri koyu renk ile belirtilmiştir.

Çizelge 4.3. Tüm Veri Setleri için Derin Öğrenme Modellerinin Sonuçları

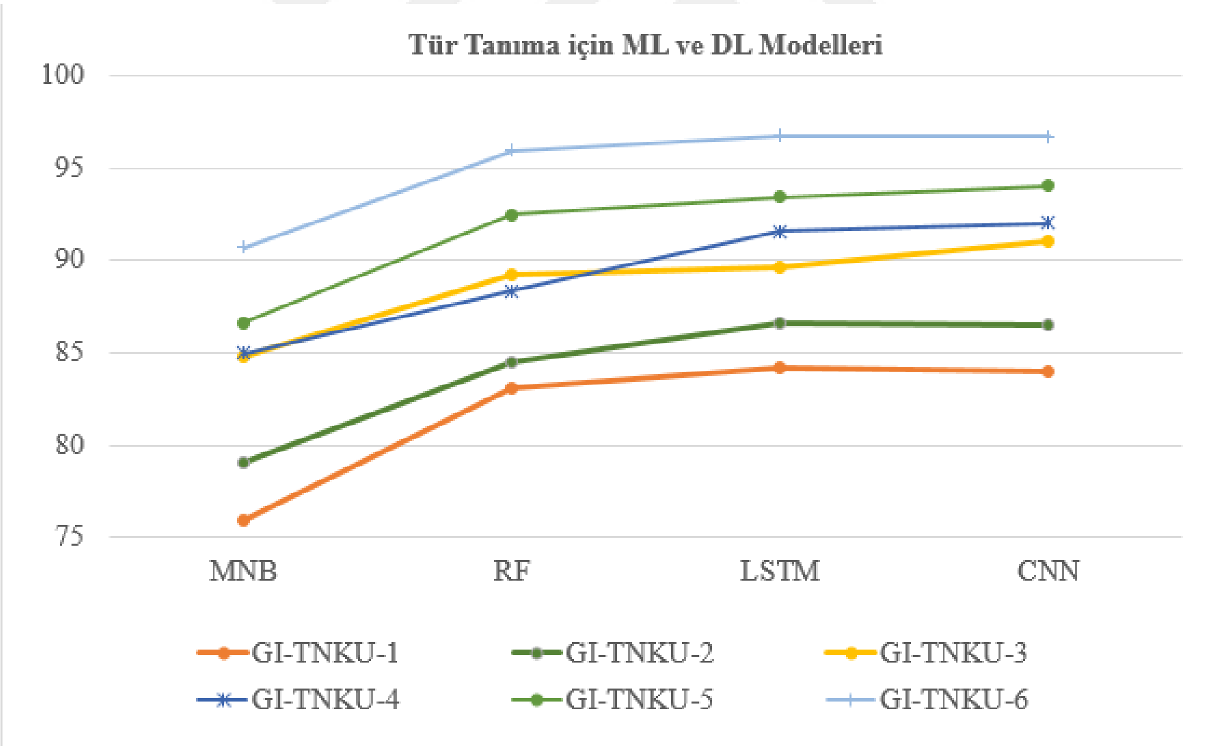
Metin Sınıflandırma	Veri Seti	Model	Doğruluk		Kesinlik		Duyarlılık	
			Eğitim	Test	Eğitim	Test	Eğitim	Test
Yazar Tanıma	AI-TNKU-1	CNN	99.51	47.48	99.77	52.50	99.77	45.38
		LSTM	97.90	55.16	99.02	61.39	99.02	53.51
	AI-TNKU-2	CNN	99.10	71.06	99.73	75.74	99.73	68.95
		LSTM	88.25	56.41	97.26	62.18	97.26	52.29
	AI-TNKU-3	CNN	99.68	77.85	99.89	80.58	99.89	76.62
		LSTM	92.86	67.46	98.00	72.46	98.00	64.85
	AI-TNKU-4	CNN	99.74	78.79	99.79	80.48	99.79	76.43
		LSTM	95.61	73.49	98.32	77.42	98.32	72.00
	AI-TNKU-5	CNN	99.64	88.18	99.94	90.17	99.94	87.26
		LSTM	98.90	**88.97**	99.64	**90.62**	99.64	**88.37**
	AI-TNKU-6	CNN	99.71	**91.73**	99.94	**92.71**	99.94	**91.29**
		LSTM	98.46	91.63	99.55	92.59	99.55	91.22
	AI-TNKU-7	CNN	99.51	**95.81**	99.73	96.22	99.73	**95.68**
		LSTM	98.88	95.81	99.38	**96.24**	99.38	95.66
Tür Tanıma	GI-TNKU-1	CNN	96.30	83.97	98.96	85.87	98.96	**83.00**
		LSTM	91.86	**84.15**	96.35	**86.81**	96.35	82.53
	GI-TNKU-2	CNN	95.45	86.54	98.08	88.42	98.08	**85.65**
		LSTM	93.52	**86.62**	96.54	**88.54**	96.54	85.53
	GI-TNKU-3	CNN	97.31	**90.98**	98.80	**91.96**	98.80	**90.53**
		LSTM	96.30	89.59	97.55	90.90	97.55	89.06
	GI-TNKU-4	CNN	94.97	**92.00**	98.17	**93.03**	98.17	**91.31**
		LSTM	94.87	91.57	97.34	92.67	97.34	90.97
	GI-TNKU-5	CNN	98.63	**94.01**	98.95	**94.11**	98.95	**93.62**
		LSTM	97.13	93.43	98.00	93.96	98.00	93.24
	GI-TNKU-6	CNN	98.89	96.69	99.46	96.77	99.46	96.69
		LSTM	98.78	**96.73**	99.16	**96.79**	99.16	**96.73**
Cinsiyet Tanıma	IAG-TNKU	CNN	90.99	88.43	94.59	88.60	94.59	88.43
		LSTM	90.29	**88.68**	93.08	**88.89**	93.08	**88.68**

