

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

MAHKEME
KARARLARININ DERİN
ÖĞRENME İLE TAHMİNİ

Berkay Bakışoğlu

Danışman

Dr. Burcu Yılmaz

Ocak, 2021

Gebze, KOCAELİ

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ

Bilgisayar Mühendisliği Bölümü

MAHKEME
KARARLARININ DERİN
ÖĞRENME İLE TAHMİNİ

Berkay Bakışoğlu

Danışman

Dr. Burcu Yılmaz

Ocak, 2021

Gebze, KOCAELİ

Bu çalışma 20/01/2021 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Bölümü'nde Lisans Bitirme Projesi olarak kabul edilmiştir.

Bitirme Projesi Jürisi

Danışman Adı	Burcu Yılmaz	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

Jüri Adı	Mehmet Göktürk	
Üniversite	Gebze Teknik Üniversitesi	
Fakülte	Mühendislik Fakültesi	

ÖNSÖZ

Bu raporun ilk taslaklarının hazırlanmasında emeği geçenlere, raporun son halini almasında yol gösterici olan sayın Dr. Burcu Yılmaz hocama ve bu çalışmayı destekleyen Gebze Teknik Üniversitesi'ne içten teşekkürlerimi sunarım.

Ayrıca eğitimim süresince bana her konuda tam destek veren aileme ve bana hayatlarıyla örnek olan tüm hocalarıma saygı ve sevgilerimi sunarım.

Ocak,2021

Berkay Bakışoğlu

İÇİNDEKİLER

ÖNSÖZ.....	iv
İÇİNDEKİLER	v
ŞEKİL LİSTESİ.....	vi
TABLO LİSTESİ.....	vii
KISALTMA LİSTESİ	viii
ÖZET	ix
SUMMARY	x
1.GİRİŞ	1
2.YÖNTEM.....	2
3.DENEYLER VE BULGULAR	5
1.MADDE EĞİTİMİ YAPILAN MODEL	5
2)BÜTÜN OLARAK EĞİTİM YAPILAN MODEL	8
4.SONUÇ	11
KAYNAKLAR.....	14

ŞEKİL LİSTESİ

ŞEKİL 1.1 Makine Öğrenmesi ve Derin Öğrenme Karşılaştırması	1
ŞEKİL 1.2 Klasik NLP, Derin Öğrenme içeren NLP farkı	2
ŞEKİL 2.1 RNN Sinir Ağlarının Yapısı	3
ŞEKİL 2.2 LSTM Sinir Ağlarının Yapısı.....	3
ŞEKİL 3.1 Madde Yöntemi İçin Başlangıç Testi Modeli Katman Yapısı.....	5
ŞEKİL 3.2 Madde Yöntemi İçin Başlangıç Kayıp ve İsabet Sonuçları.....	5
ŞEKİL 3.3 Seyreltme Katmanlı İçeren Madde Model.....	6
ŞEKİL 3.4 Seyreltme Katmanlı Model İçin İsabet ve Kayıp Sonuçları	6
ŞEKİL 3.5 Öğrenme Değeri Üstel Olarak Düşen Modelin İsabet ve Kayıp Sonuçları	7
ŞEKİL 3.6 Daha Uzun Veri Kesitleri Alan Modelin İsabet ve Kayıp Sonuçları	7
ŞEKİL 3.7 LSTM Birim Sayısı Azaltılmış Model İçin İsabet ve Kayıp Sonuçları....	8
ŞEKİL 3.8 Bütün Olarak Eğitim Yöntemi İçin Başlangıç Testi Modeli Birim Yapısı	8
ŞEKİL 3.9 Bütün Olarak Eğitim Modeli İçin Başlangıç Kayıp ve İsabet Grafiği	9
ŞEKİL 3.10 Seyreltme Katmanlı Eklenmiş Bütün Olarak Eğitim Modeli Kayıp ve İsabet Grafiği	9
ŞEKİL 3.11 Üstel Azalan Öğrenme Oranlı Bütün Olarak Eğitim Modeli Kayıp ve İsabet Grafiği	9
ŞEKİL 3.12 LSTM Birim Sayısı Arttırılmış Bütün Olarak Eğitim Modeli Kayıp ve İsabet Grafiği	10
ŞEKİL 4.1 İlk Yöntem için Model Yapısı	11
ŞEKİL 4.2 İkinci Yöntem İçin Model Yapısı	11
ŞEKİL 4.3 Madde Eğitim Modeli Kayıp ve İsabet Grafiği	12
ŞEKİL 4.4 Bütün Olarak Eğitim Modeli Kayıp ve İsabet Grafiği	12
ŞEKİL 4.5 Başarı Hesaplamasında Kullanılacak Denklemler	13

TABLO LİSTESİ

TABLO 4.1 Modeller İçin Test Sonuçları	13
--	----

KISALTMA LİSTESİ

NLTK	: Natural Language ToolKit(Doğal Dil İşleme Kiti)
JSON	: JavaScript Nesnesi Gösterimi (JavaScript Object Notation)
RNN	: Tekrarlayan Sinir ağları (Recurrent Neural Network)
LSTM	: Uzun Kısa Süreli Bellek(Long Term Short Memory)
TP	: Gerçek Pozitifler(True Positives)
FP	: Yanlış Pozitifler (False Positives)
TN	: Gerçek Negatifler(True Negatives)
FN	: Yanlış Negatifler(False Negatives)

ÖZET

Bu projenin amacı,dava metinlerini kullanarak,dava sonucunu tahmin etmektir.

Okunan dava dosyaları, derin öğrenme modeline uygun hale getirilmek için ön işlemeden geçmiştir. Bu ön işlemede dava içerisindeki her maddenin, büyük-küçük harf ayrımı ortadan kaldırılmış, eş anlamlı kelimeler tek bir kelimeye dönüştürülmüş, tek başına anlam ifade etmeyen kelimeler çıkartılmış ve oluşan liste, bir dava dosyasının tamamını oluşturuyor gibi kabul edilerek, ilk yöntem olan madde sonuç ilişkisiyle eğitim için oluşturulan listeye eklenmiştir. Aynı ön işlemler kullanılarak işlenen bütün halindeki dava metinleri ise, ikinci yöntem olan bütün olarak işleme modeli için oluşturulan listeye eklenmiştir.7100 dava dosyasından elde edilen veriler eğitim için kullanılırken,1380 dava dosyası doğrulama testi için,2998 dava dosyası ise test için kullanılmıştır.

