CUKUROVA UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

IDENTIFICATION OF CYBERBULLYING USING MACHINE LEARNING TECHNIQUES

ALI NAJIB

Computer Engineering Department

June, 2024

CUKUROVA UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS APPROVAL

IDENTIFICATION OF CYBERBULLYING USING MACHINE LEARNING TECHNIQUES

ALI NAJIB

Computer Engineering Department

This Doctorate Thesis was evaluated by the following Jury Members on …/.../…… and was approved by unanimity / majority of votes.

Jury : Prof. Dr. Selma Ayşe ÖZEL (Advisor) ……

: Assoc. Prof. Dr. Ali İNAN ……

: Assoc. Prof. Dr. Serkan KARTAL ……

: Assoc. Prof. Dr. Fatih KILIÇ ……

: Asst. Prof. Dr. Mehmet SARIGÜL ……

This Thesis was written in the Department of Computer Engineering, Institute of Natural and Applied Sciences.

Thesis Number:

Prof. Dr. Sadık DİNÇER
Director
Institute of Natural and Applied Sciences

CONTENTS

CUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

PhD THESIS

IDENTIFICATION OF CYBERBULLYING USING MACHINE LEARNING TECHNIQUES

Ali NAJIB

Advisor: Prof. Dr. Selma Ayşe ÖZEL

Department of Computer Engineering

ABSTRACT

The pervasive utilization of social media platforms has introduced a multitude of threats, among which cyberbullying stands as a significant concern. Cyberbullying is defined as the repetitive use of social and electronic media, to perpetrate immoral actions with the intent to inflict harm upon a victim. In both Turkey and globally, cyberbullying is recognized as a pressing issue necessitating urgent attention due to its profound emotional and psychological repercussions, which can include depression, stress, anxiety, and in extreme cases, suicide. For this concern, numerous researchers and scientists have endeavored to develop solutions leveraging machine learning (ML), deep learning (DL), and natural language processing (NLP) techniques. These efforts aim to create models capable of identifying cyberbullying within textual contexts, with the ultimate goal of either detecting and removing such content or preventing its dissemination. However, the large number of studies in this context have been conducted in English, with a dearth of research in Turkish language. Moreover, existing studies often segregate their analyses between English and Turkish, overlooking potential cross-linguistic nuances. Motivated by these gaps in the literature, this thesis endeavors to address the detection of cyberbullying in both English and Turkish languages. Framed as a binary text classification problem, the research scrutinizes the efficacy of standard techniques across languages, aiming to discern whether a unified approach can effectively detect cyberbullying in diverse linguistic contexts. To achieve this objective, comprehensive experiments are conducted, including the comparative evaluation of ML, DL, and Large Language model (LLM) Furthermore, an array of feature extraction techniques, including traditional feature weightings and word embeddings, are rigorously assessed. Additionally, the efficacy of an optimization technique LoRA, applied to LLM, is thoroughly evaluated. Furthermore, recognizing the problem created by labeled data shortage, a novel semi-supervised learning technique is proposed. Specifically, a self-training with LLM is implemented, demonstrating its potential to enhance classification performance by leveraging a blend of both labeled and unlabeled data, thereby mitigating the resource-intensive nature of manual labeling processes. This research contributes to advancing cyberbullying detection methods and encourages more inclusive approaches across languages to combat cyberbullying in the digital sphere.

Keywords: Cyberbullying detection, Natural Language Processing, Machine Learning, Deep Learning, social media.

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DOKTORA TEZİ

MAKİNE ÖĞRENMESİ TEKNİKLERİNİ KULLANARAK SİBER ZORBALIĞIN TESPİTİ

Ali NAJIB

Danışman: Prof. Dr. Selma Ayşe ÖZEL

Bilgisayar Mühendisliği Anabilim Dalı

ÖZ

Sosyal medya platformlarının yaygın kullanımı, aralarında siber zorbalığın da önemli bir endişe kaynağı olduğu çok sayıda tehdidi beraberinde getirmiştir. Siber zorbalık, bir mağdura zarar vermek amacıyla ahlak dışı davranışlarda bulunmak için elektronik veya dijital medyanın tekrarlı kullanımı olarak tanımlanmaktadır. Hem Türkiye'de hem de dünyada siber zorbalık, depresyon, stres, anksiyete ve aşırı durumlarda intiharı da içerebilen derin duygusal ve psikolojik yansımaları nedeniyle acil dikkat gerektiren bir konu olarak kabul edilmektedir. Bu endişe için çok sayıda araştırmacı ve bilim insanı makine öğrenimi (ML), derin öğrenme (DL) ve doğal dil işleme (NLP) tekniklerinden yararlanarak çözümler geliştirmeye çalışmıştır. Bu çabalar, siber zorbalığı metinsel bağlamlar içinde tanımlayabilen modeller oluşturmayı ve nihai hedef olarak bu tür içeriği tespit edip kaldırmayı ya da yayılmasını önlemeyi amaçlamaktadır. Ancak, bu alandaki çalışmaların büyük çoğunluğu İngilizce dilinde yapılmış olup, Türkçe dilinde yapılan araştırma sayısı oldukça azdır. Dahası, mevcut çalışmalar analizlerini genellikle İngilizce ve Türkçe arasında ayırmakta ve potansiyel diller arası nüansları göz ardı etmektedir. Literatürdeki bu boşluklardan hareketle bu tez, hem İngilizce hem de Türkçe dillerinde siber zorbalığın tespitini ele almaya çalışmaktadır. İkili metin sınıflandırma problemi olarak çerçevelenen araştırma, birleşik bir yaklaşımın farklı dilsel bağlamlarda siber zorbalığı etkili bir şekilde tespit edip edemeyeceğini anlamayı amaçlayarak diller arasında standart tekniklerin etkinliğini incelemektedir. Bu amaca ulaşmak için ML, DL ve büyuk dil modelinin (LLM) karşılaştırmalı değerlendirmesi de dahil olmak üzere kapsamlı deneyler gerçekleştirilmiştir. Ayrıca, geleneksel özellik ağırlıkları ve kelime katıştırmaları da dahil olmak üzere bir dizi özellik çıkarma tekniği titizlikle değerlendirilmiştir. Ek olarak, LLM'ye uygulanan bir optimizasyon tekniği olan LoRA'nın etkinliği kapsamlı bir şekilde değerlendirilmiştir. Ayrıca, etiketli verilerin azlığının yarattığı zorluğun farkına varılarak, yeni bir yarı denetimli öğrenme tekniği önerilmiştir. Özellikle, kendi kendini eğiten büyük bir Dil Modeli uygulanmış, etiketli ve etiketsiz verilerin bir kombinasyonundan yararlanarak sınıflandırma performansını artırma potansiyeli gösterilmiş ve böylece manuel etiketleme işlemlerinin kaynak yoğun doğası hafifletilmiştir. Bu araştırma, siber zorbalık tespit yöntemlerinin geliştirilmesine katkıda bulunmakta ve dijital alanda siber zorbalıkla mücadele etmek için diller arasında daha kapsayıcı yaklaşımları teşvik etmektedir.

Anahtar Kelimeler: Siber zorbalık tespiti, Doğal Dil İşleme, Makine öğrenmesi, Derin öğrenme, sosyal medya.

GENİŞLETİLMİŞ ÖZET

Son yıllarda, teknolojik gelişmeler, güçlü internet hizmetlerinin yaygınlaşması ve akıllı telefonlar, tabletler ve bilgisayarlar gibi dijital cihazların yaygınlaşmasına bağlı olarak sosyal medya ağlarının küresel kullanımında kayda değer bir artış olmuştur. Bu platformlar kişiler arası bağlantıyı, açık etkileşimi ve içerik paylaşımını kolaylaştırmaktadır. Bunlar arasında öne çıkanlar Facebook, YouTube, Instagram, TikTok, WhatsApp ve Telegram'dır.

Sosyal medyanın sunduğu fırsatlara rağmen, yaş sınırlamaları söz konusu olduğunda etkili bir kontrol eksikliği vardır. Hem Türkiye hem de dünya nüfusunun önemli bir bölümünün sosyal medyada oldukça aktif olan çocuk ve gençlerden oluştuğunu belirtmek gerekir. Buna ek olarak, sosyal medya platformları genellikle sıkı bir sansürden yoksundur ve bu da insanları, kişisel görüşlere saygı gösterilmemesi ve diğer insanların mahremiyetine müdahale edilmesi nedeniyle günümüz dijital ortamında küresel bir tehdit haline gelen siber zorbalık sorununa karşı savunmasız hale getirmektedir.

Siber zorbalık, elektronik ortamda bir kişi veya grup tarafından, kurbanın kendisini koruyamadığı ve zaman içinde tekrarlanan kasıtlı, saldırgan bir davranış olarak tanımlanır (Emmery vd., 2021). Hastalık Kontrol ve Önleme Merkezi (CDC), siber zorbalığı ciddi bir kamu sağlığı tehdidi olarak belirlemiş ve genel halkı uyarıda bulunmuştur (Aboujaoude vd., 2015). Siber zorbalık farklı şekillerde ortaya çıkabilir ve en yaygın olanları mesaj gönderme ve yorum yapma yoluyla gerçekleşir; bu, kurbanların kasten herhangi bir sosyal medya platformundan çıkarılmaları anlamına gelen dışlama, internette yayınlanan saldırgan mesajları içeren Taciz, izinsiz olarak birini açıkça aşağılama eylemi olan gezi, izleyicilerin kurbanları taciz etmesiyle sonuçlanan siber takip ve bir kişinin bir sosyal medyaya giriş yapması ve uygunsuz içerik yayınlayarak kişiyi taklit etmesi olan hesap şakası içerir (Beale & Hall, 2007; Beran & Li, 2008).Siber zorbalık, geleneksel zorbalığa kıyasla kalıcı etkileri olan daha geniş bir kitleye ulaşma potansiyeline sahiptir. Depresyon, düşük öz saygı, yorgunluk ve hepsinden önemlisi intihar eğilimi gibi duygusal etkilerle ilişkilidir (Özel vd., 2017).

Türkiye'nin de aralarında bulunduğu bazı ülkeler siber zorbalıkla mücadele için yasal tedbirler almış olsa da, bu tedbirlerin uygulanması zor olmaya devam etmektedir. Benzer şekilde, sosyal medya platformları siber zorbalığı ele almak için kılavuz ilkeler ve politikalar getirmiştir, ancak bu önlemler yetersiz raporlama ve etkisiz uygulama mekanizmaları nedeniyle genellikle yetersiz kalmaktadır. Sonuç olarak, sosyoloji, bilgisayar bilimleri gibi çeşitli alanlardan araştırmacılar da dahil olmak üzere özel kuruluşlar, hükümet ve akademik kuruluşlar siber zorbalık çalışmalarına yoğun ilgi göstermiştir (Xiao vd., 2012). Bilgisayar bilimlerinden gelen araştırmalar, kelime torbasi (BOW) ve N-gram ile geleneksel makine öğrenimi (ML) kullanarak siber zorbalığın tanımlanmasına yöneliktir. BOW modeli, kelimeleri metindeki görünüm sıralarını veya anlamlarını dikkate almadan özellik olarak değerlendirir (Çoban, ve Özel, 2018; Najib vd., 2023). N-gram ise

metindeki n ardışık karakter veya kelimeyi almayı içerir. N-gram modelleri, kelime analiz edilmeden önce n-1 kelimenin varlığını tahmin eder (Mossie ve Wang, 2020). Son zamanlarda, donanım maliyetlerinin düşmesi ve Word2vec (Mikolov vd., 2013) gibi modern kelime gömme yöntemlerinin ortaya çıkması nedeniyle derin öğrenme (DL) yöntemleri de uygulanmaktadır. GloVe (Jeffrey Pennington, Richard Socher, 2014) ve FastText (Bojanowski vd., 2017) gibi yöntemler, kelimeleri temsil etmek için yoğun vektörler elde etmek ve kelimeler arasındaki semantik ilişkiyi korumak için kullanılmaktadır.

Siber zorbalığın gördüğü ilgiye rağmen, araştırma çabaları ağırlıklı olarak İngilizce diline odaklanmış ve Türkçe 'de sınırlı sayıda araştırma yapılmıştır. Bu boşluğu ele alan bu tez, farklı morfolojik yapılarını tanıyarak hem İngilizce hem de Türkçe dillerinde siber zorbalık tespitinin kapsamlı bir analizini yapmayı amaçlamaktadır.

Bunu, yalnızca bir mesajın zorbalık içerip içermediğiyle ilgili olduğu ikili bir metin sınıflandırma problemi olarak ele aldık. Analizimizi ikisi İngilizce (Kaggle Parsed Dataset, Formspring Dataset) ve ikisi Türkçe (Cukurova Unversity Dataset, Abusive Turkish Comments) olmak üzere dört açık kaynak veri kümesi üzerinde gerçekleştirdik. NLTK yardımıyla kelime bölütleme (tokenization), durak kelimelerin çıkarılması (stop word removal), gövdeleme (stemming) ve küçük harfe dönüştürme gibi standart veri ön işleme yöntemlerini uyguladık.

Makine öğrenimi (ML), derin öğrenme (DL) ve büyük dil modellerinin (LLM'ler) performansını karşılaştırmalı olarak değerlendirmek için kapsamlı deneyler gerçekleştirdik. ML değerlendirmelerimizde Destek Vektör Makinesi (SVM), Lojistik Regresyon (LR), Multinomial Naïve Bayes, Rastgele Orman (RF), Karar Ağacı (DT) ve K-En Yakın Komşu (KNN) dahil olmak üzere yaygın olarak kullanılan sınıflandırıcıları kullandık. DL için, Uzun Kısa Süreli Bellek (LSTM), Çift Yönlü LSTM (BiLSTM), Geçitli Tekrarlayan Birim (GRU), Çift Yönlü GRU, Evrişimli Sinir Ağı (CNN) ve Tekrarlayan Sinir Ağı (RNN) mimarilerini değerlendirdik. Ayrıca, CNN-LSTM ve LSTM-CNN gibi hibrit mimarileri araştırdık. Ek olarak, ikili, terim sıklığı (TF) ve terim sıklığı-ters belge sıklığı (TF-IDF) özellik ağırlıklandırmasının yanı sıra Word2vec, GloVe ve FastText gibi kelime gömme yöntemlerini kapsayan çeşitli özellik çıkarma tekniklerini değerlendirdik. Büyük dil modellerinin farklı varyantlarını, özellikle de BERT'i hem İngilizce hem de Türkçe dillerinde kapsamlı bir şekilde değerlendirdik. Geleneksel makine öğrenimi yöntemlerinin performansını artırmak için, daha sonra makine öğrenimi sınıflandırıcılarında kullanılan yüksek kaliteli kelime vektörleri üretmek üzere LLM'leri entegre ederek benzersiz bir strateji geliştirdik. Ayrıca, LLM'ler ile Düşük Sıralama Uyarlaması (LoRA) kullanarak parametre optimizasyonu gerçekleştirdik.

Metin sınıflandırma görevlerinde etiketli verinin elde edilmesi zorluğu göz önüne alındığında, Kendi Kendine Eğitim (Self-Training) ve Birlikte Eğitim (Co-Training) gibi yarı denetimli öğrenme yöntemlerini araştırdık. Son olarak, sınıflandırma performansını artırmayı amaçlayan yeni bir kendi kendini eğiten büyük dil modeli yarı denetimli öğrenme tekniğini tanıttık.

Çalışmamız, siber zorbalık tespit metodolojilerinin çeşitli yönlerini araştırmayı amaçlayan birkaç ardışık deney şeklinde yapılandırılmıştır. İlk aşamada, terim ağırlıklandırma yöntemlerini klasik makine öğrenimi (ML) sınıflandırıcıları ile birleştirerek terim ağırlıklandırmanın geleneksel ML sınıflandırıcılarının performansı üzerindeki etkisini araştırdık. Bu adım, geleneksel ML sınıflandırıcılarla entegre edildiğinde terim ağırlıklandırma tekniklerinin etkisini aydınlatmayı amaçlamıştır. Daha sonra, makine öğrenimi tekniklerini kelime gömme yöntemleriyle birleştirmenin etkinliğini değerlendirmek için bir deney gerçekleştirdik. Buradaki amaç, hem eğitimli hem de ön eğitimli kelime gömmenin geleneksel makine öğrenmesi sınıflandırıcılarının performansını nasıl etkilediğini değerlendirmekti. Bu incelemede, kelime gömme tekniklerinin bir araya getirilmesiyle elde edilen sınıflandırmadaki potansiyel iyileştirmeler ele alınmıştır. Bunu takiben, siber zorbalık tespiti üzerindeki birleşik etkilerini araştırmak için kelime gömme vektörlerinin yanı sıra derin öğrenme (DL) metodolojilerini kullandık. Bu aşama, DL yöntemlerinin kelime gömme vektörleriyle birlikte, özellikle de geleneksel makine öğrenimi yaklaşımlarına kıyasla etkinliğini ortaya çıkarmayı amaçlamıştır. Bir sonraki adımda, odak noktamız BERT modelinin önceden eğitilmiş çeşitli varyantlarını değerlendirmeye ve bunların performansını çalışmamızda kullanılan ML ve DL modelleriyle karşılaştırmaya yönelmiştir. Bu karşılaştırmalı analiz, siber zorbalık tespiti bağlamında BERT varyantlarının göreceli etkinliğini belirlemeye çalıştı. Beşinci deneyimizde, makine öğrenimi algoritmalarını BERT modelleriyle entegre ederek performanslarını artırmaya çalıştık. Bu adım, makine öğrenimi algoritmalarının sınıflandırma doğruluğunu artırmak için BERT modellerinin yeteneklerinden yararlanmayı amaçlıyordu. Ayrıca, altıncı deneyimizde LoRA parametre optimizasyon tekniğini kullanarak BERT modellerinin farklı varyantlarını uyguladık. Bu deney, parametre ayarlama yoluyla BERT modellerinin performansını optimize etmeyi amaçlamıştır. Ayrıca, etiketsiz verilerin bolluğu ve etiketli verilerin azlığı göz önüne alındığında, hem etiketli hem de etiketsiz verilerden yararlanmak için yarı denetimli öğrenme tekniklerini, özellikle de Kendi Kendine Eğitim ve Birlikte Eğitim tekniklerini kullandık. Bu deneyin bir parçası olarak, siber zorbalık tespit görevlerinde sınıflandırma doğruluğunu artırmak için tasarlanmış yeni bir kendi kendini eğiten büyük dil modeli tabanlı yöntem önerdik.

Alandaki kapsamlı araştırmalarımıza dayanarak ve bildiğimiz kadarıyla, mevcut hiçbir çalışma, tek bir araştırma kapsamında hem İngilizce hem de Türkçe dillerinde siber zorbalık tespiti için makine öğrenimi (ML) ve derin öğrenme (DL) yöntemlerinin detaylı karşılaştırmalı bir değerlendirmesini yapmamıştır. Çeşitli deneyler gerçekleştirmemize rağmen, bulgularımız tutarlı bir şekilde FastText kelime gömme kullanan DL modellerinin hem İngilizce hem de Türkçe veri kümelerinde kayda değer bir etkinlik gösterdiğini ortaya koymaktadır. Bir dil modeli olan BERT'in önceden eğitilmiş varyantları tarafından sergilenen olağanüstü performansı özellikle dikkat çekicidir. Buna karşılık, eğitilmiş vektörlerin performansı önceden eğitilmiş modellerle karşılaştırıldığında yetersiz kalmaktadır. Ayrıca, analizimiz DL modellerinin ML modellerine

kıyasla üstün performansının altını çizmektedir. Özellikle, büyük dil modellerinin (LLM'ler) ML teknikleriyle birleştirilmesi, hem İngilizce hem de Türkçe dilleri için, tek başına ML yaklaşımlarının etkinliğini aşan gelişmiş sınıflandırma sonuçları vermektedir. Ayrıca, LLM'lerle, özellikle de Düşük Sıra Uyarlama (LoRA) parametre optimizasyon tekniğiyle entegre edilmiş BERT varyantlarıyla yaptığımız deneyler, genel LLM sonuçlarıyla karşılaştırılabilir performans sağlamaktadır. Kendi kendini eğiten büyük dil modeli tabanli yontemimuz, hem İngilizce hem de Türkçe dillerinde siber zorbalık tespitinde olağanüstü etkinlik göstermekte ve bu alanda gelişmiş DL metodolojilerinin faydasını daha da desteklemektedir.

Özetle, bu tez hem İngilizce hem de Türkçe dillerinde siber zorbalık tespiti için MLve DL yöntemlerinin karşılaştırmalı bir değerlendirmesine katkıda bulunarak siber zorbalık ve duygu analizi alanındaki araştırmacılar için içgörüler sunmaktadır. Gelecekteki araştırmalar, bu bulguları diğer dillere ve uygulama alanlarına genişletmeyi ve böylece dünya çapında siber zorbalığın daha geniş bir şekilde anlaşılmasına ve azaltılmasına katkıda bulunmayı amaçlamaktadır.

ACKNOWLEDGEMENTS

Pursuing a doctoral degree can be a challenging endeavor. However, with the proper support, it can become a profoundly transformative experience. Therefore, having a dedicated supervisor is essential for success in doctoral studies. In this context, I would like to express my sincere gratitude to my supervisor Prof. Dr. Selma Ayşe Özel for having given me the opportunity to work under her guidance and learn from her experience. She has always supported and guided me throughout the research work from title selection to finding the results and compiling the thesis. Her mentorship, encouragement, and invaluable insights have played a pivotal role in shaping my research and academic growth though out these years.

I extend my heartfelt appreciation to the members of the thesis committee Associate Prof. Dr. Ali İnan, Associate Prof. Dr. Serkan Kartal for their invaluable contributions, constructive feedback, and dedication to excellence. Their expertise and commitment to academic rigor have been instrumental in shaping the trajectory of my research and ensuring its quality.

I extend my gratitude to all the professors, research assistants, and administrative staff at the Department of Computer Engineering, Çukurova University for letting me fulfill my dream of completing a Ph.D. degree by enabling me to take advantage of the departments' facilities.

I am deeply thankful to Associate Prof. Dr. Önder Çoban from the Department of Computer Engineering Atatürk University, Erzurum, for his valuable assistance throughout my experimental work. He played a crucial role in reviewing my codes, offering suggestions that greatly contributed to my research. This collaboration was hand in hand with my supervisor, Prof. Selma Ayşe Özel.

I express my gratitude to the Council of Higher Education Turkiye for providing me with a Ph. D scholarship (YÖK SCHOLARSHIP). Their support has been instrumental in the successful completion of my doctoral studies.

Finally, I want to extend my sincere thanks to every member of my family who has consistently provided unwavering moral support and continuous encouragement throughout my years of education. Their encouragement, sacrifices and belief in my abilities have sustained me through the ups and downs, inspiring me to persevere and strive for excellence.

LIST OF TABLES

LIST OF FIGURES

SYMBOLS AND ABBREVIATIONS

ADAM : Adaptive Moment Estimation
API : Application Programming Interface
ANN : Artificial Neural Network
BERT : Bidirectional Encoder Representation from Transformers
BiLSTM : Bidirectional Long Short-Term Memory
BiGRU : Bidirectional Gated Recurrent Unit
BOW : Bag-of-Words
CBOW : Continuous Bag-of-Words
CPU : Central Processing Unit
CNN : Convolutional Neural Network
DL : Deep Learning
DT : Decision Tree
GloVe : Global Vectors
GPU : Graphics Processing Unit
GRU : Gated Recurrent Unit
KNN : K-Nearest Neighbors
LR : Logistic Regression
LLM : Large Language Model
LSTM : Long Short-Term Memory
ML : Machine Learning
NB : Naive Bayes
MNB : Multinomial Naive Byes
NLP : Natural Language Processing
NLTK : Natural Language Toolkit
NN : Neural Network(s)
RELU : Rectified Linear Unit
RF : Random Forest
RNN : Recurrent Neural Network
SSL : Self-Supervised Learning
SMOTE : Synthetic Minority Oversampling Techniques
SVM : Support Vector Machine
TC : Text Categorization (or Classification)
TF : Term Frequency
TF*IDF : Term Frequency*Inverse Document Frequency
TPU : Tensor Processing Unit

1. INTRODUCTION

1.1. Bullying

Bullying can be described as targeted intimidation or humiliation (Juvonen & Graham, 2014). Bullying is a form of mean behavior where a physically stronger or socially more prominent person repeatedly threatens, intimidates, demeans, or belittles another person who is in most cases a less powerful (Juvonen & Graham, 2014; Patchin & Hinduja, 2006; Smith et al., 2008). Bullying is identified by three key attributes. In the first place, the presence of hostile intent (indicating aggressive behavior from the bully towards the victim), secondly, a repetitive nature (occurring consistently over a period), and lastly an inherent power imbalance (where the bully possesses physical strength, can disclose humiliating information, or is recognized for causing harm to others) (Smith et al., 2008).

Various forms of bullying exist, including physical, verbal, and social (Dracic, 2009; Juvonen & Graham, 2014). Physical bullying involves acts such as hitting, fighting, shouting, spitting, tripping, pushing, kicking, pinching, and shoving while verbal bullying encompasses behaviors like name-calling, threatening, and taunting (Dracic, 2009). Social bullying, on the other hand, involves actions aimed at embarrassing someone publicly, such as spreading rumors, excluding individuals deliberately, laughing at, or mocking them (Yang & Salmivalli, 2013). A majority of bullying victims are males, demonstrating a higher likelihood of males being subjected to bully compared to females and this may be partly because the focus of the types of bully are not very often displayed by females (Kistner et al., 2006). Young children and adults both participate in bullying. Findings from survey research indicate that around 25-50% of young people are directly engaged in bullying either as perpetrators, victims, or in some cases, both (Juvonen & Graham, 2014). Likewise, younger age groups tend to primarily participate in direct verbal forms of bullying, while older age groups predominantly involve themselves in behaviors like exclusion, sabotage, and gossip (Privitera & Campbell, 2009; Vaez et al., 2004).

Bullying is associated with both immediate and enduring effects, including physical harm, difficulties in social functioning, diminished self-confidence, feelings of depression and anxiety, absenteeism from school, decreased academic performance, and a heightened risk of turning to self-medication as a primary issue of concern (Juvonen & Graham, 2014; Smokowski & Kopasz, 2005; Xu et al., 2012). The detrimental impact of bullying extends across various aspects of an individual's life, affecting not only their physical health but also their emotional and social dimensions.

1.2. Social Media

In the last few years, a significant global surge has been observed in the usage of social media networks. This can be attributed to advancements in technology, which are complemented by the availability of robust internet services and the widespread use of digital devices such as smartphones, mobile tablets, and modern computers. The evolution of social media, transitioning from the traditional web (web 1.0) to the Internet of Things (web 3.0), has revolutionized the way users retrieve data, engage with others, and search for information. Social media comprises a range of online platforms tailored to facilitate and enrich social interactions. These platforms represent a set of internet-based tools that have developed from the foundational principles of both the conceptual and technological aspects of web 2.0. These platforms enable the creation and sharing of user-generated content, fostering a dynamic environment for communication and collaboration on the Internet (Kaplan & Haenlein, 2010). As of October 2023, global statistics indicated that there were approximately 5.3 billion internet users worldwide, constituting around 65.7% of the total population. Among these users, approximately 4.95 billion people, accounting for 61.4% of the world's population, were active on social media platforms (*Anonymous 1*). Additionally, more than 80% of teenagers possessed at least one form of modern media technology, such as a cellphone, personal digital assistant, or computer for internet access. They are increasingly utilizing this technology for activities like texting, instant messaging, emailing, blogging, and accessing social networking websites (David-Ferdon & Hertz, 2007).

Researchers spanning disciplines from sociology, psychology, and computer science have shown keen interest in social media research (Xiao et al., 2012). Within this, sociologists find social media data to be a crucial resource for comprehending and analyzing human behavior (Wong et al., 2014). In computer science, social media research tackles challenges like information extraction (Siwag et al., 2014) user behavior analysis, and identification of fake profiles, (Terrana et al., 2014). It also explores trends in opinions and content-sharing behavior, sentiment analysis, cyberbullying (Mfenyana et al., 2013).

Social media serves as an excellent platform for exchanging information and thoughts, fostering collaboration, and facilitating communication among individuals. Consequently, it enables easy access to vast amounts of information and immediate updates on current events. Users have the freedom to share links, photographs, and videos, maintain contact with friends regardless of their location, expand their social circles, participate in online forums, and gain exposure to diverse perspectives from around the globe (Mfenyana et al., 2013).

Examples of the most generally used popular social media platforms on a global scale include Facebook, X formerly Twitter, YouTube, Instagram, TikTok, WhatsApp, Telegram, and LinkedIn among others which serve hundreds of millions of users that are connected with billions of links (Y. Li et al., 2017). As an illustration, individuals utilize LinkedIn as a platform for business and professional networking, providing information about their professional

achievements. In contrast, Facebook predominantly caters to users seeking connections with family members and friends. X (Twitter), on the other hand, functions as a microblogging tool for those interested in reading or sharing the latest news updates (Y. Li et al., 2017).

Social media is used in many diverse sectors like education, politics, and business (Raut & Patil, 2016; Smits, 2013). For instance, in business, entrepreneurs and enterprises have the opportunity to engage with their customer base, promote and sell their products, introduce new services, enhance brand visibility, and bolster brand reputation, all at a relatively low cost (Vardhini et al., 2019). Research indicates that a substantial 81% of small and medium-sized businesses actively leverage various social media platforms to achieve their goals (Edosomwan, 2011; Vardhini et al., 2019). Furthermore, an interesting statistical insight reveals that 78% of individuals who voice their concerns or complaints about a brand through X (Twitter) anticipate a response within just one hour (Edosomwan, 2011; Vardhini et al., 2019). This highlights the increasing importance of social media as a potent instrument for effective and prompt customer engagement within the business realm. Additionally, governments utilize social media as a channel for informing, interacting with, and assisting citizens (Mfenyana et al., 2013). Figure 1.1 is adapted from a Pew Research Center report, indicating that while young adults were among the initial adopters of social media, usage among older adults has surged in recent times. As of 2021, 75% of adults are active on at least one social media platform.

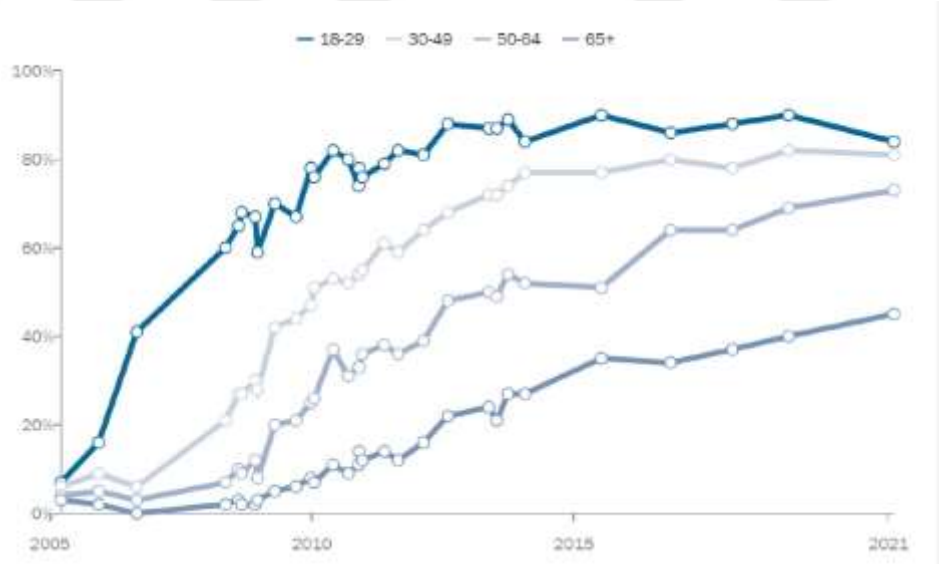


Figure 1. 1. % of U.S. adults who say they use at least one social media site, by age (*Anonymous 2*).

While social media undeniably provides significant benefits, it also presents a potential environment for cybercrimes and inappropriate online conduct, including hacking, fraudulent activities, scams, and cyberbullying. This thesis concentrates on cyberbullying, which is experiencing a global increase in prevalence.