4.3. Makine Öğrenmesi ve Derin Öğrenme Modellerinin Sonuçlarının Değerlendirilmesi

Makine öğrenmesi modellerinin sonuçlarını gösteren Çizelge 4.1'e ve derin öğrenme modellerinin sonuçlarını gösteren Çizelge 4.3'e göre, Şekil 4.30'da yazar tanıma için, Şekil 4.31'de de tür tanıma için elde edilen tüm modeller gösterilmiş ve başarıları karşılaştırılmıştır.



Şekil 4.30. Yazar Tanıma, Makine Öğrenmesi ve Derin Öğrenme Modellerinin Başarıları



Şekil 4.31. Tür Tanıma, Makine Öğrenmesi ve Derin Öğrenme Modellerinin Başarıları

Şekil 4.30 ve 4.31'deki yazar, tür ve cinsiyet tanıma modellerinin başarıları, tüm veri setleri için karşılaştırıldığında, Çizelge 4.4’de belirtildiği gibi, her bir veri seti için, en iyi modeller elde edilmiştir.

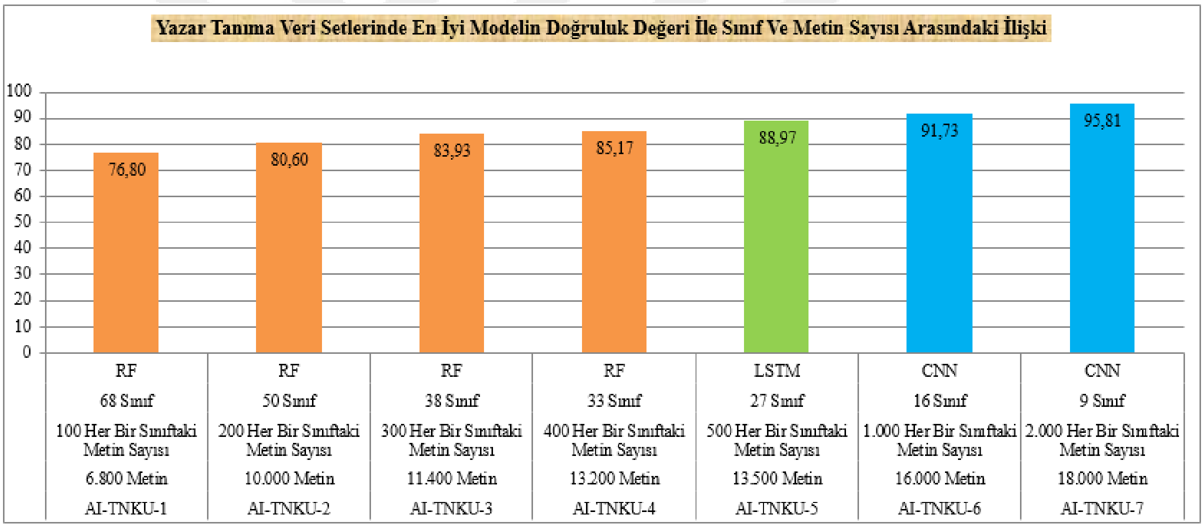
Çizelge 4.4. Tüm Veri Setleri için En İyi Model Sonuçları