Model eğitimleri için en uygun model türü ve hiper parametreler araştırılmış ve test edilmiştir. Araştırmalar sonucunda model bir ardışık model olarak, “Embedding” ve LSTM katmanı kullanılarak oluşturulmuştur. Her iki model içinde üstel azalan öğrenme oranları kullanılmış, daha uzun veri girişinin yapıldığı bütün olarak eğitim modeli için ise seyreltme katmanı uygulanmıştır.

Test aşamasında bu eğitilen modeller ve test veri seti kullanılmış, modellerden gelen tahminler sonuçları kullanılarak, isabet, kesinlik, doğruluk ve F1 puanı, hesaplanmıştır. Bu puanlar literatürdeki çalışma[13] ile karşılaştırılmıştır.

SUMMARY

The aim of this project is to predict the result of the judgement using the case texts.

The case files were read, pre-processed to make them suitable for the deep learning model. In this preprocessing, for every fact in case, words converted into lower case. words with a synonym converted into one word, stop words are removed, and this fact is considered as one complete case. It added to the list made for training with the first method. In second method, the whole case texts with the same preliminary proceedings are used for constructing training list. 7100 case files, 1380 case files are used for verification testing, and 2998 case files are used for testing.

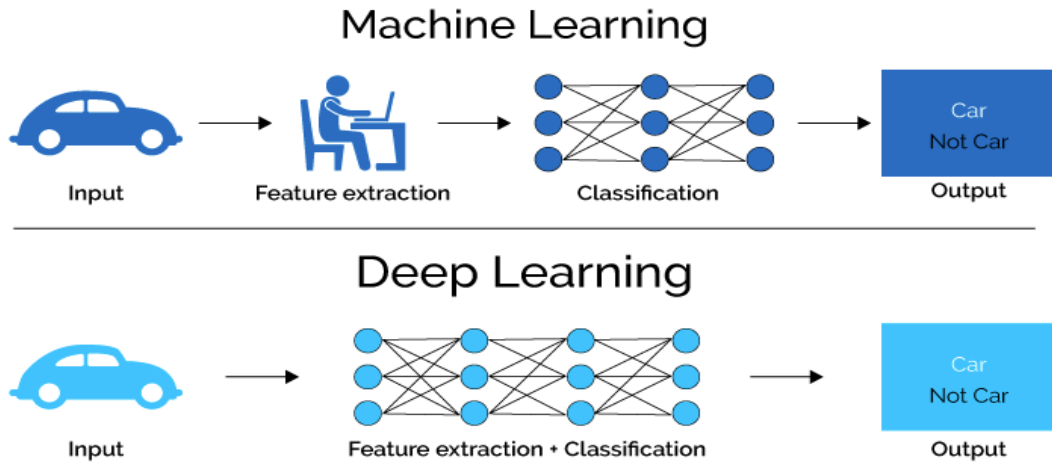
The most suitable model type and hyper parameter for model trainings were researched and tested. As a result of this researches, created model is a sequential model with Embedding and LSTM layers. Both models used exponential decay learning rates. The complete training model has a dropout layer as a result of its long input datasets.

In test case these two models and test data set were used. Predictions from both models were used for calculate accuracy, precision, recall and F1 score values. All calculated values used for comparison with research[13] in literature.

1.GİRİŞ

Bir davanın içeriğini ele alarak, bu dava hakkında karara varmak, adalet ve kanunlar üzerine eğitim almış insanlar için dahi tartışmalı olabilirken, bu işin bir bilgisayar tarafından yapılması yapay zekâ alanında zorlayıcı bir görevdir. İnsan hakları ve adalet açısından, bir dava dosyasını adil bir şekilde tahmin edebilmek hem devletler hem de bu devletin bireyleri için olmazsa olmazdır. Bu proje, yapay zekâ yardımıyla, dava içeriklerinin ret veya Kabul edildiğinin tahminini amaçlamaktadır.

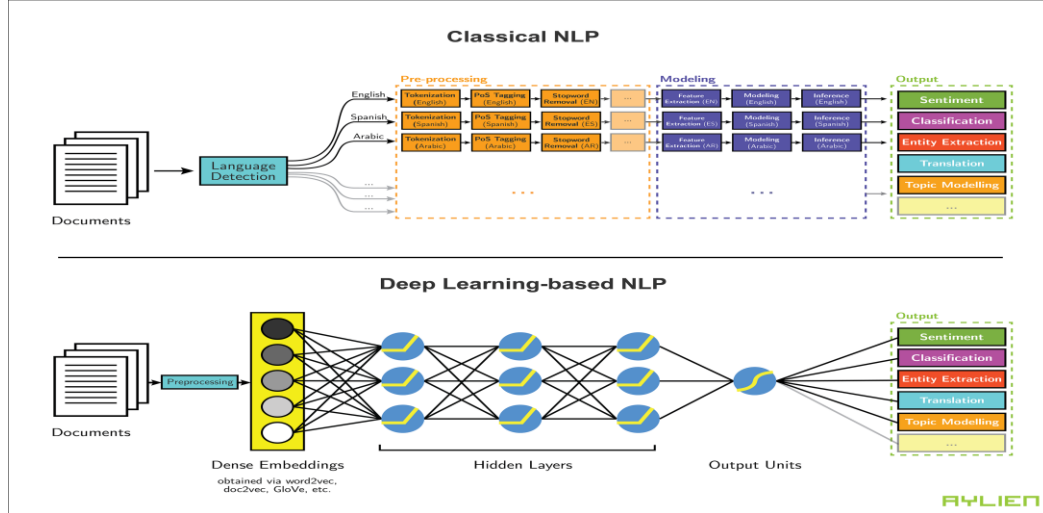
Bir metinden edinilen bilgilerle metni sınıflandırma işlemi, genelde makine öğrenmesi veya derin öğrenme ile gerçekleştirilmektedir[1]. İki yönteminde birbirine kesin bir üstünlüğü olmamakla birlikte, beraber kullanıldıklarında da başarılı olduğu görülmektedir.[2]



ŞEKİL 1.1 Makine Öğrenmesi ve Derin Öğrenme Karşılaştırması

Derin öğrenme veya diğer adıyla sinir ağları, özellik çıkarılmadan, etiketlenmiş ham veriyi girdi olarak kabul ederek öğrenen yapay zekâ alt dalıdır. Modele verilen veriyi iterasyonlar uygulayarak, iterasyonlar sırasında parametreleri düzenleyerek öğrenimi sağlayan bir yaklaşımdır.

