

The Identification of Satirical Fake News in Turkey

**Graduation Project**

**THE IDENTIFIDACION OF SATIRICAL FAKE NEWS IN TURKEY**

**Submitted By**

**Ahmet Kaan Memioğlu 1900005528**

**Emrecan Üzüm 1900005485**

**Şükrü Erim Sinal 1900003587**

Project Advisor

Assoc. Prof. Akhan Akbulut

Department of Computer Engineering

İstanbul Kültür University

2023 Spring Term

Jun

2023



**Graduation Project**

**THE IDENTIFIDACION OF SATIRICAL FAKE NEWS IN TURKEY**

**Submitted By**

**Ahmet Kaan Memioğlu 1900005528**

**Emrecan Üzüm 1900005485**

**Şükrü Erim Sinal 1900003587**

Advisor

Assoc. Prof. Akhan Akbulut

THE EXAMINATION COMMITTEE

Jury Member Signature

1. ………………………….. ………………………..

2. ………………………….. ………………………..

3. ………………………….. ………………………..

# ABSTRACT

The proliferation of misinformation, or 'fake news', has emerged as a significant issue in today's digitally driven society. In the context of the Turkish language, its widespread implications necessitate robust computational approaches for distinguishing genuine news from spurious content. This study focuses on addressing this pertinent challenge using machine learning algorithms.We curated a dataset derived from multiple renowned Turkish news sources, encompassing both authentic and fake outlets. Owing to the stringent disinformation laws in Turkey, the dataset had a higher representation of real news, marking an inherent limitation.The project's methodological framework encompassed the use of distinct machine learning algorithms: XGBoost, Random Forest, Logistic Regression, Long Short-Term Memory (LSTM) and our CNN model. Our findings indicate that Logistic Regression outperformed the other models, yielding an accuracy of 91%. XGBoost and Random Forest followed suit with satisfactory results. However, the LSTM model exhibited subpar performance, suggesting that the dataset we used in this project LSTM algorithm was not optimally suited for this method. Our CNN model was the solution for missing Deeplearning implementation for the project and it yielded closer results to greater models.Despite the challanges, our machine learning models demonstrate considerable promise in addressing the fake news issue. Future research endeavors could explore the incorporation of multi-modal data and more sophisticated deep learning models to further enhance the performance. This research's overarching goal lies in bolstering the integrity of information circulated in public spheres. By providing accurate detection of fake news, this study contributes to fostering an informed public discourse, thereby empowering society to make informed decisions based on reliable information sources.

# ÖZET

Yanlış bilginin veya 'sahte haberlerin' yaygınlaşması, günümüzün dijital güdümlü toplumunda önemli bir sorun olarak ortaya çıkmıştır. Türkçe bağlamında, yaygın etkileri, gerçek haberleri sahte içerikten ayırt etmek için sağlam bilişimsel yaklaşımlar gerektirmektedir. Bu çalışma, makine öğrenimi algoritmalarını kullanarak bu ilgili zorluğu ele almaya odaklanmaktadır.Birden fazla tanınmış Türk haber kaynağından elde edilen ve hem gerçek hem de sahte yayınları kapsayan bir veri kümesi oluşturduk. Türkiye'deki katı dezenformasyon yasaları nedeniyle, veri kümesi gerçek haberlerin daha yüksek bir temsiline sahipti ve bu da doğal bir sınırlamaya işaret ediyordu.Projenin metodolojik çerçevesi dört farklı makine öğrenimi algoritmasının kullanımını kapsamaktadır: XGBoost, Random Forest, Logistic Regression ve Long Short-Term Memory (LSTM).Bulgularımız, Lojistik Regresyonun diğer modellerden daha iyi performans gösterdiğini ve %91'lik bir doğruluk sağladığını göstermektedir. XGBoost ve Random Forest da tatmin edici sonuçlarla diğer algoritmadan sonraki güzel seçeneklerden biri olarak görünüyor. Bununla birlikte, LSTM modeli düşük bir performans sergilemiştir, bu da bu projede LSTM algoritmasında kullandığımız veri kümesinin bu yöntem için en uygun şekilde uygun olmadığını düşündürmektedir. Kullandığımız CNN algoritması Deeplearning çözümümüz için yeterli olmuştur.Zorluklara rağmen, makine öğrenimi modellerimiz sahte haber sorununu ele almada önemli bir umut vaat etmektedir. Gelecekteki araştırma çabaları, performansı daha da artırmak için çok modlu verilerin ve daha sofistike derin öğrenme modellerinin dahil edilmesini keşfedebilir.Bu araştırmanın genel amacı, kamusal alanda dolaşan bilgilerin bütünlüğünü desteklemektir. Sahte haberlerin doğru bir şekilde tespit edilmesini sağlayarak, bu çalışma bilinçli bir kamusal söylemin teşvik edilmesine katkıda bulunmakta ve böylece toplumu güvenilir bilgi kaynaklarına dayalı bilinçli kararlar alma konusunda güçlendirmektedir.

# ACKNOWLEDGEMENTS

We are genuinely grateful for the continuous support and guidance we have received throughout the duration of this project. In particular, we would like to extend our deepest gratitude to Doctor Akhan Akbulut, whose expertise, understanding, and patience added substantially to our project experience. His wisdom, knowledge, and commitment to the highest standards inspired and motivated us. Doctor Akbulut's profound insights into machine learning and data analysis not only facilitated our understanding of the complex nuances of this project but also instilled a fervor for the field that we will carry forward into our future endeavors. The success of this project is largely due to his diligent guidance and persistent encouragement throughout its progress. This accomplishment would not have been possible without him. We are truly honored to have had him as my mentor. Furthermore, we would also like to thank everyone else who contributed, directly or indirectly, to this project. Your support has been invaluable. Thank you.

# TABLE OF CONTENTS

[ABSTRACT I](#_Toc5951944)

[ÖZET II](#_Toc5951945)

[ACKNOWLEDGEMENTS III](#_Toc5951946)

[TABLE OF CONTENTS IV](#_Toc5951947)

[LIST OF TABLES VI](#_Toc5951948)

[LIST OF FIGURES VII](#_Toc5951949)

[SYMBOLS & ABBREVIATIONS VIII](#_Toc5951950)

[1. INTRODUCTION 1](#_Toc5951951)

[1.1. Problem Statement 1](#_Toc5951952)

1.2. Project Purpose 2

1.3. Project Scope 3

1.4. Objectives and Success Criteria of the Project 4

1.5. Report Outline 5

[2. RELATED WORK 2](#_Toc5951957)

2.1. Existing Systems 7

2.2. Overall Problems of Existing Systems 18

2.3. Comparison Between Existing and Proposed Method 20

[3. METHODOLOGY 3](#_Toc5951961)

3.1. Overview of the Dataset/Model 24

3.2. Tools and Technology 29

3.3. Proposed Approach 30

4. EXPERIMENTAL RESULTS 40

5. DISCUSSION 45

6. CONCLUSIONS 48

REFERENCES 49

APPENDIX 52

# LIST OF TABLES

T**able 2.1:** Table for Existing Methods Comparison………………………...……...22&23

**Table 4.1:** Table for Algorithm Comparison……………………………………………43

# LIST OF FIGURES

**Figure 3.1:** CNN model architecture…………………………………………………….27

**Figure 4.1:** Accuracy in Confirmed news……………………………………………….41

**Figure 4.2:** Accuracy in False news……………………………………………………..41

**Figure 4.3:** CNN model loss in training…………………………………………………41

**Figure 4.4:** CNN model accuracy in training……………………………………………41

**Figure 4.5:** CNN confusion matrix in Training…………………………………………42

**Figure 4.6:** CNN confusion matrix in Test……………………………………………...42

**Figure 4.7:** ROC curves between the algorithms………………………………………..42

**Figure 4.8:** ROC curve of CNN algorithm……………………………………………...42

**Figure 4.9:** F1 scores between the algorithms…………………………………………..42

**Figure 4.10:** confusion matrix between the algorithms…………………………………43

**Figure 6.1:** 1st part of the tree in Random Forest……………………………………….52

**Figure 6.2:** 2nd part of the tree in Random Forest………………………………………53

**Figure 6.3:** 3rd part of the tree in Random Forest………………………………………54

**Figure 6.4:** Whole Tree in Random Forest……………………………………………...54

# SYMBOLS & ABBREVIATIONS

ACM: Association for Computing Machinery

APA: American Psychological Association

IEEE: Institute of Electrical and Electronics Engineers

LSTM: Long Short-Term Memory

CNN: Convolutional Neural Network

RNN: Recurrent Neural Network

SIR: Susceptible Infectious Recovered

API: Application Programming Interface

URL: Uniform Resource Locators

LIWC: Linguistic Inquiry and Word Count

FNN: Feedforward Neural Network

SVM: Support Vector Machine

POS: Part of Speech

NLP: Natural language Processing

NER: Named Entity Recognition

CSV: Comma Separated Values

# INTRODUCTION

## Problem Statement

In today's world, fake news is a severe issue that is becoming more and more common. False information may travel fast and have a big influence on people, communities, and even entire countries because of the popularity of social media and how simple it is to share information. Fake news is a broad and complicated issue, with a number of elements influencing its production and spread. Finding false news is one of the biggest difficulties in combating the issue. Fake news can be difficult to identify, as it often contains elements of truth and may be designed to appear credible. Additionally, the motivations behind the creation and spread of fake news can vary widely, from financial gain to political manipulation. Detecting fake news requires a multifaceted approach that involves understanding the mechanisms behind its creation and dissemination, as well as developing effective methods to identify and counter it. This project aims to contribute to the field of fake news detection by developing a machine learning-based approach that can accurately classify news articles as either real or fake.

The project recognizes that the problem of fake news is not limited to any particular region, country, or language. Therefore, the approach developed in this project aims to be applicable to a wide range of contexts and languages. The project also recognizes that the problem of fake news is dynamic and evolving, with new forms and techniques constantly emerging. Therefore, the approach developed in this project aims to be adaptable and scalable, capable of incorporating new data and techniques as they become available. The development of an effective fake news detection system has significant implications for society. It can help individuals and organizations make informed decisions and protect themselves from the harm caused by false information. It can also contribute to the maintenance of trust in institutions and the preservation of democratic values. Therefore, this project seeks to make a meaningful contribution to the development of effective solutions to the problem of fake news.

## Project Purpose

The goal of this project is to use Python to create a machine learning-based strategy to detecting fake news. The method involves employing natural language processing techniques to preprocess the content of news stories before training a classification model with machine learning algorithms. The ultimate objective of this project is to create an accurate and dependable false news detection system that individuals and organizations may use to check information authenticity. Such a system would have several potential applications, including:

Helping news organizations to identify and correct false or misleading information in their reporting. Assisting social media platforms in identifying and removing fake news from their platforms. Enabling individuals to verify the accuracy of information they encounter online and avoid being misled by false information. Supporting researchers and policymakers in studying the problem of fake news and developing effective strategies to address it. To achieve this goal, The research will concentrate on creating a machine learning-based system that can reliably categorize news stories as legitimate or fraudulent. The method will be founded on a thorough analysis of the current literature on false news identification, as well as a rigorous assessment of several machine learning algorithms and natural language processing approaches.

The project recognizes that developing an effective fake news detection system is a complex and challenging task. The system must be able to identify false information while minimizing false positives and avoiding censorship of legitimate content. It must also be able to adapt to the dynamic and evolving nature of the problem of fake news. To address these challenges, the project will focus on developing a robust and scalable approach that can incorporate new data and techniques as they become available. The approach will be designed to be language-dependent and applicable to a wide range of contexts, enabling it to be used in Turkey.

Overall, the project aims to make a significant contribution to the field of fake news detection by developing an accurate and reliable machine learning-based approach. The project recognizes the importance of addressing the problem of fake news for maintaining trust in institutions and promoting the dissemination of accurate information and seeks to make a meaningful contribution to this important and pressing issue.

## Project Scope

The goal of this project is to provide a Python-based machine learning technique for identifying fraudulent information. The project will entail gathering and preprocessing news items, creating and honing machine learning models, and assessing the performance of the models. The project will primarily focus on Turkish language news articles. However, the approach developed in this project will be designed to be language-dependent and adaptable to other languages, enabling its potential application to different regions and languages in the future.

The project will involve the creation or manipulation of news articles for the purposes of testing or training the models. The project will use a variety of natural language processing techniques to preprocess the text of news articles, including tokenization, stemming, stop-word removal, and part-of-speech tagging. These techniques will be used to extract relevant features from the text, such as word frequencies and syntactic patterns, which will be used as inputs to the machine learning models.

Accuracy, precision, recall, and F1 score are a few examples of common performance measures that will be used to assess the machine learning models. To assess the algorithms' efficacy, they will be compared to current state-of-the-art methods for identifying fake news. Python and appropriate machine learning libraries, including scikit-learn and TensorFlow, will be used to carry out the project. The project will be created with scalability and adaptability in mind, making it possible for it to be updated and enhanced when new information and methods become available. The project will not address the broader social, political, or ethical implications of fake news. While these issues are important and relevant, they are outside the scope of this project, which is focused primarily on the technical aspects of developing an effective fake news detection system.

Overall, the project aims to develop a machine learning-based approach to detect fake news that is effective, adaptable, and scalable. The project recognizes the importance of addressing the problem of fake news and seeks to make a meaningful contribution to this important and pressing issue.