Metin Sınıflandırma	Veri Seti	Sınıf	Sınıf Başına Metin	Toplam Metin	Model	Doğruluk		Kesinlik		Duyarlılık	
						Eğitim	Test	Eğitim	Test	Eğitim	Test
Yazar Tanıma	AI-TNKU-1	68	100	6.800	RF	100.0	76.80	100.0	76.74	100.0	76.80
	AI-TNKU-2	50	200	10.000	RF	100.0	80.60	100.0	80.62	100.0	80.60
	AI-TNKU-3	38	300	11.400	RF	100.0	83.93	100.0	84.39	100.0	83.93
	AI-TNKU-4	33	400	13.200	RF	100.0	85.17	100.0	85.47	100.0	85.17
	AI-TNKU-5	27	500	13.500	LSTM	98.90	88.97	99.64	90.62	99.64	88.37
	AI-TNKU-6	16	1.000	16.000	CNN	99.71	91.73	99.94	92.71	99.94	91.29
	AI-TNKU-7	9	2.000	18.000	CNN	99.51	95.81	99.73	96.22	99.73	95.68
Tür Tanıma	GI-TNKU-1	7	3.188	22.316	LSTM	91.86	84.15	96.35	86.81	96.35	82.53
	GI-TNKU-2	6	3.343	20.058	LSTM	93.52	86.62	96.54	88.54	96.54	85.53
	GI-TNKU-3	5	3.848	19.240	CNN	97.31	90.98	98.80	91.96	98.80	90.53
	GI-TNKU-4	4	4.064	16.256	CNN	94.97	92.00	98.17	93.03	98.17	91.31
	GI-TNKU-5	3	4.760	14.280	CNN	98.63	94.01	98.95	94.11	98.95	93.62
	GI-TNKU-6	2	5.831	11.662	LSTM	98.78	96.73	99.16	96.79	99.16	96.73
Cinsiyet Tanıma	IAG-TNKU	2	21.646	43.292	LSTM	90.29	88.68	93.08	88.89	93.08	88.68

Çizelge 4.4’e göre, yazar tanıma için, AI-TNKU-1, 2, 3 ve 4 isimli veri setleri için RF algoritması kullanılarak oluşturulan modellerin, diğer modellere göre daha yüksek doğrulukta başarı oranı elde ettiği anlaşılmaktadır. AI-TNKU-1, 2, 3 ve 4 isimli veri setleri için RF algoritması kullanılarak oluşturulan modellerin doğruluk başarıları, sırasıyla, % 76,80, % 80,60, % 83,93 ve %85,17 olarak elde edilmiştir. AI-TNKU-5 veri seti için, LSTM algoritmasının kullanıldığı model ile % 88,97 doğrulukla en yüksek başarı, AI-TNKU-6 ve

AI-TNKU-7 veri setleri içinse en iyi modellerin CNN algoritmasının kullanılmasıyla, sırasıyla, % 91,73 ve % 95,81 doğruluk başarıları değerleri bulunmuştur.

Yazar tanıma için veri setlerindeki, sınıf sayıları ve her bir sınıfta yer alan metin sayılarına bakıldığında, Şekil 4.32'de de gösterildiği gibi, 68 (AI-TNKU-1), 50 (AI-TNKU-2), 38 (AI-TNKU-3) ve 33 (AI-TNKU-4) adet sınıf içeren veri setleri için, RF algoritmasının kullanıldığı modellerin performansının, diğer modellere göre daha başarılı olduğu görülmüştür. Bu veri setleri, sırasıyla, toplam 6.800, 10.000, 11.400 ve 13.200 adet metin içermektedir. 27 (AI-TNKU-5), 16 (AI-TNKU-6) ve 9 (AI-TNKU-7) sınıflı veri setlerinde ise derin öğrenme algoritmalarının (CNN ve LSTM) kullanıldığı modellerin, makine öğrenmesi modellerine (MNB ve RF) göre daha başarılı olduğu görülmüştür. Bu veri setlerinde ise, sırasıyla, toplam 16.000 ve 18.000 adet metin bulunmaktadır. Sınıf sayısının azalması ve metin sayısının artmasıyla, derin öğrenme modellerinin performanslarının arttığı görülmüştür.