Klasik dil işleme algoritmalarından ayrı olarak, kelime vektörlerinden özellik çıkarımı yapmadan, bu vektörleri yapay sinir ağlarına göndererek öğretmeyi amaçlar.



ŞEKİL 1.2 Klasik NLP, Derin Öğrenme içeren NLP farkı

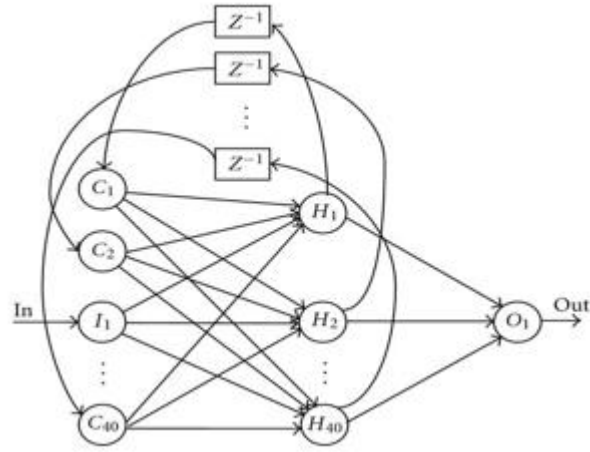
Bir giriş katmanına gönderilen veri, buradan gizli katmanlara gönderilir, gizli katmanların yaptığı hesaplamaların sonucunu çıkış katmanına iletir ve bu katman, girmiş olduğumuz verinin sonucunun tahminini verir.

Mahkeme kararlarını tahmin etme alanında ise son yıllarda çalışmalar, derin öğrenme algoritmalarının ve teknolojinin gelişmesiyle birlikte artmıştır. Amerika Birleşik Devletleri, Çin ve bazı Avrupa ülkelerinde bu konu üzerine çalışmalar yapılmaktadır. Çin’de düzenlenen yarışmada alınan farklı yarışmacıların yüksek tahmin oranı, bu alanda çalışmaların oldukça gelişmekte olduğunu göstermektedir.[3][4]

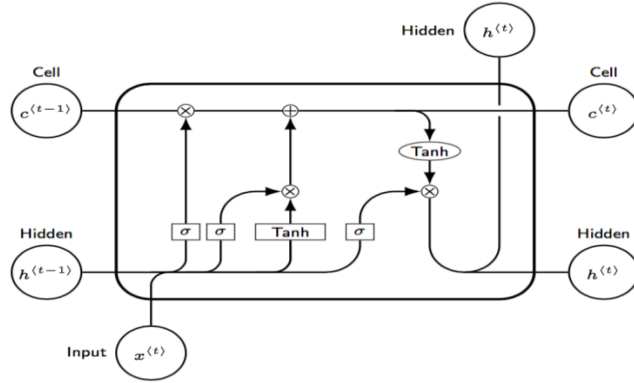
2.YÖNTEM

ŞEKİL 2.1 de yapısı görülen RNN sinir ağları bir girdiyi, daha önceki girdileri de göz önünde bulundurarak öğrenen bir sinir ağı çeşididir. Belli bir düzende uygulanan

girdiler, çıktı üzerine etki eder. Fakat uzun sayıda girdi için, RNN geçmişteki girdileri hatırlayamamaya başlar. ŞEKİL 2.2 de gösterilen LSTM ağ yapısı ise, RNN'lerdeki bu sorunu çözebilmek adına geliştirilmiş bir yöntemdir. Daha uzun girdileri hafızasında tutarak, kelimeler arasında daha iyi bağlantı kurar, buna göre LSTM algoritması kullanılarak modelin eğitilmesi ve bu model üzerinden tahmin yapılması, model için uygun görülmüştür.



ŞEKİL 2.1 RNN Sinir Ağlarının Yapısı



ŞEKİL 2.2 LSTM Sinir Ağlarının Yapısı

Tahmin sonuçları iki kategoriye ayrıldığından, ikili sınıflandırma yapılmıştır. JSON formatındaki dava dosyalarından, dava içeriği ve davada ihlal edilen maddeler okunur. Model için girdi dosyaları iki şekilde hazırlanmıştır.

İlk yöntemde, dava içeriğinin her maddesi önışlemden geçer, bu işlem sırasında, cümleler kelimelerine ayrılır, kelimelerdeki büyük harfler küçük harfe dönüştürülerek

ayrılık ortadan kaldırılır, son olarak eş anlamlı kelimeler aynı kelimeye dönüştürülür. İşlenmiş olan bu dava maddesi, eğitime girecek listeye eklenir. Maddenin içinde bulunduğu dosyadaki dava kararı ise, eğitime girecek sonuç listesine eklenir. Böylece bütün olarak öğrenmeye göre daha etkin bir öğrenme olması, modelin cümle sonuç ilişkisi geliştirmesi amaçlanır.

İkinci yöntemde ise, dava içeriği bir bütün olarak önışlemden geçer, bu işlem sırasında, cümleler kelimelerine ayrılır, kelimelerdeki büyük harfler küçük harfe dönüştürülerek ayrılık ortadan kaldırılır, Son olarak eş anlamlı kelimeler aynı kelimeye dönüştürülür. İşlenmiş olan bu dava metni, tamamı kullanılarak, eğitime girecek listeye eklenir. Dosyadaki dava kararı ise, eğitime girecek sonuç listesine eklenir.