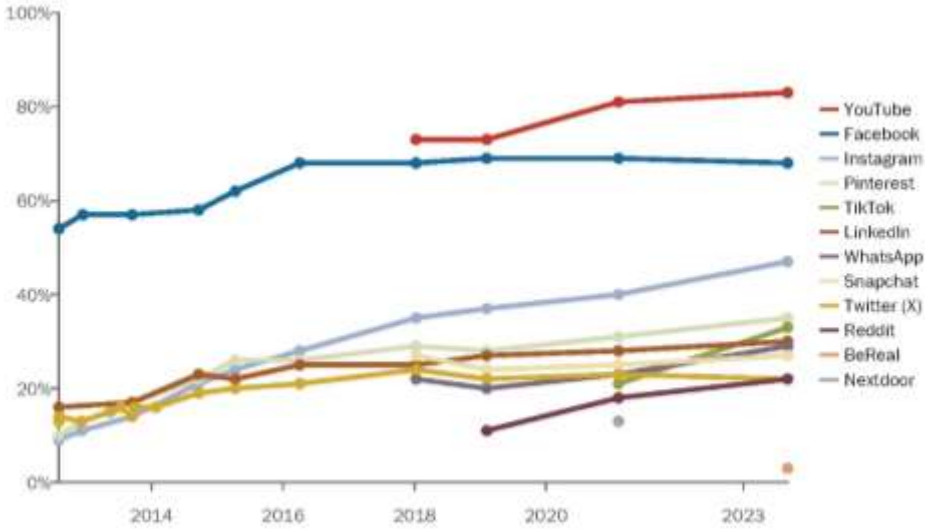


Figure 1. 2. Comparisons of Social Media Usage based on Popular Platforms (Anonymous 3).

Figure 1.2 depicts the insight into social media usage by US adults based on the popularity of these platforms. Youtube, Facebook are among the most popular social media sites in America. According to this, adults report using Instagram, while smaller percentages use platforms like TikTok, LinkedIn, Twitter (X), and BeReal.

1.3. Cyberbullying

As more people connect through social media networks, online threats like cyberbullying (CB) are becoming a very serious problem in today's world. CB has been recognized by the Centers for Disease Control and Prevention (CDC) as a serious public health issue, and the CDC has cautioned the public about it (Aboujaoude et al., 2015). CB is a type of virtual bullying in which individuals or groups regularly send offensive or aggressive messages to other people through electronic or digital media with the intention of hurting or upsetting them (Tokunaga, 2010). This definition of CB emphasizes crucial elements such as the technological aspect, the hostile nature of the behavior, and the intention to cause harm (Aboujaoude et al., 2015). In a more recent study, cyberbullying is defined as intentional and aggressive behaviors displayed on the internet, repeated over time, and directed towards a victim who is unable to defend themselves (Emmery et al., 2021).

Apart from being conducted online, CB entails sending or posting derogatory or harmful text or images using information and communication technology (ICT) devices. Depending on the context, CB is referred to by various terms such as internet bullying, cyber harassment, electronic bullying, online bullying, and digital bullying, all of which encompass the concept of CB (Kowalski & Limber, 2007; Samara et al., 2017). Cyberbullying can take various forms, including text, images, collages, memes, and other content (Hasan et al., 2023).

CB exhibits similar traits to traditional bullying: there exists a power imbalance between the perpetrator and the target, the harm inflicted is deliberate, and the behavior is repetitive over time (Corcoran et al., 2015; Dehue et al., 2008). However, due to the internet serving as the communication medium, additional variables have surfaced, rendering geographical location, timing, and physical presence irrelevant factors in the perpetration of CB. Unique aspects of CB encompass the anonymity of the bully, who conceals their identity behind a virtual mask, making it difficult to identify them or discern their location of origin. The larger potential audience for the abuse being carried out since online information is viewable by any one and can be liked or shared as many times as possible, the information can keep resurfacing at any time 24/7 and is very hard to remove or delete once shared chaired (Emmery et al., 2021). Nevertheless, not every cyberbully operates under the veil of anonymity. Instant messaging platforms such as WhatsApp, text messaging, and emails can all serve as mediums for cyberbullying if individuals involved suffer harm from the content conveyed. Some cyberbullies convey threats and harassment to their targets through instant messaging, SMS, or email. Others propagate rumors or spread gossip, inciting public animosity and even organizing anti-victim campaigns.

Publicity is an important factor in CB described as the size of the audience communicating through social media, either privately or publicly (Sticca & Perren, 2013). Publicity attacks can take private forms such as email or public forms such as Twitter and Facebook. According to a study conducted by Ditch the label, one of the world's leading anti-bullying organizations, the top social media sites where people experience cyberbullying are Facebook, Instagram, Snapchat, YouTube, and Twitter (*Anonymous 4*). Public cyberbullying is regarded as more detrimental than private cyberbullying (Sticca & Perren, 2013). An instance of private bullying is the silent treatment, a form of "relational aggression" demonstrated by a partner (Young & Nelson, 1999). In this form of private social exclusion, a partner is deliberately ignored and rebuffed; for instance, an individual may send numerous text messages to someone and receive no response. This can lead to psychological harm. This is very common with most people communicating on WhatsApp these days, as if you send a message to someone and the recipient of the message has read the message as it is appeared on WhatsApp with a blue tick, the recipient may end up not responding to the messages. This will in turn put the sender under psychological stress.

Technological advancements have significantly contributed to the widespread occurrence of cyberbullying. For instance, the proliferation of cell phones, including camera phones, has led to instances where compromising photos of students taken in school locker rooms or restrooms are circulated via email throughout the school or posted on various websites (Raskauskas & Stoltz, 2007). Such incidents are unfortunately commonplace among youth worldwide on a daily basis. For instance, a student may take a picture of their classmate in the shower and distribute it throughout the school using social media networks. Another example involves sixteen-year-old David Knight, who faced teasing and taunting from peers at school. His situation worsened when a

friend sent him a link to a webpage titled “Welcome to the page that makes fun of David Knight,” which contained derogatory comments directed at David and his family (Raskauskas & Stoltz, 2007).

The goal of CB actions is mainly to threaten, harm, humiliate, and endanger fear and helplessness in the victim (Young & Nelson, 1999). However, it should also be noted that the reason for cyberbullying, as indicated by cyberbullies and non-bullies, encompasses factors such as low self-esteem or the need to enhance one’s self-image, a craving for power and dominance, finding amusement in the act and seeking revenge (Smith et al., 2008). In a research study by Comparitech (*Anonymous 5*), which interviewed by asking parents about whether their children are exposed to cyberbullying, the results obtained showed that cyberbullying has steadily grown and rapidly increased from 2011-2018 as is shown in Figure 1.3.

Figure 1.3. The percentage of children being cyberbullied based on countries between 2011-2018 (Bozyiğit et al., 2021)

According to research conducted in Canada, approximately one-quarter of students have encountered cyberbullying (Q. Li, 2006). Other studies from various countries indicate that between 10% and 25% of young individuals experience cyberbullying (Agatston et al., 2007; Ortega et al., 2009; Patchin & Hinduja, 2006; Rivers, I., & Noret, 2010). Similarly, in a survey conducted on cyberbullying in American middle schools, findings revealed that 18% of the total sample (N = 3767), 25% of female participants (n = 1915), and 11% of male participants (N = 1852) reported experiencing cyberbullying at least once in the year preceding the study. Among the children who disclosed experiencing cyberbullying, 53.2% stated that the perpetrator was a fellow student from their school, while 37% identified the cyberbully as their friends (Kowalski & Limber, 2007).

An analysis of survey research involving 384 respondents under the age of 18 revealed that 11% acknowledged engaging in cyberbullying, 29% claimed to have been victims of cyberbullying, and over 47% reported witnessing instances of cyberbullying. The primary platforms for cyberbullying were found to be chat rooms, followed by computer text messaging and email, respectively (Patchin & Hinduja, 2006). In another study conducted in 2009, which

involved a sample of 1,671 adolescents aged 12 to 17, 5% reported experiencing both traditional bullying and cyberbullying (Ortega et al., 2009). According to the 2014 EU Kids Online report, approximately 20% of children aged 11 to 16 are susceptible to cyberbullying.

According to a study conducted by the Cyberbullying Research Center spanning from 2007 to 2023, an average of 31.2% of students involved in the studies reported being victims of cyberbullying at some point in their lifetime. Moreover, the study reveals a notable increase in the rate of cyberbullying from 18.8% in 2007 to 54.6% in 2023.

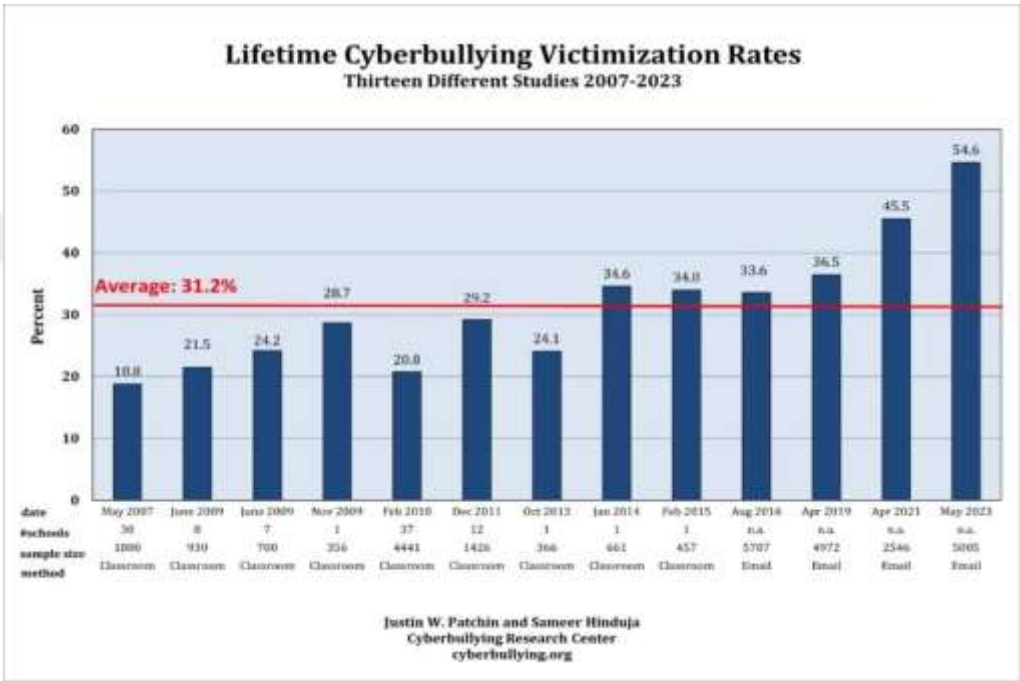


Figure 1.4. Life time Cyberbullying victimization Rates (*Anonymous 6*)

The enhanced internet infrastructure and widespread accessibility of affordable devices, such as computers and tablets, have led to the widespread adoption of social media, subsequently giving rise to the alarming phenomenon of cyberbullying. While these technologies offer undeniable benefits, their misuse poses significant challenges.

1.3.1. Components of Cyberbullying

Cyberbullying comprises various elements that shape its occurrence and effects. Research on cyberbullying varies depending on these elements. At its core, the phenomenon involves several actors, including the bully, victim, bystander, and platform (Emmery et al., 2021; Sticca & Perren, 2013). These actors can be defined as follows:

- Bully: This term generally denotes an individual who intentionally uses profanity, intimidation, or aggression to dominate or evoke fear and distress in others.

- Victim: The person battered by the bully. Victims often struggle to defend themselves and are frequently subjected to the power dynamics established by the bully.
- Bystander: An observer of the bullying incident who is not directly involved. Bystanders can demonstrate support for the victim by providing positive feedback and endorsing their actions.
- Platform, Context, and Modality: An additional important consideration is the platform on which cyberbullying occurs, as well as the modality used, which may include the publishing of abusive images, videos, or text messages.

Blogs and social media networking sites, which offer more open space on the internet after the last decade, have become the most common avenue for CB. Everyone can use these virtual settings for free. Individuals can develop and post messages, and blogs without limitation. These spaces, on the other hand, can be used to mock and shame other people. (Bhat, 2008) conducted a research study that found that cyberbullies publish the victims' blog entries with remarks pertaining to their physical appearance, intellectual capabilities, well-being, and sexual orientation. An additional study conducted by (Slonje & Smith, 2008) revealed that the most prevalent form of cyberbullying identified by participants (approximately 46 percent) is cyberbullying through comments, videos, and images. In this study, bully was considered a major factor since it's the one that purposefully uses profanity, intimidation, or hostility to assert control or instill fear and anxiety in others.

1.3.2. Forms of Cyberbullying

Cyberbullying can occur in a variety of ways, such as sharing content, or sending intimidating and insulting messages using ICT tools. According to researchers, cyberbullying can be categorized into 7 different forms including; flaming, denigration, impersonation, harassment, outing, cyberstalking, and exclusion (Beale & Hall, 2007; Beran & Li, 2008; Emmery et al., 2021; Van Hee, Jacobs, Emmery, Desmet, et al., 2018). A brief explanation of the cyberbullying forms is given below;

- Denigration: Denigration entails the dissemination of gossip, rumors, or false information about an individual, constituting a prevalent manifestation of CB in recent times. The pervasive nature of this phenomenon is largely attributed to the widespread utilization of social media platforms. In the contemporary digital landscape, individuals engaging in denigration often exploit the expansive reach and instantaneous connectivity afforded by various online platforms to spread misleading narratives about their targets.

- Flaming: Flaming often entails sending or posting aggressive, angry, or irritating remarks with the intent of inflaming the emotions and sensitivities of others. These statements or communications do not promote or contribute to the subject at hand but rather aim to socially or mentally damage another person and impose dominance over others. The majority of flame occurs on online discussion boards, forums, and comment sections of news stories (e.g., sports, celebrities, current events), etc.
- Impersonation: This behavior is predominantly observed on popular social media platforms like Facebook, Twitter, and Instagram. The perpetrator of cyberbullying often employs a deceptive tactic by assuming the guise of a cybercrime victim, fabricating a fake account, and impersonating the individual they intend to target. Through this method, the cyberbully places the victim in a precarious position and proceeds to transmit inappropriate messages to others, exacerbating the harm caused by their actions.
- Harassment*:* Threats and harassment share similarities as forms of cyberbullying. Harassment involves the use of profanity, sending explicit video images, or delivering insulting text messages. The key distinction in harassment lies in the sustained and persistent nature of this bullying behavior.
- Outing: Outing pertains to the act of revealing confidential and embarrassing personal details about the target, such as their phone number, photos, or videos, without obtaining the individual's consent. The cyberbully intimidates the victim by making threats of harm, violence, or even death, often coupled with demands for the victim's address information.
- Exclusion: This category of cyberbullying contains the prevention, undesirability, and exclusion of the victim from various online spaces, including social media platforms, forums, electronic message groups, and online gaming. The objective is to cause emotional suffering to the victim by deliberately excluding and isolating them from significant online communities.

Direct and indirect cyberbullying are the two main classifications into which various forms of cyberbullying can be classified. While indirect cyberbullying happens without the victim's knowledge or swift notice, direct cyberbullying entails the victim being specifically targeted. Direct cyberbullying, for example, would be removing someone from an internet platform; indirect cyberbullying would be things like sharing someone's personal information online or publicly publicizing it (Vandebosch & van Cleemput, 2009).

Once cyberbullying content is published online, it becomes challenging to remove, and even if removed, it may resurface later, causing continuous worry and anxiety for the affected

individuals. Therefore, detecting cyberbullying is crucial to address its harmful effects (Emmery et al., 2021).

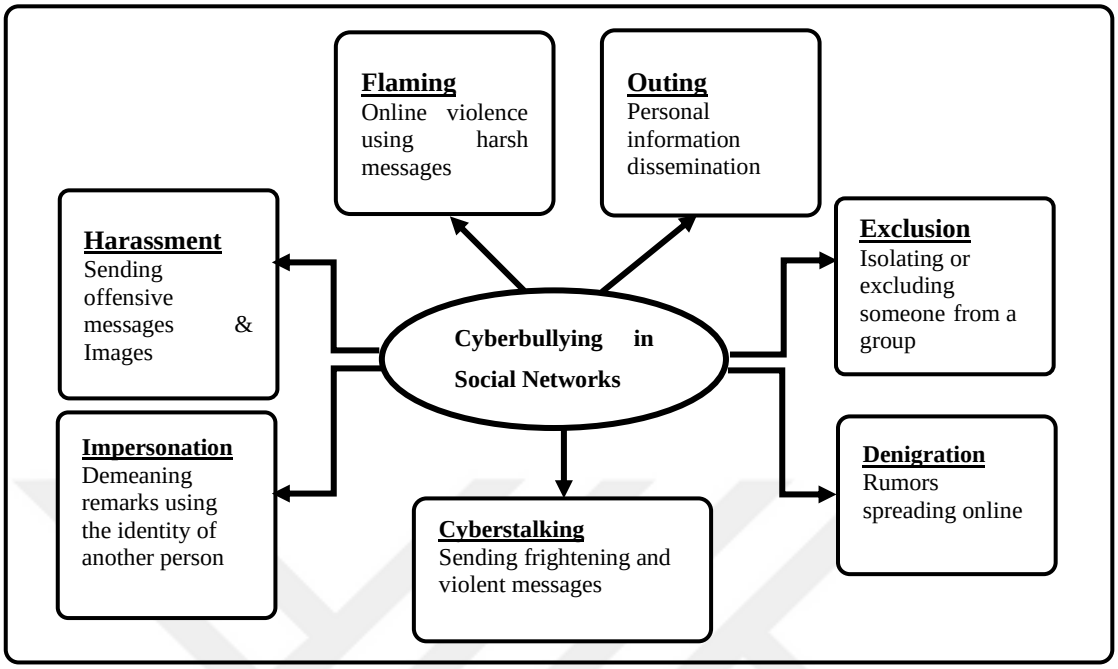


Figure 1.5. Summary of Different Forms of Cyberbullying in Online Social Media Networks

1.3.3. Cyberbullying Tools

One of the researcher's main goals in attempting to explain the nature of cyberbullying is to determine which instruments are employed to carry out this bad conduct. According to a study conducted by (Bhat, 2008), the most commonly used cyberbullying tools include social networking sites like Facebook, Twitter, and Instagram, etc., instant messaging, email, chat rooms, blogs, websites, text messages, pictures, and video clips.

In the last decade, blogs and social networking platforms, which expanded the open spaces on the internet, have transformed into potential sources of cyberbullying material. These virtual spaces offer unrestricted freedom to individuals to create and share materials. However, it is noteworthy that these environments can also be exploited for humiliating, ridiculing, and launching attacks against people individually or in the form of groups.

1.3.4. Effects of Cyberbullying

The consequences of CB can vary based on the bully's motivations, age, and the gender of the victim. Both empirical and non-empirical research consistently highlight the severe impact of Cyberbullying on victims. Research indicates that CB victimization can lead to numerous severe consequences, including heightened suicidal tendencies, diminished self-esteem, and various emotional reactions such as fear, frustration, anger, depression, and a desire to retaliate (Emmery et al., 2021; Hinduja & Patchin, 2007, 2010; Patchin & Hinduja, 2006; Van Hee, Jacobs, Emmery,

DeSmet, et al., 2018). Consequently, victims often withdraw from socializing with friends and participating in activities, which is the primary objective of the bully person. Cyberbullying has the potential to drive a victim to commit suicide. In the United States, there have been many reported examples of cyberbullying leading to a teen's suicide (Bhat, 2008; Hinduja & Patchin, 2010).

Mental and emotional Impact

Previous studies have documented various emotional repercussions linked to cyberbullying. For instance, a study involving 20 students in Grades 8 to 12, conducted by (Kowalski & Limber, 2007; B. Spears, 2009), examined the effects of cyberbullying on participants' well-being. The research identified three main themes:

- Experience of intense negative emotions such as unhappiness, sadness, embarrassment, powerlessness, loneliness, depression, and anxiety.
- Fear related to engaging in social situations, leaving the house, and the breach of previously safe spaces.
- Harming some one’s reputation and self-esteem, self-perception, feelings of rejection, and the potential for public humiliation.

Comparable emotional responses were noted in larger sample sizes. Some studies indicated that, in the range of 35% to 43%, self-reported victims of cyberbullying asserted they experienced no emotional repercussions (Hinduja & Patchin, 2007; Ortega et al., 2009). This indicates that over half of the victims experience negative consequences. This could potentially occur due to the reluctance of cyberbullying victims to disclose the emotional impact they have experienced or their lack of awareness regarding the effect they have been subjected to.

The severity of bullying may correlate with its impact on victims. Among those who report experiencing emotional consequences, a significant portion, up to 93% of participants, describe feelings such as anger, frustration, fear, increased paranoia, suspicion, or a blend of these emotions (Cassidy et al., 2012; Hinduja & Patchin, 2007; Ortega et al., 2009; Raskauskas & Stoltz, 2007). Research also indicates that depression and depressive symptoms are common reactions to cyberbullying. For instance, a study found that victims of cyberbullying exhibited more depressive symptoms compared to a control group (Raskauskas, 2009). Moreover, in a study examining the impact of electronic bullying on 41 adolescents, 60% of victims reported feelings of sadness, hopelessness, and depression (Raskauskas & Stoltz, 2007). Similarly, another study involving a larger sample found that 13% of adolescents reported symptoms of depression attributable to cyberbullying (Ortega et al., 2009). These findings underscore the significant concern of depression in cyberbullying, leading to fear and distress.

Suicidal tendencies has been widely a great concern in CB and researches have investigated a link between cyberbullying and suicidal tendencies (Cassidy et al., 2012; Hinduja & Patchin, 2010). Research studies indicate that approximately 20% to 42% of victims reported experiencing suicidal thoughts subsequent to being cyberbullied (Cassidy et al., 2012; Hinduja & Patchin, 2010; Ortega et al., 2009). In another study conducted to investigate the effect of CB, 19% of respondents acknowledged having attempted to commit suicide and in their findings, researchers emphasized that CB were twice likely to have committed suicide (Hinduja & Patchin, 2010).

Mediation models have been employed to explore the direct and indirect connections between CB and suicidal ideation. In a survey involving 426 students aged 10-21, participants were queried about self-control, mood, deliberate self-harm, suicidal tendencies, and experiences of cyberbullying victimization. The findings revealed that the association is not straightforward but is influenced by adverse mood, emotional responses, and self-control. Additionally, it was found that a positive mood and higher levels of self-control are inversely associated with self-harm (Hay & Meldrum, 2010).

Different media have reported about suicidal tendencies as a result of CB. The most famous one is of a 14- year old Hannah Smith who decided to hang herself after negative comments were posted on her ASK.FM page (Milosevic, 2016). Hanah received sequence of rude messages on her ASK.FM website, providing instructions to "ingest bleach," "acquire cancer," and "perish". Hanna's experience is a proving point that cyberbullying can lead to suicide. Similarly, Alex Teka of New Zealand passed away in 2006 at age of 12. According to media reports, prior to her passing, she purportedly endured CB, receiving abusive and threatening emails and texts from classmates at her school (*Anonymous 7*). Allegedly, the bullying intensified after her mother raised concerns with school authorities. In the United States, Megan Meier, took her own life in 2006 at the age of 13. She had developed an online relationship with someone she believed to be a 16-year-old boy named "Josh" on the social networking site Myspace. However, "Josh's" profile was allegedly created by the mother of one of Megan's former friends, as a ruse to gain Megan's trust. Shortly before Megan's death, "Josh" sent her cruel and hurtful messages Theguardian (2008). In another case, Ryan Patrick Halligan ended his life at the age of 13 in 2003. His father shared his story after his passing, which was documented on a website created by his parents in his honor. Leading up to his death, Ryan faced challenges including rumors about his sexuality and unwanted online sexual advances from an unidentified person posing as a boy, which led to rejection by his female classmates (*Anonymous 8*). These interactions occurred through instant messaging. In 2011, Jamey Rodemayer an American adolescent aged 14, killed himself because he has faced CB for years due to sexuality. Before his passing, he regularly received messages on his social media containing derogatory remarks targeting his sexual orientation, such as "Jamie is foolish, homosexual, overweight, and unattractive; he should perish!" Another post stated, "I wouldn't

mind if you were to die, nobody would, so why not just do it? It would bring joy to everyone!". Eventually, he ended his life as a result of constant threatening messages of which he couldn't bear (*Anonymous 9*). Similarly, in 2012, 14-year-old Şeniz Erkan, a Turkish living in Australia jumped Infront of a speeding train after being targeted by cyberbullies on her social media website. It was reported that prior to her death, unidentified people hacked into her social media and sent disgusting messages to her friends (*Anonymous 10*). Additionally, in 2018, a 14-year-old, an Australian by the name Dolly Everett who was also a former child model, murdered herself after being a targeted by CB. As reported by her family, she was constantly involved in social media and received online threats which resulted to her death (*Anonymous 11*). All of these cases reported above highlight the sort of injury and severe harm that are caused as a result of CB, and the nature of misuse of Information and Communication Technology (ICT). However, it should also be noted that most of the cases go unreported.

CB has profound mental and emotional effects on individuals, often leading to heightened levels of anxiety and depression, as well as a significant decline in self-esteem. The relentless nature of online harassment can induce feelings of isolation and worthlessness, exacerbating existing mental health challenges. In extreme cases, cyberbullying has been associated with suicidal ideation, as victims grapple with the overwhelming emotional burden imposed by the constant threat and humiliation. The psychological toll of cyberbullying underscores the urgent need for preventive measures, support systems, and mental health resources to mitigate the severe consequences on individuals' emotional well-being.

Academic Impact

The relationship between CB/ cyber victimization and academic performance is seen in various ways as described below;

When it comes to academics, learners need to concentrate during lessons and after lessons in order to perform at their best in school, However, being cyberbullied will make it harder for students to have maximum concentration in order to pay more attention in class. In a study investigating the impact of bullying on students and academic achievement, it was found that 62% of cyberbullying victims reported difficulty focusing on schoolwork, with 5% admitting to frequently thinking about cyberbullying incidents during class (Tjavanga, 2012). Additionally, research by (Slonje & Smith, 2008) revealed that experiencing cyberbullying is associated with lower academic achievements. Comparative studies were conducted to assess differences between cyberbullied and non-bullied students, which yielded significant findings. Cyberbullied students exhibited behaviors such as carrying weapons to school (13% vs. 0.6%), receiving detentions or suspensions (0%-21% vs. 11%), and skipping classes (20%-33% vs. 4%) (Hinduja & Patchin, 2007; Ybarra et al., 2007) Poor academic performance is also linked to cyber victimization, as evidenced by lower scores in exams, both overall and in recent assessments, as well as engagement

in cheating behavior (Hay & Meldrum, 2010; Hinduja & Patchin, 2007, 2010; Ybarra et al., 2007). Similarly, victims of cyberbullying struggle to concentrate during exams and lessons, leading to a decline in academic performance (Cassidy et al., 2012).

CB significantly impacts academic performance by causing emotional distress, social isolation, and decreased self-esteem among victims, hindering their ability to concentrate and engage in learning. The resulting physical health issues and avoidance behaviors, such as skipping classes, contribute to missed assignments and falling grades. Prolonged exposure to cyberbullying can impair cognitive functioning, affecting memory and concentration. Additionally, the negative school climate fostered by cyberbullying can disrupt the overall learning environment, making it challenging for students to focus on their studies. It is imperative for schools and communities to implement anti-bullying measures and support systems to address the emotional well-being of those affected by cyberbullying and create a positive atmosphere conducive to academic success.

Social Impact

Several studies have explored the connection between cyberbullying victimization and social standing. For instance, in a study involving 1,700 students in grades 5-11, (Fetchenhauer & Belschak, 2009) found that decreased social popularity in chat rooms was linked with experiencing cyberbullying. Participants rated their popularity in both chat rooms and classrooms, as well as reported the percentage and nature of cyberbullying they encountered. There was, however, no link between social popularity at school and cyber victimization. Whereas results revealed that CB corelated with cybervictimization, there was no relationship between participating in traditional bullying and cybervictimization (Fetchenhauer & Belschak, 2009; Rivers, I., & Noret, 2010; Vandebosch & van Cleemput, 2009). It has also been discovered that female students, but not male students, are more likely to experience more than one cyber victimization position (victim, bully, bystander) (Rivers & Noret, 2010).

Several researchers have delved into the influence of self-perceptions on the social implications of cyberbullying. Studies with small sample sizes have shown a strong association between cyber victimization and avoidance of social interactions (Barbara Spears et al., 2009). However, findings from studies with larger sample sizes present conflicting results. For instance, numerous studies have discovered that many victims of cyberbullying do not perceive significant trauma in terms of their peer relationships (Cassidy et al., 2012). In fact, when traditional forms of victimization and cyberbullying are taken into account, cyberbullying does not necessarily indicate lower social competency skills. Instead, some cyberbullies exhibit well-developed social skills that they utilize to manipulate others (Vandebosch & van Cleemput, 2009).

As per the finding of the study, it shows that most victims of cyberbullying do not perceive themselves as encountering social difficulties. However, for some individuals, there exists an emotional aspect to cyber victimization that holds significance. Further investigation is required to

explore the connection between cyber victimization and social well-being. Nevertheless, it is imperative to recognize that the academic setting serves as a crucial context where the emotional and social repercussions of cyberbullying hold significance. Consequently, educators should be attentive to the potential impacts on students and consider providing support to those who may be vulnerable to cyberbullying. The importance of understanding and addressing the emotional and social dimensions of online victimization is underscored, emphasizing the responsibility of teachers in fostering a supportive and safe learning environment.

1.3.5. Cyberbullying Awareness and Prevention

Cyberbullying has gained significant prominence because a large portion of the population, including both the younger generation and adults, is actively engaged in social media. Many individuals have become targets of the detrimental effects of cyberbullying, leading to internationally reported incidents of life-threatening experiences for instance in a 2016 UNESCO report which highlighted cases and policies from across the globe, emphasizes that CB is not solely a public health apprehension but indeed a violation of human rights (Haber, 2016). There is an urgent need to thoroughly investigate and study cyberbullying for the purpose of prevention and detection. Challenges in addressing and preventing cyberbullying include the timely identification of incidents, communication with internet service providers (ISPS), law enforcement agencies (LEA) and the identification of both victims and perpetrators. Diverse theoretical measures have been proposed to address and engage people in reducing CB threats. Also, some research has been done from social perspective. Some of the prevention measures are listed below:

- *Sensitization*: Various sensitization campaigns to produce responsiveness about the various forms of cyberbullying, its consequences, and the importance of fostering a positive online culture have been put in place. These campaigns often target schools, communities, and online platforms to inform individuals about responsible online behavior and the potential harm caused by cyberbullying. For example, every third Friday in June is designated as National Bullying and Cyberbullying Awareness Day, during which schools and organizations come together to promote awareness of bullying and cyberbullying and their adverse effects (*Anonymous 12*). Additionally, various national and international institutions have been established with the goal of enhancing individual cybersafety and security, such as Kiva and the Finnish Cyberbullying Prevention Program (*Anonymous 13*) among others.
- *Reporting mechanisms*: Cyberbullying is considered violation in many social media platforms. Many social media sites, including Facebook and Twitter, have reporting guidelines that can assist users in reporting bullying and, as a result, promote a secure

online environment. These consist of determining the target audience settings, blocking specific users, reporting inappropriate behavior (Krutka et al., 2020).

- Some Schools usually contemplate various forms of disciplinary measures for cyberbullying, including detention after-school, suspension, and expulsion, depending in some extreme scenarios on the extent of the incident. In certain U.S. educational institutions, there is a mandate to provide education to students about cyberbullying and its consequences. This is integrated into the curriculum.
- *Parental Involvement*: Engaging parents in cyberbullying awareness is crucial. Educating parents about online safety, checking their kids' online activities, and encouraging open communication can contribute to a supportive environment both at home and in the virtual space.
- *Encouraging Empathy and Digital Citizenship*: Promoting empathy and positive digital citizenship is fundamental in preventing cyberbullying. Teaching individuals to respect others online, understand the impact of their words and actions, and promote a culture of kindness contributes to creating a more inclusive and supportive online community.
- *Policy implementation*: Policy implementation is essential to discourage or prohibit cyberbullying, necessitating the establishment of effective reporting mechanisms to law enforcement. Penalties, in the form of fines, should be imposed on individuals found guilty of engaging in cyberbullying. As an illustration, in the United States, with the exception of Montana, every state has enacted legislation addressing bullying, and a significant majority have broadened their anti-bullying laws to encompass cyberbullying. Almost half of the states explicitly prohibit cyberbullying, with some providing clear definitions of severe forms, including threats and cyberstalking, treating them as criminal offenses reportable to law enforcement. Louisiana, for example, imposes penalties on cyberbullies aged 17 and above, including fines of up to $500, imprisonment, or a combination of both (*Anonymous 14*).

Several studies in the arenas education and psychology have explored CB and its defining traits (Smith et al., 2008; Wright et al., 2009). These researchers are dedicated to offering guidance to parents and school authorities on how to address incidents of cyberbullying and the appropriate procedures for reporting such occurrences. However, very little attention has been devoted to automatic identification of Cyberbullying. There have also been few studies in the domain of automated cyberbullying detection. Machine learning techniques have been utilized in the automatic detection of cyberbullying.