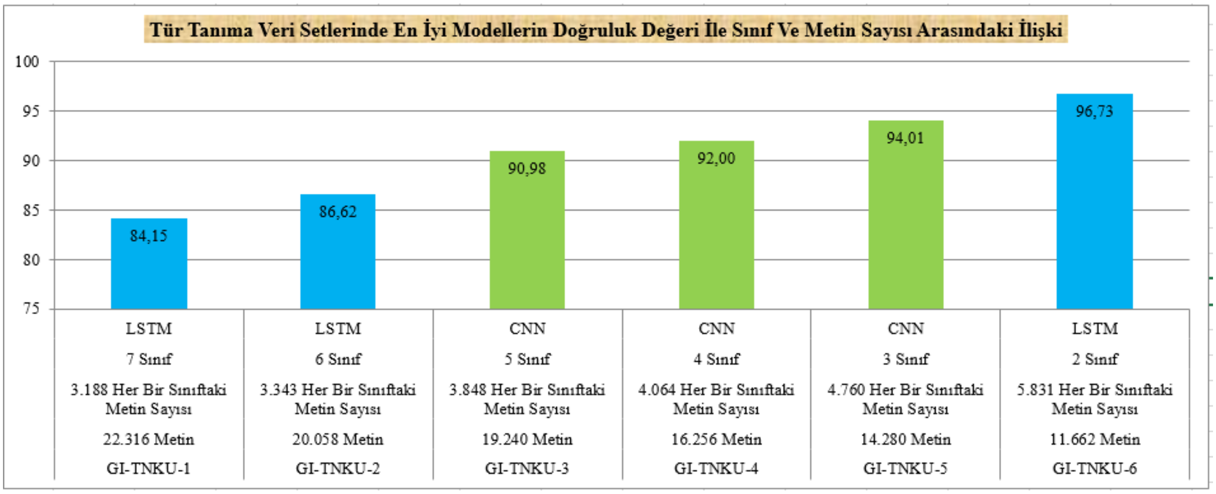


Şekil 4.32. Yazar Tanıma için Sınıf ve Metin Sayılarına Göre En İyi Model Başarıları

Yazar tanıma için, çok sınıflı, daha az sayıda metin içeren veri setleri için makine öğrenmesi modellerinin, az sınıflı daha çok sayıda metin içeren veri setleri için de derin öğrenme modellerinin, daha uygun olacağı sonucuna ulaşılmıştır.

Çizelge 4.4'e göre, tür tanıma için bulunan en iyi modellerin, derin öğrenme algoritmalarının kullanıldığı modeller olduğu görülmüştür. Bu çizelgede de görüldüğü gibi, GI-TNKU-1, 2 ve 6 adlı veri setlerinin en iyi modelleri LSTM algoritması kullanılarak oluşturulan modellerdir ve bu modellerin doğruluk başarıları, sırasıyla, %84,15, %86,62, ve %96,73 doğruluk olarak elde edilmiştir. GI-TNKU-3, 4 ve 5 veri setleri içinse CNN algoritmasını kullanan modellerin daha başarılı olduğu görülmüştür ve alınan doğruluk başarıları, sırasıyla, %90,98, %92,00 ve %94,01 olarak bulunmuştur. Derin öğrenme modellerinin başarılarının, makine öğrenmesi modellerinin başarılarına kıyasla daha yüksek olması, her bir sınıftaki metin sayılarının daha fazla olması ve sınıf sayılarının daha az olmasından dolayı olduğu söylenebilir.

Tür tanıma için veri setlerindeki, sınıf sayıları ve her bir sınıfta yer alan metin sayılarına bakıldığında, Şekil 4.33'de de gösterildiği gibi, GI-TNKU-1 veri setinden GI-TNKU-6 veri setine doğru sınıf sayısının azaldığı, her bir sınıftaki metin sayısının arttığı ve aynı zamanda da model başarılarının da arttığı görülmüştür.