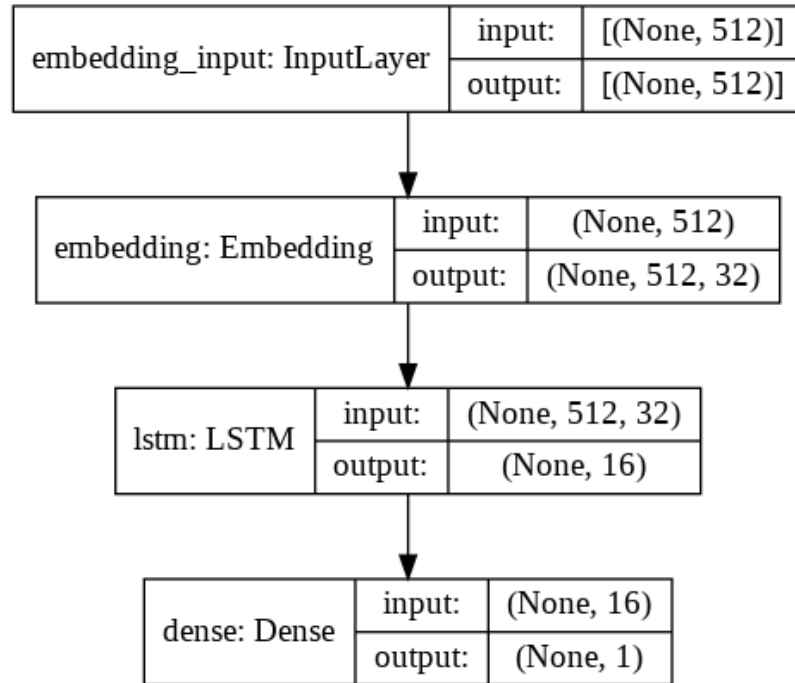
Bütün dosyalar okunduktan sonra Keras[5] kütüphanesi içerisinde bulunan Tokenizer objesi oluşturmuş, ve bu objenin en fazla bulunan 5000 kelimeyi alması sağlanmıştır. Daha sonra objenin tanıdığı bu kelimeler kullanılarak, veri seti kelimelerin sayı kodlarından oluşan listelere dönüştürülmüştür. Bu listelere bütün liste elemanlarının aynı uzunlukta olmasını sağlamak için işlem uygulanmış,512 kelimededen küçük listeler başına sıfır eklenerek uzatılmış, büyük listelerin ise son 512 kelimesi kullanılmıştır. Numpy[6] kütüphanesi kullanılarak, bu listeler modele giriş için hazır hale getirilmiştir. Makine öğrenimi için, keras kütüphanesinden ardışık bir model oluşturulmuş, LSTM katmanı iki model için de farklı sayılarda birimler kullanılarak eklenmiş, çıkış katmanı olarak tek katman kullanılmış ve aktivasyon fonksiyonu olarak en yüksek verim beklenen sigmoid[7]seçilmiştir.

Deneyler sonucunda oluşturulan modellerin katman yapıları, ŞEKİL 4.1 ve ŞEKİL 4.2 de gösterilmiştir. Keras Embedding katmanı giriş katmanı olarak kullanılmıştır. Modelin sinir ağı olarak Keras kütüphanesinin LSTM algoritması, veri seti olarak Avrupa İnsan Hakları Mahkemesi davaları[8] kullanılmıştır. Veri setini ön işlemeden geçirmek için NLTK[9] kütüphanesi, kelimelerin kelime vektörlerine dönüşümü için ise Keras Preprocessing kütüphanesi kullanılmıştır. Veri seti %65 eğitim %10 doğrulama

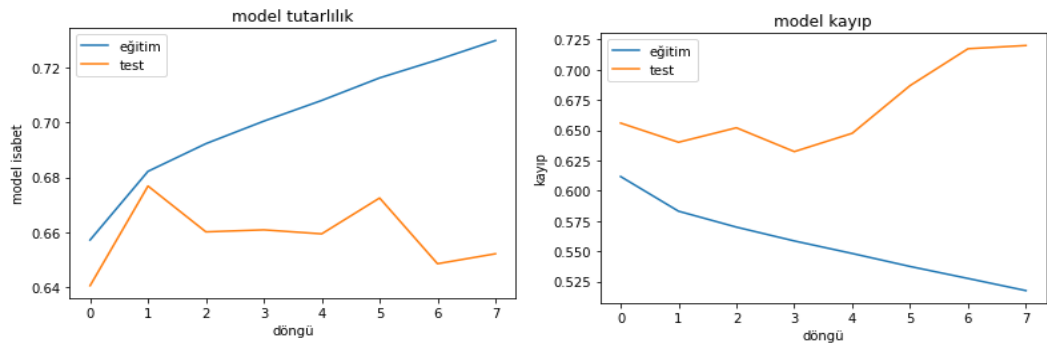
testi, %25 son test olacak şekilde ayrılmıştır. Model eğitiminde iyileştirici olarak Adam algoritması kullanılmıştır.

3.DENEYLER VE BULGULAR

1.MADDE EĞİTİMİ YAPILAN MODEL



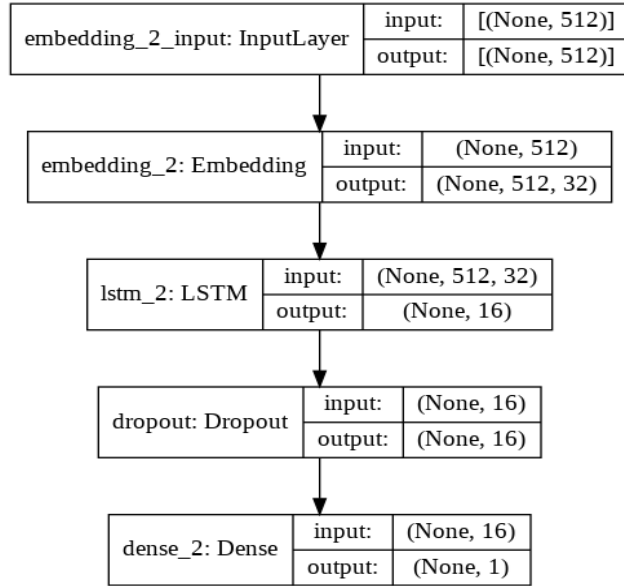
ŞEKİL 3.1 Madde Yöntemi İçin Başlangıç Testi Modeli Katman Yapısı



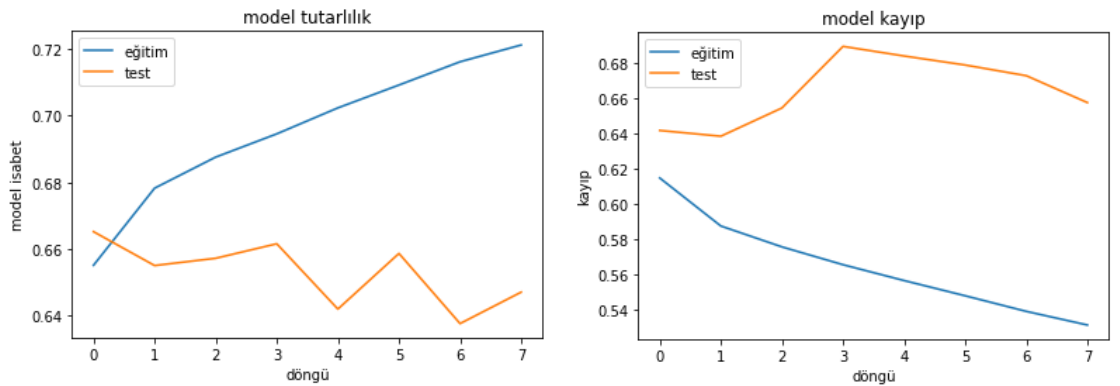
ŞEKİL 3.2 Madde Yöntemi İçin Başlangıç Kayıp ve İsbet Sonuçları