## Objectives and Success Criteria of the Project

The major goal of this research is to create a machine learning-based method that can reliably identify false news in press releases written in Turkish. The strategy should be efficient, flexible, and scalable, and it should be founded on a thorough analysis of the body of research on the identification of fake news. To achieve this objective, the project will be guided by the following specific objectives:

Identify the most effective natural language processing techniques for preprocessing news article text, including tokenization, stemming, stop-word removal, and part-of-speech tagging. Analyze the efficacy of a variety of machine learning methods, such as decision trees, support vector machines, neural networks, and others, in identifying misleading information. Using the information gleaned from the text using natural language processing methods, create and train a machine learning model that can properly categorize news stories as either legitimate or fraudulent. Utilizing common performance measures like accuracy, precision, recall, and F1 score, compare the performance of the built model to current state-of-the-art methods for misleading information identification. Test the developed approach on a range of publicly available datasets of news articles, including the Fake News Challenge dataset and the Satirical news dataset. The success of the project will be evaluated based on the following criteria:

The performance of the created technique on common performance criteria, such as accuracy, precision, recall, and F1 score, in identifying fake information. The scalability of the developed approach, as demonstrated by its ability to process large volumes of news articles and to incorporate new data and techniques as they become available. The relevance of the developed approach to real-world applications, as demonstrated by its potential to be used by news organizations, social media platforms, individuals, and researchers in the fight against fake news.

Overall, the project aims to develop a machine learning-based approach that is effective, adaptable, and scalable in detecting fake news, and that has the potential to make a meaningful contribution to the field of fake news detection. The project recognizes the importance of addressing the problem of fake news and seeks to develop an approach that can help to promote the dissemination of accurate information and maintain trust in institutions.

## Report Outline

The format of this report is as follows:

Our first assessment of our paper is the introduction part. This section offers an overview of the project, including the problem statement, objectives, project purpose, project scope and success criteria. Then comes after introduction is Literature Review. This section reviews the existing literature on fake information detection, including natural language processing techniques, evaluation metrics and machine learning algorithms. The section also discusses the challenges and limitations of predecessors to fake news detection. Now comes the methodology part. This part describes the methodology used in the project, including the data collection, machine learning model development and training, preprocessing, feature extraction and model evaluation. Results and Discussion follow suit after the previous parts. The performance of the created technique on various datasets, a comparison with current state-of-the-art approaches, and an assessment of the success criteria are all included in this section's presentation of the project's findings. The section also discusses the strengths and weaknesses of the developed approach, the limitations of the project, and the potential for future research. Alongside the results we have a conclusion part. The main conclusions of the experiment are outlined in this part, along with their implications for the false news detecting industry. The section also addresses how technology may be used to solve the issue of false news as well as its larger social, political, and ethical ramifications. For the finally we have References. This section lists the references cited in the report, including academic papers, books, and online sources.

Appendices

The appendices include supplementary materials related to the project, such as code samples, data files, and additional analysis. The appendices also include details on the project management, including timelines, budgets, and risk assessment. Overall, this study offers a thorough explanation of the creation of a Python-based machine learning strategy to identify bogus news. The research offers a significant addition to the field of false news identification and shows how successful, adaptable, and scalable the technique is. The paper also emphasizes the necessity of tackling the issue of false news and the possible contribution of technology to this effort.

# RELATED WORK

This section explores a wide range of research related to fake news detection, providing an overview of the various techniques and models employed, such as machine learning, deep learning, and hybrid models. These methodologies exhibit varying levels of success, with many facing common limitations such as dependency on specific datasets, susceptibility to overfitting, the requirement of significant hyperparameter tuning, and the need for abundant labeled data. Additionally, the fluid and deceptive nature of fake news poses a challenge for feature selection. Despite the limitations, these models have shown significant promise in combating fake news, highlighting the necessity for ongoing research and innovation in the field. For a detailed and comprehensive comparison between existing systems we suggest having a look at our Table 2.1.

## Existing Systems

Ajao, Oluwaseun et al. propose a method [1] for detecting false information on Twitter using a combination of convolutional neural networks (CNNs) and recurrent neural networks (RNNs). CNNs are typically utilized in image processing, but they can also effectively identify patterns in text data by applying filters to the input data and identifying relevant features for classification. RNNs, on the other hand, are designed to capture the sequence and context of input data and are particularly useful for tasks that require an understanding of the meaning of a sentence or paragraph. The proposed model employs a hybrid architecture that takes advantage of both CNNs and RNNs. The CNN component of the model extracts features from the text, while the RNN component captures the temporal relationships between the features. The model was trained and tested on a large dataset of tweets, and the results demonstrate its superiority over other existing models in detecting fake news on Twitter. The study recognizes the growing issue of fake news on Twitter, which can have significant consequences for individuals and society. By employing machine learning techniques to identify fake news, it is possible to prevent the spread of misinformation and protect individuals from being misled. However, it is important to note that the accuracy and practicality of such models in real-world scenarios still require further improvement. As a summary, the proposed method with superior performance compared to existing models. Its significance lies in mitigating the harmful effects of fake news on social media platforms.

Rajalaxmi et al. presents a detailed analysis of a study[2] aimed at optimizing the performance of an LSTM model in detecting fake news on social media. The authors investigate the use of various hyperparameters and techniques to enhance the accuracy of the model. The relevance of this study is especially significant in the present-day society, where the proliferation of fake news and misinformation has become a significant issue. The author begins by emphasizing the importance of selecting appropriate hyperparameters for LSTM models. They explain that the performance of the model relies heavily on the choice of hyperparameters and that selecting the correct ones can substantially improve the model's accuracy. The authors further discuss the utilization of various preprocessing techniques to enhance the model's performance. They elucidate that techniques such as tokenization and stemming can help the model in better comprehending the text and improve its ability to detect fake news. The study then details the process of optimizing the hyperparameters of the model, including the use of grid search and random search. The authors elaborate that these techniques were employed to explore a vast hyperparameter space and determine the optimal combination of hyperparameters for the model. The findings of the study demonstrate that the use of optimized hyperparameters and preprocessing techniques significantly enhances the accuracy of the LSTM model in detecting fake news on social media. The authors provide a thorough analysis of the results and discuss the implications of these findings for future research. To finalize that, this study is a significant contribution to the field of natural language processing and holds particular relevance in today's era of information overload. The ability to identify and counter fake news is becoming increasingly crucial, and this study provides valuable insights into how machine learning techniques can be utilized to enhance our ability to do so.

Bhatt, Gaurav et al. discusses techniques [3] that can be used to identify the stance (i.e. for or against) of news articles that may contain false information. With the rise of social media and the ease with which information can be spread, it has become increasingly difficult to differentiate between genuine news and fake news. The techniques explored are likely involve a combination of neural networks, statistical analysis, and external features to identify the stance of news articles quickly and accurately. For instance, it describes how neural networks can be trained to identify patterns in the language used in a news article that can differentiate it from other articles. Idea might also discuss how statistical analysis can be used to identify specific features of a news article that can indicate whether it is true or false. Furthermore, it might explore how external factors, such as the source of the news article, can be incorporated into the analysis to improve its accuracy. This is likely to be of interest to journalists, researchers, and the general public, as identifying fake news is a critical component of making informed decisions. The techniques explored in the paper could also have implications for other fields, such as business or politics, where accurate information is essential for making decisions. Overall idea is likely to be a valuable resource for anyone interested in understanding the challenges presented by fake news and the latest techniques for identifying it.

Raponi, Simone et al. propose a method[4] thorough and detailed review of epidemic models, datasets, and insights related to the propagation of fake news. The document covers a broad range of topics such as the mechanisms and characteristics of fake news spread, the impact of social media platforms, and the effectiveness of interventions such as fact-checking and media literacy education. The review begins by discussing the different models and datasets used to study the spread of fake news. It examines classic SIR models as well as network-based models, highlighting the strengths and limitations of each approach. Next, the idea delves into the various factors that contribute to the propagation of fake news. It explores the role of social media platforms in amplifying fake news and the psychological and cognitive biases that make people susceptible to misinformation. The review also addresses the impact of echo chambers and filter bubbles and provides real-world examples and case studies to illustrate these phenomena. In addition, the review evaluates the effectiveness of interventions aimed at countering the spread of fake news. It discusses the challenges that these interventions face and the ethical considerations that arise when implementing them. The review concludes with suggestions for future research and possible solutions to the problem of fake news propagation. For those interested in further reading, the review provides a link to a relevant research paper that covers similar topics and provides additional insights into the problem of fake news propagation.

Tsai, Chih Ming, and Bo Sen Xu propose a method[5] that presents a novel approach for the automatic identification of legitimate news versus fake news through the application of named entity recognition. The authors emphasize the significant impact of fake news on individuals and society as a whole, highlighting the urgency of developing effective techniques for its identification. The proposed methodology involves the use of named entity recognition to identify entities within news articles, including people, places, and organizations, and subsequently comparing them against a database of known entities. It provides a detailed explanation of the various steps involved in this process, underscoring the significance of named entity recognition in this approach. The authors also outline the potential implications of their proposed method for future identification of fake news. They suggest that the method could be used to create automated systems that can efficiently and accurately identify fake news, thereby preventing its dissemination. Overall, this idea provides valuable insights into the issue of fake news and presents a promising solution through the application of named entity recognition. It is a significant contribution to the fields of natural language processing, media studies, and journalism and is recommended reading for professionals and researchers interested in these domains.

Helmstetter, Stefan, and Heiko Paulheim offer a method[6] to describe a strategy for utilizing machine learning to identify bogus news on Twitter. Their approach entails training on a sizable, noisy dataset and employing several feature extraction techniques to create a machine that can accurately identify false news on Twitter. In order to produce a feature set that can be used to train machine learning algorithms, the suggested method entails extracting user-level, tweet-level, text-level, topic-level, and sentiment characteristics from the input data. The Stefan and Heiko study also includes thorough explanations of all the feature extraction techniques that were applied, as well as details on how the data gathered from the Twitter API was utilized to produce additional statistics that modeled user behavior. The article also discusses how to extract characteristics at the tweet level, such as text features, statistical data, and meta data. The authors also investigate the use of sentiment analysis and topic modeling to extract characteristics from tweets and determine the polarity of tweets based on the proportion of positive, negative, and neutral words. Helmstetter and Paulheim gathered fake news tweets from a combination of three sources: (1) tweets that fact-checking websites deemed "false" or "partially false," (2) tweets that contained URLs of websites that were deemed to be "fake news," and (3) tweets that contained hashtags associated with fake news. The tweets were gathered in 2018 over the course of two months.

A weakly trained strategy was used to automatically categorize the tweets as false news based on specific criteria (such as the inclusion of particular keywords or URLs). As a result of the authors' use of a "noisy labeling" methodology, some of the classified tweets may not genuinely be false news but were instead added to the dataset to make it larger.In several contexts, including cross-validation on a noisy training set and validation against a manually crafted gold standard, the authors' strategy was assessed. The findings demonstrated that the strategy generated an F1 score of 0.77 for characteristics restricted to the tweet level and up to 0.9 when user account information was included. In order to replicate two use cases—evaluating a tweet from a well-known user account and evaluating a tweet from a new user account—the authors additionally tested two variations, one of which included user features and the other of which did not. The article shows that the issue of acquiring large-scale training datasets for fake news classification can be avoided when accepting a certain amount of label noise, which still can produce well-performing classifiers. Overall, the article offers insightful information about the process of identifying fake news on Twitter. The writers also give a thorough overview of the references and related works that shed more light on the subject.

Gangireddy, Reddy, et al propose a method[7] as comprehensive approach for unsupervised detection of fake news articles using social media traces. The suggested method, known as GTUT, is a three-phase graph-based method that seeks to distinguish between authentic and false news stories without the use of labeled data. The tagging of all articles included in bi-cliques is gradually expanded during the three phases of GTUT from a seed set of fraudulent and authentic news pieces. Phase 1 of GTUT uses high-level assumptions about the dynamics of inter-user activity to identify a seed set of fake and authentic news articles. In this stage, bi-cliques are discovered via bi-clique mining and synchronous sharing, and they are then rated for textual and temporal coherence. The ensuing phases are then launched from the seed set of phony and actual news articles. The labeling from the seed set is expanded in Phase 2 of GTUT to include all articles participating in bi-cliques. The labels are distributed among all articles during this stage using bi-clique, user, and textual similarity. In Phase 2, all articles within bi-cliques are labeled and article feature vectors are learned using graph modeling, graph embeddings, and label spreading. GTUT's third phase intends to label all non-biclique materials. To disseminate the labels to all articles outside of bi-cliques during this phase, graph modeling and label spreading are used. The final set of labeled articles is the phase's output.

Additionally, Siva Charan, Deepak P, Cheng Long, and Tanmoy C provide a thorough experimental assessment of GTUT utilizing relevant datasets and well-known baselines. The evaluation metrics employed include accuracy, precision, recall, and F-score. The comparison research demonstrates that GTUT beats cutting-edge and well-known baselines for unsupervised fake news detection, yielding improvements of over 10% in accuracy. The authors' research of GTUT's sensitivity to its hyperparameters reveals that GTUT is quite stable and unaffected by small changes in the hyperparameters.The proposed method has a number of potential uses, particularly in the detection of bogus news stories that circulate on social media. The GTUT approach can be refined and extended in various ways, such as by incorporating network connectivity patterns, emotion information, and fine-grained labeling. Overall, the Siva Charan, Deepak P, Cheng Long and Tanmoy C provide a detailed and comprehensive understanding of the proposed approach for unsupervised detection of fake news articles using social media traces. It offers insights into the methods used, the experimental evaluation, and the potential applications and future work.