Şekil 4.33. Tür Tanıma için Sınıf ve Metin Sayılarına Göre En İyi Model Başarıları

Sonuç olarak, sınıf sayısının azalması ve sınıf başına metin sayısının artmasıyla derin öğrenme modellerinin performansının arttığı görülmüştür. Makine öğrenmesi modellerinin birden çok sınıflı ve daha az metin içeren veri kümeleri için daha uygun olacağı sonucuna varılmıştır. Bununla birlikte, derin öğrenme modelleri, daha az sınıf içeren daha fazla metin içeren veri kümeleri için yüksek performans ile daha kullanışlı olacaktır.

Çizelge 4.4'e göre cinsiyet tanıma için bulunan en iyi modelin, LSTM algoritmasının kullanıldığı model olduğu görülmüştür ve bu modelin başarısı %88,68 doğruluk olarak bulunmuştur.

5. SONUÇLAR

Bu çalışmada, veri olarak Türkçe haber metinlerinin seçildiği ve bu verilerin yazar, tür ve cinsiyete göre sınıflandırılabilmelerini sağlayan ve öğrenme algoritmalarının sınıflandırıcı olarak kullanıldığı bir modelleme çalışması yapılmıştır.

Çalışmanın ilk aşamasında, bir gazetenin köşe yazarlarına ait köşe yazılarını içeren, yazar tanıma, tür tanıma ve cinsiyet tanıma işlemlerinde kullanılabilecek, büyük ölçekli ve çoklu sınıflara sahip, toplam 14 adet yeni veri seti oluşturulmuştur. Yazar tanıma için 7, tür tanıma için 6 ve cinsiyet tanıma için de 1 adet olan bu veri setleri, Türkçe diline özel doğal dil işleme adımlarından geçirilerek, tanıma işlemlerinin yapılacağı sınıflandırıcıların uygulandığı ve en yüksek doğruluk başarılarının araştırıldığı, modelleme aşaması için hazır hale getirilmiştir.

Modelleme aşamasında, Türkçe metinlerde yazar tanıma, tür tanıma ve cinsiyet tanıma problemlerinin çözümüne yönelik makine öğrenmesi algoritmalarında Multinominal Naive Bayes (MNB) ve Random Forest (RF) algoritmaları ve derin öğrenme algoritmalarından da Convolutional Neural Networks (CNN) ve Long Short Term Memory (LSTM) algoritmaları, sınıflandırıcı olarak veri setlerine uygulanmış ve bu sınıflandırıcılardan en yüksek performansın alındığı hiper parametre değerleri, uzun deneysel çalışmalar sonucunda bulunmaya çalışılmıştır. Modelleme sonucunda, her bir veri seti ve sınıflandırıcı için en iyi modeller, yani en iyi doğruluk, kesinlik ve duyarlılık değerlerine sahip modeller elde edilmiştir.

Yazar tanıma için elde edilen en iyi modellerin sonuçlarına bakıldığında, kullanılan 7 farklı veri setinin sınıf sayısı, her bir sınıfta bulunan veri sayısı ve toplam veri sayısı gibi birbirinden farklı özelliklerinin bulunması bu çalışma kapsamında bu özelliklerin etkilerini incelememizi ve karşılaştırmamızı sağlamıştır. AI-TNKU-1, 2, 3 ve 4 adlı veri setleri için, en iyi model olan RF algoritmasının kullanıldığı modelle, sırasıyla, %76,80, %80,60, %83,93 ve % 85,17 doğruluk başarı oranları alınmıştır. AI-TNKU-5 veri seti için, LSTM algoritmasının kullanıldığı model ile en yüksek %88,97 doğruluk başarısı, AI-TNKU-6 ve 7 veri setleri içinse, CNN algoritmasının kullanıldığı modelle, sırasıyla, %91,73 ve %95,81 doğruluk başarı oranları elde edilmiştir. Bu veri setlerinin sınıf sayısı ve her bir sınıfta bulunan veri sayısı dikkate alındığında, sınıf sayısı arttıkça ve her bir sınıfta bulunan veri sayısı azaldıkça makine öğrenmesi modellerin sonuçlarının, derin öğrenme modellerine kıyasla daha yüksek olduğu görülmüştür. Bununla birlikte, sınıf sayısının en az olduğu ve her bir sınıftaki metin sayısının arttığı veri setlerinde de, derin öğrenme modellerinin daha başarılı olduğu tespit edilmiştir.