İlk test sonucunda modelin 4.döngüden sonra, doğrulama testinde aşırı uyum gösterdiği gözlemlendi. Bu sorunu çözmek için kullanılabilecek uygulamalar araştırıldı.[10]

Aşırı uyum sağlamaya çözüm olarak 0,3 oranında seyreltme katmanı[11] kullanıldı. Bu katmanın, iterasyonlar boyunca rastgele düğümleri yok sayması ve eğitimin genele yayılmasını sağlaması amaçlandı. ŞEKİL 3.3'deki model yapısı oluşturularak eğitimi gözlemlendi. Şekil 3.4 gözlemleriyle birlikte isabetli tahminlerin azaldığı ve kayıp oranı iterasyonlar boyunca artmaya devam ettiği tespit edildi.



ŞEKİL 3.3 Seyreltme Katmanı İçeren Madde Model

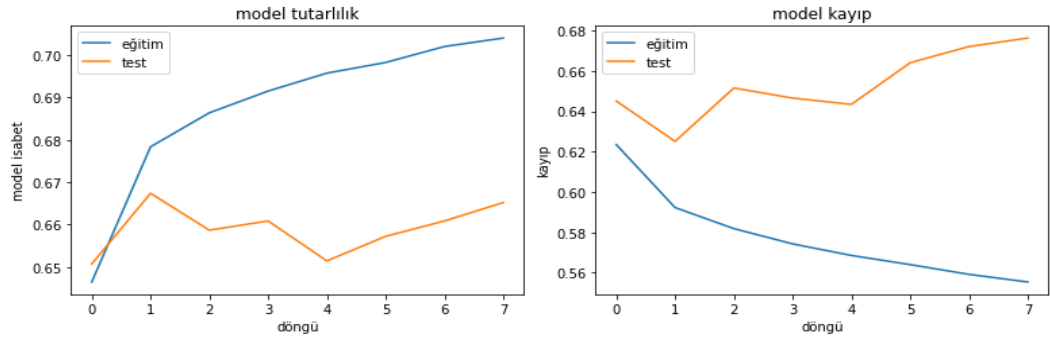


ŞEKİL 3.4 Seyreltme Katmanlı Model İçin İsabet ve Kayıp Sonuçları

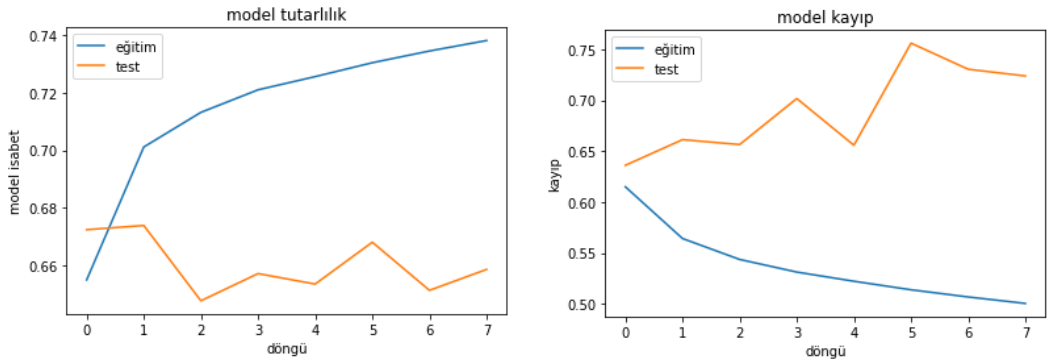
L2 ve L1 düzenleyicileri [12] kullanılarak yapılan eğitim de bu kayıp artışını önlemekte etkili olmadı. Düzenleyicilerin uygulandığı eğitimlerde, modelin başlarında dahi isabet azalması ve kayıp görüldü. Bu sebeple modele düzenleyici eklenmemesine karar verildi.

Veri setine eklenecek yeni dava bulunmadığından, aynı zamanda yapay veri eklenmesi mahkeme dosyaları gibi katı bir alanda uygulanması sakıncalı olduğundan, veri seti büyütülerek kayıp değerinin azaltılması mümkün değildi. Bu nedenle modelin öğrenme katsayısı, Adam algoritması için başlangıç değeri olan 0.001 değerinden 0.0005 değerine düşürüldü. İterasyonlar boyunca öğrenme oranı sürekli azaltılarak, modelin daha etkin bir öğrenme sağlanması hedeflendi.

ŞEKİL3.5 de bulunan test sonuçları gözlemlendiğinde, seyreltme katmanı kullanmanın ve bu yöntemin birlikte kullanımında, yeterli olmamasına rağmen gelişmiş sonuçlar elde edildi.

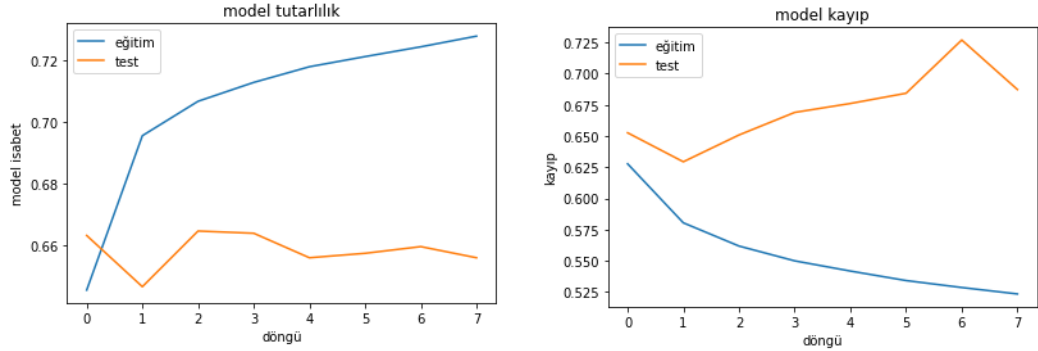


ŞEKİL 3.5 Öğrenme Değeri Üstel Olarak Düşen Modelin İsabet ve Kayıp Sonuçları



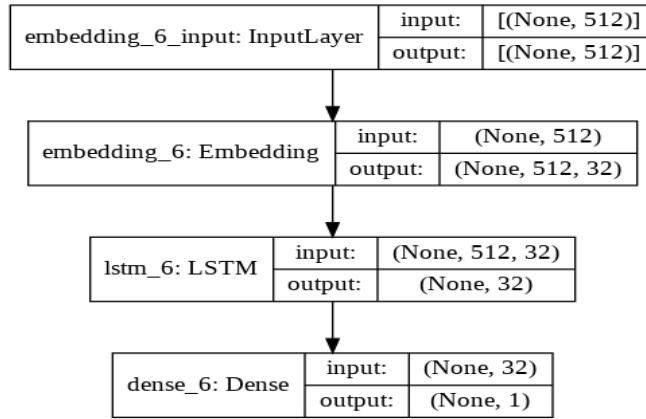
ŞEKİL 3.6 Daha Uzun Veri Kesitleri Alan Modelin İsabet ve Kayıp Sonuçları