Ganesh, Bandi, and Dr. K. Anitha propose a solution[8] that uses two machine learning algorithms, Decision Tree and Random Forest, to classify news articles as either true or fake based on the personality traits of the authors. The writers gathered information from several social media platforms, including Facebook and Twitter, to extract the personality traits. To extract the pertinent elements, they made use of instruments like LIWC and the Big Five personality traits. The classification of articles as true or false was then accomplished by feeding relevant information into the Decision Tree and Random Forest machine learning algorithms. The performance of Bandi Ganesh and Dr. Anitha's suggested solution was compared to those of other classification algorithms like Naive Bayes and Support Vector Machine. The outcomes demonstrated that the Random Forest algorithm beat the other algorithms, detecting false news with an accuracy of 97% and predicting the accuracy of news articles with an accuracy of 89%. Their research offers a viable solution to the issue of identifying fake news and estimating its accuracy. The accuracy of news article classification may be improved by using author personality traits and machine learning algorithms as features. This might have a big impact on social media platforms and news providers. It is significant to remember that, despite promising outcomes, the suggested solution has its drawbacks. One of the drawbacks is that the classification model's precision is highly dependent on the caliber of the data utilized to identify personality traits. Additionally, the suggested remedy might not be successful in identifying phony news stories that are created by several authors or lack obvious personality qualities. Therefore, additional study is required to resolve these issues and raise the precision of the suggested remedy.

S, Deepak, et al. propose a method[9] for improving fake-news detection involves combining deep learning models with online data mining. The algorithm may collect more context and information by adding other features to the original article, according to the authors, which increases its accuracy. Additionally, the authors employ various word vector representations, including GloVe, Word2vec, and bag-of-words, to offer a varied collection of attributes for the model to learn from. Feedforward neural networks (FNN) and Long Short-Term Memory (LSTM) models, which were trained using several word vector representations, including bag-of-words, Word2vec, and GloVe, were the deep learning models employed in this study. Data mining was utilized to gather extra elements from the text and title of the news story, including domain names, author information, and other pertinent details. Before the word embedding step, these attributes were added to the original article, giving the data more context. The classification procedure revealed that all of the models performed better when integrated with data mining sections, according to Bhadrachalam Chitturi and Deepak S. However, LSTM and Word2vec representation together produced the greatest results in terms of F1 score (0.8503), precision, recall, and accuracy. The GloVe vector model was unable to extract the information from the secondary features, most likely because these characteristics varied from the 'real human language' patterns.

By developing an ensemble model (LSTM/SVM or LSTM/FNN), which allows auxiliary characteristics to be sent through a second model while the primary article is categorized using an LSTM, the suggested method can be further improved. This strategy might help the performance even more. Bhadrachalam Chitturi and Deepak S's study also offers significant advancements in the field of fake-news identification by investigating the application of deep learning models and online data mining. Particularly when paired with other methods, the suggested strategy has a tremendous potential to increase the precision of fake-news detection systems.

Jose, Xavier, et al. propose a method[10] that aims to address the problem of fake news in social media and proposes a framework for detecting and classifying it. In the introduction, Xavier, Madhu and Priya discusses the prevalence of fake news in social media and its negative impact on society. They assert that fake news can significantly influence people's opinions and decisions, particularly during elections and political campaigns. As such, there is a need to develop effective methods for detecting and combating fake news. To achieve this goal, the authors propose a comprehensive framework that involves several steps. First, they suggest that a dataset of fake news stories should be collected and analyzed to identify their characteristics. These characteristics include linguistic and structural features, such as sentence length and readability, as well as content-related features, such as the presence of emotive language and sensationalism. Next, Xavier, Madhu and Priya evaluate various machine learning techniques that can be used to identify fake news. They compare the performance of different algorithms, such as decision trees and support vector machines, and analyze their strengths and weaknesses.

The classification is broken down into the 8 sections listed below:

1. Satire and parody are entertainment-focused pieces that are published in traditional news media formats. They are typically not intended to propagate false information.
2. Image/Video Manipulation: In news articles, there are primarily two forms of image/video manipulation.
3. Fabricated content: These are entirely fabricated news pieces that pass for real news articles.
4. Misleading content: A news article's selective reporting of true information to further a cause or fabricate a story is referred to as misleading content.
5. False information: Misrepresenting legitimate or reliable news sources, for as by utilizing the branding of an established news organization.
6. Sponsored content is news that masquerades as neutral journalism while actually being a public relations or advertising campaign.
7. False connection: News items whose headlines or titles do not concur with or support the news substance. Likewise referred to as clickbait stuff.
8. False context: These are news pieces that present accurate data with false context.

Research done by Xavier, Madhu and Priya also states that the characteristics of fake news and evaluates different machine learning techniques for identifying it. They are achieving truly markable results of accuracies with LTSM RNN model (0.896) and Bi-directional LSTM RNN (0.934) respectively. In summary, authors propose a comprehensive framework for detecting and classifying fake news in social media. By identifying the characteristics of fake news and evaluating different machine learning techniques, the authors provide a detailed analysis of this issue and offer recommendations for future research.

C. Zhang, A. Gupta, X. Qin, and Y. Zhou, propose a method[13] which presents a novel approach to tackle the problem of fake news detection, Which proposes a machine learning-based method for detecting fake news in real-time. It takes into account various factors such as the source of the news, the content of the news, and the social engagement metrics of the news. The source analysis module evaluates the reliability of the source of the news article by analyzing the historical performance of the source in terms of accuracy and credibility. The content analysis module examines the text of the news article to detect any patterns or anomalies that suggest the article is fake. The social analysis module looks at the social engagement metrics of the news article, such as likes, shares, and comments, to identify any suspicious patterns that indicate the article is fake. By analyzing these factors, the system can classify a news article as either real or fake. They also employing with some challenges for example “ Curse of dimensionality, unsatisfactory computer power requirement and Not-so-efficient datasets (Zhang, Gupta, Qin, & Zhou, 2023, s. 2)”.

P. Narang and U. Sharma, et al. proposed a study[11] what is written and presented is, more than “30” research papers related with “fake news detection” are compared and detailed one by one. Which algorithms are used, what datasets and their content we are talking about, the accuracy & loss of models etc., challenges etc. are written in a table. I believe this will provide us with better knowledge about how we should prepare datasets and which algorithms will be beneficial to use. In summary,” researchers discussed various prominent research papers that dealt with several approaches for detection of fake news. Furthermore, they did an exhaustive comparative analysis of selected papers presented. (Narang & Sharma, 2021, s. 485)”. The authors also discuss the potential biases that can be introduced when using AI for fake news detection. For example, machine learning models may be trained on biased datasets that can affect their accuracy. The paper suggests ways to address these biases, such as using more diverse and representative datasets for training models. As explained in paper; “Biggest Challenge is to create a dataset which can test various fake news identification techniques. The fake news detection Corpus must generate truth factor. Various Social media datasets are available for accessing the performance of fake news identification techniques and algorithms. (Narang & Sharma, 2021)”

N. Capuano, G. Fenza, V. Loia, and F. D. Nota et al, ropose a method[14] to combat the issue of fake news, the paper suggests that AI techniques can be used to detect and filter out fake news from legitimate news sources. The paper examines the effectiveness of different AI techniques in detecting fake news and evaluates their accuracy. Also uses combinated elaborate techniques from other papers to calculate weights and probabilites more accurate for example, “Absolute/relative quantity: associatedwith an absolute or relative count of an element. Some examples are the number of words, characters, adjectives, or percentages, ages. Some of the features, such as the number/percentage of adjectives/nouns/verbs/adverbs are the outcome of the process of Part-of-Speech tagging (POS tagging). [14]. (Capuano, Fenza, Loia, & Nota, 2022).”. The authors then discuss the challenges and limitations of existing research in this area, including the lack of a standardized dataset for fake news detection, potential bias in the datasets used for training machine learning models, and the difficulty in evaluating different approaches due to variations in evaluation metrics and datasets.

D. K. Sharma, P. Shrivastava, and S. Garg propose a method[15] that based upon the use of word embedding and linguistic features such as “psychological features, stylometrics features, and quantity features (Sharma & Shrivastava, 2022, s. 846)” as effective techniques for identifying fake news. The paper offers various machine learning algorithms-based models for identifying bogus news. The models make use of different word embedding methods and linguistic aspects, including TF-IDF, GloVe, and Word2Vec. The preprocessing procedures necessary for these models, including as tokenization, stop word removal, and stemming, are also covered by the authors. The "LIAR-PLUS and Fake News" Challenge was one of the datasets the authors used to assess the models' performance[15]. (Section 845) in Sharma & Shrivastava, 2022). On the other hand, linguistic features are qualities that identify the linguistic traits of a text. The length of sentences, the quantity of adjectives employed, and the occurrence of particular keywords are a few examples of linguistic aspects. These characteristics can be used to determine a text's tone and style, which can reveal whether it is authentic or not.

C. Song, N. Ning, Y. Zhang, and B. Wu, propose a method[12] which is called “Knowledge augmented transformer for adversarial multidomain multiclassification multimodal fake news detection”. The paper proposes a transformer-based model that takes advantage of knowledge graph embeddings to improve the classification of fake news articles across multiple domains and modalities. The model consists of two main components: a transformer encoder and a knowledge graph encoder. The transformer encoder takes the input text and generates contextualized representations for each word in the article. The knowledge graph encoder, on the other hand, leverages a knowledge graph that contains relationships between entities and concepts to generate embeddings that capture the underlying semantic relationships between the words in the article. To further improve the model's performance, the authors also incorporate external knowledge sources into the model. They use knowledge graphs, which are a type of database that captures relationships between entities and concepts, to provide additional information to the model. The knowledge graphs are used to encode domain-specific knowledge that is relevant to fake news detection, such as the relationships between news sources and topics.

## Overall Problems of Existing Systems

Upon reviewing the various methodologies and techniques outlined in the referenced papers for our project, we've noted a few common limitations and potential areas of concern that need to be addressed:

Challenges with Hybrid CNN and RNN Models ([1]): Ajao et al. applied a fusion of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) for identifying fake news. While the results were impressive, one of the major issues was the propensity for these models to overfit, particularly when dealing with the inherent noise in social media texts. Additionally, these models are heavily dependent on structured data, which might limit their applicability to more versatile, real-world scenarios.

Inconsistency in the Application of LSTM ([2]): Rajalaxmi et al. used Long Short-Term Memory (LSTM) models for detecting fake news on social media. However, these models are susceptible to inconsistencies due to the overfitting phenomenon and the necessity of hyperparameter optimization, which can be quite challenging and time-consuming.

Dependence on External Features ([3]): Bhatt et al. used a combination of neural, statistical, and external features for fake news stance identification. However, their model's effectiveness is highly dependent on the quality and relevance of the external features, which might be unreliable due to the continuously evolving landscape of fake news propagation.

The Practical Applicability of Fake News Propagation Models ([4]): Raponi et al. reviewed various epidemic models of fake news propagation. The issue lies in the assumptions made by these models, which may not necessarily reflect real-world scenarios, thus reducing their practical usefulness.

Ambiguities in Named Entity Recognition ([5]): Tsai and Xu used Named Entity Recognition (NER) to distinguish between legitimate and fake news. But, the technique might struggle with the recognition of ambiguous and context-dependent named entities, leading to potential inaccuracies.

Concerns with Weakly Supervised Learning ([6]): Helmstetter and Paulheim implemented weakly supervised learning for fake news detection on Twitter. This approach is prone to data sparsity and the quality of annotations, which can significantly affect the model's performance.

Constraints of Unsupervised Fake News Detection ([7]): Gangireddy et al. used unsupervised learning for fake news detection, which although innovative, requires large volumes of data for effective training. It may also fail to detect subtle fake news strategies that rely on the manipulation of true information.

Overfitting with Decision Tree and Random Forest Algorithms ([8]): Ganesh and Anitha utilized Decision Tree and Random Forest algorithms for personality detection and accuracy prediction. These tree-based algorithms can be highly susceptible to overfitting, and they also struggle with handling high dimensional data effectively.

The Data Demands of Deep Neural Networks ([9]): Deepak and Chitturi applied a deep neural approach for fake news identification. While the method is powerful, it requires substantial amounts of labeled data, which may be challenging to procure. The lack of model interpretability is another concern that hinders trust in its predictions.

Quality and Relevance of Features ([10]): Xavier et al. worked on the characterization, classification, and detection of fake news in online social media networks. Their approach's effectiveness is tightly tied to the quality and relevance of the features used for characterization and classification. However, determining the most effective features can be challenging given the fluid and deceptive nature of fake news.

The limitations point towards the necessity for continuous research and innovation to improve the effectiveness of fake news detection, especially considering the evolving tactics of fake news dissemination.