Tür tanıma için, elde edilen en iyi modellerin başarıları incelendiğinde, derin öğrenme modellerinin performanslarının, makine öğrenmesi modellerine göre daha yüksek olduğu tespit edilmiştir. LSTM algoritmasının kullanıldığı en iyi modeller ile GI-TNKU-1, 2 ve 6 veri setleri için, sırasıyla, %84,15, %86,62 ve %96,73 doğruluk başarı oranlarına ulaşılmıştır. CNN algoritmasının kullanıldığı en iyi modeller ile de GI-TNKU-3, 4 ve 5 veri setleri için, %90,98, %92,00 ve %94,01 doğruluk başarısı oranları elde edilmiştir. Tür tanıma için veri setlerindeki, sınıf sayıları ve her bir sınıfta yer alan metin sayılarına bakıldığında, GI-TNKU-1 veri setinden GI-TNKU-6 veri setine doğru sınıf sayısının azaldığı, her bir sınıftaki metin sayısının arttığı ve aynı zamanda da model başarılarının da arttığı görülmüştür. Derin öğrenme modellerinin başarılarının, makine öğrenmesi modellerinin başarılarına kıyasla daha yüksek olması, her bir sınıftaki metin sayılarının daha fazla olması ve sınıf sayılarının daha az olmasından dolayı olduğu söylenebilir.

Cinsiyet tanıma için, yine elde edilen en iyi modellerin başarıları incelendiğinde, LSTM algoritmasının sınıflandırma test doğruluk başarısının diğer algoritmalara kıyasla %88,68 doğruluk ile daha yüksek olduğu görülmüştür. CNN algoritmasının sınıflandırma başarısının %88,43 doğruluk ile LSTM algoritmasına çok yakın olduğu belirlenmiş ve bu sonuç CNN algoritmasının da doğal dil işleme problemlerinde başarılı bir şeklide kullanılabileceğini ortaya koymuştur.

Bu tez çalışması sonucunda, veri setindeki sınıf sayısı arttıkça ve her bir sınıftaki veri miktarı azaldıkça, derin öğrenme modellerindeki doğruluk başarısının azaldığı ve makine öğrenmesi modellerinin sınıflandırma başarılarının da derin öğrenme modellerinin başarılarını geçtiği gözlemlenmiştir. Veri setindeki sınıf sayısının azaldığı ve her bir sınıftaki veri miktarının arttırıldığı durumda ise derin öğrenme modellerinin sınıflandırma başarısının makine öğrenmesi modellerine göre daha yüksek olduğu görülmüştür. Elde ettiğimiz sonuçlar ışığında, derin öğrenme için veri miktarının çokluğunun hem modelin öğrenme başarısını hem de test başarısını arttırdığı ve bu sayede makine öğrenmesi modellerinden, daha başarılı bir şekilde sınıflandırma işlemini gerçekleştiğini söyleyebiliriz. CNN algoritmasının sınıflandırma başarısının LSTM algoritmasına çok yakın olması hatta bazı durumlarda daha yüksek doğruluk değerlerine ulaşmış olması, bu algoritmanın doğal dil işleme problemlerini çözmek için kullanılabileceğini göstermiştir.

KAYNAKLAR

Abdallah, E., Alzghoul, J., & Alzghool, M. (2020). Age and Gender prediction in Open Domain Text. *Procedia Computer Science*, 563-570.

Akın, A. A., & Akın, M. D. (2007). Zemberek, an open source NLP framework for Turkic Languages. 1-5.

Al-Salemi, B., Ayob, M., Kendall, G., & Noah, S. (2019). Multi-label Arabic Text Categarization: A Bencmark And Baseline Comparison of Multi-label Learning Algorithm. *Information Processing & Management*, *56*(1), 212-227.

Alsmearat, K., Al-Ayyoub, M., Al-Shalabi, R., & Kanaan, G. (2017). Author gender identification from Arabic text. *Journal of Information Security and Application*, 85-95.

Amasyalı, M., & Diri, B. (2006). Automatic Turkish Text Categorization in Terms of Author, Genre and Gender. *Springer*, 221-226.

Aytekin, Ç., Sütçü, C., & Özfidan, U. (2018). Karar Ağacı Algoritması ile Metin Sınıflandırma. *Journal of International Social Research*.

Bisong, E. (2019). Google Colaboratory. *In Buildng Machine Learning and Deep Learning Models on Google Cloud Platform* (s. 59-64). içinde Springer.