1.4. Machine Learning

Machine learning (ML) involves the development of computing algorithms designed to mimic human intelligence. It is defined as the capacity of a computer to autonomously learn how to make decisions using available data and experiences (Mitchell, 2006). ML is characterized as an automated system that utilizes past experiences to inform future decision-making processes. ML, which sits at the interface of statistics and computer science, seeks to create algorithms that can learn from data and make predictions based on those insights (Provost & Kohavi, 1998). These algorithms enable computers to construct models from sample inputs, facilitating data-informed predictions and decisions, rather than strictly adhering to programmed instructions. The data utilized in ML, referred to as training data, serves as the basis for making decisions, which may include classification or prediction for new objects or data. Computers use learning algorithms to categorize incoming data, which allows them to use ML in a variety of fields, including banking, entertainment, medical, pattern recognition, computer vision, spacecraft engineering, and more (Mitchell, 2006; Naqa & Murphy, 2015). Based on the type of data labeling, ML is divided into three categories: semi-supervised learning, unsupervised learning, and supervised learning. Through learning algorithms, computers classify new data, thereby leveraging ML in various domains such as pattern recognition, computer vision, medicine, spacecraft engineering, finance, entertainment, and more.

1.4.1. Supervised Learning

Using a set of known input-output pairs, supervised learning entails inferring an unknown relationship between input and outcome. This method's labeled output makes it appropriate for problems like regression and classification, in which the objective is to classify or predict using given samples. In supervised learning, every piece of data is labeled, resulting in instance-label pairs (X and Y values in a function, f(X)=Y, for example). This data is separated into two categories: testing data, which is used to compare predicted class labels with real ones in order to assess the correctness of the model, and training data, which is used by the learning algorithm to build a model.

Supervised learning generates models using labeled training instances (François Chollet, 2017). The main challenge with supervised learning lies in obtaining a huge number of precisely labeled instances from every class, as annotated data can be challenging to acquire, often requiring human experts or costly experiments (L. Liu et al., 2003; Zhou & Li, 2005). Classification and regression are two common tasks in supervised learning. In the current study, our emphasis is on the classification problem related to cyberbullying. We apply several state-of-the-art ML classifiers to automatically find instances of cyberbullying (CB).

1.4.2. Unsupervised Learning

In unsupervised learning, the class of the training data are unknown or not given i.e. only input samples are given to the learning system (Mitchell, 2006). For instance, in the domain of identifying CB, a dataset may exist and annotated as either harmful or not harmful. In cases where the training data lacks such labels, the algorithms rely on this unlabeled data falls under the category of unsupervised learning. The training data is the primary distinction between supervised learning and unsupervised learning. In case of unsupervised learning, the class of the training data are unknown, indicating the absence of a supervisor, while in supervised learning, the class of the training data are known. Unsupervised learning is primarily employed in tasks such as clustering, outlier detection, and dimensionality reduction scenarios.

1.4.3. Semi Supervised Learning

Both supervised and unsupervised learning strategies are combined in semi-supervised learning (SSL) driven by its pragmatic advantages in constructing more effective classifiers compared to traditional supervised learning, all while mitigating the expenses associated with labeling. The process of labeling an observation demands human involvement and, at times, specialized equipment, thereby amplifying the overall cost. Although there is an abundance of unlabeled data available, the challenge lies in harnessing their potential effectively. In response to these issues inherent in numerous learning problems, the concept of the SSL has emerged as a strategic solution (Goldberg, 2010; Van Engelen & Hoos, 2020). Semi supervised is used in various applications such as Natural Language Processing, Image categorization, spam filtering among others (Goldberg, 2010; Sadarangani & Jivani, 2016).

.

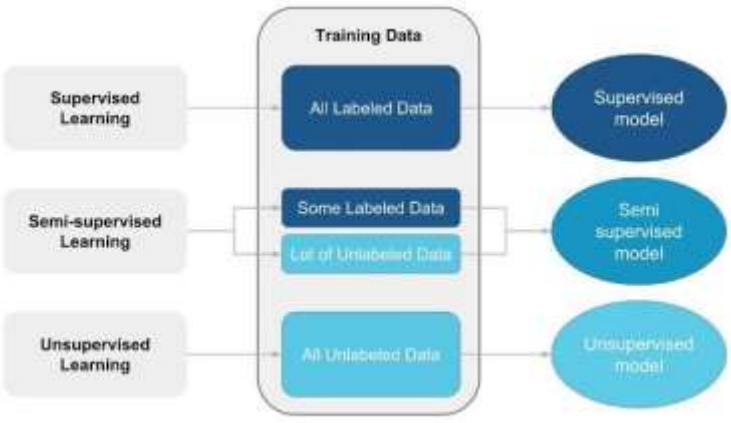


Figure 1.6. Outline of different Machine Learning approaches (*Anonymous 15*)

1.5. Deep Learning

Deep learning (DL), a subset of ML, relies on artificial neural network structures (François Chollet, 2017; Ian et al., 2016). The term "deep" doesn't imply a deeper understanding achieved by

the approach but rather signifies the idea of successive layers of representations, indicating how many layers contribute to a model or the representation of multiple layers to learn hierarchical representations of data (François Chollet, 2017). Alternative terms for deep learning could have been "layered representations" or "hierarchical representations learning". Neural networks are inspired by the structure of neurons in the human brain and their connections. Over the past decade, neural architectures based on DL have significantly enhanced the performance of various intelligent applications, including image and speech recognition, as well as machine translation. Consequently, there has been a notable rise in the adoption of deep learning in industry solutions, particularly in the field of natural language processing. A classifier based on neural networks consists of interconnected units, where input units represent terms and output unit(s) represent categories. During the classification of a test document, the input units are assigned the term weights of the document. Subsequently, the information traverses the network, and the output unit(s) determine the category to which the document belongs.

1.6. Natural Language Processing

The speech and language processing related studies have an extensive background and has been explored across diverse fields such as computer science, electrical engineering, linguistics, and psychology. The collaboration of these distinct disciplines has played a pivotal role in advancing speech recognition, computational linguistics, natural language processing, and computational psycho-linguistics (Jurafsky, 2019; Meurers, 2012).

Natural Language Processing (NLP) is a field of artificial intelligence focused on enabling computers to comprehend human language (Kyle, 2021). NLP allows machines to interpret and respond to natural language inputs by converting human language into structured representations. Its applications span various domains, including language translation, voice-controlled systems, chatbots, and speech-to-text software. NLP is extensively employed in consumer products and increasingly in enterprise solutions to enhance business operations and productivity (Kyle, 2021). Part-of-speech tagging, word sense disambiguation, speech recognition, named entity recognition, co-reference resolution, sentiment analysis, and natural language production are just a few of the many tasks it includes. These challenges help computers comprehend and produce human language more precisely. NLP, ML, and DL relationship is described in the Figure 1.7 (Sowmya et al., 2020). The development of NLP applications has been heavily influenced by ML methods, and more recently, DL has been frequently utilized in building NLP applications.

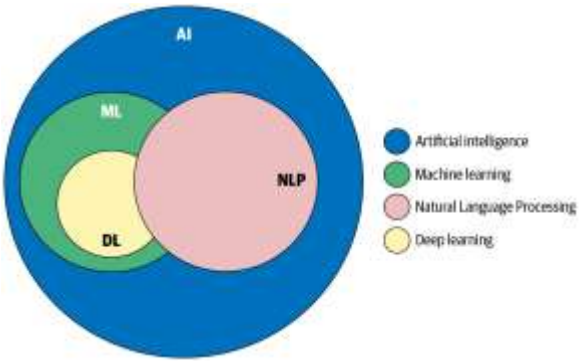


Figure 1.7. How NLP, ML, and DL are related (Sowmya et al., 2020).

1.7. Text Classification

The widespread adoption of the internet and advancements in communication technologies have resulted in a noteworthy increase in the amount of text shared online, particularly through social media platforms. Consequently, managing this vast volume of online information has become increasingly challenging. Text classification is a method used in text categorization to provide accurate and timely results. Classification entails assigning predefined class labels to existing data. To accomplish this, a classifier is trained using a set of labeled data, and the trained model is then applied to label unseen data. This process of classification is commonly known as supervised learning (Michel, 1997). Classification can be divided into binary and multiclass classification depending on the number of classes involved. In binary classification, there is a single class label, and the classifier evaluates an instance to determine whether it belongs to that specific class or not. Instances that belong to the specific class are deemed relevant, while those outside it are labeled as non-relevant. Conversely, if there are more than two classes, the classification is termed as multiclass classification (Qi & Science, 2007). Text classification specifically involves assigning predefined class labels to given text or documents. This process encompasses various types, including sentiment analysis, cyberbully detection, web classification, blog classification, etc. The goal of text categorization is to categorize documents into specific predefined categories by utilizing document features.

1.8. The Aims and Objectives of this Thesis

In Turkey, and the world in general, CB is considered a serious problem that needs serious attention since its effect can result into emotional and psychological effects like depression, stress, anxiety and in a more severe cases suicide as presented in the previous sub sections. The ease and speed with which CB can propagate globally is another terrible consequence. The majority of the time, incidents of CB are found after the victim has already suffered harm.

To address the threats of CB, numerous researchers and scientists have proposed solutions using ML, DL, and NLP to develop models capable of identifying CB within textual contexts. From the literature, it is evident that the a good number of CB detection studies have focused on

the English language, with fewer studies dedicated to the Turkish language. Additionally, existing studies typically analyze or investigate either English or Turkish independently. Turkish is an agglutinative language, meaning that its morphological structure is different from that of non-agglutinative languages like English. As a result, the grammatical differences in Turkish may affect how well texts are classified. Inspired by this discrepancy, the goal of this thesis is to identify instances of cyberbullying in the English and Turkish languages while emphasizing their distinctions due to their different morphological structures. The objective is to evaluate how standard state-of-the-art methods affect both languages.

Another gap in text classification arises from the common occurrence of having much more unlabeled data compared to labeled data. Labeled data annotation is an expensive, labor-intensive process that requires experience from an expert. Therefore, leveraging a large size volume of unlabeled data alongside a smaller set of labeled data could help mitigate the labeling effort. We explored how these unlabeled data can be effectively utilized, specifically examining whether semi-supervised learning can be employed across different languages. We developed and introduced a novel semi-supervised learning approach for detection of cyberbullying in Turkish and English, aiming to enhance classification accuracy.

1.9. Contributions of this Thesis

The thesis makes several significant contributions, enhancing the understanding and application of techniques in cyberbullying detection:

- It provides a thorough comparison of machine learning, deep learning, and large language models (BERT) for detecting cyberbullying in both English and Turkish. This includes evaluating various classifiers and methods such as SVM, LR, MNB, DT, RF, K-NN, CNN, RNN (LSTM, GRU, BiLSTM, BiGRU), CNN-LSTM, and LSTM-CNN, utilizing benchmark datasets. Furthermore, it delves into different variants of BERT tailored for English and Turkish, conducting comprehensive analyses.
- The study evaluates feature weighting techniques such as TF-IDF, TF, Binary with N-gram, and BOW features, elucidating their impact on machine learning classifiers.
- The study examines the influence of popular word embedding methods, including Word2Vec, FastText, and GloVe, in both custom-trained and pre-trained forms, across machine learning and deep learning methods.
- It thoroughly compares the effectiveness of various BERT variants with machine learning and deep learning approaches for cyberbullying identification.
- The study evaluates the impact of integrating BERT variants with machine learning to enhance classification performance.

- The study explores the impact of large language models with Low-Rank Adaptation (LoRA), an optimization technique to enhance performance.
- Additionally, the study evaluates and compares the effectiveness of traditional semi-supervised learning methods such as co-training and, self-training and proposed a novel semi-supervised approach utilizing self-training with large language models (BERT) for its ability to improve classification performance while reducing the dependency on labeled data. Through these experiments, the study aims to contribute to the advancement of cyberbullying detection across languages and improve the efficiency of automatic labeling efforts in text classification tasks.

1.10. Outline of the Thesis

This thesis comprises five chapters structured as follows:

Chapter One: This chapter provides a comprehensive overview of bullying and cyberbullying, encompassing various types, effects, and tools utilized in perpetrating them. Additionally, it introduces key techniques such as machine learning, deep learning, and natural language processing.

Chapter Two: This chapter offers a synthesis of pertinent literature concentrating on cyberbullying detection in both English and Turkish languages. It reviews existing studies to contextualize the research landscape.

Chapter Three: Detailed descriptions of the datasets utilized in this study and the methodologies employed are presented in this chapter. Furthermore, the proposed model architecture is elaborated upon, outlining its components and functionalities.

Chapter Four: This section presents the results derived from the experiments conducted throughout the study. The impact of these results is thoroughly discussed, supported by visual aids such as charts to enhance comprehension.

Chapter Five: The conclusions drawn from the study and its key findings are articulated and deliberated upon in this chapter. Additionally, avenues for future research pertinent to the thesis are outlined, providing direction for further inquiry in the field.

2. PRELIMINARY WORK

In this section of the thesis, the research work related to CB detection which includes ML and DL is presented.

2.1. Cyberbullying Detection

In the existing literature, numerous studies have been dedicated to the detection of CB, aiming to monitor and prevent harmful online activities. However, the majority of these studies concentrate on the English language, while the number of studies addressing CB in Turkish is notably scarce. In this thesis, we provide a comprehensive review focusing on previous research endeavors that have tackled the task of CB detection within both English and Turkish contexts.

2.1.1. Cyberbullying Related Literature for English Datasets

After examining existing literature, it becomes evident that the pioneering investigation into automated CB detection was conducted by Yin et al. (2009). Their study utilized various feature extraction techniques such as N-grams with TF-IDF, alongside a linear kernel support vector machine (SVM) classifier, to pinpoint instances of online harassment across three distinct datasets: Kongregate, Slashdot, and MySpace. Kongregate is an online gaming platform featuring chat-rooms for player interaction, Slashdot serves as a hub for discussions on technology, and MySpace provides forums for registered users to engage in topic-based discussions. The study achieved its most notable result, an F-score of 0.313, when applied to the MySpace dataset. The researchers concluded that identifying online harassment is feasible by complementing TF-IDF with sentiment and contextual feature attributes. It's worth mentioning that while the experiment's outcomes may not cover all scenarios exhaustively, they provided a catalyst for subsequent research endeavors.

Further investigations in this field were carried out by Reynolds et al. (2011). They employed ML algorithms like C4.5, knearest neighbor (K-NN), and SVM classifiers utilizing a dataset comprising user comments extracted from Formspring.me, a website structured around questions and answers. The evaluation results revealed that the C 4.5 classifier, and instance-based learner from the data mining tool like Weka exhibited superior performance compared to both K-NN and SVM classifiers, achieving an accuracy rate of 78.5%.

Dinakar et al. (2011) introduced an automated CB detection system utilizing ML classifiers such as SVM, JRIP, C4.5 on comments from YouTube, categorized into three CB topics of, race, intelligence, sexuality. The authors created models for binary classification focusing on separate labels as target classes, as well as multiclass classification models training all labels simultaneously. They observed that binary classification methods focusing on labels for individual

goals yielded better results in classifying each CB topic's labels, with accuracies ranging between 70% and 80%, compared to the multiclass classification models.

Munezero et al. (2013) explored the effectiveness of conventional ML methods such as Multinomial Naïve Bayes (MNB), SVM, and Decision Tree (DT) for identifying antisocial textual behavior within Movie review and Wikipedia datasets. Their experiments demonstrated that both SVM and MNB attained an F-score of 0.98 using the Weka platform. Dadvar et al. (2013) utilized a hybrid methodology to detect instances of CB within a dataset sourced from YouTube. Their approach integrated expert system techniques with ML methods, incorporating three feature categories: content, activity, and user attributes. The optimal performance for the hybrid approach was attained, yielding a notable AUC value of 0.76. Al-garadi et al. (2016) Identified instances of CB in Twitter posts by employing four distinct feature categories: content, activity, user characteristics, and network attributes. They employed various classifiers including Naïve Bayes (NB), SVM, Random Forest (RF), and KNN. The most promising outcome was achieved using a RF classifier, with an F-score of 0.936, indicating its effectiveness in identifying cyberbullying incidents.

Chen et al. (2017) Identified instances of online harassment using two ML classifiers, namely MNB and SVM, alongside TF-IDF for feature extraction across three datasets: Formspring, Kongregate, and Slashdot. The researchers achieved the highest recall of 0.78 using SVM on the Myspace dataset. Zhang et al. (2017) suggested a method for identifying CB employing DL models. The authors utilized word pronunciation as features for a Convolutional Neural Network (CNN) model, which was tested on social media posts gathered from Formspring.me and Twitter. The accuracy scores demonstrate that the pronunciation-based CNN surpasses the baseline CNN using randomly produced word embeddings.

Hani et al. (2019) proposed an approach to detect CB using ML techniques. They utilized sentiment analysis algorithms and the TF-IDF method for feature extraction. Classification tasks were performed using NN and SVM classifiers with various n-gram language models. The neural network outperformed SVM, achieving an accuracy of 92.8% with a 3-gram model, compared to SVM's accuracy of 90.3% with a 4-gram model. However, the study highlighted the challenge of limited training data size in CB detection, suggesting the necessity of a larger dataset to improve model performance.

Muneer & Fati (2020) conducted a comparative analysis of ML techniques for identification of CB using a Twitter dataset. They explored various machine learning and ensemble methods, including NB, Logistic Regression (LR), SVM, RF, Adaboost, SGD, and LightGBM classifiers, along with word embedding techniques. Word2vec, TF-IDF were employed for feature extraction. The authors found that LR achieved the highest F1-Score of 0.928 when used with Word2vec embeddings.

Shah et al. (2020) evaluated the effectiveness of ML in identifying CB using a dataset from Twitter. They employed TF-IDF for feature extraction along with ML algorithms such as SVM, MNB, LR, RF, and SGD. LR emerged as the most effective classifier, achieving an accuracy of 0.930. Islam et al. (2020) conducted automated cyberbullying detection using four ML classifiers—DT, SVM, RF, NB—utilizing BOW, TF-IDF for vectorization on Twitter and Facebook datasets, respectively. They reported the highest accuracy of 0.81 with the SVM classifier. Jain et al. (2021) investigated the efficacy of ML classifiers for identification of cyberbullying on social media platforms, employing RF, LR SVM, as ML classifiers. For vectorization, they used BoW, TF-IDF, along Word2vec on datasets from Wikipedia and Twitter. LR as well as SVM achieved the highest F1-Score of 0.939.

Alotaibi (2021) Introduced an automated approach for CB detection integrating features from three distinct DL models—transformer block, CNN, and Bidirectional Gated Recurrent Unit (BiGRU)—compared with two ML algorithms, RF and LR. The method aimed to categorize Twitter comments as offensive or non-offensive. The suggested framework achieved an accuracy of 87.99% by combining three renowned datasets and evaluating the effectiveness of the proposed approach. Mahmud et al. (2022) analyzed the efficacy of five distinct ML models—RF, LightGBM, XGBoost, LR, and RF—for CB identification utilizing textual features from a Twitter dataset. TF-IDF was employed for feature extraction. The authors reported that LightGBM yielded the best result, attaining an F-score of 0.844 and surpassing the other models. This achievement underscores LightGBM as the most effective model among the evaluated ML algorithms, showcasing its potential for enhancing cyberbullying detection. Alam et al. (2021) proposed an ensemble based classification approach for detection of cyberbullying. Their approach incorporates DT, LR, and a Bagging ensemble model as classifiers, employing TF-IDF and Bigrams for feature extraction. By applying this methodology to a Twitter dataset, the authors observed that the Bagging ensemble model achieved the highest accuracy, reaching an impressive 0.96%.

Balakrishnan et al. (2020) implemented a robust model for detecting CB, utilizing users' psychological attributes and employing machine learning ML techniques on an extensive Twitter dataset. Their objective was to categorize tweets into specific groups based on their characteristics, with a particular emphasis on identifying instances of CB. The classification was established across four primary categories: 'bully,' 'aggressor,' 'spammer,' and 'normal,' defined by psychological traits extracted from the tweets, including sentiment and emotions. To conduct their experiments, the authors investigated the efficacy of three ML classifiers—NB, RF, and J-48. Results indicated that the J-48 classifier outperformed its counterparts, achieving an impressive accuracy score of 0.928. Dharma et al. (2022) conducted a text classification task on news articles using CNN along with three word embedding techniques: Word2vec, GloVe, and FastText. They observed that CNN coupled with FastText yielded an accuracy of 0.972. Nevertheless, the impact of GloVe besides

Word2vec was relatively marginal compared to FastText. Therefore, the authors came to the conclusion that the datasets used may have an impact on the word embedding methods chosen.

Bharti et al. (2022) conducted a comparative study of ML algorithms, including DT, NB, SVM, and DL methods such as BiLSTM, utilizing BOW alongside GloVe for feature extraction to detect CB in tweets. They found that BiLSTM combined with GloVe yielded the most favorable result, achieving an F-score of 0.942. Iwendi et al. (2020) investigated and developed a DL-based solution for identification of CB on social media. They collected data from Twitter, Instagram, Facebook and experimented with DL algorithms including Bidirectional Long Short-Term Memory (BiLSTM), Gated Recurrent Unit (GRU), and Recurrent Neural Network (RNN), using Word2vec for feature extraction. BiLSTM proved the best with an accuracy of 0.821, surpassing LSTM, GRU, and RNN.

Singhs et al. (2022) conducted a comparative evaluation to assess the effectiveness and performance of ML algorithms such as NB, LR, SVM, RF, XGBoost and DL models including LSTM, and GRU on a Twitter dataset, utilizing TF-IDF and GloVe word embeddings for CB identification. They found that GRU emerged as the top performer, obtaining F-1 score of 0.924 in the overall assessment.

Banerjee et al. (2020) introduced a NN-based method for identifying CB, incorporating CNN with GloVe on a Twitter dataset. They achieved an accuracy of 0.937 in their study. (Paul & Saha (2022) proposed the utilization of BERT for CB detection. They introduced a Fine-tuned BERT model and evaluated it on three distinct datasets from Twitter, Formspring, and Wikipedia. The performance of the BERT model was compared with ML and DL including; SVM, LR, CNN, LSTM, and RNN. The BERT model achieved the overall F1-Score of 0.94 specifically on the Twitter dataset.

Maslej-Krešňáková et al. (2020) conducted a comparative analysis of DL models for automated detection and recognition of toxic comments. They explored various DL architectures and word embedding models, including CNN, LSTM, BiLSTM, GRU, and BERT, using TFIDF, Word2vec, and FastText for feature engineering. BERT exhibited the highest overall performance, achieving an accuracy of 0.9741. Raj et al. (2021) investigated cyberbullying detection using a combination of ML such as XGBOOST, SVM, NB, LR and DL algorithms including CNN-LSTM, LSTM, BiLSTM, GRU, BiGRU, Attention-BiLSTM) with TF-IDF, GloVe, and FastText word embeddings on datasets sourced from Wikipedia. The BiGRU model with GloVe word embeddings demonstrated superior performance, achieving an accuracy of 0.9898.

Mazari et al. (2023) proposed a method for detecting hate speech on social media platform using a multi-aspect approach to text classification. They utilized BiLSTM and BiGRU models stacked on GloVe together with FastText to construct DL models, achieving a ROC-AUC score of 98.63% in detecting hate speech. Marzieh et al. (2020) evaluated the efficiency of BERT in

identifying cyberbullying on social media using Twitter datasets. They reported an F1-score of 92%, highlighting the efficacy of employing BERT for cyberbullying detection.

Pericherla & Ilavarasan (2021) performed a study to assess the effectiveness of various WE techniques and LLM in identifying CB using a Twitter dataset labeled for a racism, sexism. They found that LLM achieved notably high F1 scores compared to conventional semantic WE models. Agarwal et al. (2020) used BiLSTM with attention layers over the Wikipedia dataset for CB detection. Their proposed method surpassed existing approaches in terms of precision, recall, and F- score, achieving scores of 89%, 86%, and 88%, respectively.

Murshed et al. (2022) proposed a hybrid method called Dolphin Echolocation Algorithm (DEA), which cartels Elman-type RNN with an optimized DEA to reduce training time for CB detection on Twitter. They compared the performance of their method with BiLSTM and RNN models, demonstrating that their proposed approach achieved higher accuracy, precision, recall, and F1 score, with scores of 90.45%, 89.52%, 88.98%, and 89.25%, respectively. Teng & Varathan (2023) recommended a Transfer learning approach for CB identification, conducting a comprehensive study, comparing the performance of ML methods like LR, SVM, and various transfer learning methods such as DistilBERT, RoBERTa, and Electra. These models were trained with diverse features like Word2vec, GloVe, and FastText using a substantial dataset extracted from the widely-used ASK.FM platform among adolescents. Their results showed that DistilBERT achieved the highest accuracy of 0.974, surpassing traditional ML methods, underscoring the significant impact of transfer learning in enhancing model performance.

In a study by Yu (2013), the effectiveness of self-training for sentiment analysis across various domains was examined using datasets from movie reviews, news articles, and blog posts. The findings proved promising results, indicating that self-training could be a viable approach for opinion classification when labeled data are scarce. However, the study noted that the effectiveness of self-training may vary depending on the domain. Additionally, the study highlighted the challenge of applying Semi-Supervised Learning (SSL) for domain adaptation. Nahar et al. (2014) proposed a SSL method to enhance training data for CB detection. They utilized a fuzzy SVM algorithm and applied it to datasets from Myspace, Kongregate, and Slashdot. Their approach involved automatically extracting and expanding the training set from unlabeled text, while learning was initiated with a small labeled set. Comparative analysis with supervised methods like RF, NB, and LR revealed that K-FSVM achieved the highest performance, with an F1-score of 0.82.

In their study, Kiritchenko & Matwin (2001) investigated the application of Co-training for email classification, highlighting the importance of selecting the appropriate classifier to enhance the performance of Co-training algorithms. Their results indicated that Co-training could effectively be utilized for email classification, with SVM outperforming NB as a learning algorithm. Meanwhile, Xia et al. (2015) proposed a Co-Training SSL sentiment classification

method based on Dual View Bag of Words representation. They conducted experiments on four datasets, including Books, DVDs, Electronics, and Kitchen. Their approach involved automatically generating anonymous reviews and representing review texts using pairs of BOW with contrasting viewpoints. When combining Co-Training with LR, exhibited the highest performance on the kitchen dataset, achieving an average accuracy of 0.780.

2.1.2. Cyberbullying Related Literature for Turkish Datasets

The majority of studies on detecting CB focus primarily on analyzing English text, with fewer investigations conducted for Turkish. A notable study by Özel et al. (2017) examined CB detection in Turkish using Instagram and Twitter datasets, employing ML classifiers like MNB, DT, SVM and K-NN along BOW for word vectorization. They enhanced classifier accuracy through chi-square and information gain feature selection methods, achieving better results when considering both emoticons and words as features. MNB classifier showed the highest accuracy, reaching 0.84 with feature selection.

In another study by Bozyigit et al. (2019), they used data from Twitter (Tweets) to make a dataset in Turkish language for CB detection. They evaluated various neural network models, applying information gain feature ranking to reduce dimensionality and enhance performance, achieving a 91% F-measure after feature selection. Kilimci & Akyokus (2019) assessed word embedding models in conjunction with DL algorithms for Turkish text classification using news datasets. Their findings indicated that employing LSTM with FastText achieved the best accuracy result of 0.935, demonstrating the effectiveness of integrating DL algorithms with word embedding models.

On & Yeniterzi (2020) investigated the use of CNN in CB detection in Turkish, comparing it with ML algorithms like MNB, SVM, and LR. They utilized pre-trained word embeddings and found CNN with FastText to be the most effective, reaching a 0.937 accuracy. Zampieri et al. (2020) in their study have focused on abusive language detection in Turkish, combining datasets and utilizing SVM to achieve an F-score of 0.773.

Bozyiğit et al. (2021) examined various ML methods, including SVM, LR, MNB, AdaBoost, and RF, for identifying CB in Turkish text utilizing social media attributes. They employed BOW, TF-IDF for vectorization. Among these algorithms, AdaBoost achieved the highest accuracy of 0.901, albeit with slower prediction times due to the integration of multiple SVM models. AdaBoost was favored for its high recall metric in identifying cyberbullying posts, underscoring a significant association between CB and specific features such as sender followers.

Cihan Ates et al. (2021) conducted CB detection in Turkish text by evaluating nineteen ML algorithms, including KNN, SVM, LGBM, SGD, RF, Voting Classifier, DT, XGBoost, LR, Extra trees, LSVM, Gradient Boosting, Perceptron, Bernoulli NB, Adaboost, MNB, LDA, and QDA. The process of feature extraction employed TF-IDF and BOW, resulting in the highest

performance of 0.909 F1-Score when paired with LGBM. Çelik & Koç (2021) evaluated the categorization of a varied Turkish news dataset into six predetermined groups. They utilized pre-trained Word2Vec, FastText, and TF-IDF for feature extraction, along with NB, RF, LR, SVM and ANN algorithms. The highest accuracy of 95.75% was attained with SVM.

Karayiğit et al. (2021) conducted a comparative evaluation for the detection of offensive and abusive Instagram posts in Turkish language using SVM, DT, RF, and LR as ML classifiers and CNN as a DL classifier. They utilized oversampled and non-oversampled versions of the Instagram dataset and employed vectorization techniques like TF-IDF and Word2vec. The best performance, with an average F-score of 0.974, was reported using CNN on the oversampled dataset. Özberk & Çiçekli (2021) proposed the use of BERT for identifying offensive language in Turkish tweets, exploring various fine-tuned BERT models, including BERT Turk case, Distil Bert Turk, and ConvBERT Turk. These were compared with ML algorithms such as LR, DT, RF, and SVM, with BERT Turk achieving the most effective performance, scoring 0.823. The study highlighted the potential for performance enhancement through pre-training the BERT model on Turkish tweets using character-level pre-trained models.

Guven (2022) Conducted a comparative analysis using diverse pre-trained version of BERT models, including, Electra, Distilbert, Albert for sentiment analysis on Turkish hotel and review datasets. They proposed a text filtering technique to eliminate words conveying opposing sentiments in positively and negatively labeled text. Electra demonstrated the best overall in performance, with 0.923 F-Score. Koksal & Akgul (2022) performed a comparative analysis using various pre-trained DL algorithms to classify the TTC 3600 News Turkish dataset. DL algorithms included pre-trained DNN, CNN, LSTM, and GRU, along with FastText for feature vectorization. Achieving an F-score of 0.96, CNN with FastText was deemed the most effective approach for news article classification.

In their study, Coban et al. (2023) investigated the identification and cross-domain evaluation of CB within Turkish Facebook activity contents. They utilized several ML algorithms including LR, MNB, SVM, and DL models utilizing CNN, and various RNN such as LSTM, BiLSTM, GRU, BiGRU, and BERT. Feature extraction involved considering term weighting methods like Binary, TF, TF-IDF, as well as word embeddings such as Word2vec. By employing BERT, they attained a macro-average F1 score of 0.928

In a study conducted, Najib et al. (2023) for comparative evaluation for detection of CB for on English and Turkish language, authors utilized traditional feature weighting techniques such as binary, TF, TF-IDF, along with word embeddings including Word2vec, GloVe, FastText with traditional ML, DL and LLM techniques. Research findings revealed better performance on both languages with 0.9775 and 0.9892 F-score on English and Turkish when used with BERT variants

2.2. Comparison of this Thesis with the Previous Work

The prevailing academic research extensively focuses on studies centered on identification of CB, primarily within the English and Turkish languages. Scholars commonly collect data from prominent social media like Twitter, Instagram, Facebook alongside other web repositories like, UCI, Kaggle, Wikipedia, among various others. Our research stands out from previous studies in terms of both the datasets utilized and the methodologies employed. While most prior studies rely on publicly available datasets, our investigation is distinct in that we conducted experiments on four publicly accessible datasets: two for Turkish and two for English languages. Based on through exploration in the field and to best of our research in this domain, no existing study has conducted a comparative evaluation involving ML as well as DL and LLM methods for CB detection across both languages of English and Turkish within a single investigation.

As depicted in Table 2.1, previous studies mainly used pre-trained model such as Word2Vec, in conjunction with DL models. In contrast, our study showed the impact of BoW and N-grams in cross linguistic domain, and explores the impact of both custom and existing pre-trained word embeddings including FastText, GloVe, and Word2vec on ML as well as DL algorithms. Additionally, earlier studies rarely incorporated LLM (BERT) variants fully for both languages within a single model, a gap that our research addresses. To enhance the performance of ML, we integrated BERT with ML classifiers. Furthermore, we applied LoRA, a parameter optimization technique that aimed to optimize the performance of BERT models through parameter tuning. Moreover, considering the surplus of unlabeled data compared with the deficiency of labeled data, we utilized semi-supervised learning methodologies, notably self-training and co-training, to capitalize on both labeled and unlabeled datasets. Furthermore, we introduced an innovative self-training approach employing a large language model (BERT) aimed at augmenting the classification accuracy in tasks related to CB detection.