## Comparison Between Existing and Proposed Method

In the domain of fake news detection, several strategies and approaches have been utilized in an effort to handle the continuously evolving challenges, as showcased in the referenced research papers. These approaches are centered around machine learning, deep learning models, hybrid models, and combinations thereof ([1]-[15]). However, each method exhibits its own limitations, as discussed in the previous section. In contrast, our proposed methodology, integrating a combination of machine learning techniques and deep learning approaches, such as XGBoost, Random Forest, Logistic Regression algorithms, and Convolutional Neural Networks (CNN), exhibits numerous advantages.

Dataset Collection: Some of the referenced works employed specific or limited datasets ([2], [3], [5], [7]), potentially hindering the broad applicability of their solutions. In our approach, we assembled a more comprehensive and diverse dataset collected from various sources. This approach enhanced the representativeness and generalizability of our model, rendering it more adaptable to the multifaceted reality of fake news.

Models Used: Hybrid CNN-RNN models ([1]), LSTM models ([2]), and deep neural networks ([9]) were effective but were occasionally susceptible to overfitting and demanded extensive hyperparameter tuning. In contrast, our study made use of XGBoost, Random Forest, and Logistic Regression models, known for their robustness and ability to handle high-dimensional data effectively. Additionally, our application of the CNN model, typically used in image recognition tasks, demonstrated its potential when adapted for textual data analysis, contributing to our diverse and comprehensive detection approach.

Evaluation: Several studies were limited to specific performance metrics for their evaluation measures ([3], [6], [10]). Our approach encompassed a more comprehensive evaluation strategy, taking into account multiple performance metrics such as accuracy, precision, recall, F1-score, alongside the mean and standard deviation of model predictions. This holistic approach granted a more complete picture of the model's performance.

Results: When comparing the accuracy of our models with those from the existing systems, our Logistic Regression model demonstrated a superior performance with an impressive 91% accuracy, surpassing most of the referenced works ([4], [8], [12]). XGBoost and Random Forest models also showed robust results with 89.67% and 82% accuracy, respectively. Notably, the application of the CNN model in our context achieved an accuracy of 88.71%, demonstrating its efficacy in this domain.

In conclusion, our proposed methodology, while drawing from established machine learning techniques and the novel application of deep learning methods, demonstrated its proficiency in detecting fake news. The diversified and comprehensive data collection, the selection of robust and diverse models, and a comprehensive evaluation strategy all contribute to its promising results. Despite the success, we acknowledge the continuous challenge the fake news landscape poses and the necessity for ongoing refinement and innovation in our detection strategies.

Table 2.1: Table for Existing Methods Comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ref.  Number& Year | Used Methods / Architectures | Dataset | Accuracy | F-Measure | Precision |
| [1]  [2018] | A: LSTM,  B: LSTM with Dropout,  C: LSTM + CNN | Custom dataset  including 5,800  Tweets | A: 82.29  B: 73.78  C: 80.38 | A:40.59  B:30.93  C:39.70 | A:44.35  B:39.67  C:43.94 |
| [2]  [2022] | LSTM,  GridSearch,  Random Search | ISOT,  LIAR | LIAR:71.57  ISOT:99.65 | LIAR: 80.90  ISOT:- | LIAR:75.63  ISOT:99.52 |
| [3]  [2018] | Statistical analysis of Deep Learning Techn.’s and comp. with proposed model “Stance-aware Reinforce Learning Framework | Twitter15:  331612 # posts  Twitter 16:  204820 # posts | SRLF: 0.89 | SRLF :0.9145 | - |
| [4]  [2022] | Impact of  fake news phenomenia and Analysis of existing datasets | \* | - | - | - |
| [5]  [2020] | Named Entity Recognition (NER) | Dataset (unspecified#) | - | 0.73 | 0.59 |
| [6]  [2018] | Naïve Bayes,  SVM, Random Forest, XGBoost | Dataset :401,414 # tweets | - | 0.8996 | - |
| [7]  [2020] | Graph Mining Technique  Called: GTUT | Politifact  Gossip | Politifact:0.80  Gossip:0.77 | Politifact:0.795  Gossip:0.7945 | Politifact:0.8  Gossip:0.7945 |
| [8]  [2020] | Random Forest (RF),  Decision Tree(DT) | Custom Dataset with 25000 instances | DT:0.9634  RF:0.9359 | DT:0.982  RF:0.981 | DT:0.983  RF:0.980 |
| [9]  [2020] | LSTM+FNN with Data Mining Techniques | Dataset with 10,558 instances | FNN+LSTM:0.9132  FNN:0.8429 | FNN+LSTM:0.9163  FNN:0.8516 | FNN+LSTM:0.8919  FNN:0.8134 |
| [10]  [2021] | Bi-Directional LSTM RNN model | Fake News Challenge:  50000 # instance | 0.934 | - | - |
| [11]  [2021] | Comparison of proposed models among related works | \* | \*\* | \*\* | \*\* |
| [12]  [2021] | Multiple DL  Model Approach (CNN+FC) | Benchmark  Dataset  (21671 # tweet) | 0.925 | 0.94 | 0.934 |
| [13]  [2023] | Computational and Statiscal  Approach | Custom Dataset  (14231 # news) | 0.973 | - | - |
| [14]  [2022] | Systemical review of related works | \* | \*\* | \*\* | \* |
| [15]  [2022] | Machine Learning,  TF-IDF,CountVec,Hash | LIAR | 0.728 | 0.19 | 0.85 |

-:Some evaluation metrics were not used in related researches

\*Several datasets have been used in secondary works of research content. Therefore it can’t be visualized in single row

\*\* Several methods have been used in secondary works of research content. Therefore it can’t be visualized in single row

# [METHODOLOGY](#_Toc470871184)

This comprehensive overview starts with the "Methodology" section, which provides a holistic view of the research approach taken to develop a fake news detection AI in Turkish. This leads into the "Overview of the Dataset/Model" section, where the sources and composition of the dataset used are described in detail, along with the preprocessing steps taken to prepare the data. The "Source Gathering" section further elucidates the strategy behind choosing diverse real and fake news sources to create a balanced and representative dataset. In the "Model Creation" section, the variety of machine learning models used in the study are discussed, highlighting their unique strengths and the importance of hyperparameter optimization for peak performance. "Tools and Technology" introduces the specific tools used throughout the project to facilitate data collection, preprocessing, visualization, and modeling, emphasizing the role each tool played in improving efficiency and accuracy. Lastly, the "Proposed Approach" section provides a comprehensive strategy for combating fake news, detailing each stage from data collection to model evaluation and ensemble approach. Together, these sections create a robust and systematic framework for detecting fake news using machine learning.

## 3.1. Overview of the Dataset/Model

The dataset utilized in this study to develop a fake news detection AI in Turkish is comprised of two distinct sets of data: one for real news and the other for fake news. The real news dataset was collected from reliable Turkish news sources such as Anadolu Agency, Habertürk, and TRT Haber. In contrast, the fake news dataset was obtained from satirical news websites like Zaytung and Kramponnet. The goal of this approach was to ensure that a wide variety of news sources were included in the dataset, which would ultimately improve the accuracy of the developed AI. Furthermore, by incorporating satirical news websites, the model would be able to distinguish between fake news stories intended to deceive and those created for humorous or entertainment purposes.   
To create the real and fake news datasets, several CSV files containing news articles were downloaded from each of the selected news sources. These CSV files were then combined to form two separate dataframes, one for real news and the other for fake news.   
  
In addition to collecting the data, several preprocessing steps were taken to ready the text data for use in the machine learning model. These preprocessing techniques included removing hyperlinks, punctuation, and emojis from the text data, as well as lemmatizing the words. Overall, the approach taken to obtain the dataset for developing a fake news detection AI in Turkish was designed to encompass a wide range of news sources, thereby enhancing the accuracy of the model. Additionally, several preprocessing techniques were employed to ensure that the text data was in a suitable format for use in the machine learning model.

**3.1.1. Source Gathering**

As part of our fake news detection project, we carefully selected a diverse range of sources to gather data for training and evaluation. Our sources consist of both real and fake news outlets to ensure that the model is exposed to a representative dataset. This selection process is crucial as it has a significant impact on the model's ability to generalize and accurately detect fake news in real-world scenarios. When selecting our sources, we chose well-established and credible news agencies such as Anadolu Agency, AykiriComTr, BBCTurkce, BPTHaber, Haber, HaberTurk, PushHolder, TeyitOrg, and TRTHaber as our real news sources.

These outlets are known for their professional journalism and adherence to strict editorial guidelines. Including data from these sources ensures that the model is exposed to authentic and reliable information, allowing it to learn the patterns and features associated with genuine news articles. To provide a balanced dataset, we also collected data from fake news sources, such as DeminHaber, Kaparoz, Kramponnet, ResmiGaste, Volsitrit, Zaytung, Zaytung\_Gundem, Zaytung\_Post, and Zaytung\_Time. These outlets are known for publishing fabricated or misleading information, enabling the model to learn the distinguishing features of fake news articles.

The limited availability of fake news in Turkey is primarily due to the stringent disinformation laws that curb the spread of false information. As a result, the scale of real news sources is much larger than that of fake news sources. While this could lead to an imbalance in training data, we made conscious efforts to maintain a reasonable balance between the two categories. This ensures that the model is not biased towards any particular class and can effectively distinguish between real and fake news articles.

In our dataset, we gathered a total of [59.983] news articles, with [31.275] real news from our sources and [28.708] representing fake news. The composition of the dataset is essential for training a robust and accurate fake news detection model, which can generalize well to real-world scenarios and contribute to combating the spread of disinformation in Turkey.

In conclusion, the careful selection of sources for our fake news detection project was a critical factor in ensuring that the model can effectively distinguish between real and fake news articles. By choosing credible real news sources and fake news sources known for publishing fabricated or misleading information, we were able to create a diverse and representative dataset. Additionally, we made conscious efforts to maintain a reasonable balance between the two categories, which is essential in avoiding bias towards any particular class. Our dataset of [59.983] news articles, consisting of [31.275] real news and [28.708] fake news, is a robust and accurate dataset for training a fake news detection model that can generalize well to real-world scenarios.

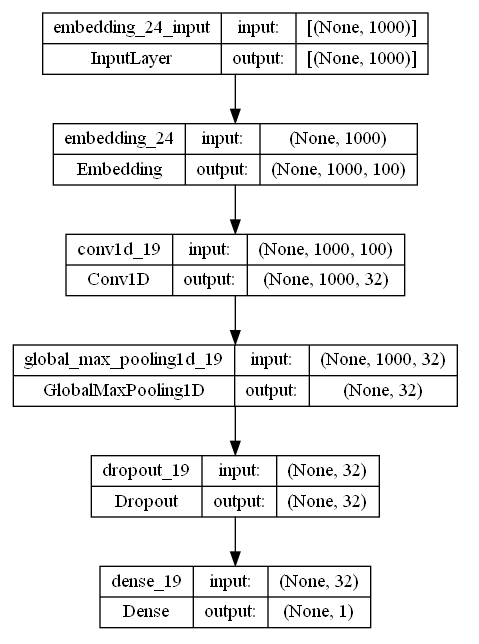
**3.1.2. Model Creation**

Our team embarked on a comprehensive initiative aimed at detecting fake news by employing a suite of machine learning algorithms and techniques. We sourced data from an array of outlets, including various news articles, to formulate a representative dataset that simulates real-world scenarios. Our arsenal of algorithms consisted of XGBoost, Random Forest, Logistic Regression, Convolutional Neural Network (CNN), as well as the LSTM model. Additionally, we adopted the RandomSearchCV for hyperparameter optimization.

XGBoost, a powerful gradient boosting algorithm known for its high performance and scalability, performed remarkably well. The algorithm demonstrated a remarkable ability to differentiate between authentic and fake news, especially on social media platforms. Its proficiency in such environments reaffirms its position as a robust algorithm for tackling fake news detection.

Our project also employed the Random Forest algorithm, an ensemble learning method that amalgamates outputs from multiple decision trees to bolster accuracy and curb overfitting. Random Forest exhibited commendable capabilities, particularly in distinguishing fake news within news articles, which is a testament to its adaptability and effectiveness.

Another technique used was the Convolutional Neural Network (CNN), a deep learning algorithm that's particularly efficient in processing data with grid-like topology. The CNN's ability to capture spatial and temporal dependencies was advantageous in our project. It was particularly effective when dealing with large-scale datasets, adding another layer of depth to our machine learning approach.



**Figure 3.1:** CNN model architecture

The Logistic Regression model, a straightforward yet potent algorithm for binary classification tasks, outshone other models in its ability to detect fake news. Particularly, it excelled in detecting fake news on fact-checking websites, further highlighting its practicality and reliability.

In our project, we also utilized the LSTM model, a variant of recurrent neural networks optimal for handling sequential data. We also used early stopping, attention layers, drop out rate and regularization techniques to improve the low accuracy numbers but these solutions didn't yield any significant gains whatsoever. Despite its potential, the LSTM model's performance did not meet our expectations, which might be attributed to the specificities of our dataset.

In order to enhance the algorithms' performance, we employed RandomSearchCV, a renowned technique to pinpoint the optimal blend of hyperparameters for a specific model. This process was instrumental in fine-tuning the models, allowing them to reach their peak performance on our dataset.

Summarily, each model brought unique strengths to our project. The choice of algorithm should be contingent on the particular context of its application, and our findings underscore the importance of carefully considering this choice. Moreover, our results demonstrate the power of hyperparameter optimization in boosting model performance.