Daha önce 512 kelime uzunluğuna indirilmiş girdi verileri, 1024 uzunluğuna getirilerek yapılan testte de (ŞEKİL 3.6) ilerleyen iterasyonlarda, isabet kaybı gözlemlendi. Son olarak modelin kompleks yapısını azaltmak amacıyla LSTM birimlerinin sayısı 4'te 1'ine indirildi. ŞEKİL 3.7 de eğitim durumunda, gelişme olmadığı tespit edildi. Buna göre ŞEKİL 3.1 de bulunan model yapısının, kayıp yaşamadan önceki azami döngü sayısında ve azalan öğrenme oranıyla test için kullanılmasına karar verildi.

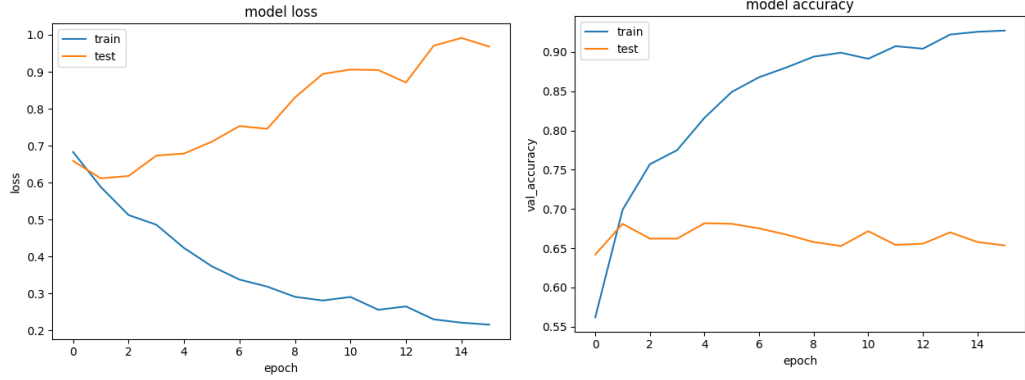


ŞEKİL 3.7 LSTM Birim Sayısı Azaltılmış Model İçin İsabet ve Kayıp Sonuçları

2) BÜTÜN OLARAK EĞİTİM YAPILAN MODEL

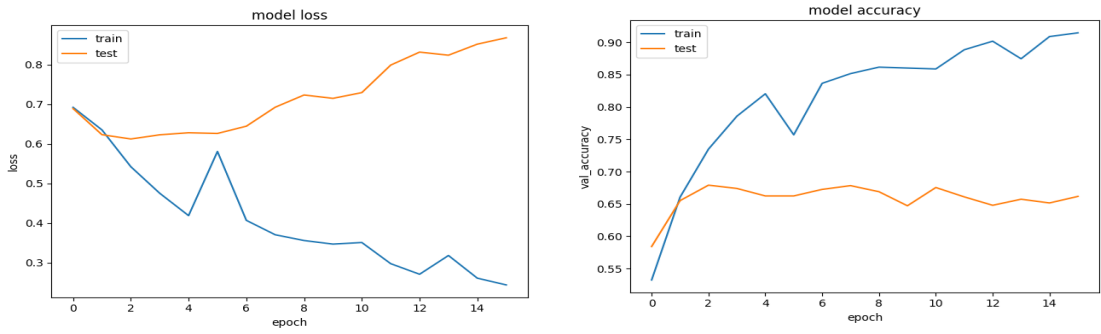


ŞEKİL 3.8 Bütün Olarak Eğitim Yöntemi İçin Başlangıç Testi Modeli Birim Yapısı

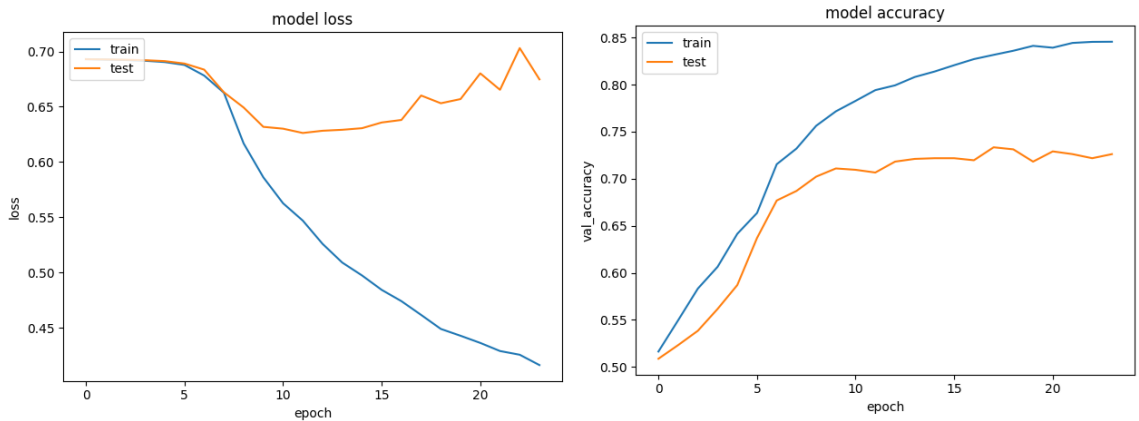


ŞEKİL 3.9 Bütün Olarak Eğitim Modeli İçin Başlangıç Kayıp ve İsabet Grafiği

İlk modelde olduğu gibi, bu modelde de düşük döngü sayılarında modelin kayıp değerlerinin artmaya başladığı ve isabet kaybı yaşadığı gözlemlendi. Bu duruma çözüm olarak, 0.3 değerinde bir seyreltme katmanı uygulanarak model gelişimi gözlemlendi.