Table 2.1. Comparisons of the methods used in this thesis with previous studies

Studies in English					
Study	Dataset	BoW	Pre-trained word embeddings	Features	Classifier
(Yin et al., 2009)	Congregate, Slashdot, Myspace		✓	TF-IDF	SVM
(Reynolds et al., 2011)	Formspring.me	✓		TF-IDF	C4.5,Jrip,KNN, SVM
(Dinakar et al., 2011)	Youtube	✓		TF-IDF	Jrip, C4.5, SVM, NB
(Munezero et al., 2013)	Formspring	✓		TF-IDF	MNB, SVM and J-48 (DT)
(Yu, 2013)	News articles, movie reviews, blog posts	✓		TF-IDF	Self training
(Nahar et al., 2014)	Myspace, Kongregate, Slashdot	✓		TF-IDF	Self training F-SVM, RF, NB, LR
(Xia et al., 2015)	Books	✓		TF-IDF	Co-training LR
(Al-garadi et al., 2016)	Twitter			TF-IDF	SVM, RF, KNN, MNB

(Chen et al., 2017)	Congregate, Slashdot, Myspace	✓			TF-IDF	MNB, SVM
(X. Zhang et al., 2017)	Twitter				Word2vec	CNN
(Hani et al., 2019)	Twitter		✓		TF-IDF	SVM, NN
(Muneer & Fati, 2020)	Twitter	✓		✓	TFIDF, Word2vec	NB, LR, SVM, RF, ADB, SGD, LGBM
(Shah et al., 2020)	Twitter	✓			TF-IDF	SVM, LR, MNB, RF, SGD
(Islam et al., 2020)	Twitter, Facebook	✓			BOW, TF-IDF	DT, SVM, RF, NB
(Jain et al., 2021)	Wikipedia, Twitter	✓			BoW, TFIDF, Word2vec	SVM, RF, LR, MLP
(Kiritchenko & Matwin, 2001)	Email ✓				TF-IDF	Co-training SVM, NB
(Alotaibi, 2021)	Twitter	✓			TF-IDF	RF, LR, CNN, BiGRU, Transformer
(Mahmud et al., 2022)	Twitter	✓			TFIDF	RF, LGBM, XGBoost, LR, RF
(Alam et al., 2021)			✓		TFIDF	DT, LR, Bagging ensemble
(Balakrishnan et al., 2020)	Twitter				NA	NB, RF, J-48
(Dharma et al., 2022)	News stories			✓	Word2vec, GloVe, FastText	CNN
(Bharti et al., 2022)	Twitter	✓		✓	TF-IDF, GloVe	DT, NB, TF, SVM, BiLSTM
(Iwendi et al., 2020)	Facebook Instagram Twitter			✓	Word2vec	BiLSTM, GRU, LSTM, RNN
(Singhs et al., 2022)	Twitter			✓	TF-IDF, GloVe	NB, LR, SVM, XGBOOST, LSTM, GRU
(Banerjee et al., 2020)	Twitter		✓		GloVe	CNN
(Paul & Saha, 2022)	Twitter Wikipedia Formspring			✓	TF-IDF	CNN, LR, LSTM, BiLSTM, BERT, SVM,
(Maslej-Krešňáková et al., 2020)	Wikipedia			✓	TF-IDF, Word2vec, FastText, BERT	GRU, LSTM, BiLSTM+CNN, CNN
(Raj et al., 2021)	Wikipedia	✓		✓	FastText, TF-IDF GloVe	GRU, XGBOOST, SVM, NB, LR CNN, BiLSTM, BIGRU,CNN-LSTM, Attention-BiLSTM), LSTM
(Mazari et al., 2023)				✓	GloVe, FastText	BiLSTM and BiGRU
(Marzieh et al., 2020)				✓		BERT+CNN
(Pericherla & Ilavarasan, 2021)				✓	GloVe, FastText, Word2vec	Roberta, XL net, ALbert
(Agarwal					GloVe	BiLSTM with

et al., 2020)						attention layers
(Murshed et al., 2022)		✓			Word2vec,TF-IDF	BiLSTM, RNN, DEA-RNN, SVM, MNB, RF
(Teng & Varathan, 2023)	Ask.fm	✓		✓	Word2vec, GloVe, FastText	Transfer Learning, ML
			Studies in Turkish			
(Özel et al., 2017)	Instagram Twitter	✓			TF-IDF	SVM, MNB, KNN, DT
(Bozyigit et al., 2019)	Twitter		✓		TF-IDF,Word2vec	ANN
(Kilimci & Akyokus, 2019)	Haber, Miliyet, Huriyet, Haber			✓	Word2vec, GloVe, FastText	CNN, LSTM
(On & Yeniterzi, 2020)	Twitter			✓	FastText, Word2vec	CNN, MNB, SVM, LR
(Zampieri et al., 2020)	Twitter	✓			TF-IDF	SVM
(Bozyiğit et al., 2021)	Twitter	✓			TF-IDF	SVM, LR, MNB, Adaboost
(Cihan Ates et al., 2021)	Twitter	✓			TF-IDF	KNN, SVM, LGBM, SGD, RF, DT, XGBoost, MNB, ADB, QDA
(Çelik & Koç, 2021)	Twitter	✓			TF-IDF,Word2vec,FastText	SVM, MNB, LR, RF, ANN
(Karayiğit et al., 2021)	Instagram	✓			TF-IDF, Word2vec	CNN, SVM, DT, RF,LR
(Özberk & Çiçekli, 2021)	Twitter	✓			TF-IDF	BERT Turk case, Distil Bert Turk, ConvBERT Turk, LR, DT, RF, SVM
(Guven, 2022)	Hotel and Review					BERTturk,DistrilBertturk,ConBERTTurk
(Koksal & Akgul, 2022)	TTC 3600 News	✓			TF-IDF, FastText	DNN, CNN, LSTM, GRU
(Coban et al., 2023)	Facebook	✓			Binary, TF, W2vec, TF-IDF,	LR,MNB,SVM,CNN, LSTM, BiLSTM, GRU, BIGRU
(Najib et al., 2023)	CUD, ATC, KPD, FSD English and Turkish	✓	✓	✓	✓	Binary, TF, TF-IDF, Word2vec, GloVe, FastText	MNB,LR,SVM,RF, DT, KNN, BiLSTM, CNN-LSTM, BIGRU, LSTM-CNN, LSTM, GRU
In this thesis,	CUD, ATC, KPD, FSD English and Turkish	✓	✓	✓	✓	Binary, TF, TF-IDF, Word2vec, GloVe, FastText	MNB,LR,SVM,RF, DT, KNN, BiLSTM, CNN-LSTM, GRU LSTM-CNN, BiLSTM, LSTM, BERT+ML, BERT+PEFT (LoRA), Selftraining, Co-training, Self training+BERT

3. MATERIALS AND METHODS

In this section, we begin by examining the datasets employed in this thesis. Following that, we delve into the Python libraries employed for conducting this research, and subsequently explore the hardware utilized. Subsequent topics include a discussion on preprocessing techniques, followed by an examination of feature extraction methods. Lastly, we cover the methodologies encompassing machine learning, deep learning, and large language models.

3.1. Datasets

In this study, we performed cyberbullying detection focusing on English and Turkish languages. We considered four datasets with each representing a specific language. The implemented datasets are named as Cukurova University Dataset (CUD), and Abusive Turkish Comments (ATC) for Turkish language, and Kaggle_parsed_dataset (KPD), and Form Spring (FSD) for English language.

3.1.1. Kaggle (KPD)

Kaggle parsed dataset is an open source English language dataset that was constructed using a variety of social-media networks including Twitter, Wikipedia, YouTube (Elsafoury, 2020). The dataset contains 2806 bullying comment and 5993 non-bullying comments, as shown in Table 3.1, Table 3.2 includes samples from both classes while Figure 3.1 illustrates unique bully and non-bully words used with respect to KPD.

Table 3.1. Distribution of KPD

Distribution	**# of posts**
Cyberbullying Positive	2806
Cyberbullying Negative	5993
Total	8799

Table 3.2. Sample text from Kaggle (KPD) dataset

Text	**Label**
"If you throw trash out your window, you get a yellow star bumper sticker."	0
"@sdnoriko LOL as if Capcom listens. Watch them fuck this up too. I hope I am wrong. I love Okami."	0
"Allinit123, your\xa0hypocrisy\xa0is sickening, you ignorant loser."	1
"suck my dick yeaah babby pussy"	1
"@SlipResistantShoes\xa0In a few years, they'll make a damn movie about it?\n\xa0\nUncle Daddy Rides Again?"	0
"FACT.. YOU ARE A MORON"	1
"Yea, damn those Shriners and their little cars!"	0

Figure 3.1. Illustration of unique bully and non-bully words used in KPD

3.1.2. Form spring (FSD)

Formspring.me (FSD) comprises an XML file encompassing 13,158 messages submitted by 50 distinct users on the formspring.me platform, as detailed in a study by (Yin et al., 2009). As the dataset was originally in XML format, it was necessary to convert it to CSV format for our research purposes. The dataset is categorized into "CB Positive" and "CB Negative" sections. Messages categorized as negative denote those devoid of cyberbullying content, while positive messages entail instances of cyberbullying. The dataset contains 892 messages in the CB Positive category and 12,266 messages in the CB Negative category, as shown in Table 3.3, Table 3.4 includes samples from both classes while Figure 3.2 illustrates unique bully and non-bully words used with respect to FSD.

Table 3.3. Distribution of FSD

Distribution	# of posts
Cyberbullying Positive	892
Cyberbullying Negative	12266
Total	13158

Table 3.4. Sample text from FSD

Text	Label
Q: Q: Hey. Why you such a bitch ? A: Why thank yuh!	Yes
Q: iLove your nose ring! A: Awhh i love ur earrings!	No
Q: Your to young what love is bitch so do you use it! ! ! Dumb whore put your boobs away! ! ! A: Get some boobs a life !	Yes
Q: Your ugly x) ! A: Thanks but I think you need Jesus in your life	Yes
Q: Do u wnt 2 suk my dick ? A: Umm i don ' t wanna catch anything :/ sorry(:	Yes
Q: What was the last book you read ? A: The Miracle Worker	No
Q: What was the worst concert you went to ? A: Never been 2 1 tht was bad.	No

Figure 3.2. Illustration of unique bully and non-bully words used in FSD

3.1.3. Cukurova University Dataset (CUD)

CUD stands for Cukurova University Turkish Dataset, a Turkish Cyberbullying dataset which was created by a group of people who have expertise in Turkish language and overseen by a professor from the computer engineering department at Cukurova University (Öztürk, 2019). The dataset was collected from four prominent social media platforms: Twitter, YouTube, Facebook, and Instagram. Collection involved a combination of manual and automated browsing of these platforms, with a focus on Turkish text containing instances of cyberbullying. A JavaScript code was utilized to extract comments featuring cyberbullying expressions such as insults, swearing, and sexual assaults. The gathered text underwent manual inspection and classification, with labels assigned based on the presence or absence of cyberbullying. The dataset comprises a total of 15,658 Turkish text comments, evenly split with 7,998 labeled as positive and 7,999 as negative, resulting in a balanced distribution across classes, as presented in Table 3.5, Table 3.6 depicts samples from both classes while Figure 3.3 illustrates unique bully and non-bully words used with regard to CUD.

Table 3.5. Distribution of CUD

Distribution	**# of posts**
Cyberbullying Positive	7998
Cyberbullying Negative	7999
Total	15658

Table 3.6. Sample text from CUD

Text	Label
dünya hala dönüyorsa bu güzel insanlar?n hat?r?na dönüyor	0
orospu	1
pis sap?k senin gibi sap?k görmedi türkiye	1
piç kurusu	1
geber pislikk	1
yemekte ne var	0
ben senin kalbini yerim ya	0

Figure 3.3. Illustration of unique bully and non-bully words used in CUD

3.1.4. Abusive Turkish Comments (ATC)

The ATC, a publicly accessible Turkish Cyberbullying dataset intended for non-commercial research purposes, was developed by Turkish scholars (Karayiğit et al., 2021). This dataset was constructed by gathering content from Instagram accounts associated with various entities such as a Turkish magazine page (@2.sayfaofficial), football teams (@fenerbahce, @galatasaray), and football players (@ardaturan, @1volkandemirel), known for hosting contentious discussions and opinions where abusive and hateful comments are prevalent. Each entry in the dataset underwent manual classification as either abusive or non-abusive. It encompasses 10,528 abusive comments and 19,826 non-abusive comments posted between 2017 and 2019. Data extraction from Instagram was performed using the Python programming language and a dedicated Application Programming Interface (API) developed for this specific purpose.

As presented in Table 3.7, Table 3.8 depicts samples from both classes while Figure 3.4 illustrates unique bully and non-bully words used in line with ATC.

Table 3.7. Distribution of ATC

Distribution	**# of posts**
Cyberbullying Positive	10528
Cyberbullying Negative	19826
Total	30354

Table 3.8. Sample text from ATC dataset

Text	Label
Yerim hahahaha zengin boku sonuçta kriz ?????????? @officialyork	1
Sevgi bir duygu de?il, bilinç halidir. Sevgi, bilgi ve zaman enerjilerinin dura??d?r. Durak eylem sever...	0
ilikoazade Am?na koydum hassiktir git geymisin nesin dudaklar?n tavuk götü sanki	1
Senin kadar saçmalayan ba?ka bir adam daha yok bu memlekette. Ona ra?men hala seni seviyorum. Amk nas?l bir adams?n ??	1
ercan_toraman Hazar can?m ne yap?yorsun böyle kendine gel.	0
defol git tak?mdan amik embesili	1
moonapisa@moonapisa Bir Ömür Boyu Mutluluklar Dilerim ????	0

Figure 3.4. Illustration of unique bully and non-bully words used in ATC

3.2. Handling Class Imbalance problem

An imbalanced dataset refers to a dataset lacking a balanced distribution of classes, wherein the primary class of interest is underrepresented with only a few instances, while the negative class dominates. Put differently, it describes a scenario where classes within a dataset are distributed unequally (Han, J. K., & Kamber, 2001; Mohammed et al., 2020). Imbalanced datasets are frequently encountered across various practical domains such as text classification and detection of fraud in many fields (Nguyen et al., 2011). Addressing imbalanced data poses

challenges for building classifiers because many machine learning methods assume the same quantity of examples for each proposed class. To mitigate this issue, diverse approaches are being explored. Some techniques operate at the data level, aiming to rectify the imbalance by either oversampling the minority class (adding new instances) or under sampling the majority class (removing instances). Alternatively, at the algorithm level, methods involve adapting existing classifiers to better handle imbalanced datasets (Nguyen et al., 2011). As shown in Figure 3.5, under sampling entails reducing the number of majority class instances, while oversampling involves increasing the number of minority class instances, either by generating new instances or replicating existing ones (Mohammed et al., 2020).

Figure 3.5. Differences between undersampling and oversampling

In simple oversampling and undersampling techniques, the addition or removal of target instances is typically carried out randomly. However, this can often lead to the elimination of important occurrences or the addition of insignificant new cases (Krawczyk, 2016). Consequently, more sophisticated data-level strategies have been developed to preserve structural groups or generate new data in line with underlying distributions (Chawla.. et al., 2002). A largely excepted and popular oversampling methods include Random Oversampling, Synthetic Minority Oversampling Technique (SMOTE) as described in (Chawla.. et al., 2002), ADASYN (He et al., 2008), and Borderline SMOTE (Han et al., 2005). Meanwhile, notable undersampling methods encompass Random Undersampling, Condensed Nearest Neighbor (Hart, 1968), Tomek links (Tomek, 2010), and Edited Nearest Neighbors (Wilson, 1972). In this thesis, we utilize the SMOTE oversampling method with the assistance of the imbalanced-learn library as represented in (Lemaˆıtre et al., 2017) to address the issue of imbalance. SMOTE involves the random creation of synthetic minority class instances along the line segments connecting each minority class with a number of its neighbors (Chawla, 2002; Nguyen et al., 2011).

3.3. Used Hardware, Software Tools and Services

In this thesis, we employ a hardware and arrange of software, library, tools to build, customize and design our methods which are described as follows:

- Hardware: The proposed system was developed using a laptop computer Intel(R) Core-i5-3210M CPU @2.50GHz, Windows 10 Operating System and a remote Machine with GPUs (Google colab).
- Google Colab: Google Colab, also known as Collaboratory, is a research initiative designed for the experimentation of ML and DL models using high-performance hardware such as GPUs and TPUs (Bisong, 2019). This platform is freely accessible, offering a complimentary Tesla T4 GPU with a capacity of approximately 15GB. Users are allowed to initiate a session for a duration of 12 hours, as there is a concern about potential misuse (Ponnappa, 2019). Once the 12-hour time frame elapses, it is necessary to restart the session. For this thesis, we used Google colab, a freely available one and Google colab Pro, is one of the paid versions offered by Google that has higher availability of RAMs, GPUs and TPUs for advanced computationally expensive tasks and are required for more demanding DL tasks.
- Pandas: Pandas, a Python library, is an essential tool in data science, providing integrated routines for conducting common data manipulations and analyses (McKinney, 2011). It seamlessly integrates with other popular Python libraries like NumPy and Matplotlib, enhancing its capabilities and making it a multipurpose for all-inclusive data analysis and visualization tasks.
- NumPy: NumPy serves as a library for scientific computing in Python, featuring a powerful feature known as an ndarray (Harris et al., 2020; Van Der Walt et al., 2011). This multidimensional array is highly proficient in executing various mathematical operations efficiently. It enables the efficient handling of large datasets, supports advanced mathematical operations, and serves as a foundation for numerous scientific and mathematical Python-based packages (Harris et al., 2020).
- RE: Regular expression, commonly referred to as RE, is a robust and versatile method for manipulating patterns within strings, allowing for tasks such as pattern matching, substitution, and splitting. It facilitates tasks like eliminating URLs through expressions like re.sub(r'http\S+', '', text) and removing non-alphanumeric characters using re.sub(r'[^\w\s]', '', text) (Kaur, 2014).
- NLTK: NLTK is a collection of libraries for Python language and algorithms tailored for both symbolic and statistical NLP tasks (Bird, 2006). It provides a user-friendly platform that allows programmers, including beginners, to engage in NLP tasks without requiring extensive time for study or resource gathering (Lobur et al., 2011). NLTK finds widespread application in linguistics, artificial intelligence, and machine learning projects.

- Imbalanced Learn: Imbalanced-learn is a Python toolbox, open-sourced, specifically designed to provide a diverse range of methods to address the challenge of learning from imbalanced datasets, a common occurrence in machine learning tasks (Lemaˆıtre et al., 2017).
- Scikit-learn: Scikit-learn is one of the popular open-source ML library in Python that supports both supervised and unsupervised learning. It offers numerous pre-built ML algorithms spanning classification, regression, clustering, dimensionality reduction, and related tasks (Pedregosa, 2011).
- Keras: Keras is a high-level neural network API in Python designed to operate on established ML frameworks like TensorFlow and Theano (Abadi et al., 2016; François Chollet, 2017). This API facilitates straightforward prototyping of neural network algorithms and is compatible with both CPU and GPU architectures.
- Gensim: Gensim is a platform-independent, open-source tool written in Python that offers a variety of algorithms for constructing and examining models using textual data as input (Rehurek & Sojka, 2010).
- TensorFlow: TensorFlow, a popular platform, created by the Google Brain team, is an open-source tool designed for mathematical computation and extensive research in ML and DL. It operates on various CPUs and GPUs and offers interfaces in multiple programming languages like Python, C++, and Java. TensorFlow functions as a Python library, enabling the creation of computations in graph form of data flows, where nodes represent mathematical operations and edges signify the flow of data or tensors, which are multidimensional arrays (Abadi et al., 2016).
- Transformer: The Transformers Library, developed by Hugging Face, is a widely adopted Python library known for delivering cutting-edge models and tools for NLP. It has revolutionized NLP by introducing the Transformer architecture, founded on attention mechanisms. This architecture excels in managing extensive dependencies within textual data. The Transformers library offers a plethora of pre-trained models designed to perform tasks across various modalities, including text, vision, and audio (Wolf, 2020).

3.4. Preprocessing

Preprocessing data is a crucial step before extracting features, especially in social media where redundant and unimportant information abound. By eliminating irrelevant data, the model's performance enhances as it focuses solely on pertinent information. In this thesis, we conducted experiments involving commonly used pre-processing techniques: tokenization, removal of stop

words, conversion to lowercase, and stemming. These methods are frequently employed in data retrieval and text mining approaches (Kowsari et al., 2019; Naseem et al., 2021)

Tokenization involves the segmentation of text into tokens, which can represent words, phrases, or significant units. In this process, tokens can be derived solely from alphabetic characters or include alphanumeric characters, punctuation, and special characters separated by non-alphanumeric characters.

Stop words are words lacking meaningful content, such as prepositions, conjunctions, digits, and formatting tags. In certain contexts, like text classification or information retrieval, including stop words does not contribute positively to classification or search performance.

The third preprocessing stage includes transforming all uppercase letters into lowercase equivalents. This conversion ensures normalization as the meaning of a word remains unaffected by its case. Essentially, the entire text undergoes conversion to lowercase, thereby aligning words with various cases to a consistent lowercase format.

In the final preprocessing stage, stemming is applied to transform words into their root forms or stems, aiming to minimize the feature space. We utilized the famous stemmer (Porter Stemmer) for the English and (Snowball Stemmer) for Turkish from the NLTK to reduce words to their root forms. Stemming, particularly through the Porter stemmer, is instrumental in simplifying word representation by capturing essential linguistic roots, thereby reducing overall feature dimensionality.

3.5. Feature Extraction

During feature extraction, text is transformed into numerical representations that algorithms can readily comprehend for classification purposes. To capture the most relevant features, we employed both traditional text representation methods, which rely on term weighting or manually crafted features such as Binary, TF, and TF-IDF, as well as commonly used Word Embedding techniques like Word2vec, GloVe, and FastText, which are based on neural network architecture. It's important to note that BOW and N-gram are utilized for feature extraction in traditional term weighting methods.

3.5.1. BOW and N-GRAM

Based on the desired features to be extracted and the pre-processing techniques employed, BOW and N-gram are commonly utilized as traditional feature models or manually crafted features. The BOW model considers the words as features regardless of their placement or order of appearance or semantic meanings in the text (Çoban & Özel, 2018; Najib et al., 2023). It represents a collection of unique identifiable words observed in the associated text or document.

On the other hand, N-gram involves selecting n-consecutive characters or words in the text, where the model predicts the existence of n-1 words before analyzing the current word (Mossie & Wang, 2020). N-gram can be applied in two different ways: character-level and word-level. The distinction lies in the fact that character n-gram addresses misspellings and abbreviations as it is language-independent (Kanaris et al., 2007). In this thesis, we considered both BOW and N-gram by examining three consecutive characters in the text.

3.5.2. Term Weighting Methods

Information retrieval and text mining generally revolve around converting documents into numerical representations within a vector space. The initial step in creating a numeric document representation entail extracting terms, followed by assigning numerical weights to these terms. Therefore, the crucial process of term weighting involves allocating new weights to terms based on their importance within the corpus (Çoban et al., 2021). In this thesis, we employed various widely recognized term weighting techniques, including Binary, TF, and TF-IDF, for feature weighting. Additionally, we utilized BOW and N-gram methods for feature extraction.

Binary Weighting (BW)

In the binary weighting approach, rather than tallying the frequency of words in a document, we solely acknowledge the presence or absence of terms in the document like 1 as present and 0 as absent (Najib et al., 2023). For example, if term *t* is detected in document *D*, its weight is assigned as 1; conversely, if term *t* is not present, the weight for that term in document *D* is denoted as 0. This method effectively captures the binary aspect of term presence in a document, offering a straightforward indication of term inclusion.

Term Frequency (TF)

Term frequency refers to how often a specific feature appears within a document. This approach focuses solely on frequency information, without considering the sequence of the terms. Although a feature with a higher frequency may appear significant, it may not necessarily contribute substantially to the document, particularly if it is a common stop word. Moreover, due to variations in document lengths, a term's frequency in a longer document will naturally be higher compared to its frequency in shorter documents. To address this issue and mitigate the impact of document length on term frequency values, it is advisable to normalize the term frequency. This normalization entails dividing the term frequency by the document's length, which corresponds to the total number of occurrences of all terms in the document. Consequently, the TF of a term *t* for a document *D* is calculated as illustrated in equation 3.1. This normalization process ensures a more equitable representation of term frequency across documents with different lengths.

$$TF(t)\ \frac{\#\ of\ times\ term\ t\ appears\ in\ document\ D}{Total\ occurances\ of\ all\ terms\ in\ document\ D} \quad (3.1)$$

TF-IDF

TF-IDF, which stands for Term Frequency-Inverse Document Frequency, merges the concepts of Term Frequency (TF) and Inverse Document Frequency (IDF). It evaluates the ratio of the total number of samples in the dataset to the number of samples containing a particular word, which is frequent in a specific document yet rare across the corpus. Terms meeting this criterion receive higher numerical values, emphasizing their significance within the content of that document. The computation of the TF-IDF weight for a term *c* observed in a document *d* is depicted as follows in equation 3.2 (Hamdan et al., 2016).

$$TF - IDF\ (c,d) = TF(c,d) \times log\frac{N}{DF(c)} \quad (3.2)$$

Here, TF(c,d) denotes the raw term frequency of term *c* in document *d*, while DF(c) signifies the document frequency of term *c* across the entire collection. Lastly, *N* denotes the number of documents in the given collection.

3.5.3. Word Embedding Methods

Traditional feature weighting or term weighting methods encounter several challenges, including the loss of word order in documents, the creation of high-dimensional and sparse vector representations, and the inability to capture the semantic meanings of words. Consequently, word embeddings are recommended as an alternative. Word Embedding, a relatively new techniques compared to term weighting, is trained on large corpora and is based on the hypothesis of distribution, which proposes that words with similar meanings tend to appear in similar contexts. The concept of word embedding was initially introduced in 2003 by (Bengio et al., 2003), and in 2009, (Collobert, 2009) demonstrated that word embeddings trained on large corpora contain both syntactic and semantic information, proving to be a successful approach. Subsequently, research on word embeddings and their applications proliferated, with representation methods such as Word2vec (Mikolov et al., 2013), FastText (Bojanowski et al., 2017), and GloVe (Jeffrey Pennington, Richard Socher, 2014) and others emerging. In word embedding, relationships are typically unsupervised, meaning that the model is induced using unlabeled data in an unsupervised manner and learned from collections of documents using neural networks (Zhang et al., 2016). In this thesis, we utilized both custom trained and popular pre-trained word embeddings, including Word2vec, GloVe, and FastText, in ML and DL models for both English and Turkish languages.

Word2vec

Word2vec, pioneered by (Mikolov et al., 2013), is recognized as a groundbreaking embedding model that ushered in a new era in distributed semantics. Its objective is to derive high-quality vectors for the vocabulary from extensive datasets containing millions of words. Essentially, it operates on the principle that words with similar meanings are positioned closely together in the vector space, acknowledging that words may exhibit varying degrees of similarity. Word2vec employs an artificial neural network structure consisting of three layers: input, projection, and output layers. During network training, stochastic gradient descent and backpropagation techniques are utilized (Mikolov et al., 2013). For each word, which is represented by one-hot encoding in the input layer, a word vector of a specified size is generated. Importantly, the word vectors for words with similar meanings are grouped together in the vector space. Two log-linear architectures are employed for word embedding: Continuous Bag of Words (CBOW) and skip-gram, as depicted in Figure 3.6.

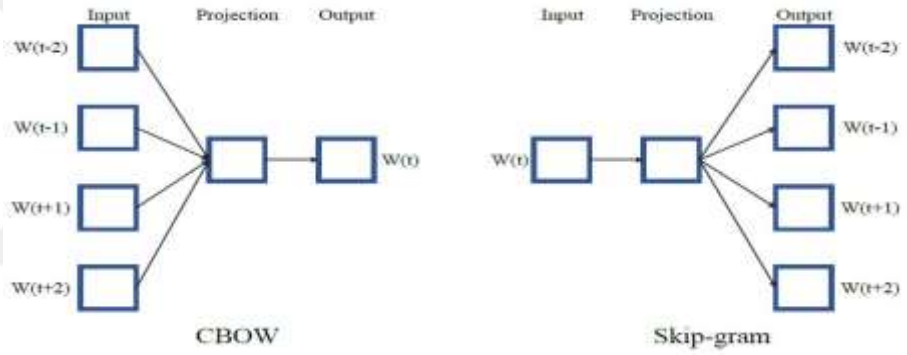


Figure 3.6. CBOW and Skip-gram architectures for Word2vec model (Mikolov et al., 2013)

In CBOW, the current word is forecasted based on its surrounding context, as depicted in Figure 3.6. Specifically, the prediction of the word (w(t)) is derived from the context provided by surrounding words (w(t-2), w(t-1), w(t+1), w(t+2)). Conversely, skip-gram predicts the surrounding context based on the current word. In other words, it forecasts the context (w(t-2), w(t-1), w(t+1), w(t+2)) from the given word (w(t)).

GloVe

Word2vec primarily emphasizes local context window knowledge, overlooking the effective utilization of global information statistics. In addressing this constraint, Global Vectors for Word Representation (GloVe) was introduced by (Jeffrey Pennington, Richard Socher, 2014). Unlike Word2vec, which focuses solely on local context, GloVe does not examine the entire dataset but instead learns from global statistics concerning the occurrence of words within a

specific document. This approach allows for the capture of diverse linguistic patterns and showcases proficiency in problem-solving based on similarity principles.

GloVe combines count-based matrix factorization with local context window techniques to achieve a more robust representation. Essentially, GloVe employs matrix factorization to derive consolidated global word-word co-occurrence statistics from a collection of documents. Serving as an extension of Word2vec, GloVe efficiently learns word vectors by predicting words based on their surrounding context. Similar to Word2vec, GloVe provides pre-trained word embeddings across various dimensions (100, 200, 300) that are trained on extensive corpora.

FastText

Word2vec and GloVe assign unique representations to each word, which poses a challenge for languages containing sub-word level information, resulting in out-of-vocabulary words. To address this, Bojanowski et al. (2017) introduced FastText, which utilizes character n-grams to create vector representations. Initially built on the skip-gram model, FastText represents each word by summing the vectors of its character n-grams. For instance, with the word "embedding" and N=3, FastText produces a representation of character tri-grams as shown in Figure 3.7 (Hasan et al., 2023). FastText training is faster compared to other methods because it can generate vectors for words absent in the training corpus. Moreover, leveraging character n-grams enables richer vector representations, particularly beneficial for morphologically complex languages like Turkish. In summary, FastText can generate vectors for out-of-vocabulary words that are absent in the training corpus, with its ability to operate at the sub-word level, a capability lacking in Word2vec and GloVe.

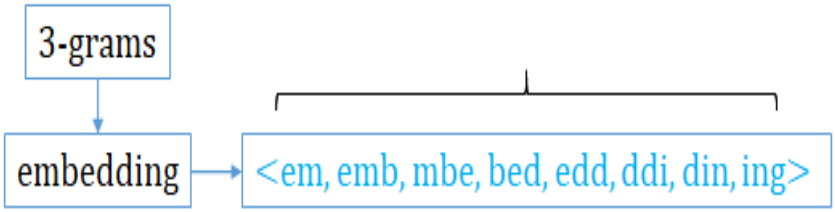


Figure 3.7. FastText word embedding for 3 grams

Facebook employs FastText model to enhance adverts. For instance, if a user shares a status update about purchasing a bike on their Facebook timeline, they may notice bike-related advertisements shortly afterward (Hasan et al., 2023). This capability stems from Facebook's status as a social media platform. Leveraging FastText, Facebook can deliver more personalized and relevant ads to its users.

3.6. Text Classification Methods

In this thesis, text classification methods are categorized into two groups: traditional classifiers and methods based on deep learning.

3.6.1. Traditional Supervised Classification Methods

In traditional classification approaches, we utilized widely recognized supervised ML algorithms, including Support Vector Machine (SVM), Logistic Regression (LR), Multinomial Naive Bayes (MNB), Decision Tree (DT), K-Nearest Neighbor (K-NN), and Random Forest (RF). Default parameters were employed, with the exception of SVM, which utilized a linear kernel, and Logistic Regression, where a maximum of 1000 iterations was set. These implementations were carried out using the Scikit-learn Library in Python.

SVM

The Support Vector Machine (SVM) is a supervised machine learning algorithm applicable to both classification and regression tasks. Its primary objective, in the context of binary classification problems, is to identify an optimal hyperplane from a given set of training features and their associated classes (Vapnik, 1999). Support vectors, defined as data points closest to this hyperplane, are considered crucial elements within the training dataset. The margin, denoting the distance between support vectors and the hyperplane, plays a significant role in SVM's decision-making process. While multiple hyperplane solutions might exist, SVM selects the one with the maximal margin, known as the optimal separating hyperplane (OSH), to ensure superior generalization performance (Holmström & Koistinen, 2010). Notably, each SVM's optimal hyperplane is unique (Holmström & Koistinen, 2010).