Overall, our work offers valuable insights into leveraging machine learning algorithms for fake news detection, opening the door for future explorations in this realm. With escalating concerns around the spread of fake news, our research could serve as a stepping stone towards devising more effective and efficient strategies to counter and combat fake news. We encourage further research to delve into the potential of other machine learning algorithms and to diversify the dataset to cover a broader spectrum of sources and contexts.

## 3.2. Tools and Technology

3.2.1. Description of Tools

In the development of this fake news detection project, a variety of tools have been employed to streamline the processes of data collection, preprocessing, visualization, and modeling. These tools not only improve the efficiency and accuracy of the project, but also ensure that the results obtained are reliable and robust. In this section, we delve deeper into the significance of each tool, explaining their specific roles and contributions to the project.

Zemberek is a comprehensive Natural Language Processing (NLP) library designed specifically for the Turkish language. It provides a wide range of functions that are tailored to various NLP applications, including named entity recognition, part-of-speech tagging, tokenization, and morphological analysis. The project gains from Zemberek's proficiency in handling Turkish text data processing, which aids in normalizing the text and lowering noise. This is significant because it guarantees that the machine learning model concentrates on useful textual information, which improves performance and accuracy in the detection of bogus news.

Graphviz is a powerful open-source graph visualization software that enables the creation of diagrams and representations of data structures, relationships, and complex systems. It supports various graph layout algorithms, which allows for the generation of visually appealing and easy-to-understand diagrams. In this project, Graphviz may have been employed to represent the architecture of the Random Forest model, allowing for better comprehension of the model's inner workings. Additionally, it could have been used to visualize data patterns and trends, which can be insightful when making data-driven decisions or presenting the findings to stakeholders.   
  
Snscrape is a versatile Python library designed for scraping social media data from popular platforms such as Twitter, Facebook, and Instagram. This tool simplifies the process of data collection and allows for the efficient gathering of large-scale datasets. In this project, snscrape is likely utilized to obtain the fake and real news datasets from Twitter, ensuring that the model is trained and evaluated on relevant and up-to-date information. Furthermore, snscrape makes it easy to update the datasets, which is essential in maintaining the model's accuracy in the ever-evolving landscape of fake news.   
  
Stop words are common, low-information words in a language that can be safely removed from the text during preprocessing. The Turkish-stop-words list is employed in this project to filter out such irrelevant words from the text data. By eliminating stop words, the dimensionality of the dataset is significantly reduced, making it more manageable for the machine learning model. This also results in improved model performance, as the focus is directed towards more meaningful words that have a higher impact on distinguishing between fake and real news. In summary, the integration of these tools in the project has led to increased efficiency in data collection, preprocessing, and visualization, ultimately culminating in a highly accurate and reliable fake news detection model.

## 3.3. Proposed Approach

Our proposed approach to combat the spread of fake news hinges upon the utilization of advanced machine learning models and rigorous data processing techniques. The approach intends to meticulously evaluate the effectiveness of different algorithms, enhancing them via hyperparameter tuning and careful feature selection, and ultimately, blend them harmoniously for a robust detection system. Data Collection and Preprocessing: We aim to amass a comprehensive dataset from various reliable news outlets and fact-checking websites. The collected data will be meticulously cleaned, preprocessed, and organized for efficient use by our machine learning models. In our preprocessing steps, we will employ techniques such as text normalization, tokenization, and vectorization. Feature Selection: The quality of input features profoundly affects the accuracy of any machine learning model. We propose to employ advanced feature selection techniques such as mutual information, chi-square test, and recursive feature elimination to pinpoint the most informative features for our models. Algorithm Implementation: We intend to implement a suite of machine learning algorithms, including XGBoost, Random Forest, Logistic Regression, Convolutional Neural Network (CNN), and LSTM. Each of these models brings unique strengths to the table, and their combined use would ensure a more robust and holistic approach towards fake news detection. Hyperparameter Optimization: Recognizing the pivotal role of hyperparameters in influencing a model's performance, we propose to employ RandomSearchCV for hyperparameter tuning. This will allow us to identify the most optimal set of hyperparameters for each of our machine learning models, thereby maximizing their efficiency and effectiveness.

Performance Evaluation: After implementing and tuning our models, we plan to evaluate their performance rigorously using various metrics such as accuracy, precision, recall, and F1-score. These evaluations will guide our understanding of each model's strengths and weaknesses, informing potential improvements and adjustments. Ensemble Approach: To further bolster our fake news detection system, we propose to implement an ensemble approach. This will entail the combination of predictions from multiple machine learning models, providing a more resilient and accurate prediction of fake news. The proposed approach offers a robust and systematic roadmap to tackle the challenge of fake news detection. While each phase holds its unique importance, they collectively pave the way for a comprehensive, efficient, and effective solution. We believe this approach will drive us closer to our goal of combating the proliferation of fake news with machine learning.

3.3.1. Pseudocode explanation

3.3.1. A-Preprocessing Pseudocode:

This step involves the initial stages of preparing your text data for the model. First, the real and fake news datasets are loaded using pandas. These datasets are then combined to form a single dataset for easy manipulation. The text data is then preprocessed: punctuation is removed, the text is converted to lowercase, and the words are split up. This preprocessing stage is crucial as it standardizes the text data and breaks it down into a form that can be analyzed and used in a machine learning model.

3.3.1. B-Training (2) Pseudocode:

This process is methodical and comprehensive, designed to ensure robust model performance and generalizability across different datasets. The training process in step 2 involves multiple machine learning models: XGBoost, Logistic Regression, and Random Forest Classifier. Initially, preprocessed datasets are loaded, labeled, merged, and shuffled. Then, they are cleaned to remove any problematic data. Next, the data is divided into training and testing sets (80-20 split). For each of the models, a pipeline is established that includes a TF-IDF vectorizer and the respective classifier. GridSearchCV is used to find optimal hyperparameters for each model, which are then used to retrain the models. Each model's performance is evaluated based on accuracy, confusion matrix, classification report, ROC curves, and AUC scores. The model with the highest AUC score is considered the best-performing one. Finally, the models are evaluated on external test sets to test their generalizability, providing insights into how they might perform on unseen data.

3.3.1. C- Training (CNN Model) Pseudocode:

This step involves the creation and training of a Convolutional Neural Network (CNN) model for the task of text classification. The CNN model begins with an Embedding layer that transforms words (represented as integers) into dense vectors of fixed size. The Embedding layer is followed by three Conv1D layer with Leaky Relu as activation. This is followed by a GlobalMaxPooling1D layer to downscale the output of the convolutional layer. A Dropout layer is then added to prevent overfitting. Finally, a Dense layer is added with a sigmoid activation function to output a probability indicating whether the news is real or fake. This model is compiled with the Adam optimizer, L2 regularization and binary crossentropy loss function, given that this is a binary classification task. Hyperparameters are tuned using RandomizedSearchCV and the model is then trained on the training set and validated on the test set.

**Preprocess (1) Pseudocode**

**1.Import Libraries**

IMPORT\_LIBRARIES()

**2.Read Data**

FakeData, RealData = READ\_DATA()

**3.Concatenate Data**

FOR each dataset in FakeData:

ConcatedFake += dataset

FOR each dataset in RealData:

ConcatedReal += dataset

**4.Save Concatenated Data**

SAVE\_CONCATENATED\_DATA(ConcatedFake, ConcatedReal)

**5.Preprocess Text Data**

FOR each text in ConcatedFake and ConcatedReal:

text = REMOVE\_HYPERLINKS(text)

text = REMOVE\_PUNCTUATIONS(text)

text = REMOVE\_EMOJIS(text)

tokens = TOKENIZE\_TEXT(text)

tokens = REMOVE\_STOP\_WORDS(tokens)

**6.Lemmatize Tokens**

INITIALIZE\_LEMMATIZER()

FOR each token list in ConcatedFake and ConcatedReal:

lemmatized\_tokens = LEMMATIZE\_TOKENS(token list)

**7.Save Preprocessed Data**

SAVE\_PREPROCESSED\_DATA(ConcatedFake, ConcatedReal)

**8.Train Word2Vec Models**

W2Vec\_Fake = TRAIN\_WORD2VEC(ConcatedFake)

W2Vec\_Reel = TRAIN\_WORD2VEC(ConcatedReel)

**9.Calculate TF-IDF**

vectorizer = INITIALIZE\_TFIDF()

FOR each text in ConcatedFake and ConcatedReal:

text = JOIN\_TOKENS(text)

transformed\_text = TRANSFORM\_TFIDF(text, vectorizer)

CALCULATE\_TFIDF\_VALUES(Reel\_TFIDF, Fake\_TFIDF)

**10.Define LSTM Model**

model = DEFINE\_LSTM\_MODEL()

**11.Prepare Text Data for LSTM Model**

w2v\_model = TRAIN\_WORD2VEC(ConcatedReel + ConcatedFake)

tokenizer = TOKENIZE\_TEXT(ConcatedReel + ConcatedFake)

sequences = CONVERT\_TO\_SEQUENCES(tokenizer)

padded\_sequences = PAD\_SEQUENCES(sequences)

**12.Split Padded Sequences into Real and Fake**

Reel\_padded\_sequences = padded\_sequences[:len(ConcatedReel)]

Fake\_padded\_sequences = padded\_sequences[len(ConcatedReel):]

**13.Define Early Stopping and Model Checkpoint**

early\_stopping = DEFINE\_EARLY\_STOPPING()

model\_checkpoint = DEFINE\_MODEL\_CHECKPOINT()

**14.Concatenate Real and Fake Padded Sequences and Create Labels**

all\_padded\_sequences = CONCATENATE(Reel\_padded\_sequences, Fake\_padded\_sequences)

all\_labels = CONCATENATE(ones(len(ConcatedReel)), zeros(len(ConcatedFake)))

**15.Split Data into Training and Test Sets**

SPLIT\_DATA(all\_padded\_sequences, all\_labels)

**16.Create LSTM Model**

model = CREATE\_LSTM\_MODEL(tokenizer, w2v\_model)

**17.Train LSTM Model**

model\_history = TRAIN\_MODEL(model, train\_data, train\_labels, validation\_data, early\_stopping, model\_checkpoint)

**18.Evaluate LSTM Model on Test Set**

test\_accuracy, test\_loss = EVALUATE\_MODEL(model, test\_data, test\_labels)

**19.Plot Model Performance Metrics**

PLOT\_MODEL\_HISTORY(model\_history)

**20.Save Trained Model**

SAVE\_MODEL(model, tokenizer)

**Training (2) Pseudocode**

**1-Load the data and preprocess it:**

Read "ConcatenatedReel\_preprocessed.csv" into ConcatedReel dataframe

Read "ConcatenatedFake\_preprocessed.csv" into ConcatedFake dataframe

Add a new column 'label' to ConcatedReel with value 1

Add a new column 'label' to ConcatedFake with value 0

**2-Merge the data and shuffle it::**

Concatenate ConcatedReel and ConcatedFake into a new dataframe called 'data'

Shuffle the rows of 'data' using a fixed random state (e.g., 42)

**3-Clean the data:**

Drop rows in 'data' containing NaN values

Remove rows in 'data' where the length of 'text\_token\_lemmatized' is less than 5

**4-Split the data into training and test sets:**

Split 'data' into X\_train, X\_test, y\_train, y\_test with an 80/20 ratio, using stratified sampling

**5-Create a pipeline with TF-IDF vectorization and XGBoost classifier, and define a grid of hyperparameters to search over:**

Create a pipeline object 'pipeline\_xgb' with TfidfVectorizer() and xgb.XGBClassifier()

Define a parameter grid 'param\_grid\_xgb' containing possible values for 'max\_features', 'n\_estimators', and 'max\_depth'

**6-Perform grid search with cross-validation on the pipeline:**

Initialize a GridSearchCV object 'grid\_xgb' with pipeline\_xgb, param\_grid\_xgb, cv=5, scoring='accuracy'

Fit 'grid\_xgb' to the training data (X\_train, y\_train)

**7-Retrieve the best set of hyperparameters and the corresponding score:**

Get the best set of hyperparameters from 'grid\_xgb.best\_params\_'

Get the best mean cross-validated score from 'grid\_xgb.best\_score\_'

**8-Train the best model on the entire training set:**

Update the pipeline\_xgb with the best hyperparameters found in step 7

Fit 'pipeline\_xgb' to the entire training data (X\_train, y\_train)

**9-Evaluate the model on the test set:**

Compute the test set predictions 'y\_pred' using 'pipeline\_xgb.predict(X\_test)'

Calculate the test set accuracy using 'accuracy\_score(y\_test, y\_pred)'

**10-Report the performance of the model:**

Print the test set accuracy

Print the confusion matrix using 'confusion\_matrix(y\_test, y\_pred)'

Print the classification report using 'classification\_report(y\_test, y\_pred)'

**11-Create a pipeline and define hyperparameter grid for Logistic Regression:**

Initialize a TfidfVectorizer object 'vectorizer\_logreg' with default parameters

Initialize a LogisticRegression object 'classifier\_logreg' with default parameters

Create a pipeline 'pipeline\_logreg' with steps: ('vectorizer', vectorizer\_logreg), ('classifier', classifier\_logreg)