Breiman, L. (2001). *Random Forests.* Machine Learning.

Canbek, G., Sagiroglu, S., Temizel, T., & Baykal, N. (2017). Binary classification performance measures/metrics: A comprehensive visualized roadmap to gain new insights. *International Conference on Computer Science and Engineering (UBMK)* .

Cheng, N., Chandramouli , R., & Subbalakshmi, K. (2011). Author gender identification from text. *Digital Investigation*, *1*(8), 78-88.

Chollet, F. (2019). *Python ile Derin Öğrenme.* Ankara: Buzdağı Yayınevi.

Çevik, F., & Kilimci, Z. (2019). Derin öğrenme yöntemleri ve kelime yerleştirme modelleri kullanılarak Parkinson hastalığının duygu analiziyle değerlendirilmesi. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*.

Elnagar, A., Al-Debsi, R., & Einea, O. (2020). Arabic text classification using deep learning models. *Information Processing & Management*.

Goenawan, R., Chanrico, W., Suhartono, D., & Purnomo, F. (2019). Gender Demography Classification on Instagram based on User's Comments Section. *Procedia Computer Science*, 64-71.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks* , 602-610.

Hossin, Mohammad, & Sulaiman, M. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process, 1*.

Hunter, J. (2007). Matplotlib: A 2D graphics enviroment. *Computing in Science & Engineering* , 90-95.

Hussein, S., Farouk, M., & Hemayed, E. (2019). Gender identification of egyptian dialect in twitter. *Egpyptian Informatic Journal*, 109-116.

İnan Acı, Ç., & Çırak, A. (2019). Türkçe Haber Metinlerinin Konvolüsyonel Sinir Ağları ve Word2Vec Kullanılarak Sınıflandırılması. *Bilişim Teknolojileri Dergisi*.

Kaban, Z., & Diri, B. (2008). Genre and author detection in Turkish texts using artificial immune recognition systems. *16. Signal Processing and Communications Applications Conference (SIU).* Aydın.

Kalaycı, T. E. (2018). Kimlik hırsızı web sitelerinin sınıflandırılması için makine öğrenmesi yöntemlerinin karşılaştırılması. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, 870-878.

Kowsari, K., Meimandi, J., Heidarysafa, K., Mendu, M., Barnes , S., & Brown, D. (2019). Text Classification Algorithms: A Survey. *Information* .

Lee, Y.-B., & Myaeng, S. (2004). Automatic Identification of Text Genres and Their Roles in Subject-Based Categorization. 37th Hawaii International Conference on System Sciences.

Levent, V., & Diri, B. (2014). Türkçe Dokümanlarda Yapay SinirAğları İle Yazar Tanıma. *Akademik Bilişim*, (s. 5-7). Mersin.

Loper, E., & Bird, S. (2002). NLTK: The Natural Language Toolkit. *arXiv:cs/0205028* .

Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam Filtering with Naive Bayes – Which Naive Bayes? . *CEAS*, 28-69.

Nergiz, G., Safali, Y., Avaroğlu, E., & Erdoğan , S. (2019). Classification of Turkish News Content by Deep Learning Based LSTM Using Fasttext Model. *International Artificial Intelligence and Data Processing Symposium (IDAP)*, (s. 1-6).

Nizam, H., & Akın, S. (2014). Sosyal medyada makine öğrenmesi ile duygu analizinde dengeli ve dengesiz veri setlerinin performanslarının karşılaştırılması. İzmir: XIX. Türkiye'de İnternet Konferansı .

Oliphant, T. (2006). *A Guide to NumPy.* Trelgol Publishing.

Picard, R., & Cook, R. (tarih yok). Cross-validation of regression models. *ournal of the American Statistical Association, 79*(387), 575-583.

Ritesh, C. (2018). Word Representations For Gender Classification Using Deep Learning. *Procedia Computer Science*, 614-622.

S, D., & Diri, B. (2010). Türkçe Dökümanlar için N-gram Tabanlı Yeni Bir Sınıflandırma (Ng-ind): Yazar, Tür ve Cinsiyet. *Journal of Turkey Informatics Foundation of Computer Science and Engineering, 3*(1), 11-19.

Sboev , A., Moloshnikov, I., Gudovskikh, D., Selivanov, A., Rybka , R., & Litvanova, T. (2018). Deep Learning neural nets versus traditional machine learning in gender identification of authors of RusProfiling texts. *Procedia Computer Science, 1*(123), 424-431.