Figure 3.8 illustrates the binary classification scenario employing the SVM classifier (Holmström & Koistinen, 2010). SVM offers advantages such as robustness against noise, minimal risk of overfitting, and capability to handle complex classification tasks due to kernel functions (Fatima & Pasha, 2017). However, notable drawbacks include the requirement for formulating problems as 2-class classifications, longer training durations, and its black box nature, making evaluation challenging (Renjith et al., 2020).

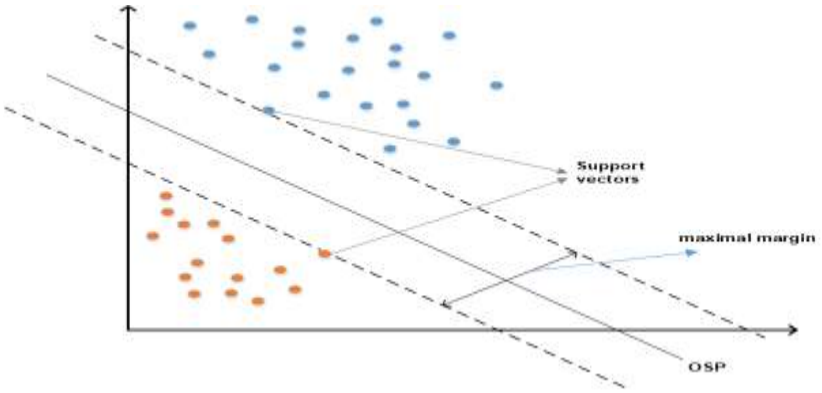


Figure 3.8. The Support Vector Machine Classifier

LR:

Logistic regression (LR) stands out as one of the most widely used supervised machine learning classification methods, employed to model a categorical outcome variable, Y, based on a set of features, X. It utilizes the sigmoid function to handle binary dependent variables and maximizes likelihood estimation to refine the model (Zafarani et al., 2014). The logistic regression predicts outcomes by computing a linear combination of multiple feature values, which serve as arguments for the sigmoid function, producing probabilities ranging between 0 and 1 (Zafarani et al., 2014). The model's prediction can be represented by the equation:

$$Y = P(Y = 1|X) = \frac{1}{1+e^{W^T X+b}} \quad (3.3)$$

where Y represents the probability of the output being 1 given the input features X, b denotes the bias, and W signifies the learnable parameter vectors associated with the input feature vectors X. In binary classification, the threshold lies at the midpoint of the predicted values, where values less than 0.5 are assigned to class 0, while values equal to or greater than 0.5 belong to class 1. One notable advantage of logistic regression is its ease of interpreting the text categories learned from the training set compared to other classifiers. However, logistic regression typically requires a larger volume of data to yield robust classification results compared to alternative classifiers (Segura-Bedmar et al., 2018).

Naive Bayes

Naive Bayes (NB) stands as one of the most effective and widely recognized classification techniques (Minsky, 1961; Sulzmann et al., 2007). It operates as a straightforward probabilistic classifier grounded in Bayes' theorem, incorporating strong independence assumptions (Khan, 2007). This method demonstrates efficacy across both numerical and textual datasets and boasts ease of implementation computationally when compared to alternative classifiers (Kibriya et al., 2005). At its core, the premise is based on the idea that the presence of a feature in one category

does not depend on its presence in another category. For example, the likelihood of a document D being classified into a category C can be calculated using equation 3.4.

$$P(C|D_j) = \frac{P(D_j|C)P(C)}{P(D_j)} \quad (3.4)$$

In order to estimate that a document D_j words $(a_i' \dots\dots a_d')$ belongs to a specific category, the prediction is given by Equation next

$$\theta(D_j)$$

$$\theta(D_j) = argmax_{yk \in C} P(y_k) \prod_{i=1}^{d} P(a_i'|y_i) \quad (3.5)$$

The advantages of NB algorithms are simplicity, robustness, and interpretability(Hmeidi et al., 2015).

MNB

Multinomial Naive Bayes (MNB) is coming from type of its predecessor Naive Bayes Classifier that uses multinomial distribution for each individual feature (Rennie et al., 2003). MNB is a parametric model that is commonly used in text classification (Su et al., 2011). Given $X = (x_1, x_2, .. x_m)$ as values of m features and Y = $(y_1, y_2, \dots\dots, y_n)$ as values of class attribute, class of instance X can be calculated as follows:

$$Y^* = arg\ max_{yi} \frac{P(y_i) \prod_{j=1}^{m} P(x_j|y_i)^{f_i}}{P(X)} \quad (3.6)$$

Where f_i is the number of occurrences of the features x_i in a sample X_i, $P(x_j|y_i)$ is the conditional probability that a feature x_j may be observed in a sample X in class y_i, and m is the number of unique features in sample X. Whereas $p(y_i)$ is the prior class probability that a sample with the class attribute y_i may be observed among all samples (Kibriya et al., 2005; Su et al., 2011).

DT

The decision tree (DT), introduced by Quinlan in 1986, stands as one of the earliest classification algorithms (Quinlan, 1986). It arranges attributes into a hierarchical tree structure, sorting them based on their values, where nodes represent attribute groups for classification and branches depict potential attribute values (Mahesh, 2018). Operating through a top-down approach with if-then conditional expressions, this algorithm efficiently partitions data. Within the tree

structure, leaf nodes signify the class with the highest number of training examples reaching that node (Çelebi, 2016). The classifier's architecture comprises root, decision, and leaf nodes, symbolizing the dataset, computation, and classification respectively (Mahesh, 2018).

During training, the classifier acquires the necessary decisions to distinguish between labeled categories. When classifying an unknown instance, data traverses the tree, and a specific feature from the input is compared with the fixed feature known during training. At each decision node, a comparison is conducted between the selected feature and the fixed one, resulting in a binary division in the tree based on the feature's prominence compared to the fixed one. This process continues until the data reaches a leaf node that assigns it a class. The decision tree classifier presents several advantages, including minimal hyperparameter tuning requirements, ease of description, clear interpretability through visualizations, and effective feature handling capabilities (Hmeidi et al., 2015).

KNN

The K-Nearest-Neighbor (KNN) algorithm represents one of the simplest supervised ML approaches applicable to both regression and classification tasks. It operates on the premise that similar instances tend to cluster together in the feature space, implying that neighboring examples are likely to be similar to each other. Therefore, in KNN as is seen in Figure 3.9, proximity or distance serves as the measure of similarity, computed as the distance between points on a graph. Consequently, when a new sample is introduced, the classifier examines the labels of its nearest data points and assigns the most frequently occurring label to the new sample. Essentially, KNN classifies the test instance based on the class with the highest number of instances among its nearest neighbors. Each neighbor contributes equally to the decision, and the class with the highest number of votes among the neighbors is selected (Alpaydın, 2014).

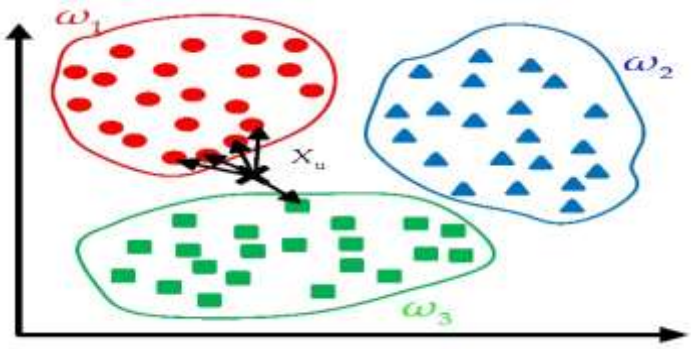


Figure 3.9. KNN Classifier (Bai et al., 2023)

RF

Random forest (RF), a supervised ML method introduced by (Breiman, 2001), is renowned for its adaptability in addressing both classification and regression tasks. In essence, it creates numerous decision trees and combines their outcomes to enhance predictive accuracy. These

decision trees are subsets randomly sampled from the dataset, with randomness playing a significant role in various aspects of the approach, including node splitting and attribute selection. Random forest can be seen as a particular case of bagging, an ensemble learning technique that merges predictions from multiple classifiers through bootstrapping, a process of randomly sampling with replacement from the training data (Khoshgoftaar et al., 2007). Each tree in a random forest is generated using bootstrapping, where approximately two-thirds of the instances from the training data are randomly chosen for tree construction. Furthermore, random forest introduces randomness by randomly selecting a subset of attributes to consider at each node of every tree. In the random forest framework, each tree contributes to the prediction by casting a unit vote for the most commonly predicted class for a given input vector. The final classification of a sample is determined by selecting the most frequently predicted class among all decision trees in the forest. Random forest offers several advantages, including short training and estimation times, applicability to multi-class classification and regression problems, a low number of parameters for training, and suitability for large-scale problems (Cutler & Cutler, 2012).

3.6.2. Semi Supervised Classification Methods

The issue of having abundant unlabeled data alongside limited labeled data is a common challenge in text classification tasks. A promising solution to this challenge is semi-supervised learning (SSL), which leverages both labeled and unlabeled data during the training phase. SSL techniques exploit the combined information from labeled and unlabeled data to enhance classification performance, particularly in scenarios where labeled data is scarce (Balcan & Blum, 2010; Goldberg, 2010). Unlike supervised approaches that rely solely on labeled samples for training, semi-supervised learning reduces this dependency by striking a balance between supervised and unsupervised learning (Haffari, 2006).

Traditional supervised learning methods necessitate all training instances to be labeled, which can be time-consuming and costly due to the involvement of data labeling experts. Semi-supervised learning, however, alleviates this burden by incorporating a portion of unlabeled data alongside labeled data to construct more effective models. While various semi-supervised learning techniques exist for classification, many are built upon conventional supervised classification algorithms (Witten et al., 2005). These techniques typically aim to expand the initial label set iteratively and may involve one or multiple classifiers, as well as single or multiple views (Triguero et al., 2015). In this thesis, two commonly utilized semi-supervised algorithms, namely Self-training and Co-training, were employed (Blum & Mitchell, 1998). These algorithms offer effective strategies for leveraging both labeled and unlabeled data to enhance classification performance.

Self-Training

Self-training is a SSL technique aimed at gradually enlarging the labeled training dataset (M. Li & Zhou, 2005). Initially, the model undergoes training using a set of labeled data samples, followed by making predictions on unlabeled data. Subsequently, the unlabeled data with highly confident predictions are chosen and progressively added to the labeled training dataset along with their predicted labels (Nartey et al., 2020). In subsequent iterations, the classifier is re-trained using the enlarged labeled examples. This method leverages a supervised model to generate pseudo-labels for unlabeled data samples, integrating those with the highest confidence into the training dataset along with their generated pseudo-labels, thereby expanding the size of the training data. This iterative process continues until the model converges (Nartey et al., 2020) (Ünal & Özel, 2023).

Co-Training

Co-training, introduced by (Blum & Mitchell, 1998), is a widely used method in SSL. Similar to Self-training, Co-training operates on a training set consisting of a small number of labeled examples and a larger number of unlabeled examples. In its basic form, the Co-Training algorithm employs different perspectives or "views" to represent the dataset. Each view's training examples are expanded by utilizing the classifier from the other view. The method assumes that these views are conditionally independent given the class label (independent assumption) and that each view alone can make accurate classifications with sufficient labeled data (compatibility assumption). In the Co-Training approach, a learning method such as MNB can be trained on two distinct views, known as feature sets, where two classifiers are employed. These classifiers collaborate by labeling the unlabeled set and adding the most confident predictions to the labeled set. The essence of this algorithm lies in the reciprocal teaching process: the first classifier supplements the labeled set with examples used for learning by the second classifier based on the second view, and vice versa (L. Liu et al., 2003).

3.6.3. Deep Learning Classification Methods

Being a subset of ML, DL models extract useful features automatically and are considered robust when dealing with high-dimensional feature space. The main benefit of employing deep learning models as opposed to traditional machine learning methods is that they do not necessitate manually crafted features rather, they automatically discern and select the most effective features. In this thesis, we used CNN, RNN including LSTM, GRU, BiLSTM, BiGRU and a combination of CNN-LSTM and LSTM-CNN.

CNN

CNN is a DL framework frequently employed for hierarchical document categorization (Jaderberg et al., 2016; Siwei et al., 2015). Originally developed for image processing (LeCun et al., 1998), CNN has proven to be highly effective for NLP tasks as well (Kim, 2014). It comprises a feedforward neural network with input, convolutional, pooling, and fully connected layers. CNN typically employs a loss function such as softmax on the final fully connected layer (Uysal & Murphey, 2017). Through convolution filters, CNN extracts feature maps, while pooling layers downsample these maps to retain pertinent information. The final decision is made via a fully connected layer with dropout regularization. Below is a generalized CNN architecture tailored for NLP applications.

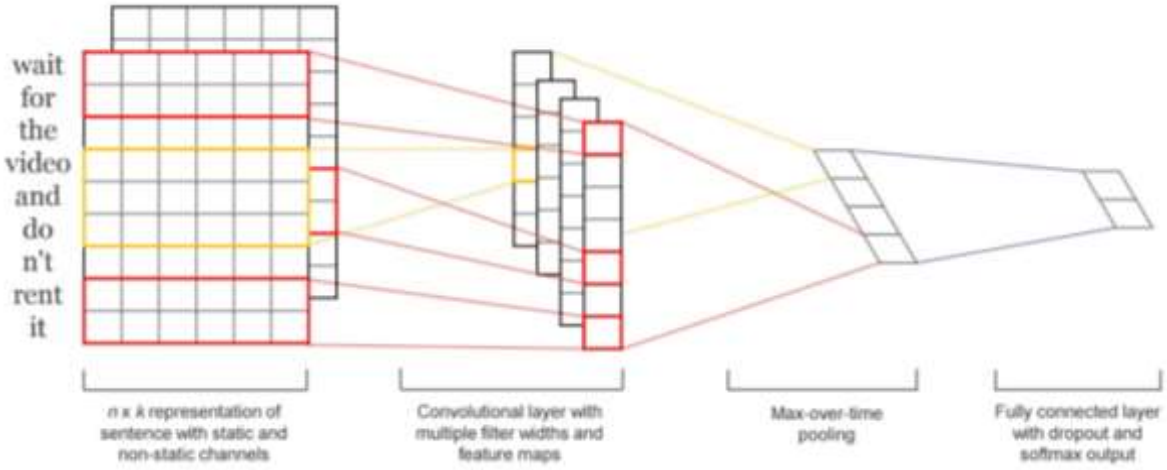


Figure 3.10. Basic CNN structure for NLP with two channels (Kim, 2014)

In order to develop this model on text data, each text with a length of N is represented as following

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n \quad (3.7)$$

Where $\oplus$ is the concatenation operator. Each convolution step involves a filter $W \in \mathbb{R}^{hk}$, in a window of h words to generate a new feature. The convolutional operation can be formulated as follows:

$$C_i = f(W \cdot x_{i:i+h-1} + b) \quad (3.8)$$

Where C_i is generated feature from window of words $x_{i:i+h-1}$, $b \in \mathbb{R}$ is the bias term and f is a non-linear activation function like Relu. A new feature map of $C = [c_1, c_2, \ldots, c_{n-h+1}]$ is obtained by applying this operation to each possible window of words in the sentence $x_{1:h}, x_{2:h}, \ldots, x_{n-h+1:n}$. a max-over-time pooling is applied over the feature map c ($with\ c \in$

$\mathbb{R}^{n-h+1}$) to generate the maximum value ($i.e., \tilde{C} = \max c$) as feature for the related filter (Kim, 2014).

RNN

The Recurrent Neural Network, abbreviated as RNN, belongs to a class of neural networks where connections between neurons form a directed cycle (Elman, 1990). RNNs are specialized in processing sequential data, earning their name from their recurrent nature, as they execute the same task for every element of a sequence, with each output relying on previous computations (Stojanovski et al., 2016). With a built-in memory component, RNNs retain information from prior computations, effectively remembering past input information (Zhang et al., 2018). The architecture of a basic RNN is illustrated in the Figure 3.11 showing an unfolded network with cycles on the left graph, while the right graph presents a folded sequence with three time steps (Zhang & Liu, 2018).

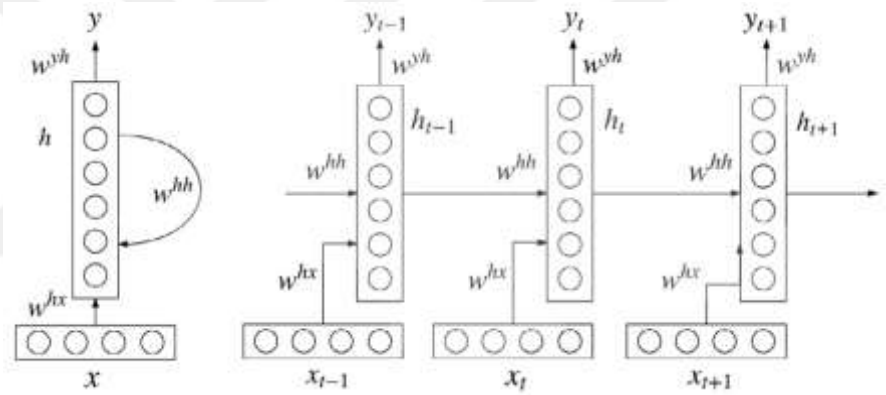


Figure 3.11. RNN architecture before (left) and after (right) the unfolding (Zhang et al., 2018)

The process of forward propagation in RNN can be explained through the following equations, as indicated by (Rysbek, 2019; Zhang et al., 2018).

$$h_t = f(w^{hx} x_t + w^{hh} h_{t-1} + b_h) \tag{3.9}$$

$$y_t = g(w^{yh} h_t + b_y) \tag{3.10}$$

w^{hx}, w^{hh}, and w^{yh} defines the learned and shared weight matrices across all steps, while f and g are the activation functions and b_h and b_y are the bias terms. Theoretically, RNN can make use of information in arbitrarily long sequences, but in practice, it is limited to looking back only a few steps hence suffers from vanishing gradient or exploding problem during back propagation (Bengio et al., 1994; Cliche, 2017; Karpathy et al., 2015). Therefore, a more

sophisticated RNN architectures like LSTM and GRU were introduced to overcome the challenges of RNN (Karpathy et al., 2015; Santur, 2019).

LSTM

Long Short-Term Memory (LSTM) networks, a specialized variant of RNN's, present significant advancements in preserving information compared to traditional RNN's (Hochreiter & Schmidhuber, 1997). LSTM effectively addresses the challenge of vanishing gradient descent commonly encountered in conventional RNN's. These networks boast a memory capacity enabling them to determine the relevance of information to pass forward and what data to discard. Employing the backpropagation algorithm, LSTM networks incorporate gated mechanisms. Key components of a basic LSTM network include an input gate (i_t), an output gate (o_t), and a forget gate (f_t), facilitating the retention of information over prolonged periods. Figure 3.12 provides an illustration of LSTM architecture.

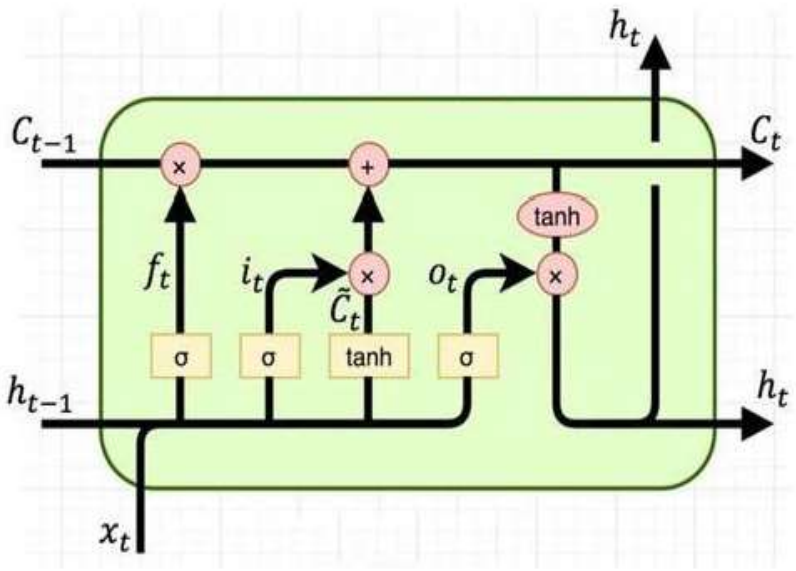


Figure 3.12. *Architecture of LSTM cells*

The following set of equations are implied to compute the hidden state of the LSTM (Cliche, 2017).

$$i_t = \sigma(W_i \,.\, x_t + U_i \,.\, h_{t-1} + b_i) \tag{3.11}$$

$$f_t = \sigma(W_f \,.\, x_t + U_f \,.\, h_{t-1} + b_f) \tag{3.12}$$

$$\tilde{C}_t = \tanh(W_c \,.\, x_t + U_c \,.\, h_{t-1} + b_c) \tag{3.13}$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \tag{3.14}$$

$$o_t = \sigma\,(W_o\,.\,x_t + U_o\,.\,h_{t-1} + b_o) \tag{3.15}$$

$$h_t = o_t \times \tanh(C_t) \tag{3.16}$$

Where by i_t, f_t and o_t representing the input gate, forget gate, and output gate. In addition, x_t and h_t stand for the input vector and hidden layer value at time step t. Meanwhile, C_t, $\tilde{C}_{t1}$, and C_{t-1} correspond to the current, candidate, and previous cell states, respectively. The activation functions utilized here are σ (i.e., sigmoid) and tanh.

GRU

The Gated Recurrent Unit (GRU) represents a streamlined version of the LSTM architecture, introduced to address short-term memory issues (Cho et al., 2014; Chung et al., 2014). GRU merges the input gate and forget gate of LSTM into a single update gate and forget gate, resulting in a more concise design (Shiri et al., 2023). Unlike LSTM, GRU does not incorporate a distinct cell state. Additionally, GRU distinguishes itself from LSTM by featuring two gates: the update gate and reset gate. These gates assist the architecture in determining the extent to which previous information from preceding time steps should be retained for future use, and how much of that information should be discarded. Figure 3.13 illustrates GRU cells with the update and reset gates, capable of retaining information over extended periods.

It's worth noting that GRU provides a simplified alternative to LSTM with reduced tensor operations, facilitating quicker training. However, the choice between GRU and LSTM depends on the specific task at hand, as their performance may vary (Abbaspour et al., 2020).

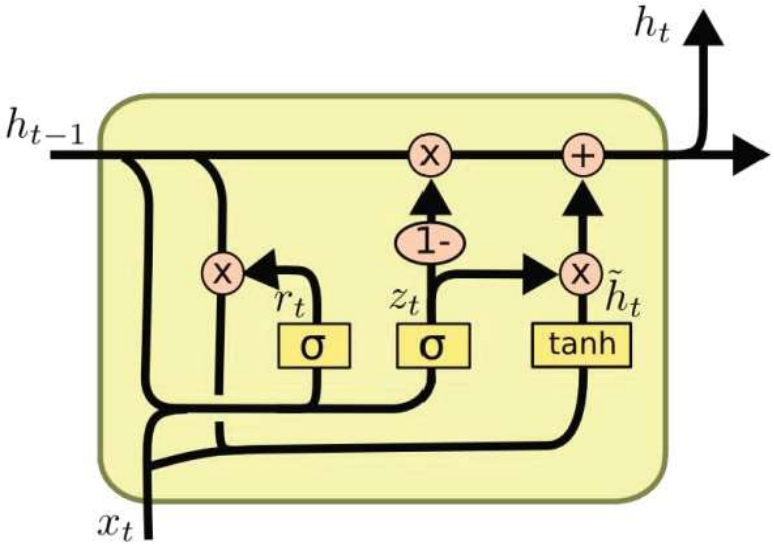


Figure 3.13. Architecture of GRU cells (Karpathy et al., 2015; Lopez, 2019).

$$Z_t = \sigma(W_z . x_t + U_z . h_{t-1} + b_z) \quad (3.17)$$

$$r_t = \sigma(W_r . x_t + U_r . h_{t-1} + b_r) \quad (3.18)$$

$$\tilde{h}_t = \sigma(W_h . x_t + U_h . (r_t \times h_{t-1}) + b_h) \quad (3.19)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t + \widetilde{h_t} \quad (3.20)$$

where r_t and z_t represent vectors of reset and update gates differently from the variables used above. In the given equation, × represent element wise (hadamard) product of two matrices/vectors.

BiLSTM

BiLSTM (Bidirectional Long Short Term Memory) networks, builds on regular LSTM models to improve how information flows within the system (Schuster & Paliwal, 1997). Unlike LSTMs that only analyze data moving forward, BiLSTM tackles this limitation by considering both the past and future of a sequence. This is achieved by training two separate LSTM layers, one that reads data forward (like normal) and another that reads it backward (Aldhyani & Alkahtani, 2021; Lipton et al., 2015). You can imagine this like having two readers, one starting at the beginning of a sentence (green arrow) and another starting at the end (red arrow) Fig 3.14 (Liciotti et al., 2020). By combining the information from both directions, BiLSTM gets a more comprehensive understanding of the sequence.

While training, the two BiLSTM layers, one processing forward and the other backward, act like independent detectives examining a crime scene. Each analyzes the clues (data points) and forms their own interpretations (prediction scores) at every step. These interpretations are then combined, like pooling their findings, to create a final conclusion (the model's output) (Liciotti et al., 2020). Because both directions are considered, BiLSTM can examine a wider range of evidence (context) and better understand the order of events within the data (temporal dependencies) (Shiri et al., 2023).

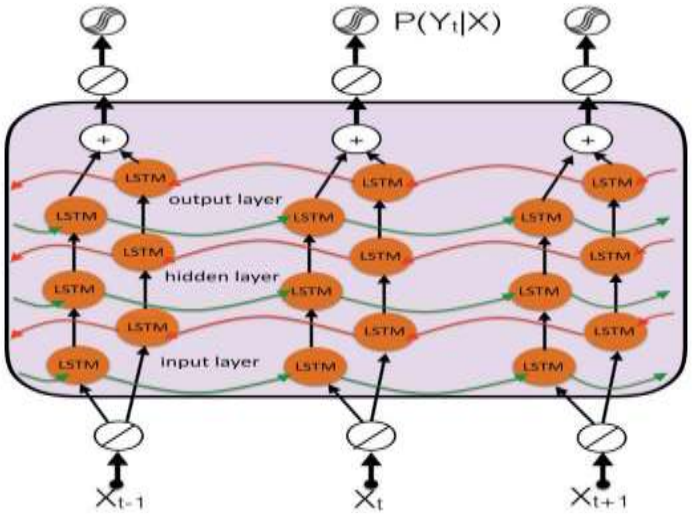


Figure 3.14. The architecture of a Bidirectional LSTM model (Liciotti et al., 2020)

BiGRU

Just like BiLSTM, BiGRU (Bidirectional Gated Recurrent Unit) is a system with two parts, one working forward and the other backward (Schuster & Paliwal, 1997). Information gets fed into the system from both directions, and the outputs from each direction are then merged to create the final result. This double analysis allows BiGRU to squeeze more information out of the data, which improves how well the whole system learns (Zhang et al., 2018). Figure 3.15 is an architecture of BiGRU with layers. Imagine it like having two analysts examining data, one looking forward and one backward (Li et al., 2020). Then, their findings are combined at each step to get a more complete picture.

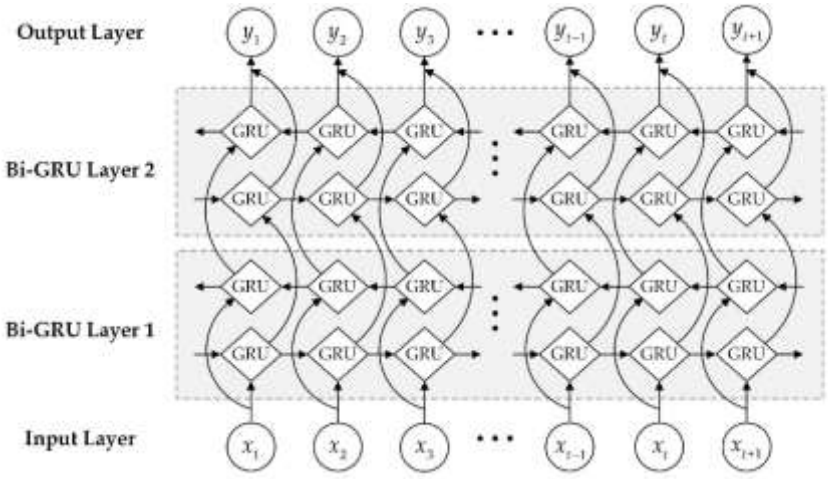


Figure 3.15. BiGRU neural network structural unit (Li et al., 2020)

BiLSTM and BiGRU which are sometimes mentioned as Bidirectional Recurrent Neural Network (BRNN) can as well be described well by the following three equations (Lipton et al., 2015).

$$h^{(t)} = \sigma(W^{hx} x^{(t)} + W^{hh} h^{(t-1)} + b_h) \quad (3.21)$$

$$z^{(t)} = \sigma(W^{zx} x^{(t)} + W^{zz} z^{(t+1)} + b_z) \quad (3.22)$$

$$\tilde{y}^{(t)} = softmax(W^{yh} h^{(t)} + W^{yz} z^{(t)} + b_y) \quad (3.23)$$

Where $h^{(t)}$ and $z^{(t)}$ represents the values of the hidden layers in the forwards and backwards directions respectively.

Combining CNN and LSTM

An extensively adopted strategy to enhance the precision of associated tasks involves integrating RNN and CNN algorithms. This widespread approach stems from the recognition that merging these two sophisticated neural network architectures can notably enhance accuracy and enable effective handling of diverse tasks (Farha & Magdy, 2019; Sosa, 2017). Consequently, in this thesis, we merged two DL models, namely CNN-LSTM and LSTM-CNN, for cyberbullying detection. The rationale behind this amalgamation is that the initial segment of the model will extract features, while the subsequent part will focus on learning from input text. All models developed for the DL are illustrated in appendix section.

3.6.4. Large Language Models

The Large Language Model (LLM) is an advanced DL model based on a transformer architecture, and extensively trained on vast datasets to generate text resembling human language, answer questions, aid in translation, and perform various NLP tasks (Kasneci & Sessler, 2023). Representing a significant breakthrough in NLP and AI, LLMs have exhibited remarkable performance. The pivotal advancement in LLMs lies in the utilization of transformer architecture (Devlin et al., 2019; Tay et al., 2022), along with the underlying attention mechanism (Ashish et al., 2017), greatly enhancing the models' ability to handle long-range dependencies in natural language texts.

Figure 3.16 shows the architecture of a transformer. A transformer comprises multiple transformer blocks or layers, encompassing self-attention layers, feedforward layers, and normalization layers. These components collaboratively decipher input and predict streams of output during inference. The transformer architecture employs the self-attention mechanism to discern the relevance of different input segments when generating predictions, enabling the model to grasp the relationships between words in a sentence irrespective of their positions (Ashish et al., 2017)

Transformers excel in LLM’s due to two key features: positional encodings and self-attention. Positional encoding embeds the order of input occurrence within a sequence, enabling non-sequential input feeding into the neural network. Meanwhile, self-attention assigns weights to

each input part during processing, signifying their importance in the context of the entire input. Consequently, models can focus on significant input parts rather than uniformly distributing attention, facilitating the analysis of subtle connections and contexts among elements over long distances in a non-sequential manner.