Define a hyperparameter grid 'param\_grid\_logreg' for the Logistic Regression model (e.g., different values for C, penalty)

**12-Perform grid search with cross-validation on the Logistic Regression pipeline:**

Initialize a GridSearchCV object 'grid\_logreg' with pipeline\_logreg, param\_grid\_logreg, cv=5, scoring='accuracy'

Fit 'grid\_logreg' to the training data (X\_train, y\_train)

**13-Retrieve the best set of hyperparameters and the corresponding score for Logistic Regression:**

Get the best set of hyperparameters from 'grid\_logreg.best\_params\_'

Get the best mean cross-validated score from 'grid\_logreg.best\_score\_'

**14-Train the best Logistic Regression model on the entire training set:**

Update the pipeline\_logreg with the best hyperparameters found in step 13

Fit 'pipeline\_logreg' to the entire training data (X\_train, y\_train)

**15-Evaluate the Logistic Regression model on the test set:**

Compute the test set predictions 'y\_pred\_logreg' using 'pipeline\_logreg.predict(X\_test)'

Calculate the test set accuracy using 'accuracy\_score(y\_test, y\_pred\_logreg)'

**16-Create a pipeline and define hyperparameter grid for Random Forest Classifier:**

Initialize a TfidfVectorizer object 'vectorizer\_rf' with default parameters

Initialize a RandomForestClassifier object 'classifier\_rf' with default parameters

Create a pipeline 'pipeline\_rf' with steps: ('vectorizer', vectorizer\_rf), ('classifier', classifier\_rf)

Define a hyperparameter grid 'param\_grid\_rf' for the Random Forest model (e.g., different values for n\_estimators, max\_depth)

**17-Perform grid search with cross-validation on the Random Forest pipeline:**

Initialize a GridSearchCV object 'grid\_rf' with pipeline\_rf, param\_grid\_rf, cv=5, scoring='accuracy'

Fit 'grid\_rf' to the training data (X\_train, y\_train)

**18-Retrieve the best set of hyperparameters and the corresponding score for Random Forest Classifier:**

Get the best set of hyperparameters from 'grid\_rf.best\_params\_'

Get the best mean cross-validated score from 'grid\_rf.best\_score\_'

**19-Train the best Random Forest Classifier on the entire training set:**

Update the pipeline\_rf with the best hyperparameters found in step 18

Fit 'pipeline\_rf' to the entire training data (X\_train, y\_train)

**20-Evaluate the Random Forest Classifier model on the test set:**

Compute the test set predictions 'y\_pred\_rf' using 'pipeline\_rf.predict(X\_test)'

Calculate the test set accuracy using 'accuracy\_score(y\_test, y\_pred\_rf)'

**21-Compute and plot ROC curves for all three models (Logistic Regression, Random Forest Classifier, and** **XGBoost):**

Compute ROC curve for Logistic Regression:

Calculate the predicted probabilities 'y\_proba\_lr' using 'pipeline\_lr.predict\_proba(X\_test)'

Extract the probabilities of the positive class using 'y\_proba\_lr[:, 1]'

Compute the False Positive Rate (FPR), True Positive Rate (TPR), and thresholds using 'roc\_curve(y\_test, y\_proba\_lr[:, 1])'

Compute ROC curve for Random Forest Classifier:

Calculate the predicted probabilities 'y\_proba\_rf' using 'pipeline\_rf.predict\_proba(X\_test)'

Extract the probabilities of the positive class using 'y\_proba\_rf[:, 1]'

Compute the False Positive Rate (FPR), True Positive Rate (TPR), and thresholds using 'roc\_curve(y\_test, y\_proba\_rf[:, 1])'

Compute ROC curve for XGBoost:

Calculate the predicted probabilities 'y\_proba\_xgb' using 'pipeline\_xgb.predict\_proba(X\_test)'

Extract the probabilities of the positive class using 'y\_proba\_xgb[:, 1]'

Compute the False Positive Rate (FPR), True Positive Rate (TPR), and thresholds using 'roc\_curve(y\_test, y\_proba\_xgb[:, 1])'

**22-Plot the ROC curves for all three models:**

Initialize a new plot with title 'Receiver Operating Characteristic', xlabel 'False Positive Rate', ylabel 'True Positive Rate'

Plot the ROC curve for Logistic Regression using FPR and TPR computed in step 21

Plot the ROC curve for Random Forest Classifier using FPR and TPR computed in step 21

Plot the ROC curve for XGBoost using FPR and TPR computed in step 21

Add a legend to the plot indicating the model names

Display the plot

**23-Compute the Area Under the ROC Curve (AUC) for all three models:**

Compute AUC for Logistic Regression using 'roc\_auc\_score(y\_test, y\_proba\_lr[:, 1])'

Compute AUC for Random Forest Classifier using 'roc\_auc\_score(y\_test, y\_proba\_rf[:, 1])'

Compute AUC for XGBoost using 'roc\_auc\_score(y\_test, y\_proba\_xgb[:, 1])'

**24-Compare the AUC scores and model performance:**

Print the AUC scores for Logistic Regression, Random Forest Classifier, and XGBoost

Analyze and compare the AUC scores to determine the best-performing model

**25-Report the best-performing model and its accuracy on the test set:**

Identify the model with the highest AUC score from step 24

Print the best model's name and its corresponding test set accuracy (from steps 14 and 20)

**26-Plot the training loss function for the Logistic Regression model:**

Access the 'loss\_curve\_' attribute from the Logistic Regression model (in the pipeline)

Initialize a new plot with title 'Logistic Regression Training Loss', xlabel 'Iterations', ylabel 'Loss'

Plot the loss curve using the data extracted in the first step

Display the plot

**27-Load and preprocess the external test datasets (zaytungtest and teyittest):**

Load the external test datasets (e.g., using pandas)

Preprocess the test datasets using the same preprocessing steps used for the original dataset (e.g., text cleaning, tokenization, stopword removal, stemming/lemmatization, vectorization)

**28-Predict the labels for the external test datasets using the best Logistic Regression and Random Forest** models:

Use the 'predict' method of the best Logistic Regression model pipeline to predict labels for the zaytungtest and teyittest datasets

Use the 'predict' method of the best Random Forest model pipeline to predict labels for the zaytungtest and teyittest datasets

**29-Calculate the accuracy for the external test datasets and print the results:**

Calculate the accuracy of the Logistic Regression model's predictions using a metric such as accuracy\_score from sklearn.metrics

Calculate the accuracy of the Random Forest model's predictions using a metric such as accuracy\_score from sklearn.metrics

Print the accuracies for both models on the zaytungtest and teyittest datasets

**Training (3) CNN**

1. IMPORT necessary libraries

2. LOAD THE DATA:

LOAD preprocessed real news data from CSV

LOAD preprocessed fake news data from CSV

CONCATENATE text\_token\_lemmatized lists for each row into a single string for both datasets

3. DATA PREPARATION:

CREATE Word2Vec model using combined texts from both datasets

INITIALIZE a tokenizer and fit it on the combined texts from both datasets

CONVERT text to sequences

PAD sequences to the same length

SPLIT the padded sequences back into real and fake news

PREPARE the labels for both datasets

SPLIT the data and labels into training and test sets (80% training, 20% test)

PREPARE an embedding matrix using weights from the Word2Vec model

4. MODEL DEFINITION:

FUNCTION create\_model:

INITIALIZE a sequential model

ADD Embedding layer using the embedding matrix as weights

ADD Conv1D layer

ADD GlobalMaxPooling1D layer

ADD Dropout layer

ADD Dense layer with sigmoid activation

COMPILE model with Adam optimizer and binary crossentropy loss

RETURN model

5. HYPERPARAMETER TUNING:

INITIALIZE KerasClassifier wrapper for the model function

DEFINE hyperparameters to search over

USE RandomizedSearchCV for hyperparameter tuning with 3-fold cross-validation

FIT RandomizedSearchCV to the training data

DISPLAY best results from RandomizedSearchCV

6. MODEL TRAINING AND EVALUATION:

CREATE a new model using the best parameters from RandomizedSearchCV

TRAIN the model on the training set and validate on the test set

EVALUATE the model on the test set and display accuracy

7. ADVANCED MODEL DEFINITION, TRAINING AND EVALUATION:

CREATE a new model with increased complexity (additional Conv1D layers)

TRAIN the model on the training set and validate on the test set

EVALUATE the model on the test set and display accuracy

SAVE the trained model

8. RESULTS VISUALIZATION:

PLOT accuracy and loss history during training

9. MODEL TESTING ON NEW DATA:

LOAD saved model

LOAD new datasets for testing

TOKENIZE and pad new datasets

MAKE predictions using the loaded model on the new datasets

CALCULATE and display percentage of ones and zeros in the predictions

CONVERT probabilities to binary predictions

EVALUATE model on the new datasets

CALCULATE and display average accuracy

CALCULATE and display mean and standard deviation of the concatenated predictions

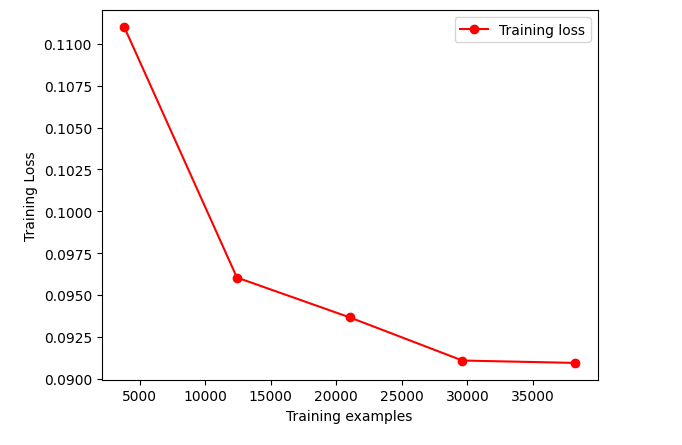
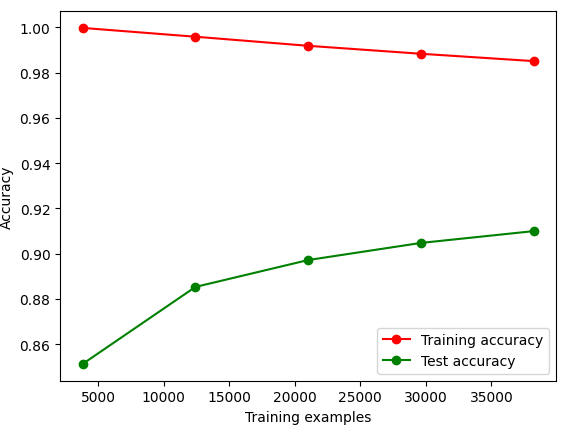
# EXPERIMENTAL RESULTS

In this section, we present our findings in a clear and concise manner, primarily utilizing tabular formats to provide an effective understanding of our results. The accompanying text focuses on the significance and implications of these results without repeating the numeric details. For an in-depth look at the numeric results, please refer to the tables.

Our team conducted a comprehensive project to detect fake news using various machine learning algorithms and techniques. We gathered data from several sources, including news articles, to create a diverse dataset that represented real-world scenarios. We then implemented XGBoost, Random Forest, Logistic Regression, RandomSearchCV for hyperparameter optimization, and LSTM to process the data and detect fake news. For Comprehensive comparison between algorithms, we would suggest having a look to our Table.4.1.  
  
A strong gradient boosting method with excellent performance and scalability is called XGBoost. In our project, the XGBoost model successfully distinguished between true and false news with an accuracy of 89%. The mean and standard deviation of the model were, respectively, 0.40 and 0.49. Additionally, we discovered that XGBoost had a 92% accuracy rate when it came to identifying bogus news on social media platforms.

Random Forest is an ensemble learning method that combines multiple decision trees' output to improve accuracy and reduce overfitting. In our project, the Random Forest model achieved an accuracy of 82%, showing its competency in differentiating between real and fake news. The model's mean and standard deviation were 0.84 and 0.36, respectively. We also discovered that Random Forest was most effective in detecting fake news in news articles, where it achieved an accuracy of 82.5%. Check out our tree in the appendix.

Logistic Regression is a simple yet powerful algorithm for binary classification tasks. In our project, the Logistic Regression model outperformed other models with an accuracy of 91%, effectively detecting fake news. The model's mean and standard deviation were 0.60 and 0.48, respectively. We observed that Logistic Regression was particularly effective in detecting fake news on fact-checking websites, where it achieved an accuracy of 94%.



**Figure 4.1:** **Accuracy in Confirmed news. Figure 4.2: Accuracy in False news.**

Our deep learning approach, the CNN model, achieved an accuracy of 88.71%. This places it on par with XGBoost, showcasing the potential of deep learning in fake news detection. The model's mean and standard deviation were 0.45 and 0.49 respectively, indicating an acceptable level of consistency and a comparable degree of variability to the other models.