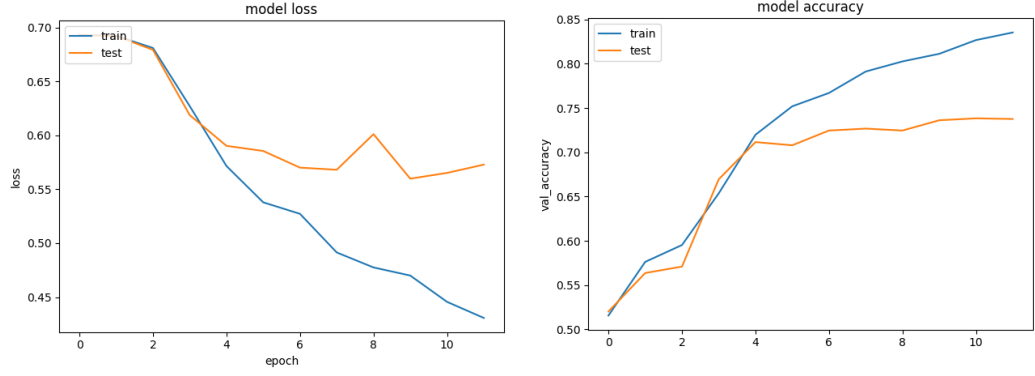


ŞEKİL 3.10 Seyreltme Katmanı Eklenmiş Bütün Olarak Eğitim Modeli Kayıp ve İsabet Grafiği



ŞEKİL 3.11 Üstel Azalan Öğrenme Oranlı Bütün Olarak Eğitim Modeli Kayıp ve İsabet Grafiği

Yalnızca seyreltme katmanı eklemenin, model üzerinde biraz olumlu etkisi olduğu gözlemlendi. Bu katmanla beraber, üstel azalan öğrenme oranı kullanılarak test yapıldı. Yaklaşık 10 döngümlük bir eğitimin, model için yeterli olduğu gözlemlendi.

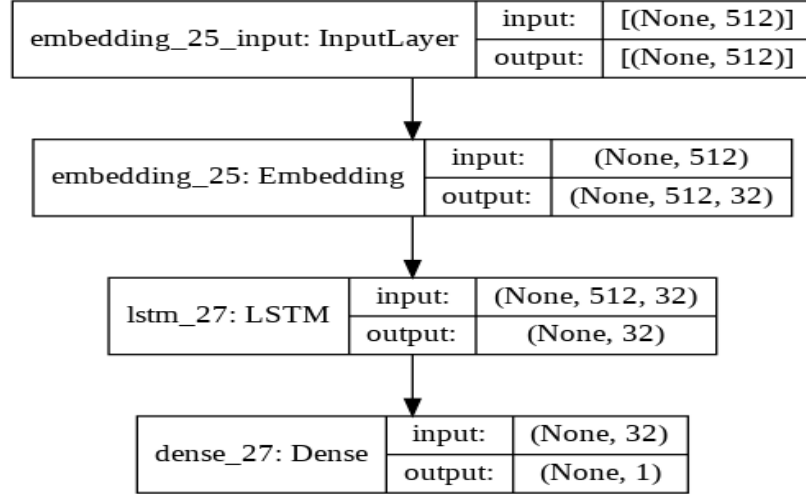


ŞEKİL 3.12 LSTM Birim Sayısı Arttırılmış Bütün Olarak Eğitim Modeli Kayıp ve İsbet Grafiği

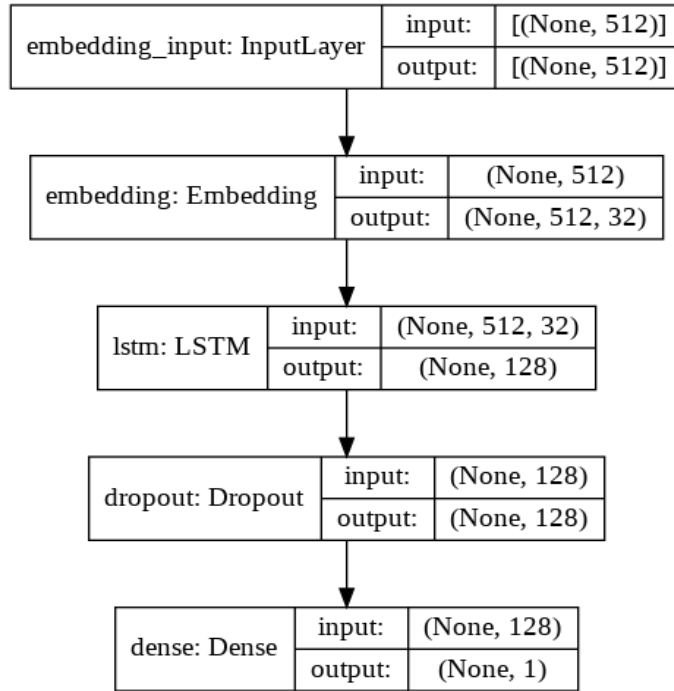
LSTM birimleri arttırılarak isabet oranında artış durumu kontrol edildi ve olumlu bir sonuç elde edildi.

Sonuçta seyreltme katmanlı ve öğrenme oranı üstel olarak düşen modelin, 128 LSTM birimi kullanılarak testte kullanılmasına karar verildi.