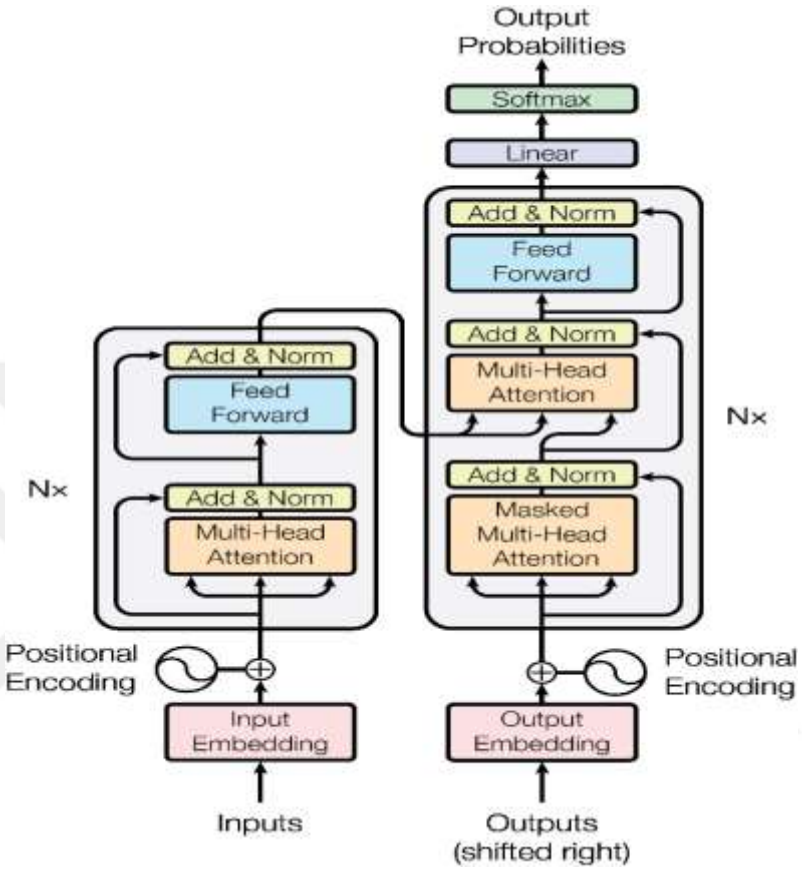


Figure 3.16. Transformer architecture (Ashish et al., 2017)

Another unique features in LLM is the use of pre-training or pre-trained, where a language model is trained on a larger dataset before being fine tune on a specific task and this has been an effective techniques for improving performance on a wide range of language tasks (Min et al., 2023). Some of the examples of LLM's include GPT3, GPT 4, Meta's Llama models and Google's Bidirectional Encoder Representation from Transformer (BERT). In this thesis, we used BERT Language model and its variants designed specifically for English and Turkish.

BERT

BERT stands for Bidirectional Encoder Representations from Transformers. It's a fancy way of saying it's a super-powered language model that understands the connections between words in text. Unlike older models that only analyzed words in order (left to right), BERT can consider both the words before and after a word at the same time (bidirectional). This lets it grasp the true meaning of words based on their context (Devlin et al., 2019). Before BERT, models like

Word2vec focused on learning how words are used in different situations (Mikolov et al., 2013; Quoc & Tomas, 2014). BERT takes things a step further by using a special technique called transformers to learn these contextual relationships (Ashish et al., 2017).

To train BERT, researchers threw a massive amount of text at it, including billions of words from Wikipedia and books (Zhu et al., 2015). BERT is trained on an extensive dataset comprising 2,500 million words from Wikipedia and 800 million words from the Book Corpus forming a substantial collection of unlabeled text utilized for pre-training. This pre-training phase is what makes BERT so powerful. There are two main versions of BERT: base and large. Base is like the mini-version, with fewer parts (parameters) than the larger and more complex large version. BERT base comprises 12 transformer blocks, 12 attention heads, and 110 million parameters, whereas BERT large features 24 transformer layers, 16 attention heads, and a total of 340 million parameters. These pre-trained BERT models can be applied to unlabeled data or fine-tuned on task-specific data derived from pre-trained models. Figure 3.17 illustrates the architecture of BERT base and BERT large. The cool thing is that you can use these pre-trained models directly or fine-tune them for specific tasks, making them even more effective.

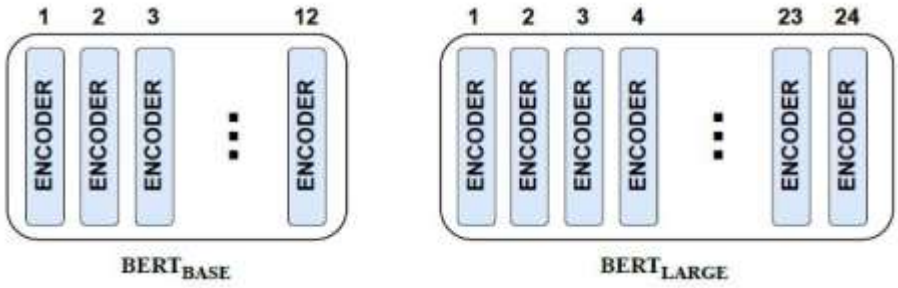


Figure 3.17. BERT architecture (BERT base and BERT large).

BERT undergoes pre-training through two unsupervised tasks. In the first task, known as the masked language model (MLM), approximately 15% of tokens within the input text are randomly masked (replaced with [MASK]), and the model is trained to predict these masked tokens, allowing it to learn about word relationships. The second task is Next Sentence Prediction (NSP), where the model is provided with pairs of sentences and trained to determine whether the second sentence logically follows the first one. This phase not only captures relationships between sentences but also aims to gather additional long-term contextual information. Figure 3.18 illustrates the input representation of BERT.

Figure 3.18. Input representation of BERT model

The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings. There have been many BERT variants developed after BERT to improve performance and solve computational capability since the original BERT was large. Some of the pre-trained BERT variants include RoBERTa, ALBERT, Electra, DistilBERT, BERTbase multilingual etc.

- RoBERTa, short for Robustly Optimized BERT Approach, presents a streamlined adaptation of BERT with reduced parameters (Y. Liu et al., 2019). RoBERTa incorporates several adjustments, including dynamically altering the masking pattern during training data, a departure from BERT's static masking pattern. Additionally, RoBERTa trains on longer sequences and eliminates the next sentence prediction task. RoBERTa leverages a diverse dataset of 160GB text, comprising Book Corpus, English Wikipedia, and OpenWebText, whereas BERT is trained solely on Book Corpus. These modifications are aimed at enhancing RoBERTa’s performance. It's noteworthy that RoBERTa adopts the same architecture as BERT and undergoes pretraining using the MLM task while omitting the NSP task. The structure of RoBERTa is depicted in Figure 3.19 (Singh, 2023).

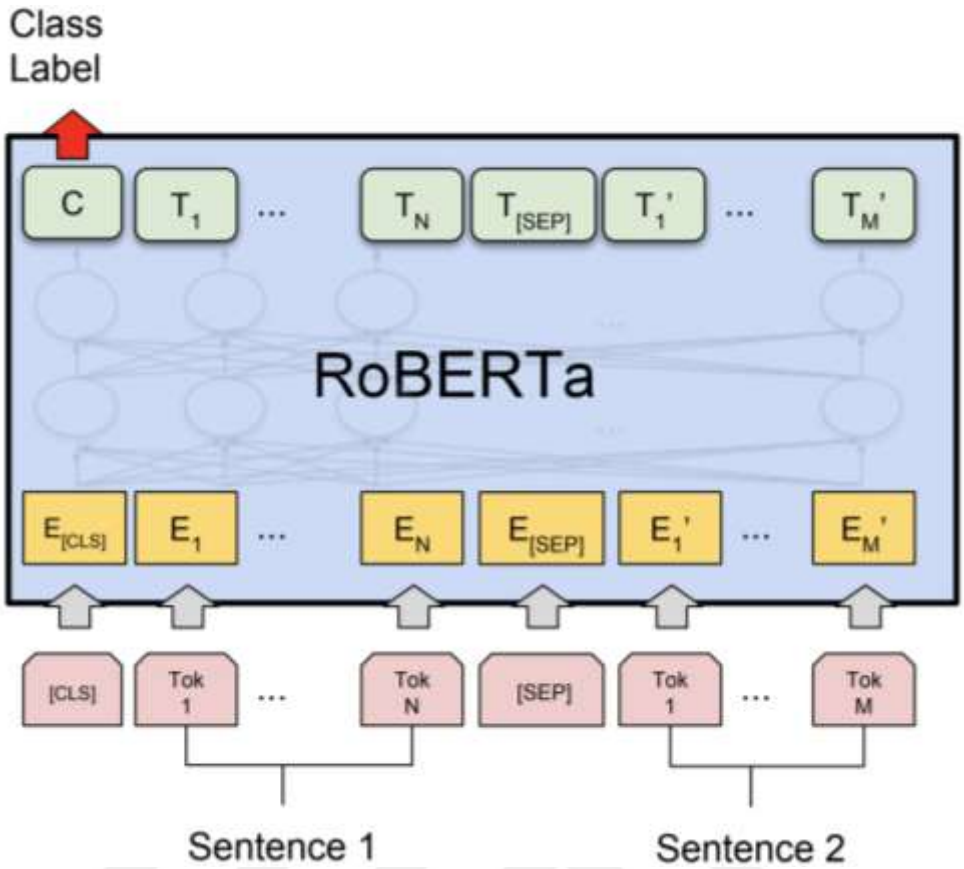


Figure 3.19. Structure of RoBERTa (Singh, 2023).

Class label is used to identify the text classified as Roberta. While the label consists of the following:

- ❖ CLS which is a special token that is used to identify the beginning of the text to be classified
- ❖ Class which is the name of the class
- ❖ Token which represents the individual words that make up the text
- ❖ SEP is a special token that is used to separate the tokens
- ❖ Mask which is a special token that is used to mask out the tokens that are not used for classification
- ❖ The class model is used by the RoBERTa model to identify the text that is to be trained on. The model learns to classify text by comparing the encoded representation of the text to encoded representation of the training text.

- ▪ ALBERT: Overcoming challenges in model performance, such as GPU/TPU memory constraints and prolonged training durations, can be daunting. ALBERT, a lightweight variant of BERT, was introduced to address these issues by reducing the model's parameters, thereby decreasing memory consumption and expediting training speed

(Lan et al., 2020). The primary advantage of ALBERT lies in its parameter efficiency, enabling it to accommodate significantly larger batches into memory for swift inference.

- DistilBERT: DistilBERT stands as a condensed, pre-trained language representation model, offering a faster alternative to the original BERT model (Sanh et al., 2019). Tailored for efficiency, this version of BERT demonstrates commendable performance across various tasks, particularly in resource-constrained environments. The overall architecture of DistilBERT mirrors that of BERT, albeit with the removal of token type embedding and the pooler, and a halving of the number of layers, resulting in substantial gains in computational efficiency. DistilBERT was distilled from large stacks with dynamic masking and prediction of the next sentence (NSP) (Sreelakshmi et al., 2023). Despite utilizing fewer resources, DistilBERT achieves performance levels comparable to the original BERT model. The approach involves knowledge distillation during the pretraining phase, reducing the size of the standard BERT by 40%, while still retaining a language understanding capability of 97%. Consequently, the model exhibits faster, more compact, and lighter characteristics during pre-training.
- BERT Multilingual: Introducing multilingual BERT aimed to incorporate various languages, with the model pre-trained on 104 languages, including Turkish (Devlin et al., 2019). Training data comprised Wikipedia articles in each language, addressing normalization issues specific to individual languages. Notably, the multilingual BERT model doesn't employ markers to indicate input language and lacks an explicit mechanism ensuring that translation-equivalent pairs possess similar representations. Two versions of the multilingual BERT model exist, differing in capitalization and punctuation (Devlin et al., 2019). This multilingual model of BERT is based on the word piece method, effectively handling unknown words to represent word vectors (Wang, Z., & Wang, 2021).

Other BERT Turkish variants

- **dbmdz/bert-base-turkish-128k-cased:**
  dbmdz/bert-base-turkish-128k-cased is a Turkish BERT Variant which is trained on a large Turkish corpus that has a size of 35GB and a vocabulary size of 128k (*Anonymous 16*).
- **ELECTRA-Tr Dbmdz/electra-base-Turkish**
  This Turkish language model, Dbmdz/electra-base-Turkish (*Anonymous 17*), is more efficient to train compared to BERT because it requires less computation. Unlike BERT, which uses a method called MLM, this model trains a part that can tell real

words from disguised ones fed by another neural network . The training data comes from a massive collection of Turkish text, including Wikipedia entries (Turkish OSCAR corpus, 35 GB).

- **DistilBERT-Tr distilbert-base-Turkish-case**

  DistilBERT-Tr distilbert-base-Turkish-case, is a lightweight Turkish version of the BERT model created by the MDZ Digital Library team. It's much smaller and faster than the original BERTurk model because it was trained on a less extensive dataset of Turkish text (7 GB) (DistilBERTurk, n.d.). Despite its size, DistillBERTurk delivers performance almost on par with BERTurk for tasks like identifying parts of speech and recognizing named entities in text.

3.6.5. Parameter-Efficient Fine-Tuning

Parameter-efficient fine-tuning (PEFT) refers to a collection of methodologies utilized to decrease the computational expenses associated with fine-tuning LLM's. The Low-Rank Adaptation of Large Language Models (LoRA) represents a parameter-efficient fine-tuning technique employed to train LLMs for specific tasks or domains (Hu et al., 2022). This approach introduces trainable rank decomposition matrices into each layer of the transformer architecture, thereby reducing the number of trainable parameters for downstream tasks while maintaining the pre-trained weights frozen.

The rationale behind LoRA is to achieve superior fine-tuning outcomes in a more resource-efficient manner. Full-parameter fine-tuning necessitates significant memory resources, and existing methods either pose impractical challenges or compromise on quality. Given that high-performance GPUs are a valuable asset, more efficient techniques can render fine-tuning more accessible and facilitate increased experimentation.

Unlike modifying the weight matrix W of a layer across all its elements, LoRA creates two smaller matrices, A and B, whose product approximates the adjustments to W. This adaptation can be mathematically expressed as W' = W + AB, where A and B represent the low-rank matrices. If W is an mxn matrix, A might be mxr and B rxn, where r denotes the rank and is substantially smaller than m and n. During fine-tuning, only A and B undergo adjustments, enabling the model to learn task-specific features (Hu et al., 2022)

3.6.6. General Optimization Techniques

Dropout

The Dropout technique functions as a safeguard against overfitting in neural networks, as observed in studies (Garbin et al., 2020; Srivastava et al., 2014). Figure 3.20 represents Comparison of NN before dropout (on the left) and after dropout (on the right). It entails randomly deactivating selected neurons within hidden layers based on a dropout probability

(Ko et al., 2017). During a specific training phase, these chosen neurons' activities are set to zero in both the forward and backward steps, effectively temporarily removing them from the network. This dynamic approach fosters more robust and reliable learning by ensuring diverse neuron activation in each iteration. While it may extend the training duration, this technique contributes to enhanced model generalization, as highlighted in research (Ko et al., 2017; Srivastava et al., 2014).

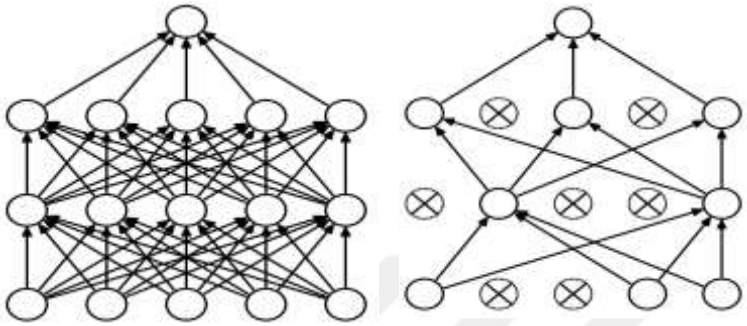


Figure 3.20. Comparison of NN before dropout (on the left) and after dropout (on the right) (Srivastava et al., 2014)

Adam Optimizer

The Adam optimizer, derived from Adaptive Moment Estimation, stands out as a widely utilized optimization algorithm in neural network training (Kingma & Ba, 2015). Adam seamlessly integrates the strengths of both AdaGrad and RMSProp (Z. Zhang, 2019). By maintaining per-parameter learning rates that dynamically adjust based on the gradients' moments, Adam efficiently and rapidly converges, particularly in scenarios involving extensive datasets and high-dimensional parameter spaces. This optimizer calculates individual adaptive learning rates for distinct parameters, rendering it particularly effective for tasks with sparse gradients. Its adaptive characteristic, coupled with its momentum functionality, equips it to adeptly manage noisy gradients and non-stationary objectives, making it a favored option across various deep learning applications (Kingma & Ba, 2015).

Dense Layer (Sigmoid Activation)

A dense layer, often referred to as a fully connected layer, serves as a foundational element within neural networks, wherein each neuron in the layer establishes connections with every neuron in the preceding layer (Bengio et al., 1994). Employing a sigmoid activation function, this layer operates by applying the sigmoid function element-wise to the weighted sum of inputs and biases, thereby compressing the output values within the range of [0, 1]. Particularly beneficial for binary classification tasks, the sigmoid function allows for interpreting the output as probabilities (Bengio et al., 1994). By incorporating the sigmoid activation function, non-linearity is introduced into the

network, empowering it to capture intricate relationships between inputs and outputs. However, it is important to note that the sigmoid function may encounter challenges such as vanishing gradients, particularly evident at the extreme ends of its range, potentially impeding the learning process, especially in deeper networks. Nevertheless, despite this drawback, the dense layer with sigmoid activation remains a prevalent choice across various neural network architectures due to its interpretability and straightforwardness.

3.7. Model Evaluation

When tackling classification problems in machine learning, it's customary to partition the dataset into three distinct subsets: the training set, validation set, and testing set. Numerous approaches have been proposed for dividing the dataset into these subsets, but when dealing with a very small dataset, cross-validation is often preferred. The fundamental concept behind cross-validation is to train and evaluate each candidate model on separate datasets to ensure unbiased performance evaluation (Lei, 2020). Cross-validation involves several steps: first, keeping one-fold separate so the model doesn't access it, then training the model on the remaining folds, and finally testing each trained classifier on the held-out fold. The results from each fold are then averaged to assess the performance of the particular parameter setting (Domingos, 2012) (Kohavi, 1995). Cross-validation can take various forms, such as K-fold and Leave-one-out (Kohavi, 1995).

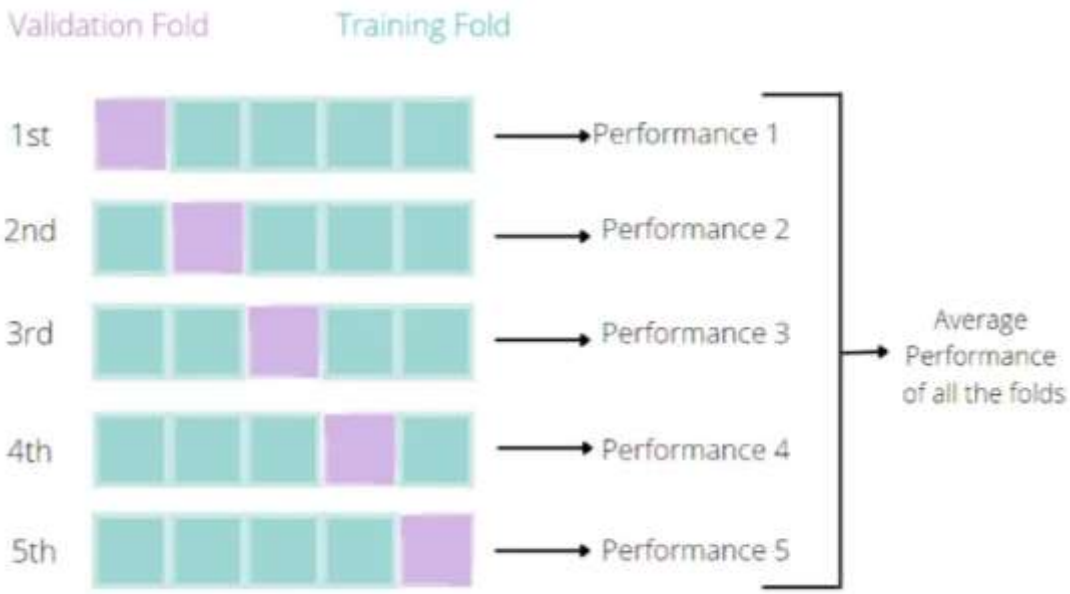


Figure 3. 21. Example of 5-fold cross validation

In this instance, the approach entails dividing the dataset into K partitions, where K-1 partitions are designated for training and one for validation testing. For instance, as depicted in the figure 3.21, the dataset can be segmented into 5 folds. During each iteration, 4 folds are utilized for

training the model, while the remaining fold is used to evaluate the model's performance. This cycle is repeated five times, with the fold used for evaluation changing position in each iteration.

To address text classification challenges, a model's effectiveness is evaluated through a confusion matrix. This matrix comprises four distinct combinations of predicted and actual values, as illustrated in Figure 3.22. Each row corresponds to predictions, while columns represent actual classes. Its primary function is to quickly identify if the system is confusing two types of data. The matrix features two dimensions labeled "actual" and "predicted," with both dimensions encompassing the same sets of "classes." Each intersection of dimension and class forms a unique variable in the table, crucial for metrics such as True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) (Narkhede, 2018).

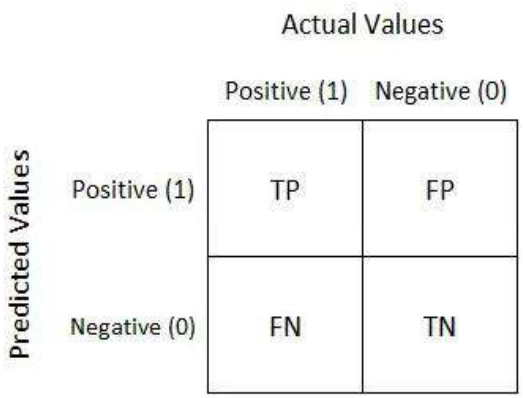


Figure 3.22. Confusion matrix (Narkhede, 2018)

- ❖ True Positive: Classified as Positive and its true
- ❖ True Negative: Classified as Negative and its true
- ❖ False Positive: Classified as Positive and its false
- ❖ False Negative: Classified as Negative and its false.

Using the definitions provided in the confusion matrix, one can calculate commonly used evaluation metrics for classification tasks. These metrics include Precision (P), Recall (R), Accuracy (ACC), and F-score, each defined as follows;

- Precision: This metric represents the proportion of true positives relative to the total number of predicted positives. It indicates the accuracy of identifying cyberbullying instances among all instances predicted as such. A higher precision signifies that the model is careful in labeling positive cases and tends to minimize false positives.

$$Precision = \frac{True\ positives}{(True\ positives + False\ positives)} \quad (3.24)$$

- Recall: This parameter gauges the model's capacity to identify true positives. Illustrated with the cyberbullying label, it measures the proportion of cyberbullying instances correctly predicted out of all cyberbullying instances. A higher recall suggests that the model is proficient at detecting positive cases, even if it occasionally misclassifies positives as negatives.

$$Recall = \frac{True\ positives}{(True\ positives + False\ Negatives)} \quad (3.25)$$

- Accuracy: This measure quantifies the ratio of correct predictions to the total number of outcomes. In the context of the cyberbullying label, it encompasses the number of correctly predicted cyberbullying instances and accurately identified non-cyberbullying instances from the entire prediction set. Accuracy offers a comprehensive assessment of the model's performance across all classes.

$$Accuracy = \frac{True\ positives + True\ Negatives}{Total\ instances} \quad (3.26)$$

- F-score: The F-score (a.k.a. F1-score, F-measure) denotes the harmonic mean of precision and recall, offering a balanced assessment of model performance. It proves particularly beneficial in handling imbalanced datasets. A higher F-measure signifies a favorable equilibrium between precision and recall.

$$F1 - score = \frac{2*(Precision*Recall)}{(Precision + Recall)} \quad (3.27)$$

4. RESULTS AND DISCUSSIONS

In this section, we report the results of our comprehensive experiments. The thesis is focused on detection of CB in the context of Turkish and English language. The study experiments involve comparison of ML and DL together with variety of feature extraction techniques. A novel semi supervised learning method that accurately detect and improve classification accuracy of CB in both languages of English and Turkish was proposed.

The summary of our work can be outlined as comprehensive study involving ML and DL for CB detection on English and Turkish language, evaluating impact of different feature extraction techniques such as traditional feature weighting method including binary, TF, TF-IDF and word embedding such as Word2vec, GloVe, FastText. BOW and N-grams features are evaluated with respect to traditional ML methods. The impact of word embeddings are evaluated with ML and DL methods. The effectiveness of large language model specifically BERT are compared for both English and Turkish variants, and the BERT variants' performances are evaluated with results generated through ML and DL models. Additionally, the performance evaluation of BERT combined with ML, BERT with LoRa, a parameter efficiency optimization technique, impact of SSL techniques including traditional self-training and co-training are deeply studied in this thesis. Finally, implementing and proposing a novel self-training with BERT, a SSL method that improves classification accuracy, is proposed and evaluated.

We believe that based on the nature of the study i.e. datasets used i.e. English and Turkish language, applied techniques and analysis can distinguish this work from existing work performed in the same ways. Our experiment was divided into the following steps:

- Term weighting methods were combined with ML classifiers
- Word embedding with ML and DL methods
- Evaluation of different LLM's specifically, BERT variants
- Combining LLM (BERT) with ML classifiers
- Evaluation of LLM (BERT) with LoRA parameter optimization technique
- Evaluation of traditional SSL techniques
- Proposing self-training with LLM (BERT)

The evaluation metrics employed include accuracy and F-score, with all experiments configured to execute using a 5-fold cross-validation. The following sub headings present results and analysis. Some of the results are represented in Appendix side.

4.1. Effect of Term Weighting and Feature Set with ML Methods

In this part, we investigated how assigning weights to terms in text data affects the performance of traditional ML models for classification tasks. First, the text data was cleaned and prepared for analysis. Two common techniques were used to extract features from the text: BOW and character-level N-grams. Then, different weighting methods were applied to these features: Binary (where a term is either present or absent), Term Frequency TF (how often a term appears), and TF-IDF (which considers both TF and how common the term is across documents). By applying these weights, the text data was converted into numerical formats suitable for classification by ML models. Five well-known traditional ML classifiers were used in this experiment: MNB, LR, SVM with a linear kernel, DT, RF, and KNN. Default settings were used for all classifiers except for SVM (which used a linear kernel) and LR (which was limited to 1000 iterations). The results of this experiment are summarized in four tables (Table 4.1 to 4.4)

Table 4.1 and 4.2 illustrates the performance of BOW features with ML. When we assess performance (Table 4.1), considering CUD dataset, it is evident that the highest F-score of 0.9113 is achieved with SVM when combined with the TF-IDF weighting method, followed by MNB with Binary at 0.9040, and MNB with TF at 0.9026. Similarly, on the ATC dataset, SVM with Binary attains the top F-score of 0.9277, closely followed by SVM with TF at 0.9275, and SVM with TF-IDF at 0.91976. Notably, SVM consistently outperforms other classifiers across all term weighting methods with BOW. TF-IDF demonstrates superior performance compared to other term weighting methods in this context.

Conversely, when analyzing (Table 4.2), regarding KPD dataset, RF with TF-IDF achieves the highest F-score of 0.7869, followed by LR with TF at 0.7689, and LR with Binary at 0.7664. On the FSD dataset, SVM with TF-IDF leads with an F-score of 0.8714, LR with TF follows at 0.8480, and SVM with Binary closes at the third position. It's noteworthy that for Turkish datasets, the best overall performance with BOW features and ML is observed with SVM on the ATC dataset, while for English datasets, it's achieved with SVM on the FSD dataset with TF-IDF.

Table 4.3 and Table 4.4 illustrates the performance of N-gram features with ML. when we consider Table 4.3, for CUD dataset, SVM with TF-IDF exhibits the highest F-score of 0.9130, followed by LR with Binary at 0.9102, and LR with TF at 0.9086. Similarly, on the ATC dataset, RF with TF-IDF leads at 0.9296, followed by SVM with TF at 0.9270, and SVM with Binary at 0.9255. SVM and RF consistently demonstrate superior performance in this category.

In contrast, evaluating the performance (Table 4.4), on the KPD dataset, LR with TF-IDF achieves the highest F-score of 0.8437, followed by MNB with TF at 0.8393, and LR with Binary at 0.8338. On the FSD dataset, LR with TF-IDF stands out with an F-score of 0.8879, followed by SVM with TF at 0.8691, and LR with Binary at 0.8442. Notably, TF and TF-IDF exhibit better performance compared to Binary in this scenario. Overall, the best-performing algorithm for Turkish datasets using N-gram features and ML is observed with RF on the ATC dataset with TF-

IDF weighting. For English datasets, the best performance is achieved with LR on the FSD dataset with TF-IDF.

Table 4.1. Results of Term Weighting and BOW with Traditional ML Classifiers on Turkish Datasets

		BOW					
		Binary		**TF**		**TFIDF**	
Dataset	**Classifier**	ACC	F-score	ACC	F-score	ACC	F-score
CUD	MNB	0.9074	**0.9040**	0.9058	**0.9026**	0.9064	0.9029
	SVM	0.8937	0.8967	0.8929	0.8958	0.9113	**0.9113**
	LR	0.8959	0.8999	0.9011	0.9011	0.8938	0.8993
	RF	0.8517	0.8393	0.8512	0.8384	0.8568	0.8469
	DT	0.8166	0.7993	0.8169	0.8004	0.8072	0.8087
	KNN	0.8225	0.8551	0.8187	0.8261	0.8249	0.8296
ATC	MNB	0.8976	0.8996	0.8970	0.8988	0.8929	0.8957
	SVM	0.9285	**0.9277**	0.9284	0.**9275**	0.9197	**0.9176**
	LR	0.9207	0.9181	0.9200	0.9174	0.8889	0.8832
	RF	0.8912	0.8990	0.8918	0.8996	0.8982	0.9048
	DT	0.8712	0.8824	0.8717	0.8829	0.8680	0.8798
	KNN	0.8440	0.8206	0.8434	0.8146	0.8145	0.8018

Table 4.2. Results of Term Weighting and BOW with Traditional ML Classifiers on English Datasets

		BOW					
		Binary		**TF**		**TFIDF**	
Dataset	**Classifier**	ACC	F-score	ACC	F-score	ACC	F-score
KPD	MNB	0.797	0.7492	0.7864	0.7518	0.8155	0.7849
	SVM	0.7892	0.7564	0.7872	0.7524	0.8089	0.7855
	LR	0.8095	**0.7664**	0.8014	**0.7689**	0.8022	0.7813
	RF	0.8029	0.7506	0.8013	0.7678	0.8016	**0.7869**
	DT	0.7424	0.7452	0.7481	0.7525	0.7546	0.7837
	KNN	0.7512	0.7388	0.7409	0.7562	0.7554	0.7745
FSD	MNB	0.9400	0.8181	0.8722	0.8175	0.8579	0.8221
	SVM	0.9265	**0.8422**	0.9203	0.8300	0.9269	**0.8714**
	LR	0.9299	0.8337	0.9264	**0.8480**	0.9226	0.8627
	RF	0.9432	0.8299	0.9443	0.8301	0.9465	0.8542
	DT	0.9235	0.7653	0.9236	0.7643	0.9149	0.8617
	KNN	0.8969	0.7646	0.8894	0.7624	0.9263	0.8022

Table 4.3. Results of Term Weighting and N-GRAM with Traditional ML Classifiers for Turkish Datasets

		N-GRAM					
		Binary		**TF**		**TFIDF**	
Dataset	**Classifier**	ACC	F-score	ACC	F-score	ACC	F-score
CUD	MNB	0.8959	0.8987	0.8939	0.8967	0.8982	0.9010
	SVM	0.8943	0.8946	0.8952	0.8956	0.9115	**0.9130**
	LR	0.9194	**0.9102**	0.9078	**0.9086**	0.9066	0.9091
	RF	0.897	0.8994	0.8972	0.8995	0.8926	0.8962
	DT	0.8364	0.8351	0.8329	0.8310	0.8337	0.8335
	KNN	0.8165	0.8071	0.8152	0.8056	0.8881	0.8885
ATC	MNB	0.8685	0.8646	0.8668	0.8624	0.8611	0.8580
	SVM	0.9216	**0.9255**	0.9256	**0.9270**	0.9275	0.9298
	LR	0.9034	0.9019	0.9035	0.9019	0.9180	0.9128
	RF	0.9182	0.9189	0.9178	0.9184	0.9201	**0.9296**
	DT	0.9121	0.9158	0.9137	0.9172	0.9289	0.9232
	KNN	0.8251	0.8664	0.8261	0.8616	0.8015	0.8955

Table 4.4. Results of Term Weighting and N-GRAM with Traditional ML Classifiers for English Datasets

		N-GRAM					
		Binary		**TF**		**TFIDF**	
Dataset	**Classifier**	ACC	F-score	ACC	F-score	ACC	F-score
KPD	MNB	0.8227	0.8142	0.839	**0.8393**	0.8363	0.8218
	SVM	0.8243	0.8116	0.8415	0.831	0.8524	0.8388
	LR	0.8187	**0.8338**	0.8279	0.8323	0.8387	**0.8437**
	RF	0.8122	0.7986	0.7845	0.7663	0.7948	0.7991
	DT	0.8041	0.7964	0.8312	0.8216	0.8414	0.8318
	KNN	0.7784	0.7502	0.7691	0.7939	0.7504	0.8026
FSD	MNB	0.8543	0.8510	0.8541	0.8458	0.8483	0.8791
	SVM	0.9201	0.8716	0.9152	**0.8691**	0.9156	0.8657
	LR	0.9206	0.8686	0.9181	0.8676	0.9127	**0.8879**
	RF	0.9447	**0.8442**	0.9451	0.8543	0.9455	0.8768
	DT	0.9062	0.8620	0.9088	0.8622	0.9038	0.8518
	KNN	0.8918	0.8612	0.8760	0.8492	0.8918	0.8869

Overall results on Term weighting with ML present the following findings;

- TF-IDF emerges as the superior feature weighting method overall when compared to Binary and TF.
- SVM performs best across the feature weighting as compared to the other classifiers
- The most effective classifiers across all languages are SVM, MNB, LR, RF, and DT.
- In the context of BOW features with ML, the highest performance for Turkish datasets is observed with SVM on the ATC dataset, achieving an F-score of 0.9277 when paired with Binary weighting. Conversely, for English datasets, optimal performance is attained with SVM combined with TF-IDF on the FSD dataset, yielding an F-score of 0.8714.
- When considering N-gram features with ML, the top performance for Turkish datasets is found with RF on the ATC dataset, achieving an F-score of 0.9296 when paired with TF-IDF. Meanwhile, for English datasets, the best performance is observed with LR on the FSD dataset, utilizing TF-IDF, and yielding an F-score of 0.8879.
- While both the N-gram model and BOW model demonstrate comparable performance, the N-gram model exhibits superior performance, making it the preferred choice for feature extraction. Apparently, the reason for N-gram as better performer lies its architecture where it is used to consider words in their sequence which makes much sense when we are using sentences as the context is important whereas BOW simply bags the words in its vocabulary irrespective for their order.