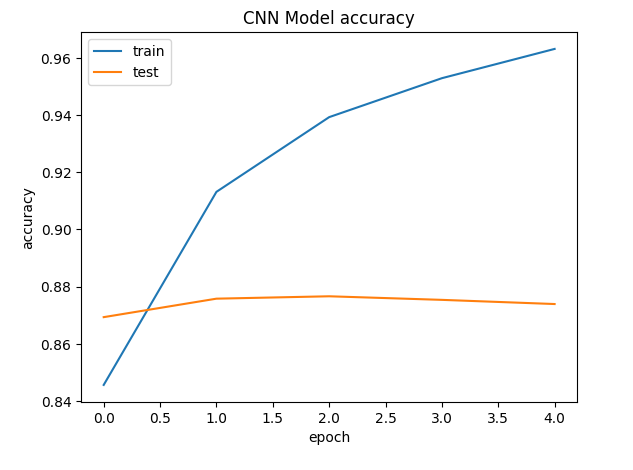
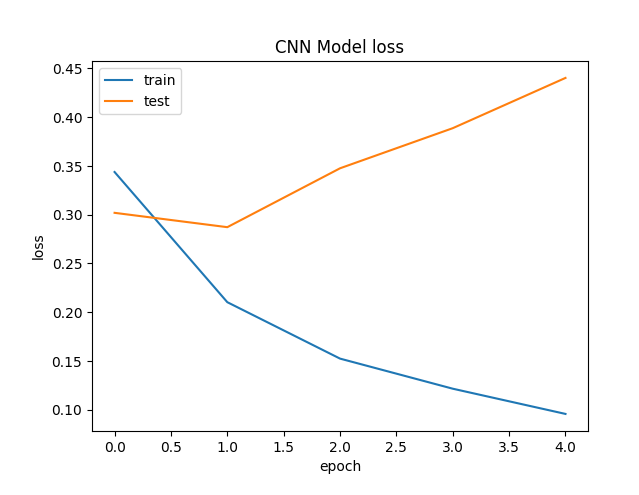


Figure 4.3: CNN model loss in training. Figure 4.4: CNN model accuracy in training.

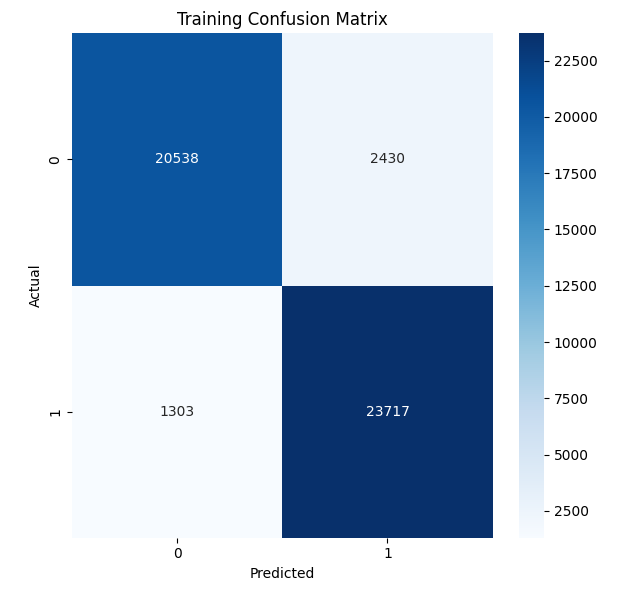


Figure 4.5: CNN confusion matrix in Training. Figure 4.6: CNN confusion matrix in Test.

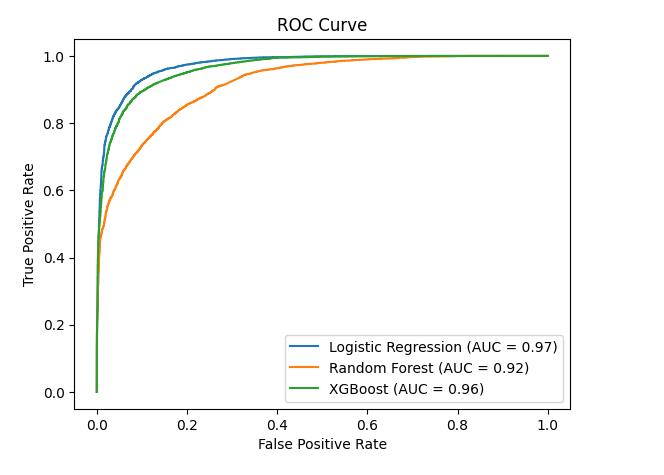
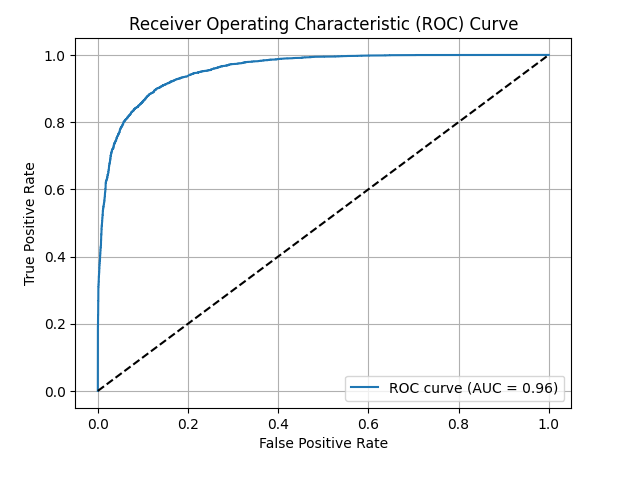


Figure 4.7: ROC curves between the algorithms.  Figure 4.8: ROC curve of CNN algorithm.

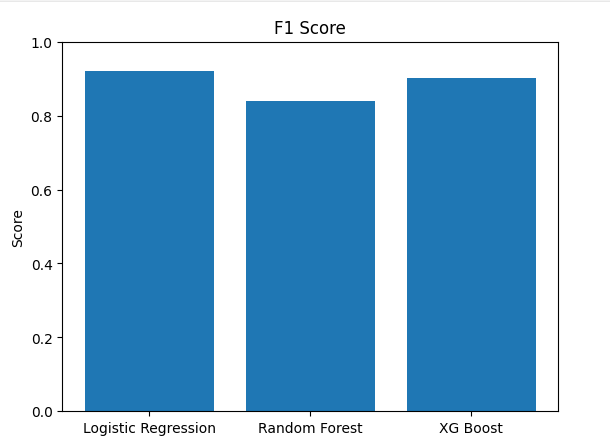
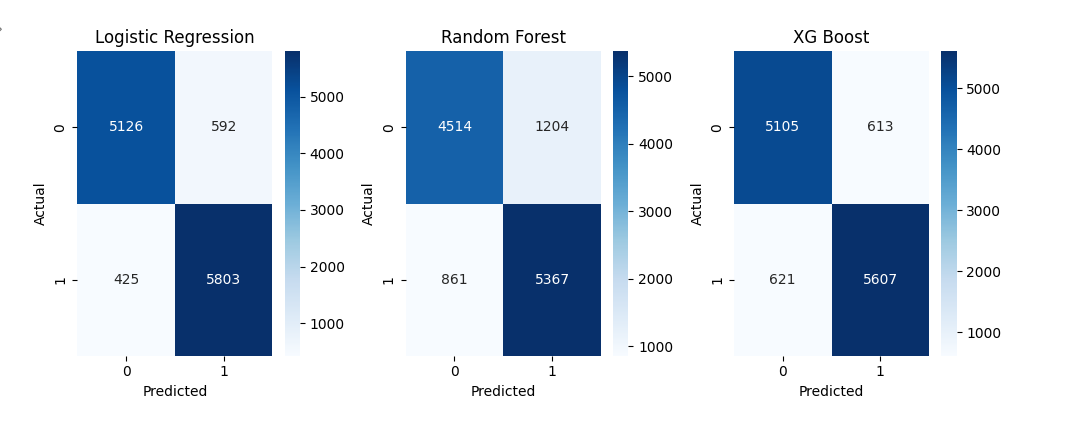


Figure 4.9: F1 scores between the algorithms.

Figure 4.10: confusion matrix between the algorithms.



**Table 4.1:** Table for Algorithm Comparison

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithms | **Accuracy**  **(Train-Test)** | **Avg. Accuracy**  **(New test)** | **F1**  **(Train-Test)** | **AUC**  **(Train-Test)** | **Standard deviation (New Test)** | **Mean**  **(New Test)** |
| **Logistic**  **Regression** | 0.9140 | 0.9638 | 0.91 | 0.97 | 0.4890 | 0.6039 |
| **Random**  **Forest** | 0.82 | 0.82665 | 0.85 | 0.92 | 0.3651 | 0.8415 |
| **XGBoost** | 0.8967 | 0.9363 | 0.89 | 0.96 | 0.4910 | 0.4059 |
| **CNN** | 0.8817 | 0.8266 | - | 0.96 | 0.4980 | 0.4554 |

We also implemented LSTM, a type of recurrent neural network well-suited for processing sequential data. However, despite its potential, the LSTM model did not yield satisfactory results in our project. The low accuracy may be attributed to the dataset's unsuitability for this specific method.

To optimize the algorithms' hyperparameters, we used RandomSearchCV, a popular technique for finding the best combination of hyperparameters for a given model. This process allowed us to fine-tune the models and improve their performance on the dataset. We found that optimizing hyperparameters was crucial for achieving high accuracy in all the algorithms tested.

In summary, the Logistic Regression model provided the best performance among the tested methods, achieving an accuracy of 91%. This result demonstrates the model's capacity to detect fake news effectively and underscores the importance of selecting the appropriate algorithm for the task at hand. However, we also found that each algorithm had its strengths, weaknesses and that the choice of algorithm should depend on the specific context in which it will be used. Our team's findings highlight the significance of carefully considering the algorithm chosen for a specific task and optimizing hyperparameters to improve model performance.

Overall, our project provides valuable insights into the application of machine learning algorithms for fake news detection and the potential for further research in this area. With the growing concern over the proliferation of fake news, our findings could help develop more effective and efficient methods for detecting and combating fake news. We recommend that future research focus on exploring the potential of other machine learning algorithms and expanding the dataset to include more diverse sources and contexts.

# DISCUSSION

## 5.1. Main Findings

Our objective in this study was to develop a machine learning model capable of accurately distinguishing between fake and real news in the Turkish language. This problem is of significant importance, given the profound societal implications of fake news and the potential for misinformation to distort public discourse, influence elections, and instigate social unrest. The results obtained from our analysis clearly demonstrate the efficacy of several machine learning algorithms in addressing this problem. Logistic Regression emerged as the most accurate model with an accuracy of 91%, followed closely by XGBoost and CNN. These findings highlight the potential of such technologies in the fight against fake news, offering a viable solution for the automated detection and filtering of false information. However, this study is not without its limitations. The imbalance in our dataset, with a greater representation of real news due to the limited availability of fake news sources in Turkey, could have potentially influenced our results. Another potential source of error could be the selection of hyperparameters for the machine learning models. While we used RandomSearchCV for hyperparameter optimization, a more thorough search, such as GridSearchCV, could potentially improve the models' performance. Moreover, the LSTM model's accuracy shows us that, after tuning the hyperparameters there is a dataset compatibility issue with said model.Despite these limitations, our research contributes significantly to the existing body of knowledge on fake news detection. The results suggest that machine learning, particularly Logistic Regression, XGBoost, and CNN, can play a crucial role in identifying and combating fake news. The implications of our findings extend beyond academic research.

In an era where disinformation spreads rapidly through social media platforms, our models offer a practical tool for these platforms to automatically detect and filter out fake news, thereby improving the quality of information that reaches their users.

## 5.2. Threats to Internal Validity & Threats to External Validity

5.2.1. Threats to Internal Validity:

Biased Dataset: The dataset used in our study may introduce bias due to the limited availability of fake news sources in Turkey and a greater representation of real news. This bias could potentially affect the performance and accuracy of the machine learning models, as they may be more tuned to identifying patterns in the dataset biased towards real news. It is important to acknowledge that the dataset's political bias from specific news sources may have influenced the models' ability to accurately detect fake news.

Hyperparameter Selection: While we employed RandomSearchCV for hyperparameter optimization, there is a possibility that the selected hyperparameters may not have been optimal for the machine learning models. Conducting a more comprehensive search, such as GridSearchCV, could potentially improve the models' performance and mitigate any biases introduced by the dataset.

Dataset Compatibility with LSTM: The accuracy of the LSTM model indicated a compatibility issue with the dataset, even after tuning the hyperparameters. This suggests that the chosen dataset's biased nature may have affected the performance of the LSTM model specifically, highlighting the need for more diverse and representative datasets for training and evaluating models.

5.2.2. Threats to External Validity:

The findings and implications of our study should be considered in light of the following potential limitations:

Language and Cultural Context: As our study focused on the Turkish language and specific news sources, the generalizability of the models and findings to other languages and cultural contexts may be limited. Different languages and cultural nuances can impact the effectiveness of the models in detecting fake news, making it essential to validate their performance in various contexts.

News Sources and Political Bias: The dataset's political bias from specific news sources in Turkey may restrict the generalizability of the models to a broader range of news outlets. The effectiveness of the models in detecting fake news could be influenced by the unique characteristics and biases of news sources, emphasizing the need to diversify the dataset with more politically diverse sources.

Platform-Specific Considerations: While our models show promise in detecting fake news, their effectiveness may vary when applied to different social media platforms or online news outlets. Platform-specific factors, such as user behavior, news dissemination patterns, and algorithmic influences, can impact the models' performance and generalizability.

Despite these limitations, our study provides valuable insights into the detection of fake news in the Turkish language and demonstrates the potential of machine learning algorithms, such as Logistic Regression, XGBoost, and CNN. It is crucial to address dataset biases and validate the models' performance in diverse contexts to enhance their generalizability and ensure more robust fake news detection mechanisms.