Sboev, A., Litvinova, T., Gudovskikh, D., Rybka , R., & Moloshnikov, I. (2016). Machine Learning Models of Text Categorization by Author Gender Using Topic-independent Features. *Procedia Computer Science, 101*(1), 135-142.

Sboev, A., Moloshnikov, I., Gudovskikh, D., Selivanov, A., Rybka , A., & Litvinova, T. (2018). Automatic gender identification of author of Russian text by machine learning and neural net algorithms in case of gender deception. *Procedia Computer Science, 1*(123), 417-423.

Solar-Company, J., & Wanner, L. (2018). On the Role of Syntactic Dependencies and Discourse Relations for Author and Gender Identification. *Pattern Recognition Letters*, 87-95.

Soucy, P., & Mineau, G. W. (2005). Beyond TFIDF Weighting for Text Categorization in the Vector Space Model . *IJCAI*, (s. 1130-1135).

Stamatatos, E. (2008). Author identification: Using text sampling to handle the class imbalance problem. *Information Processing & Management, 44*(2), 790-799.

Sun , Q., Jankovic, M., Bally, L., & Mougiakakou, S. (2018). Predicting Blood Glucose With an LSTM and Bi-LSTM Based Deep Neural Network. *14th Symposium on Neural Networks and Applications (NEUREL).* Belgrade, Serbia.

Şahin, D., Kural, O., Kılıç, E., & Karabina , A. (2018). A Text Classification Application: Poet Detection from Poetry.

Şeker, A., Diri, B., & Balık, H. (2017). Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Mühendislik Bilimleri Dergisi* , 47-64.

Tai, K., Socher, R., & Manning, C. (2015). Improved semantic representations from tree-structured long short-term memory networks.

Tantuğ, C. (2014). Metin Sınıflandırma. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*.

Tanyıldızı, E., & Demirtaş, F. (2019). Hiper Parametre Optimizasyonu. (s. 1-5). 1st International Informatics and Software Engineering Conference (UBMYK).

Tüfekci, P., & Uzun , E. (2013). Author detection by using different term weighting schemes. (s. 1-4). Haspolat: 21st Signal Processing and Communications Applications Conference (SIU).

Tüfekci, P., Uzun , E., & Sevinç, B. (2012). Text classification of web based news articles by using Turkish grammatical features. *20th Signal Processing and Communications Applications Conference (SIU).* Muğla.

Türkçe Etkisiz Kelimeler (Stop Words) Listesi 1.1. (2009, Ocak 1). Türkçe Öğretimi: http://www.turkceogretimi.com/genel-konular/t%C3%BCrk%C3%A7e-etkisiz-kelimeler-stop-words-listesi-11 adresinden alındı

Uzun , E. (2020). A Novel Web Scraping Approach Using the Additional Information Obtained From Web Pages. *IEEE Access*, 61726-61740.

VanderPlas, J. (2016). *Python Data Science Handbook Essential Tools for Working with Data.* O'Reilly Media.

Vijayakumar, B., Marvan, M., & Fuad, M. (2019). A New Method to Identify Short-Text Authors Using Combinations of Machine Learning and Natural Language Processing Techniques. *Procedia Computer Science, 159*, 428-436.

Wongso, R., Luwinda, F. A., Trisnajaya, B. C., & Rudy, O. R. (2017). News Article Text Classification in Endonesian Language. *Procedia Computer Science, 116*, 137-143.

Yasdi, M., & Diri, B. (2012). Author recognition by Abstract Feature Extraction. *0th Signal Processing and Communications Applications Conference (SIU).* Muğla.

Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, (s. 649-657).

ÖZGEÇMİŞ

Melike Bektaş, 1995 yılında Tekirdağ'da doğdu. İlkokul ve lise öğrenimini 2013 yılında tamamladıktan sonra Düzce Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümüne başladı. Bu bölümden 2018 yılında mezun oldu. 2018 yılında Tekirdağ Namık Kemal Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında yüksek lisans eğitimine başladı. 2019 yılının Mayıs ayından itibaren İstanbul Rumeli Üniversitesi Mühendislik ve Mimarlık Fakültesi Bilgisayar Mühendisliği bölümünde araştırma görevlisi olarak çalışmaya devam etmektedir.