4.2. Effect of Trained and Pre-trained Word Embeddings Models with Traditional ML Methods

The next stage of the experiment focused on combining traditional ML with word embedding to see how it affects performance. This experiment evaluated both custom-trained and pre-trained word embedding models alongside traditional ML classifiers.

After cleaning the text data, word embedding techniques like FastText, Word2Vec, and GloVe were utilized to transform the text into numerical vectors suitable for classification by ML models. Each word was converted into a vector, and the vector for the entire document was created by averaging the word vectors of all the words in the document.

Custom word embedding models were trained with a specific size (300 dimensions) and context window (5 words around a target word). Pre-trained word embeddings with the same size (300 dimensions) were also included for comparison. The results of this experiment are presented in tables (4.5 - 4.8). Table 4.5 and Table 4.6 shows the performance of trained word embeddings with traditional ML methods, while Table 4.7 and Table 4.8 explores the effectiveness of pre-trained word embeddings with these ML models.

In Table 4.5 and Table 4.6, the assessment of performance with trained word vectors and ML on the CUD dataset reveals KNN as the top performer when used with GloVe, achieving an F-score of 0.5989. SVM follows closely as the second-best performer when combined with FastText, yielding an F-score of 0.555, while LR ranks third when utilized with Word2Vec, attaining an F-score of 0.5453. Similarly, on the ATC dataset, the best result is obtained from MNB paired with FastText, garnering an F-score of 0.5972, followed by DT with GloVe at 0.5921, and KNN with Word2Vec at 0.5578.

Conversely, in the evaluation of the KPD dataset, LR with Word2Vec stands out with the highest F-score of 0.4806, followed by LR with GloVe at 0.4794, and RF with FastText at 0.4772. For the FSD dataset, MNB with FastText delivers the best result with an F-score of 0.5986, with GloVe and Word2Vec following as second and third performers, respectively. It's noteworthy that trained word vectors did not yield significant results in this experiment.

Overall, for the Turkish dataset, the highest performance with trained word vectors and ML is observed on the CUD dataset, utilizing KNN with GloVe. Conversely, for the English dataset, optimal performance in this category is achieved on the FSD dataset with MNB when paired with FastText.

Table 4.5. Results of Trained Word Embedding Models with Traditional ML Classifiers on Turkish Datasets

Dataset	**Classifier**	**Trained**					
		Word2vec		**GloVe**		**FastText**	
		ACC	F-score	ACC	F-score	ACC	F-score
CUD	MNB	0.5484	0.5036	0.5563	0.5303	0.5616	0.5222
	SVM	0.5717	0.5439	0.5628	0.5884	0.561	**0.5550**
	LR	0.5754	**0.5453**	0.5712	0.5164	0.5718	0.5158
	RF	0.5774	0.538	0.5793	0.5323	0.5791	0.5319
	DT	0.5696	0.5022	0.5717	0.5041	0.5673	0.5002
	KNN	0.5557	0.595	0.5417	**0.5989**	0.5626	0.5465
ATC	MNB	0.5798	0.5148	0.5718	0.5871	0.5594	**0.5972**
	SVM	0.5919	0.5435	0.5878	0.5317	0.5897	0.5603
	LR	0.5963	0.5506	0.5966	0.5507	0.5953	0.5534
	RF	0.5247	0.5312	0.5859	0.5718	0.5750	0.5763
	DT	0.5228	0.5535	0.5920	**0.5921**	0.6222	0.5723
	KNN	0.5517	**0.5578**	0.5554	0.5548	0.5590	0.5677

Table 4.6. Results of Trained Word Embedding Models with Traditional ML Classifiers on English Datasets

Dataset	Classifier	Trained					
		Word2vec		GloVe		FastText	
		ACC	F-score	ACC	F-score	ACC	F-score
KPD	MNB	0.3978	0.4710	0.3983	0.4715	0.3986	0.4708
	SVM	0.3937	0.4799	0.4501	0.4785	0.4535	0.3994
	LR	0.3937	**0.4806**	0.4477	**0.4794**	0.4532	0.4068
	RF	0.3964	0.4770	0.3964	0.4772	0.3964	**0.4772**
	DT	0.4499	0.4048	0.4493	0.4043	0.449	0.4040
	KNN	0.4079	0.4381	0.4279	0.4431	0.4173	0.4339
FSD	MNB	0.5759	0.4434	0.7822	0.5013	0.5986	**0.5986**
	SVM	0.5768	**0.5456**	0.5964	0.5432	0.5540	0.5931
	LR	0.5760	0.5453	0.5843	**0.5821**	0.5753	0.5778
	RF	0.5360	0.5329	0.5536	0.5658	0.5764	0.5868
	DT	0.5349	0.5353	0.5349	0.5431	0.5490	0.5432
	KNN	0.5353	0.5295	0.5353	0.5409	0.5173	0.5639

When evaluating the performance of pre-trained word vectors with ML, as illustrated in Table 4.7 and Table 4.8, focusing on the CUD dataset, reveals SVM as the top-performing algorithm when paired with FastText, achieving an F-score of 0.8470. GloVe follows closely as the second-best performer with an F-score of 0.8270 when utilized alongside SVM, and Word2Vec trails behind with an F-score of 0.8139 in combination with SVM. Notably, SVM demonstrates the best performance across all word vectors in this context. Turning to the ATC dataset, SVM paired with FastText delivers the best performance with an F-score of 0.8422, while GloVe ranks second when utilized with SVM, achieving an F-score of 0.8395, and Word2Vec performs third with an F-score of 0.8434 when combined with SVM.

In terms of the KPD dataset, the highest result is attained with FastText when used with SVM, achieving an F-score of 0.7992. Following this, RF paired with GloVe yields an F-score of 0.7944, and Word2Vec with SVM closes this ranking with an F-score of 0.7514. Transitioning to the FSD dataset, SVM stands out with the best performance, obtaining an F-score of 0.8574. KNN ranks second when utilized with GloVe, achieving an F-score of 0.8543, while DT paired with Word2Vec concludes this ranking with an F-score of 0.8353.

Overall performance concerning pre-trained word vectors with ML, for the Turkish dataset, specifically the CUD dataset, SVM paired with FastText achieved the highest F-score of 0.8470, whereas for the English dataset, the FSD dataset exhibited the best overall performance, with SVM coupled with FastText yielding an F-score of 0.8574.

Table 4.7. Results of Pre-Trained Word Embedding Models with Traditional ML Methods on Turkish Datasets

Dataset	Classifier	Pre-trained					
		Word2vec		GloVe		FastText	
		ACC	F-score	ACC	F-score	ACC	F-score
CUD	MNB	0.8028	0.8034	0.8074	0.7547	0.8047	0.8061
	SVM	0.8015	**0.8139**	0.8113	**0.8270**	0.8388	**0.8470**
	LR	0.8021	0.8123	0.8294	0.8194	0.8422	0.8423
	RF	0.7802	0.8000	0.7353	0.7400	0.8353	0.8400
	DT	0.7981	0.7087	0.7562	0.7454	0.7488	0.7521
	KNN	0.7771	0.7811	0.7013	0.7038	0.7277	0.7743
ATC	MNB	0.7687	0.7676	0.8081	0.8083	0.8168	0.8174
	SVM	0.8336	**0.8434**	0.8381	**0.8395**	0.8420	**0.8422**
	LR	0.8069	0.8023	0.8117	0.8161	0.8234	0.8232
	RF	0.8218	0.8266	0.8388	0.8382	0.8408	0.8388
	DT	0.8233	0.8299	0.8345	0.8394	0.8355	0.8388
	KNN	0.8040	0.8043	0.8180	0.8106	0.8022	0.8235

Table 4.8. Results of Pre-Trained Word Embedding Models with Traditional ML Methods on English Datasets

Dataset	Classifier	Pre-trained					
		Word2vec		GloVe		FastText	
		ACC	F-score	ACC	F-score	ACC	F-score
KPD	MNB	0.7344	0.7453	0.7414	0.7426	0.7681	0.7683
	SVM	0.7528	**0.7514**	0.7644	0.7687	0.7758	**0.7992**
	LR	0.7519	0.7404	0.7610	0.7603	0.7717	0.7761
	RF	0.7011	0.7014	0.7934	**0.7944**	0.7847	0.7859
	DT	0.7163	0.7312	0.7094	0.7258	0.7207	0.7356
	KNN	0.7273	0.7384	0.7130	0.7274	0.7205	0.7342
FSD	MNB	0.8063	0.8133	0.8191	0.8196	0.8203	0.8224
	SVM	0.8174	0.8161	0.8441	0.8355	0.8533	**0.8574**
	LR	0.8146	0.8119	0.8067	0.8058	0.8069	0.8011
	RF	0.8013	0.8013	0.8131	0.8131	0.8342	0.8467
	DT	0.8244	**0.8353**	0.8513	0.8527	0.8564	0.8573
	KNN	0.8215	0.8265	0.8571	**0.8543**	0.8608	0.8663

The comprehensive results on word embedding with ML reveal several key findings:

- FastText word embedding outperformed GloVe and Word2Vec.

- Generally, custom trained word vectors did not yield significant results.
- Pre-trained word vectors generally outperformed custom trained ones.
- SVM consistently demonstrated superior performance across all models.
- When considering custom trained word vectors with ML, the top performance for the Turkish dataset was observed on the CUD dataset, with KNN paired with GloVe achieving an F-score of 0.5989, while for the English dataset, the FSD dataset displayed the best performance, with MNB combined with FastText producing an F-score of 0.5986.
- When examining pre-trained word vectors with ML, the best performance for the Turkish dataset was found on the CUD dataset, with SVM coupled with FastText achieving an F-score of 0.8470, while for the English dataset, the FSD dataset demonstrated optimal performance, with SVM paired with FastText yielding an F-score of 0.8574.

4.3. Effect of Using Trained and Pre-Trained Word Embeddings with DL Methods

In this section of the experiment, DL methods with word vectors were employed and compared. DL utilizes word embeddings to automatically construct features, with two distinct approaches: custom embedding, trained on a specific corpus, and pre-trained embedding, already trained on extensive datasets. Impact of utilizing both custom-trained and pre-trained word vectors on DL algorithms for cyberbullying detection was investigated in this thesis. The chosen word embedding models included Word2vec, GloVe, and FastText.

For custom-trained word vectors, training was conducted with a vector size of 300 and a window context of 5. In contrast, for pre-trained models, we consistently utilized word vectors of size 300 for comparative analysis. Various architectures for deep learning for classification, particularly CNN and RNN, including LSTM, GRU, BiLSTM, BiGRU, as well as combined CNN-LSTM and LSTM-CNN architectures were investigated.

DL experiments were divided into two categories: examining the performance of CNN architecture separately and evaluating RNN with both trained and pre-trained word vectors. For CNN algorithms, we designed CNN architectures akin to those in (Kim, 2014), with parameter adjustments including filter sizes of 3, 4, and 5, 128 filters, a dropout rate of 0.5, and batch sizes of 124 over 11 epochs. Conversely, our RNN configuration comprised three stacked layers of LSTM, GRU, BiLSTM, and BiGRU, each containing 16 neurons, a dropout rate of 0.2, Adam optimizer, and batch sizes of 128 over 11 epochs. Additionally, the combination of CNN-LSTM and LSTM-CNN units were explored to assess their potential enhancement of overall performance. The developed DL architectures are depicted in appendix section (Appendix A.1 to A.7).

The results of the experiment in this domain are presented in Table 4.9 to 4.12, with the former displaying outcomes of custom-trained word vectors with DL and the latter showcasing

results of pre-trained word vectors with DL. When examining the performance of trained word vectors with DL algorithms, as shown in Table 4.9, CNN-LSTM emerged as the top performer for the CUD dataset, achieving an F-score of 0.6696 when integrated with FastText. Still CNN-LSTM performed second and third with GloVe and Word2ve giving F-score of 0.669 and 0.6593.

Similarly, for the ATC dataset, CNN-LSTM paired with FastText demonstrated the highest performance with an F-score of 0.6215.

In contrast as is seen in table 4.10, for the KPD dataset, FastText combined with CNN-LSTM yielded the best result with an F-score of 0.5698. Lastly, for the FSD dataset, BiLSTM with Word2vec attained the highest F-score of 0.6899, followed by CNN-LSTM when used with FastText with giving an F-score of 0.6898 as second best and BiGRU combined with GloVe with an F-score of 0.6897 respectively.

Overall, the top-performing algorithm for the Turkish dataset was observed on the CUD dataset, utilizing CNN-LSTM with FastText, while for the English dataset, the best performance was demonstrated on the FSD dataset, with BiLSTM paired with Word2vec.

Table 4.9. Results of Trained Word Embeddings Models with DL Methods on Turkish Datasets

Dataset	**Classifier**	**Trained**					
		Word2vec		**GloVe**		**FastText**	
		ACC	F-score	ACC	F-score	ACC	F-score
CUD	CNN	0.6508	0.6509	0.6953	0.6954	0.6530	0.6532
	LSTM	0.6499	0.6498	0.6534	0.6533	0.6558	0.6559
	GRU	0.6546	0.6545	0.6173	0.6174	0.6549	0.6549
	BiLSTM	0.6555	0.6554	0.6579	0.6578	0.6574	0.6574
	BiGRU	0.6558	0.6560	0.6678	0.6677	0.6617	0.6617
	CNN-LSTM	0.6593	**0.6593**	0.6694	**0.6694**	0.6695	**0.6696**
	LSTM-CNN	0.6543	0.6543	0.6683	0.6684	0.6682	0.6682
ATC	CNN	0.6036	0.6051	0.6135	0**.6122**	0.6126	0.6100
	LSTM	0.6021	0.6061	0.6061	0.6052	0.6081	0.6048
	GRU	0.6025	0.6064	0.6092	0.6083	0.6059	0.6060
	BiLSTM	0.6072	0.6061	0.6079	0.6066	0.6071	0.6079
	BiGRU	0.6027	0.6029	0.6091	0.6083	0.6097	0.6088
	CNN-LSTM	0.6167	**0.6183**	0.6108	0.6112	0.6227	**0.6215**
	LSTM-CNN	0.6054	0.6054	0.6111	0.6106	0.6044	0.6000

Table 4.10. Results of Trained Word embeddings Models with DL Methods on English Datasets

Dataset	Classifier	Trained					
		Word2vec		**GloVe**		**FastText**	
		ACC	F-score	ACC	F-score	ACC	F-score
KPD	CNN	0.5687	0.5674	0.5685	0.5695	0.5684	0.5680
	LSTM	0.5661	0.5653	0.5665	0.5660	0.5670	0.5675
	GRU	0.5637	0.5640	0.5616	0.5640	0.5649	0.5649
	BiLSTM	0.5695	0.5670	0.5644	0.5690	0.5663	0.5688
	BiGRU	0.5648	0.5646	0.5669	0.5640	0.5652	0.5654
	CNN-LSTM	0.5694	**0.5694**	0.5695	**0.5695**	0.5689	**0.5698**
	LSTM-CNN	0.5676	0.5669	0.5620	0.5634	0.5661	0.5657
FSD	CNN	0.6881	0.6889	0.6887	0.6885	0.6882	0.6891
	LSTM	0.6869	0.6870	0.6875	0.6873	0.6800	0.6876
	GRU	0.6892	0.6897	0.6890	0.6893	0.6881	0.6880
	BiLSTM	0.6898	**0.6899**	0.6869	0.6885	0.6895	0.6898
	BiGRU	0.6894	0.6895	0.6897	**0.6897**	0.6883	0.6881
	CNN-LSTM	0.6895	0.6895	0.6896	0.6896	0.6898	**0.6898**
	LSTM-CNN	0.6852	0.6854	0.6848	0.6846	0.6863	0.6860

In the assessment of pre-trained models with DL, taking CUD dataset as seen in Table 4.11, BILSTM paired with the FastText model demonstrated the highest performance, achieving an F-score of 0.9189. Following closely, CNN-LSTM with GloVe yielded the second-best performance with an F-score of 0.9134, and CNN-LSTM combined with Word2vec produced an F-score of 0.8963, ranking third. Moving to the ATC dataset, CNN integrated with FastText exhibited the highest score of 0.9376, followed by CNN with Word2vec at 0.9211, and CNN with GloVe at 0.9231. Notably, FastText outperformed Word2vec and GloVe, while CNN-LSTM and CNN emerged as the top-performing DL algorithms in this task.

For the KPD dataset as is seen in Table 4.12, CNN-LSTM paired with FastText achieved an F-score of 0.9694, with BILSTM combined with GloVe following closely at 0.9689, and CNN-LSTM with Word2vec achieving an F-score of 0.9660. Moving to the FSD dataset, the highest performance was seen with CNN-LSTM combined with FastText, scoring 0.9498, followed by BiLSTM with GloVe at 0.9398, and Word2vec with BiLSTM at 0.9210. CNN-LSTM and BiLSTM emerged as the top-performing algorithms, with FastText demonstrating superior performance among the word vectors.

Table 4.11. Results of Pre-trained Word Embeddings Models with DL Methods on Turkish Datasets

Dataset	Classifier	Pre-trained					
		Word2vec		GloVe		FastText	
		ACC	F-score	ACC	F-score	ACC	F-score
CUD	CNN	0.8886	0.8886	0.9128	0.9128	0.9069	0.9070
	LSTM	0.8840	0.8840	0.8904	0.8904	0.905	0.9054
	GRU	0.8868	0.8868	0.8916	0.8916	0.8920	0.8920
	BILSTM	0.8874	0.8874	0.8965	0.8965	0.9189	**0.9189**
	BiGRU	0.8854	0.8854	0.8940	0.8940	0.9023	0.9043
	CNN-LSTM	0.8963	**0.8963**	0.9134	**0.9134**	0.9188	0.9188
	LSTM-CNN	0.8863	0.8863	0.9069	0.9069	0.9102	0.9102
ATC	CNN	0.9231	**0.9231**	0.9211	**0.9211**	0.9376	**0.9376**
	LSTM	0.8860	0.8860	0.9022	0.9022	0.9203	0.9203
	GRU	0.8774	0.8774	0.9038	0.9038	0.9253	0.9253
	BILSTM	0.8895	0.8895	0.9068	0.9068	0.9227	0.9227
	BiGRU	0.8844	0.8844	0.9060	0.9060	0.9260	0.9260
	CNN-LSTM	0.8924	0.8924	0.9148	0.9148	0.9338	0.9338
	LSTM-CNN	0.8914	0.8914	0.9085	0.9085	0.9312	0.9312

Table 4.12. Results of pre-trained Word Embeddings Models with DL Methods on English Datasets

Dataset	Classifier	Pre-trained					
		Word2vec		GloVe		FastText	
		ACC	F-score	ACC	F-score	ACC	F-score
KPD	CNN	0.9663	0.9651	0.9685	0.9663	0.9679	0.9666
	LSTM	0.9525	0.9349	0.9693	0.9556	0.9541	0.9412
	GRU	0.9555	0.9544	0.9720	0.9764	0.9390	0.9384
	BILSTM	0.9574	0.9652	0.9691	**0.9689**	0.9641	0.8755
	BiGRU	0.9687	0.9644	0.9780	0.9654	0.9534	0.9463
	CNN-LSTM	0.9679	**0.9660**	0.9699	0.9684	0.9732	**0.9694**
	LSTM-CNN	0.9452	0.9444	0.9592	0.9569	0.9142	0.9036
FSD	CNN	0.9107	0.9053	0.9131	0.9061	0.9224	0.9188
	LSTM	0.8327	0.8264	0.8410	0.8502	0.9393	0.9322
	GRU	0.9089	0.9043	0.9194	0.9118	0.9439	0.9420
	BILSTM	0.9229	**0.9210**	0.9315	**0.9398**	0.9427	0.9455
	BiGRU	0.9145	0.9085	0.9315	0.9369	0.9415	0.9488
	CNN-LSTM	0.8990	0.8990	0.9278	0.9269	0.9486	**0.9498**
	LSTM-CNN	0.8856	0.8831	0.9269	0.9239	0.9320	0.9340

Key findings from results on word embedding with DL reveal several key findings:

- Pre-trained models showed significantly better performance than trained ones since pre-trained models have been trained on very large corpora of various sizes and nature to incorporate all the aspects of the concerned topic which is also evident from our experiments and in line with the general studies in this field.
- A combination of CNN-LSTM exhibited superior performance across all models.

- FastText outperformed overall best throughout the DL models since the FastText word embedding incorporates character level word representation (breaking words to character level n-grams, which is good for morphologically rich languages like Turkish) and it also incorporates the out of vocabulary (OOV) words that are rare and unseen during training. It is also good for spelling errors.
- The overall performance of pre-trained word vectors with DL revealed that for the Turkish dataset, the highest score was obtained from the ATC dataset, with CNN combined with FastText achieving a score 0.9376, while for the English dataset, the best performance was observed on the KPD dataset, with CNN-LSTM paired with FastText achieving an F-score of 0.9694.

4.4. Effect of Pre-Trained BERT Models

In this part of experimentation, the focus was on assessing various pretrained large language models, particularly BERT models, in comparison to the ML and DL models utilized previously. Nine different forms of BERT, with five dedicated to English and four to Turkish language were considered. For the English BERT variants, Bert-base-uncased, bert-base-multilingual-uncased, Roberta (Roberta-base), Distilbert-base-uncased, and Albert-base were utilized. For Turkish, bert-base-multilingual-uncased, Dbmdz/bert-base-Turkish-128k-case, Dbmdz/electra-base-Turkish, and Dbmdz/distilbert-base-Turkish-case were considered and evaluated. Additionally, runtime for these models are provided.

The results of this experiment are presented in Table 4.13 and Table 4.14, with the former displaying the outcomes of pre-trained BERT on English datasets and the latter showcasing the results on Turkish datasets. As depicted in Table 4.13, when evaluating the performance of BERT models on the KPD dataset, bert-base-multilingual achieved the highest performance with an F-score of 0.9892. Following closely, Distilbert exhibited the second-best performance with an F-score of 0.9885, and Roberta ranked third with an F-score of 0.9881. Albert-base and Bert-base followed suit with F-scores of 0.9792 and 0.9777, respectively. Shifting focus to the FSD dataset, Roberta demonstrated the best performance with an F-score of 0.9859, while Distilbert achieved the second-best performance with an F-score of 0.9784. Bert-base, bert-base-multilingual, and Albert-base followed with F-scores of 0.9777, 0.9758, and 0.9754, respectively. It is noteworthy that all five BERT variants exhibited exceptionally good performance.

Considering runtime, Albert-base showed the shortest runtime of 660.65 seconds, followed by Distilbert-base-uncased with a runtime of 669.31s. Bert-base-uncased followed with a runtime of 673.21 s, Roberta with 805.4306, and bert-base-multilingual with the longest runtime of 893.7336s.

Table 4.13. Results of Pre-trained BERT on English Dataset

	KPD			FSD		
BERT MODEL	ACC	F1-Score	Run time/sec	ACC	F1-Score	Run time/sec
bert-base-multilingual-uncased	0.9891	**0.9892**	893.73	0.9755	0.9758	1,389.13
Bert-base-uncased	0.9770	0.9777	673.21	0.9770	0.9777	1,376.41
Roberta (Roberta-base)	0.9880	0.9881	805.43	0.9847	**0.9859**	1,530.36
Distilbert-base-uncased	0.9884	0.9885	669.31	0.9776	0.9784	771.62
Albert-base	0.9794	0.9792	660.65	0.9752	0.9754	713.85

However, in assessing the performance of BERT models on the CUD dataset presented in Table 4.14, the most exceptional performance was observed with Dbmdz/bert-base-Turkish-128k-case, achieving an F-score of 0.9778, closely followed by Dbmdz/electra-base-Turkish with an F-score of 0.9777, and Dbmdz/distilbert-base-Turkish-case with an F-score of 0.9752. Lastly, bert-base-multilingual yielded an F-score of 0.9749. It's notable that all BERT variants demonstrated strong performance, with minor variations in their effectiveness. However, in line run time, Dbmdz/distilbert-base-Turkish-case had shortest run time of 512.18 seconds as compared to the others.

Considering the outcomes from the ATC dataset, the most notable result was attained with Dbmdz/bert-base-Turkish-128k-case, registering an F-score of 0.9725, followed by bert-base-multilingual with an F-score of 0.9703, Dbmdz/distilbert-base-Turkish-case with an F-score of 0.9685, and lastly, Dbmdz/electra-base-Turkish producing an F-score of 0.9634. Regarding runtime, Dbmdz/distilbert-base-Turkish-case demonstrated the shortest runtime of 852.52 seconds, followed by Dbmdz/electra-base-Turkish with a runtime of 1,159.71seconds, Dbmdz/bert-base-Turkish-128k-case with 1,373.14s, and the longest runtime was observed with bert-base-multilingual, at 1,651.44s.

Table 4.14. Results of Pre-trained BERT on Turkish Dataset.

	CUD			ATC		
BERT MODEL	ACC	F1-Score	Run time/sec	ACC	F1-Score	Run time/sec
bert-base-multilingual-uncased	0.9746	0.9749	1,201.92	0.9711	0.9703	1,651.44
Dbmdz/bert-base-Turkish-128k-case	0.9773	**0.9778**	1,285.10	0.9725	**0.9725**	1,373.14
Dbmdz/electra-base-Turkish	0.9772	0.9777	890.60	0.9657	0.9634	1,159.71
Dbmdz/distilbert-base-Turkish-case	0.9752	0.9752	512.18	0.9689	0.9685	852.52

Overall performance of different BERT variants shows that for English dataset, the highest performance was achieved on the KPD dataset, with bert-base-multilingual-uncased attaining an F-score of 0.9892. Conversely, for the Turkish dataset, the best overall performance was observed on the CUD dataset, with Dbmdz/bert-base-Turkish-128k-case yielding an F-score of 0.9778. The most efficient runtime for the English dataset was recorded on the KPD dataset using Albert-base, with a runtime of 660.65, while for the Turkish dataset, Dbmdz/distilbert-base-Turkish-case demonstrated the shortest runtime of 852.52.

In summary, the analysis on BERT models yielded the following insights:

- All BERT variants demonstrated outstanding results with minor differences.
- The results of BERT are higher than those of DL and ML
- bert-base-multilingual-uncased performed exceptionally well overall for the Turkish dataset, achieving an F-score of 0.9892, while Dbmdz/bert-base-Turkish-128k-case showed the best performance overall for the Turkish dataset with an F-score of 0.9778.
- The Turkish dataset exhibited a shorter runtime, with Dbmdz/distilbert-base-Turkish-case completing in 512.18, compared to the English dataset, which took 660.65 seconds using Albert-base.

4.5. Effect of Using Pre-Trained BERT with ML Algorithms

To further enhance the performance of ML models, we opted to integrate BERT with ML classifiers, considering the superior performance of BERT observed in previous experiments. In this approach, pre-trained BERT embeddings are utilized as features to train ML models. Various traditional ML classifiers including SVM, MNB, LR, DT, K-NN, and RF are utilized, with default parameter settings. For English, Bert-base (uncased), Bert-base (multilingual-uncased), Roberta (Roberta-base), and Distilbert (Distilbert-base-uncased) variants are considered, while for Turkish, Bert-base (multilingual-uncased), dbmdz (bert-base-turkish-128k-cased), and Dbmdz (distilbert-base-Turkish-case) are as well employed. The results for this configuration are presented in Table 4. 15 which depicts pre-trained BERT with ML on English datasets and Table 4.16 showing results of using pre-trained BERT with ML on Turkish datasets.

Upon evaluating the model performance based on Table 4.15, particularly the KPD dataset, the best performance was observed with Bert-base multilingual integrated with MNB, achieving an F-score of 0.9576. Following closely, Distilbert, when paired with LR, demonstrated the second-best performance with an accuracy of 0.9377. Roberta followed as the third-best performer with an F-score of 0.9374 when used with LR, while Bert-base obtained the lowest F-score of 0.9243 when integrated with LR. On the other hand, for the FSD dataset, Roberta performed the overall best when paired with LR, achieving an F-score of 0.9465. Bert-base, in combination with LR, followed

as the second-best performer with an F-score of 0.9443, succeeded by Bert-base multilingual with an F-score of 0.9422 when used with LR. Lastly, Distilbert, integrated with DT, achieved an F-score of 0.9287. It is noteworthy that LR, as a traditional classifier, formed better combinations with BERT models.

Table 4.15. Results of Pre-trained BERT with ML on English Dataset

		KPD		FSD	
BERT VARIANTS + ML		ACC	F1-Score	ACC	F1-Score
Bert-base-uncased	SVM	0.9237	0.9283	0.9395	0.9296
	MNB	0.9373	0.9053	0.9296	0.9397
	LR	0.9215	**0.9243**	0.9462	**0.9443**
	DT	0.9130	0.9161	0.9254	0.9253
	KNN	0.9191	0.9117	0.9120	0.9113
	RF	0.9239	0.9139	0.9378	0.9278
bert-base-multilingual-	SVM	0.9333	0.9385	0.9337	0.9378
	MNB	0.9416	**0.9576**	0.9339	0.9341
	LR	0.8396	0.9359	0.9466	**0.9422**
	DT	0.9262	0.9288	0.9361	0.9361
	KNN	0.9234	0.9324	0.9074	0.9168
	RF	0.9310	0.9356	0.9278	0.9078
Roberta (Roberta-base)	SVM	0.9441	0.8970	0.9235	0.9335
	MNB	0.9354	0.9354	0.9486	0.9367
	LR	0.9336	**0.9374**	0.9431	**0.9465**
	DT	0.9373	0.8999	0.9315	0.9315
	KNN	0.8981	0.9002	0.9249	0.9245
	RF	0.9257	0.9281	0.9232	0.9188
Distilbert (base-uncased)	SVM	0.9286	0.9286	0.9263	0.9032
	MNB	0.9348	0.9354	0.9109	0.9208
	LR	0.9443	**0.9377**	0.9238	0.9237
	DT	0.9094	0.9130	0.9333	**0.9287**
	KNN	0.9220	0.9292	0.9248	0.9244
	RF	0.9041	0.9162	0.9061	0.9161

In the context of the Turkish dataset, as illustrated in table 4.16, when examining the performance on the CUD dataset, the highest performance is achieved by dbmdz/bert-base-turkish-128k-cased attaining an F-score of 0.9492 with DT. Following this, Dbmdz/distilbert-base-Turkish-case, emerges as the second-best performer with an F-score of 0.9488 when utilized in conjunction with MNB, while bert-base-multilingual-uncased exhibits the lowest performance, yielding an F-score of 0.9246 when paired with LR. On the other hand, for the ATC dataset, Dbmdz/distilbert-base-Turkish-case, combined with SVM, proves to be the most effective model, achieving an F-score of 0.9478. Subsequently, dbmdz/bert-base-turkish-128k-cased performs as the second-best model with an F-score of 0.9376 when integrated with DT, followed by bert-base-multilingual-uncased, which yields an F-score of 0.9376 when utilized with LR.