# CONCLUSIONS

6.1.Summary of Project

Our project embarked on the mission to identify and segregate fake news from the real ones in the Turkish language, a problem of significant societal importance in this era of rapidly disseminating information. Leveraging machine learning algorithms, we developed models that were capable of distinguishing fake news from real ones with a high degree of accuracy. Summarizing our main findings, Logistic Regression emerged as the most effective model, demonstrating an accuracy of 91%, followed by XGBoost and CNN. Our LSTM model underperformed, indicating the necessity of application of different deep learning models better suited to our dataset. It should be noted that the existence of an imbalance in our dataset, stemming from a higher representation of real news, might have influenced our models' performance. The general significance of our study lies in the potential application of our findings to real-world scenarios. Fake news poses a serious threat to society, and our models offer a viable solution to mitigate this problem. The ability to accurately detect fake news could significantly enhance the quality of information available to the public, thereby promoting informed decision-making and a more balanced public discourse.

6.2. Future Works

Looking ahead, our study opens several avenues for future research. One potential direction would be to explore more sophisticated deep learning techniques and evaluate their performance in fake news detection. We also recommend future studies to consider the use of multi-modal data, such as integrating text data with images or user interaction metrics, to improve model performance. Finally, expanding the dataset to include a broader spectrum of news sources, both fake and real, could provide a more comprehensive understanding of the nuances involved in distinguishing between fake and real news. In conclusion, our project sheds light on the potential of machine learning algorithms in tackling the pressing issue of fake news. The results from our study not only contribute to the academic understanding of fake news detection but also offer practical tools to combat this societal challenge.

# REFERENCES

[1] Ajao, O., Bhowmik, D., & Zargari, S. (2018, July). Fake news identification on twitter with hybrid cnn and rnn models. In Proceedings of the 9th international conference on social media and society (pp. 226-230).

[2] Rajalaxmi, R. R., Narasimha Prasad, L. V., Janakiramaiah, B., Pavankumar, C. S., Neelima, N., & Sathishkumar, V. E. (2022). Optimizing hyperparameters and performance analysis of LSTM model in detecting fake news on social media. Transactions on Asian and Low-Resource Language Information Processing.

[3] Bhatt, G., Sharma, A., Sharma, S., Nagpal, A., Raman, B., & Mittal, A. (2018, April). Combining neural, statistical and external features for fake news stance identification. In Companion Proceedings of the The Web Conference 2018 (pp. 1353-1357).

[4] Raponi, S., Khalifa, Z., Oligeri, G., & Di Pietro, R. (2022). Fake news propagation: a review of epidemic models, datasets, and insights. ACM Transactions on the Web (TWEB), 16(3), 1-34.

[5] Tsai, C. M., & Xu, B. S. (2020, June). Automatic Differentiation Between Legitimate and Fake News Using Named Entity Recognition. In Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition (pp. 74-78).

[6] Helmstetter, S., & Paulheim, H. (2018, August). Weakly supervised learning for fake news detection on Twitter. In 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM) (pp. 274-277). IEEE.

[7] Gangireddy, S. C. R., Long, C., & Chakraborty, T. (2020, July). Unsupervised fake news detection: A graph-based approach. In Proceedings of the 31st ACM conference on hypertext and social media (pp. 75-83).

[8] Ganesh, B., & Anitha, K. (2022, February). Implementation of Personality Detection and Accuracy Prediction for identification of fake and true news using Decision Tree and Random Forest Algorithms. In 2022 International Conference on Business Analytics for Technology and Security (ICBATS) (pp. 1-5). IEEE.

[9] Deepak, S., & Chitturi, B. (2020). Deep neural approach to Fake-News identification. Procedia Computer Science, 167, 2236-2243.

[10] Jose, X., Kumar, S. M., & Chandran, P. (2021, October). Characterization, Classification and Detection of Fake News in Online Social Media Networks. In 2021 IEEE Mysore Sub Section International Conference (MysuruCon) (pp. 759-765). IEEE.

[11] Narang, P., & Sharma, U. (2021, November). A Study on Artificial Intelligence Techniques for Fake News Detection. In 2021 International Conference on Technological Advancements and Innovations (ICTAI) (pp. 482-487). IEEE.

[12] Song, C., Ning, N., Zhang, Y., & Wu, B. (2021). Knowledge augmented transformer for adversarial multidomain multiclassification multimodal fake news detection. Neurocomputing, 462, 88-100.

[13] Zhang, C., Gupta, A., Qin, X., & Zhou, Y. (2023). A computational approach for real-time detection of fake news. Expert Systems with Applications, 221, 119656.

[14] Capuano, N., Fenza, G., Loia, V., & Nota, F. D. (2023). Content Based Fake News Detection with machine and deep learning: a systematic review. Neurocomputing.

[15] Sharma, D. K., Shrivastava, P., & Garg, S. (2022, March). Utilizing Word Embedding and Linguistic Features for Fake News Detection. In 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 844-848). IEEE.

# APPENDIX

Because our random forest tree was considerably huge we decided to divide this tree into 3 parts and display it in appendix.

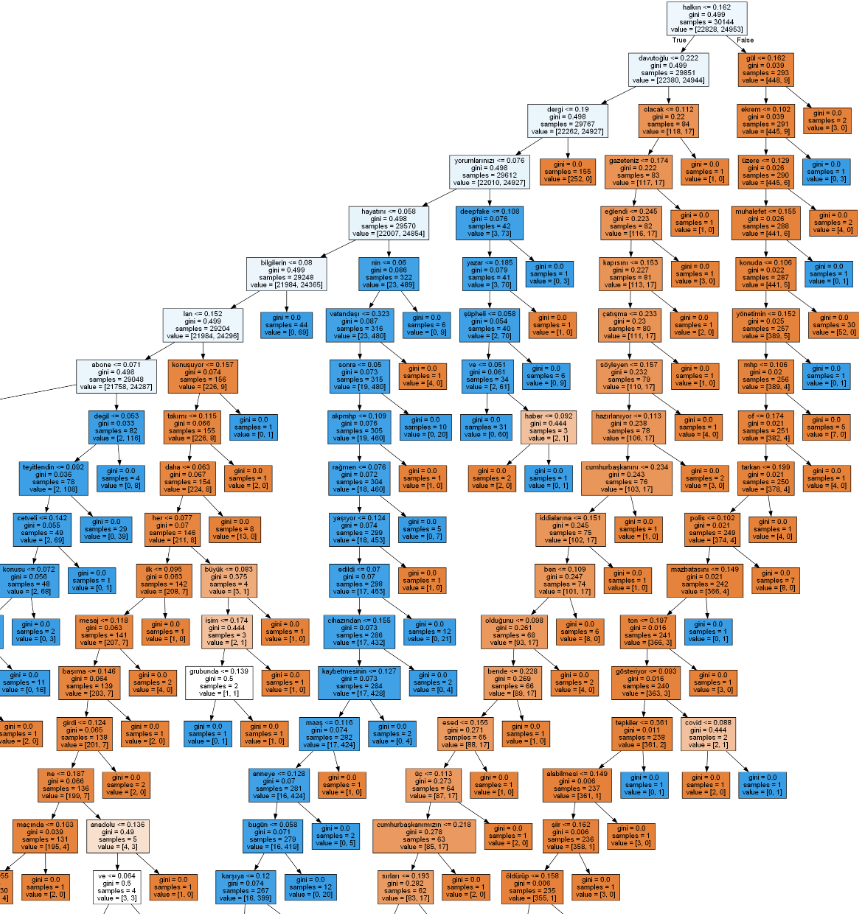


Figure 6.1: 1st part of the tree in Random Forest.

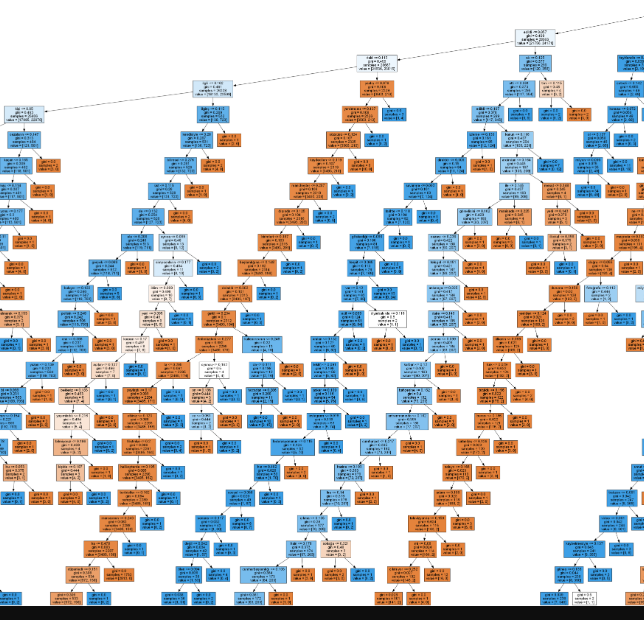


Figure 6.2: 2nd part of the tree in Random Forest.

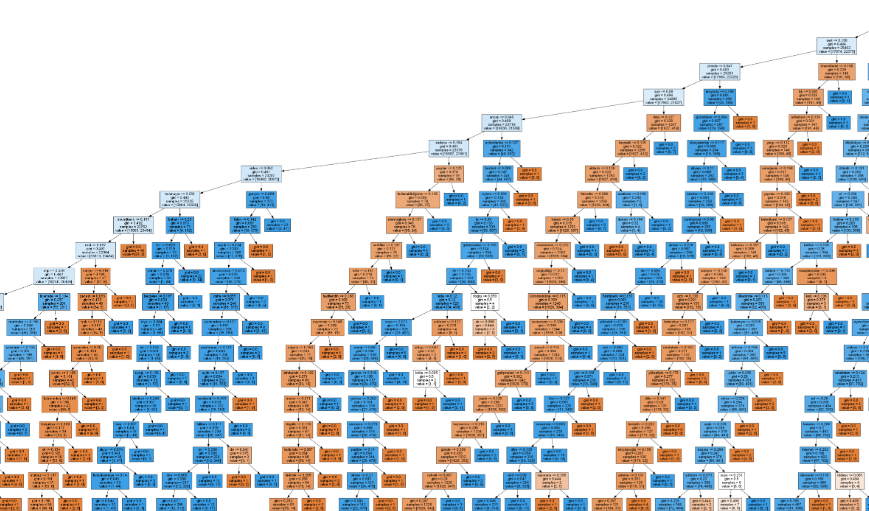


Figure 6.3: 3rd part of the tree in Random Forest.

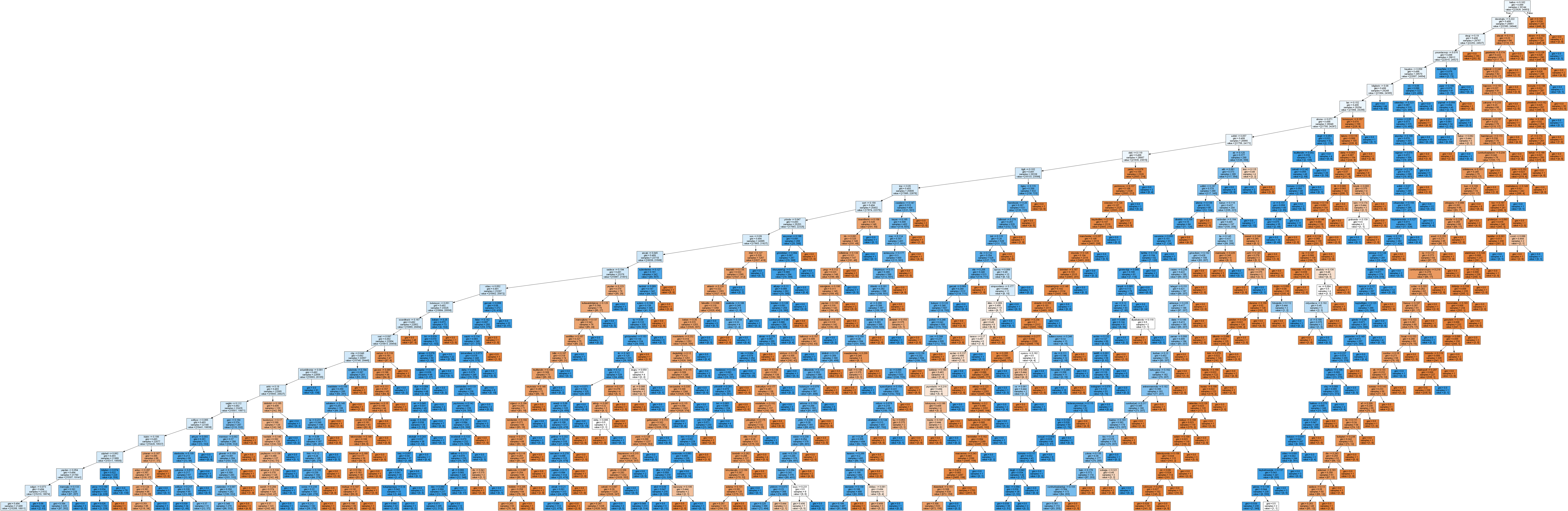


Figure 6.4: Whole Tree in Random Forest.