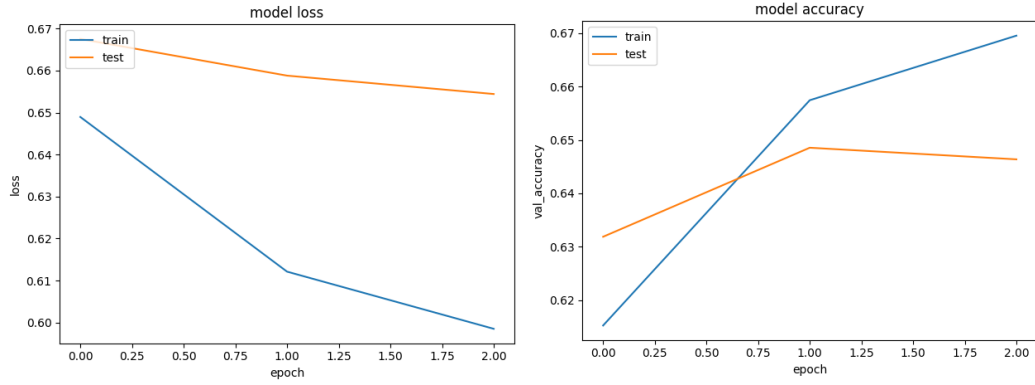
4.SONUÇ



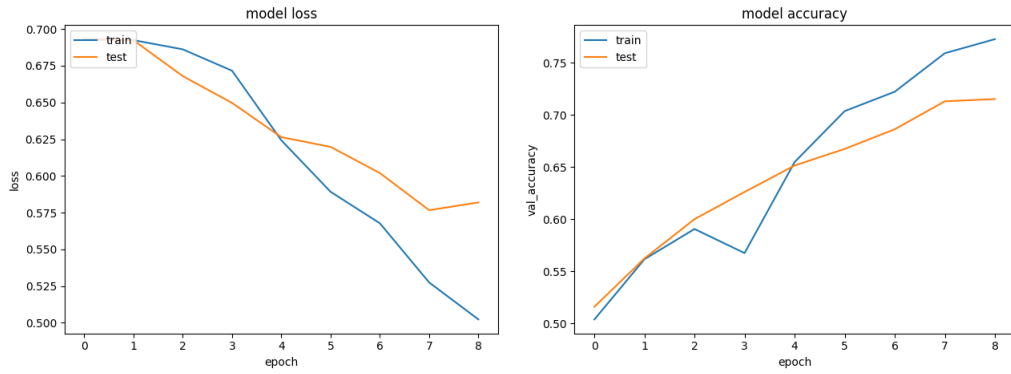
ŞEKİL 4.1 İlk Yöntem için Model Yapısı



ŞEKİL 4.2 İkinci Yöntem için Model Yapısı



ŞEKİL 4.3 Madde Eğitim Modeli Kayıp ve İsabet Grafiği



ŞEKİL 4.4 Bütün Olarak Eğitim Modeli Kayıp ve İsabet Grafiği

İki modelin eğitimi de 128'lik parçalar kullanılarak, Ryzen 5 3500X işlemci üzerinde yapıldı.

İlk modelin eğitimi yaklaşık 1 saat sürerken, ikinci model için 30 dakikalık bir eğitim yeterli oldu. Eğitimlerde kayıp oranını önlemek için, keras kütüphanesi içerisindeki erken durdurma çağrısı kullanıldı. Model başarısının ölçümünde kullanılan yöntemler ŞEKİL 4.5 deki gibidir.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + T_n}$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

ŞEKİL 4.5 Başarı Hesaplamasında Kullanılacak Denklemler

TABLO 4.1 Modeller İçin Test Sonuçları

Model	İsabet	Kesinlik	Duyarlılık	TP	FP	TN	FN	F1Puanı
1.Yöntem	%67,48	%71,26	%84,80	1674	675	349	300	%77,31
2.Yöntem	%70,25	%80,43	%72,44	1430	348	676	544	%76,22

İki yöntem karşılaştırıldığında, verileri bir bütün olarak kabul eden 2.yöntemin, daha isabetli olduğu gözlemlendi. Fakat veri setinde ceza verilmiş ve reddedilmiş dava oranında dengesizlik olduğundan, F1 puanı daha yüksek olan maddelerle eğitilmiş olan 1.modelin daha başarılı olduğu kabul edilebilir. Maddelerle eğitilmiş veri cezalandırma kararı vermeye daha yatkınken, bütün olarak eğitilen veri daha eşit dağılımlı kararlar vermiştir.

Elde edilen sonuçlar, aynı veri setini kullanan çalışmadaki sonuçlar[13] ile karşılaştırıldığında, iki modelinde önceden eğitilmiş kelime temsilleri kullanan modellerle yakın seviyede başarı gösterdiği gözlemlenmiştir. Bu LSTM algoritmasının yazı analizinde ve kelimeler arasında bağlantı kurmada başarılı olduğunun bir göstergesidir. Modellerin daha da geliştirilebilmesi için, her iki karar içinde bulunan, kişisel bilgilerin verildiği kısımların çıkartılması gibi veri üzerinde işlemler, modelin geliştirilmesine katkıda bulunabilir. Modelin tanıdığı kelime listesinde artış, yani önceden eğitilmiş kelime temsilleri kullanıldığında, modelin tahminlerinde daha isabetli olması

beklenebilir. Daha geniş bir veri seti kullanılarak eğitilen modeller, daha başarılı olabilir. Sonuç olarak bu çalışma, geliştirilmeye açıktır.

KAYNAKLAR

- [1] AMJAD, M., VORONKOV, I., SAENKO, A., & GELBUKH, A. Comparison of text classification methods using deep learning neural networks. *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, 2019.
- [2] PIETERS, M., & WIERING, M. Comparison of machine learning techniques for multi-label genre classification. *Communications in Computer and Information Science*, 823(February), 131–144, 2018.
- [3] ZHONG, H., XIAO, C., GUO, Z., TU, C., LIU, Z., SUN, M., FENG, Y., HAN, X., HU, Z., WANG, H., & XU, J. *Overview of CAIL2018: Legal judgment prediction competition*, 2018.
- [4] GUPTA, K. Y., & BHARADWAJ, M. *Predicting Justice Behavior of Supreme Court of the United States*.
- [5] *Keras Kütüphanesi* [online], <https://keras.io/api/> , [Ziyaret Tarihi: 27 Eylül 2020].
- [6] *Numpy Kütüphanesi* [online] , <https://numpy.org/doc/stable/> , [Ziyaret Tarihi: 27 Eylül 2020].
- [7] FENG, J., & LU, S. Performance Analysis of Various Activation Functions in Artificial Neural Networks. *Journal of Physics: Conference Series*, 1237(2), 2019.
- [8] *NLTK Kütüphanesi* [online], <https://www.nltk.org> , [Ziyaret Tarihi: 27 Eylül 2020].
- [9] *Veriseti kaynağı* [online], <https://archive.org/details/ECHR-ACL2019> , [Ziyaret Tarihi: 27 Eylül 2020].
- [10] YING, X. An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, 1168(2), 2019.
- [11] SRİVASTAVA N. , HİNTON G. ,KRİZHEVSKY A.,SUTSKEVER I. , SALAKHUTDİNOV R , Dropout: A Simple Way to Prevent Neural Networks from Overfitting, *Journal of Machine Learning Research* 15 ,2014.
- [12] KİNGMA, D. P., & BA, J. L. , Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15,2015.
- [13] CHALKİDİS, I., ANDROUTSOPOULOS, I., & ALETRAS, N. , Neural legal judgment prediction in English. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, 4317–4323, 2020.