Table 4.16. Results of Using Pre-trained BERT with ML on Turkish Dataset

BERT VARIANTS + ML		CUD		ATC	
		ACC	F1-Score	ACC	F1-Score
bert-base-multilingual-uncased	SVM	0.9217	0.9214	0.9343	0.9242
	MNB	0.9228	0.9223	0.9287	0.9361
	LR	0.9234	**0.9246**	0.9371	**0.9376**
	DT	0.9184	0.9182	0.9125	0.9094
	KNN	0.8977	0.8956	0.9016	0.9371
	RF	0.9222	0.9220	0.9131	0.9151
dbmdz/bert-base-turkish-128k-cased	SVM	0.9483	0.9483	0.9421	0.9420
	MNB	0.9411	0.9411	0.9331	0.9303
	LR	0.9498	0.9494	0.9202	0.9324
	DT	0.9261	0.9261	0.9468	**0.9460**
	KNN	0.9355	0.9355	0.9276	0.9173
	RF	0.9491	**0.9492**	0.9144	0.9144
Dbmdz/distilbert-base-Turkish-case)	SVM	0.9483	0.9483	0.9460	**0.9478**
	MNB	0.9422	**0.9488**	0.9344	0.9343
	LR	0.9394	0.9293	0.9370	0.9367
	DT	0.9375	0.9374	0.9205	0.9196
	KNN	0.9001	0.9000	0.9308	0.9205
	RF	0.9376	0.9476	0.9061	0.9060

The collective findings regarding BERT combined with ML are as follows:

- Upon examination of BERT integrated with ML, a significant majority of the results surpass the 90% across evaluation metrics. Furthermore, these performance metrics notably exceed those observed with Traditional ML techniques. This unequivocally demonstrates the considerable impact of BERT when employed alongside ML and shows performance improvement when BERT is integrated with ML.
- The highest overall performance on the English dataset was achieved by bert-base multilingual integrated with MNB, yielding an F-score of 0.9576. Conversely, for the Turkish dataset, the optimal performance was attained with dbmdz/bert-base-turkish-128k-cased, achieving an F-score of 0.9492 when used with DT
- Traditional ML classifiers consistently formed superior combinations with various BERT models. Generally, the BERT model itself has performed exceptionally well with all datasets. But for the sake of experimentation, we have integrated the BERT along with traditional ML in order to observe the performance enhancement of ML classifiers. It is observed from the experiments that integration of BERT with ML improves the performance of ML classifiers.

4.6. Effect of PEFT LoRA with BERT

In this phase of experiment, the concentration was on assessing Low-Rank Adaptation, abbreviated as LoRA, a technique for parameter-efficient fine-tuning (PEFT) tailored for Language

Model (LLM) adaptation. The objective was to investigate how LoRA influences BERT models. LoRA optimizes parameters by introducing trainable rank decomposition matrices into each layer of the transformer architecture, thereby reducing the number of trainable parameters for downstream tasks while keeping the pre-trained weights fixed. This reduction in trainable parameters lessens the computational burden and memory requirements on GPUs, resulting in high-quality fine-tuning outcomes. It's important to highlight that while LLMs offer superior performance, their training is computationally intensive, necessitating optimization techniques to enhance efficiency and results.

This experiment examines the impact of LoRA on various pre-trained BERT model variants, specifically evaluating its effect on performance for both English and Turkish languages. LoRA implementation was carried out using the Hugging Face Parameter Efficient Fine-Tuning (PEFT) library, setting LoRA with a rank value of 8 to define the intrinsic rank of trainable weights, an Alpha value of 0.01 serving as the learning rate, and a dropout of 0.1. Additionally, the average run time (fold) were recorded. Tables 4.17 and 4.18 depict the influence of LoRA on LLM (BERT) models for English and Turkish datasets.

Referring to Table 4.17, when assessing the performance on the KPD dataset, the highest result was achieved by bert-base-multilingual-uncased with an F-score of 0.9893. Roberta followed as the second-best performer with an F-score of 0.9883, succeeded by Distilbert-base-uncased with an F-score of 0.9791, Bert-base-uncased with an F-score of 0.9759, and finally, Albert-base with an F-score of 0.9741. Meanwhile, for the FSD dataset, the optimal result was attained by Distilbert-base-uncased with an F-score of 0.9889, with bert-base-multilingual-uncased as the second-best performer with an F-score of 0.9874, Roberta with 0.9855, and Bert-base-uncased with 0.9791. It's noteworthy that all models yielded commendable results. In terms of run time, Albert-base exhibited the shortest run time of 453.2215 with a commendable F-score of 0.9741 on the KPD dataset, while on the FSD dataset, it still maintained the shortest run time of 478.7345 with a notable F-score of 0.9851.0.9851

Table 4.17. Results of Using Pre-trained BERT with LoRa PEFT on English Dataset

	KPD			**FSD**		
BERT MODEL+ LoRA PEFT	ACC	F1-Score	Run time/sec	ACC	F1-Score	Run time/sec
bert-base-multilingual-uncased	0.9647	**0.9893**	773.74	0.9884	0.9874	1,576.65
Bert-base-uncased	0.9751	0.9759	664.05	0.9757	0.9791	1,192.92
Roberta (Roberta-base)	0.9571	0.9883	717.83	0.9550	0.9855	1,367.39
Distilbert-base-uncased	0.9791	0.9791	476.67	0.9862	**0.9889**	645.87
Albert-base	0.9737	0.9741	453.22	0.9801	0.9851	478.73

When examining the performance of the Turkish dataset as illustrated in Table 4.18, considering the CUD dataset, dbmdz/electra-base-turkish emerged as the overall top performer with an F-score of 0.9860. The second-best performer was dbmdz/bert-base-turkish-128k-cased with an F-score of 0.9777, followed by Dbmdz/distilbert-base-Turkish-case with 0.9583 and bert-base-multilingual-uncased with an F-score of 0.9579. For the ATC dataset, the most effective model was bert-base-multilingual-uncased with an F-score of 0.9834, followed by dbmdz/electra-base-turkish as the second-best performer with an F-score of 0.9822, dbmdz/bert-base-turkish-128k-cased with 0.9667, and finally, Dbmdz/distilbert-base-Turkish-case with an F-score of 0.9654. The lowest average run time observed for the CUD dataset was 408.70 seconds, achieved through Dbmdz/distilbert-base-Turkish-case, while for the ATC dataset, Dbmdz/distilbert-base-Turkish-case also had the lowest average run time of 750.88 seconds.

Table 4.18. Results of Using Pre-trained BERT with LoRa PEFT on Turkish Dataset

	CUD			**ATC**		
BERTMODEL+ LoRA PEFT	ACC	F1-Score	Run time/sec	ACC	F1-Score	Run time/sec
bert-base-multilingual-uncased	0.9580	0.9579	1,177.35	0.9535	**0.9834**	1,404.22
dbmdz/bert-base-turkish-128k-cased	0.9774	0.9777	1,144.32	0.9662	0.9667	1,224.48
dbmdz/electra-base-turkish	0.9663	**0.9860**	664.45	0.9807	0.9822	1,067.97
Dbmdz/distilbert-base-Turkish-case	0.9579	0.9583	408.70	0.9505	0.9654	750.88

The overall findings from the trained BERT models with LoRa PEFT are as follows:

- Bert-base multilingual achieved the highest overall performance with an F-score of 0.9893 on the English dataset, whereas dbmdz/electra-base-turkish exhibited the best performance with an F-score of 0.9860 on the Turkish dataset.
- All variants of Bert demonstrated commendable performance.
- Albert-base and Distilbert-base-uncased showed the shortest average run time while maintaining strong performance on the English dataset. Similarly, Dbmdz/distilbert-base-Turkish-case and dbmdz/electra-base-turkish had the shortest average run time while delivering good results for the Turkish dataset.
- In general, we have calculated the average runtime for each dataset with BERT alone and then BERT with LoRa. The results showed that the parameter optimization technique LoRa reduces the average runtime as compared with BERT alone. So, in term of efficiency, BERT with LoRa is performing comparatively better than BERT models when used alone.

- In terms of performance, when it comes to the smaller datasets, the BERT with LoRa performs slightly poor than the BERT alone but with datasets having large sample set, BERT with LoRa performs better as compared to BERT alone.

4.7. Effect of Using Semi Supervised Learning Method

In this part of experiment, semi supervised ML was explored and compared. Semi-supervised methods (SSM), leverage both labeled and unlabeled data to build robust models. These methods alleviate the challenge of relying solely on labeled data, which can be expensive and time-consuming to acquire. The main goal in this phase was to examine how SSM can enhance cyberbullying detection. Specifically, two widely used SSM approaches: Self-training and Co-training are explored. It's important to note that SSM operates in conjunction with supervised learning methods (SLM). Thus, we integrated six SLMs—SVM, LR, MNB, DT, K-NN, and RF—into the SSM framework with default parameter settings.

Initially, each dataset was split into training and testing sets, with a small fraction (5% of the training set) designated as the labeled set and the rest as the unlabeled set. Subsequently, the labeled set was expanded using SSM techniques. This augmented labeled data served as the training data to extract features and build a classification model. These features were then used to generate vectors for both the training and testing datasets. Next, class labels are assigned to the documents in the testing dataset using the trained classifier. Regular Self-training and Co-training, were set to parameters such as the number of iterations (num_iterations = 3) and selected top pseudo-labels meeting a minimum confidence threshold of 0.9 to add to the labeled set. For Co-training, we considered a viewpoint satisfying the assumptions outlined in (Blum & Mitchell, 1998) by employing BOW together with TF-IDF.

Tables 4.19 and 4.20 present the results of the Semi-Supervised Method on English and Turkish datasets, respectively. In Table 4.19, focusing on the KPD dataset representing the English dataset, Self-training with RF achieved the highest F-score of 0.6992. LR followed as the second-best performer with an F-score of 0.6855, while MNB ranked third with an F-score of 0.6779. DT and SVM obtained F-scores of 0.6295 and 0.5690, respectively. For Co-training, DT performed the best with an F-score of 0.7554, followed by KNN with an F-score of 0.7542. MNB achieved an F-score of 0.7350, ranking third.

For the FSD dataset, using Self-training, SVM yielded the highest F-score of 0.6832, followed by DT with an F-score of 0.6602 and MNB with an F-score of 0.6411. With Co-training, MNB performed the best with an F-score of 0.6955, followed by SVM with an F-score of 0.6933. LR and DT achieved F-scores of 0.6889 and 0.6832, respectively. We observed a slight improvement in performance accuracy when using the Co-training method.

Table 4.19. Results of Using Traditional Semi Supervised Methods on English Datasets

	KPD				FSD			
SSL	Self-Training		Co-Training		Self-Training		Co-Training	
	ACC	F1-Score	ACC	F1-Score	ACC	F1-Score	ACC	F1-Score
SVM	0.5685	0.5690	0.7748	0.7325	0.6728	**0.6832**	0.6843	0.6933
MNB	0.6660	0.6779	0.7343	0.7350	0.6383	0.6411	0.6989	**0.6955**
LR	0.6271	0.6855	0.7389	0.7423	0.6401	0.6401	0.6899	0.6889
DT	0.5138	0.6295	0.7599	**0.7554**	0.6583	0.6602	0.6754	0.6832
KNN	0.6430	0.6485	0.7632	0.7542	0.6328	0.6322	0.6609	0.6622
RF	0.6985	**0.6992**	0.7632	0.7346	0.6488	0.6355	0.6515	0.6555

When examining the performance of the Turkish dataset, as illustrated in Table 4.20, concerning the CUD dataset, LR exhibited the highest overall performance with an F-score of 0.7588 using the self-training method. The second-best performer was DT with an F-score of 0.6907, followed by RF with an F-score of 0.6800. In the case of utilizing the co-training method, MNB emerged as the top-performing model with an F-score of **0.7577**, trailed by LR as the second-best performer with an F-score of 0.7524, and DT with an F-score of 0.7443, and SVM with an F-score of 0.7255. As for the ATC dataset, SVM demonstrated the highest performance with an F-score of 0.7714, followed by MNB with an F-score of 0.7655, and LR with an F-score of 0.7635.

Table 4.20. Results of Using Traditional Semi Supervised Methods on Turkish Datasets

SSL	CUD				ATC			
	Self-Training		Co-Training		Self-Training		Co-Training	
	ACC	F1-Score	ACC	F1-Score	ACC	F1-Score	ACC	F1-Score
SVM	0.6583	0.6523	0.7252	0.7255	0.6520	**0.6532**	0.7834	**0.7714**
MNB	0.6683	0.6682	0.7557	**0.7577**	0.6375	0.6280	0.7674	0.7655
LR	0.7547	**0.7588**	0.7524	0.7524	0.6449	0.6429	0.7548	0.7635
DT	0.7078	0.6907	0.7442	0.7443	0.6994	0.6394	0.7432	0.7540
KNN	0.6676	0.6570	0.7143	0.7145	0.6290	0.6352	0.6908	0.6986
RF	0.6980	0.6800	0.7023	0.7115	0.6365	0.6483	0.7515	0.7447

When assessing the overall effectiveness of semi-supervised learning methods, for the English dataset, the optimal performance was observed in the KPD dataset, where DT achieved an F-score of 0.7554 with Co-training, while Self-training yielded the highest F-score of 0.6992 with RF. Conversely, for the Turkish dataset, the most proficient algorithms in this category originated from the ATC dataset, with SVM producing the best result of an F-score of 0.7714 using the co-training method.

To further enhance the performance of semi-supervised learning methods, we incorporated self-training SSL with LLM (BERT), an innovative self-training BERT approach. BERT variants,

including Roberta base and Distilbert uncased dedicated for the English language, and Bert base multilingual and Distilbert base for the Turkish language, were selected alongside SVM, MNB, and LR with default parameter settings due to their previous success in experiments. Initially, each dataset was divided into training and testing sets, with a small portion (5% of the training set) serving as the labeled set and the rest as the unlabeled set. BERT generated embeddings features for ML and employed pseudo-labeling to iteratively expand the training with confident predictions from unlabeled data. This process continued for 3 iterations (num_iterations = 3), and subsequently, the labeled set was expanded using the self-training method. Performance metrics such as accuracy and F-score were evaluated using 5-fold cross-validation, and average run times per fold were recorded. Tables 4.21 and 4.22 present the results obtained from Self-Training SSL with BERT on English and Turkish datasets, respectively.

Table 4.21. Results of Self-Training Semi Supervised Learning with BERT on English Datasets

SSL		**KPD**			**FSD**		
BERT VARIANTS		ACC	F1-Score	Runtime/sec	ACC	F1-Score	Runtime/sec
Roberta (Roberta-base)	SVM	0.8767	**0.8767**	1,511.5992	0.8409	0.8333	1,900.5592
	MNB	0.8582	0.8589	1,566.5412	0.8388	0.8314	1,895.3104
	LR	0.8525	0.8625	1,485.8822	0.8574	**0.8598**	1,824.7378
Distilbert (base-uncased)	SVM	0.8810	0.8922	875.9846	0.8612	**0.8624**	1,308.9056
	MNB	0.9000	**0.9012**	827.7638	0.8575	0.8575	1,297.1172
	LR	0.8728	0.8728	813.3312	0.8555	0.8536	1,013.7842

When analyzing the outcomes presented in Table 4.21, focusing on the KPD dataset, the most superior performance is achieved using Distilbert in conjunction with MNB, attaining an F-score of 0.9012, while Roberta paired with SVM achieves an F-score of 0.8767. As for the FSD dataset, the optimum performance is attained with Distilbert combined with SVM, resulting in an F-score of 0.8624, followed by Roberta integrated with LR, yielding an F-score of 0.8598.

Regarding the findings outlined in Table 4.22, derived from Turkish datasets, when considering the CUD dataset, the highest performance is observed with Distilbert paired with LR, achieving an F-score of 0.8894, whereas bert-base-multilingual-uncased combined with MNB emerges as the second-best model with an F-score of 0.8790. For the ATC dataset, the best performance is obtained from Dbmdz/distilbert-base-Turkish- when utilized with LR, resulting in an F-score of 0.8875. The bert-base-multilingual-uncased model secures the second position when employed with SVM, yielding an F-score of 0.8855.

Table 4.22. Results of Self-Training Semi Supervised Learning with BERT on Turkish Datasets

SSL		CUD			ATC		
BERT VARIANTS		ACC	F1-Score	Runtime/sec	ACC	F1-Score	Runtime/sec
bert-base-multilingual-uncased	SVM	0.8574	0.8676	1,711.7476	0.8786	**0.8855**	9340.319
	MNB	0.8589	**0.8790**	1,676.5226	0.8877	0.8880	9310.819
	LR	0.8626	0.8626	1,652.7104	0.8708	0.8715	9310.819
Dbmdz/distilbert-base-	SVM	0.8716	0.8716	1,074.4252	0.8823	0.8835	7277.066
	MNB	0.8869	0.8869	1,096.4252	0.8784	0.8787	7336.608
	LR	0.8745	**0.8894**	1,113.4464	0.8872	**0.8875**	7336.608

The overall outcomes of Self-Training SSL with BERT reveal the following insights:

- In the English dataset, the most notable performance is achieved by integrating Distilbert (base-uncased) with MNB, resulting in an F-score of 0.9012. Conversely, for the Turkish dataset, the optimal performance is observed when utilizing Dbmdz/distilbert-base-Turkish with LR, yielding an F-score of 0.8875.
- There is a general enhancement in performance from traditional SSL methods to the innovative self-training BERT approach. This improvement could be attributed to the integration of BERT models, given that BERT is language-specific and capable of extracting high-quality features, thereby enhancing overall performance.

5. CONCLUSIONS

In today's interconnected world, where social media platforms like Facebook, Twitter, and Instagram have become integral parts of our daily lives, the dynamics of human interaction have undergone a profound transformation. While these platforms have revolutionized communication and fostered vibrant digital communities, they have also introduced new challenges, biggest among them being the insidious phenomenon of cyberbullying. Cyberbullying, characterized by the malicious use of digital communication tools to harass or intimidate innocent people, represents a significant societal concern with far-reaching implications.

Cyberbullying knows no bounds of age or demographic, affecting individuals across the spectrum, from young children navigating their first steps on the internet to adults. Its impact extends beyond the confines of the virtual world, seeping into the fabric of everyday life and exacting a toll on mental and emotional well-being. The consequences can be devastating, leading to feelings of sadness, anxiety, isolation, and in extreme cases, culminating in tragic outcomes such as suicide.

One of the challenges inherent in combating cyberbullying lies in its elusive nature it often evades detection until irreparable harm has been inflicted upon its victims and in most instances, the content keeps re-surfacing even after a long time. Moreover, the rapid dissemination of harmful content across digital channels exacerbates the challenge, making timely intervention all the more critical. Recognizing the gravity of this issue, there is an urgent need for robust and efficient methods of cyberbullying detection to safeguard the well-being of individuals in online spaces.

In the domain of academic research, efforts to address cyberbullying detection have primarily focused on the English language, relegating other linguistic domains, such as Turkish, to the brink. Since, Turkish is an agglutinative language where by its morphological structure differs from non-agglutinative languages like English. This thesis aims to detect cyberbullying in both English and Turkish languages, highlighting their differences, as these languages possess distinct morphological structures. The objective is to assess the impact of standard state-of-the-art supervised learning techniques on both languages.

This thesis also answers the question of how unlabeled data can be utilized effectively since it’s a common problem in text classification where there’s more unlabeled data compared to labeled data and annotating labeled data is very expensive and time-consuming process that demands expertise of the specific language domain. To answer this question. We carried out extensive experiments using semi-supervised methods.

We carried a comprehensive study on cyberbullying detection for both English and Turkish language by employing sophisticated ML classifiers, including Support Vector Machines (SVM), Logistic Regression (LR), Multinomial Naive Bayes (MNB), Decision Trees (DT), Random Forests (RF), K-Nearest Neighbors (K-NN), alongside cutting-edge deep learning paradigms, such

as Convolutional Neural Networks (CNN) and Recurrent Neural Network (RNN) variants—Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Bidirectional LSTM (BiLSTM), Bidirectional GRU (BiGRU), in tandem with hybrid architectures like CNN-LSTM and LSTM-CNN. These sophisticated models are rigorously evaluated across benchmark datasets, leveraging diverse traditional feature weighting methodologies (Binary, Term Frequency (TF), Term Frequency-Inverse Document Frequency (TF-IDF), Bag-of-Words (BOW), N-gram), and state-of-the-art word embedding techniques (Word2Vec, FastText, GloVe), encompassing both pre-trained and custom-trained embeddings. Furthermore, large LLM's such as BERT and its variants are meticulously examined for their efficacy in cyberbullying detection. In order to improve on performance of ML, BERT was also combined with ML algorithms. In addition to model evaluation, an optimization technique (LoRA), that leverages LLM specifically BERT was utilized thereby enhancing both accuracy and computational efficiency.

Another important issue of text classification, where there's often much more unlabeled data than labeled data, and labeling data is costly and time-consuming. By using lots of unlabeled data with a small amount of labeled data, we can reduce the effort needed for labeling. To judge the impact of these scenarios, we implied Semi supervised learning specifically Self-Training and Co-Training. Furthermore, we also developed and proposed an innovative approach called Self-Training with BERT which improves on performance and also cuts down on the time and effort required for data labeling.

Study findings show that DL methods usually work better than traditional ML methods, but we need to carefully adjust parameters for DL to find the best setup. We also found that combining different DL methods, like CNN-LSTM or LSTM-CNN, improves classification accuracy. Large language model BERT performs better than other DL models because of its excellent understanding of the language context as these pre-trained models are learned from a variety of corpora that includes knowledge about different domains thus producing superior word representations according to the context. Combining BERT with machine learning also improves performance. Using LoRA, a parameter optimization technique, makes large language models faster and cheaper to train. In the context of the study, the experimental results also reveal that pre-trained word vectors, outperform custom-trained ones.

To evaluate the performance of popular word-embedding techniques used in this study, the FastText outperforms other embedding methods like Word2vec and GloVe in both languages including Turkish reason being the FastText is well-suited for languages with intricate word structures, like Turkish. The experiment results further demonstrate that DL most specifically with FastText word embeddings achieves good performance on both English and Turkish datasets. However, pre-trained versions of BERT, a powerful language model, delivered exceptional results. BERT achieved an average F-score of 0.9775 on Turkish data and 0.9892 on English data. The proposed method, BERT with self-Training, also yielded impressive results when integrated with

MNB on the English dataset using DistilBERT (base-uncased), it achieved an F-score of 0.9012. Similarly, on the Turkish dataset, it obtained an F-score of 0.8875 when used with Dbmdz/distilbert-base-Turkish and LR.

We hope this thesis will serve as a guide that will provide researchers in the field of Text mining, NLP with important information for designing their models, survey, in order to develop a more effective and reliable studies. In the future, we plan to evaluate the performance of the proposed models with other languages like Arabic, Spanish, African, Urdu etc. We also plan to explore feature selection techniques to improve on the models.

REFERENCES

Abadi, M., Jianmin Chen, Zhifeng Chen, A. D., Jeffrey Dean, M. D., & Michael Isard, M. K. (2016). TensorFlow: A system for large-scale machine learning. *2017 IEEE Radar Conference, RadarConf 2017*, 1222–1227. https://doi.org/10.1109/RADAR.2017.7944391

Abbaspour, S., Fotouhi, F., Sedaghatbaf, A., Fotouhi, H., Vahabi, M., & Linden, M. (2020). A comparative analysis of hybrid deep learning models for human activity recognition. *Sensors (Switzerland)*, *20*(19), 1–14. https://doi.org/10.3390/s20195707

Aboujaoude, E., Savage, M. W., Starcevic, V., & Salame, W. O. (2015). Cyberbullying: Review of an old problem gone viral. *Journal of Adolescent Health*, *57*(1), 10–18. https://doi.org/10.1016/j.jadohealth.2015.04.011

Agarwal, A., Chivukula, A. S., Bhuyan, M. H., Jan, T., Narayan, B., & Prasad, M. (2020). Identification and Classification of Cyberbullying Posts: A Recurrent Neural Network Approach Using Under-Sampling and Class Weighting. *Communications in Computer and Information Science*, *1333*, 113–120. https://doi.org/10.1007/978-3-030-63823-8_14

Agatston, P. W., Kowalski, R., & Limber, S. (2007). Students' Perspectives on Cyber Bullying. *Journal of Adolescent Health*, *41*(6 SUPPL.), 59–60. https://doi.org/10.1016/j.jadohealth.2007.09.003

Al-garadi, M. A., Varathan, K. D., & Ravana, S. D. (2016). Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter network. *Computers in Human Behavior*, *63*, 433–443. https://doi.org/10.1016/j.chb.2016.05.051

Alam, K. S., Bhowmik, S., & Prosun, P. R. K. (2021). Cyberbullying detection: An ensemble based machine learning approach. *Proceedings of the 3rd International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, ICICV 2021*, *Icicv*, 710–715. https://doi.org/10.1109/ICICV50876.2021.9388499

Aldhyani, T. H. H., & Alkahtani, H. (2021). A bidirectional long short-term memory model algorithm for predicting covid-19 in gulf countries. *Life*, *11*(11), 1–26. https://doi.org/10.3390/life11111118

Alotaibi, M. (2021). *A Multichannel Deep Learning Framework for Cyberbullying Detection on Social Media*.

Alpaydın, E. (2014). Introduction to Machine Learning. In *Machine Learning and Non-volatile Memories* (Second Edi). The MIT Press. https://doi.org/10.1007/978-3-031-03841-9_1

Anonymous 1: https://www.statista.com/statistics/617136/digital-population-worldwide. [Accessed: December 2023]

Anonymous 2: https://www.pewresearch.org/internet/fact-sheet/social-media/ [Accessed: December 2023]

Anonymous 3: https://www.pewresearch.org/internet/fact-sheet/social-media/[Accessed: December 2023]

Anonymous 4: https://extrainningsoftball.com/topical-issue-all-the-latest-cyber-bullying-statistics-and-what-they-mean-in-2022/[Accessed: December 2023]

*Anonymous 5:*https://www.comparitech.com/internet-providers/cyberbullying-statistics/ [Accessed: December 2023]

Anonymous 6: https://cyberbullying.org/summary-of-our-cyberbullying-research [Accessed: December 2023]

Anonymous 7: https://www.nzherald.co.nz/nz/teenage-bullies-hound-12-year-old-to death/5HKHTINXQKQLKM7HBORIDWIVCQ/ [Accessed:December 2023]%0A

Anonymous 8: https:// www.internetsafetystatistics.com/ryan-halligan-loses-his-life-to-taunts-rumors-and-cyberbullying/ [Accessed: December 2023]

Anonymous 9: https://abcnews.go.com/Health/jamey-rodemeyer-suicide-ny-police-open-criminal-investigation/story?id=14580832 [Accessed: December 2023]

Anonymous 10: https://www.hurriyetdailynews.com/turkish-cypriot-kills-self-after-being-cyber-bullied---11565[Accessed: December 2023]

Anonymous 11: https://www.abc.net.au/news/2018-01-11/online-bullying-suicide-sparks-national-discussion/9321226 [Accessed: December 2023]

Anonymous 12: https://stopcyberbullyingday.org/about-the-day/ [Accessed: January 2023]

Anonymous 13: https://www.kivaprogram.net/ [Accessed: Dec. 2023]

Anonymous 14: https://www.findlaw.com/education/student-conduct-and-discipline/cyberbullying-and-social-media.html [Accessed:Dec. 2023]

*Anonymous 15: https://telef*onicatech.com/en/blog/semi-supervised-learning-the-great-unknown [Accessed on Dec.2023]

Anonymous 16: https://huggingface.co/dbmdz/bert-base-turkish-128k-cased[Accessed:December2023]

Anonymous 17: https://huggingface.co/dbmdz/electra-base-turkish-cased-discriminator[Accessed:December2023]

Ashish, V., Noam, S., Niki, P., Jakob, U., Jones, G., & Łukasz, K. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, *proceedings.neurips.cc*, 4752–4758. https://doi.org/10.1145/3583780.3615497

Bai, X., Yang, Y., Wei, S., Chen, G., Li, H., Li, Y., & Tian, H. (2023). *A Comprehensive Review of Conventional and Deep Learning Approaches for Ground-Penetrating Radar Detection of Raw Data*.

Balakrishnan, V., Khan, S., & Arabnia, H. R. (2020). Improving cyberbullying detection using

Twitter users' psychological features and machine learning. *Computers and Security*, *90*, 101710. https://doi.org/10.1016/j.cose.2019.101710

Balcan, M. F., & Blum, A. (2010). A discriminative model for semi-supervised learning. *Journal of the ACM*, *57*(3). https://doi.org/10.1145/1706591.1706599

Banerjee, V., Telavane, J., Gaikwad, P., & Vartak, P. (2020). Detection of Cyberbullying Using Deep Neural Network. *2020 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019*, 604–607. https://doi.org/10.1109/ICACCS.2019.8728378

Beale, A. V., & Hall, K. R. (2007). Cyberbullying: What School Administrators (and Parents) Can Do. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, *81*(1), 8–12. https://doi.org/10.3200/tchs.81.1.8-12

Bengio, Y., Ducharme, R., & Vincent, P. (2003). A neural probabilistic language model (short version). *Advances in Neural Information Processing Systems*. https://proceedings.neurips.cc/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf

Bengio, Y., Simard, P., Frasconi, P., & Member, S. (1994). Learning Long-Term Dependencies with Gradient Descent is Difficult. *Eeexplore*, *5*(2).

Beran, T., & Li, Q. (2008). The Relationship between Cyberbullying and School Bullying. *The Journal of Student Wellbeing*, *1*(2), 16. https://doi.org/10.21913/jsw.v1i2.172

Bharti, S., Yadav, A. K., Kumar, M., & Yadav, D. (2022). Cyberbullying detection from tweets using deep learning. *Kybernetes*, *51*(9), 2695–2711. https://doi.org/10.1108/K-01-2021-0061

Bhat, C. S. (2008). Cyber bullying: Overview and strategies for school counsellors, guidance officers, and all school personnel. *Australian Journal of Guidance and Counselling*, *18*(1), 53–66. https://doi.org/10.1375/ajgc.18.1.53

Bird, S. (2006). NLTK: The natural language toolkit. *COLING/ACL 2006 - 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Interactive Presentation Sessions*, 69–72.

Bisong, E. (2019). Training a Neural Network. In *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. https://doi.org/10.1007/978-1-4842-4470-8_29

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the Annual ACM Conference on Computational Learning Theory*, 92–100. https://doi.org/10.1145/279943.279962

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146. https://doi.org/10.1162/tacl_a_00051

Bozyigit, A., Utku, S., & Nasiboglu, E. (2019). Cyberbullying Detection by Using Artificial Neural Network Models. *UBMK 2019 - Proceedings, 4th International Conference on Computer Science and Engineering*, 520–524. https://doi.org/10.1109/UBMK.2019.8907118

Bozyiğit, A., Utku, S., & Nasibov, E. (2021). Cyberbullying detection: Utilizing social media features. *Expert Systems with Applications*, *179*(April). https://doi.org/10.1016/j.eswa.2021.115001

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1007/978-3-030-62008-0_35

Cassidy, W., Brown, K., & Jackson, M. (2012). "Under the radar": Educators and cyberbullying in schools. *School Psychology International*, *33*(5), 520–532. https://doi.org/10.1177/0143034312445245

Çelebi, R. (2016). Information Sciences and Systems 2014. *Information Sciences and Systems 2016, Iscis*. https://doi.org/10.1007/978-3-319-09465-6

Çelik, Ö., & Koç, B. C. (2021). TF-IDF, Word2vec ve Fasttext Vektör Model Yöntemleri ile Türkçe Haber Metinlerinin Sınıflandırılması. *Deu Muhendislik Fakultesi Fen ve Muhendislik*, *23*(67), 121–127. https://doi.org/10.21205/deufmd.2021236710

Chawla, I. (2002). Deep Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research 16 (2002) 321–357*, *16*, 321–357. http://arxiv.org/abs/2003.09788

Chen, H., McKeever, S., & Delany, S. J. (2017). Harnessing the power of text mining for the detection of abusive content in social media. *Advances in Intelligent Systems and Computing*, *513*(September), 187–205. https://doi.org/10.1007/978-3-319-46562-3_12

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1724–1734. https://doi.org/10.3115/v1/d14-1179

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 1–9. http://arxiv.org/abs/1412.3555

Cihan Ates, E., Bostanci, E., & Güzel, S. (2021). Comparative Performance of Machine Learning Algorithms in Cyberbullying Detection: Using Turkish Language Preprocessing Techniques. *Computers and Society*.

Cliche, M. (2017). BB twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs. *Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2014*, 573–580. https://doi.org/10.18653/v1/s17-2094

Çoban, Ö., & Özel, S. A. (2018). Utilizing Language Model for Term Weighting in Text Categorization. *Nternational Conference on Artificial Intelligence and Data Processing*

(IDAP). IEEE.

Coban, O., Ozel, S. A., & Inan, A. (2023). *Detection and cross-domain evaluation of cyberbullying in Facebook activity contents for Turkish*. 1–32. https://doi.org/10.1145/3580393

Çoban, Ö., Özel, S. A., & Inan, A. (2021). Deep Learning-based Sentiment Analysis of Facebook Data: The Case of Turkish Users. *Computer Journal*, *64*(3), 473–499. https://doi.org/10.1093/comjnl/bxaa172

Collobert, A. (2009). General Deep architecture for NLP. *ICML*.

Corcoran, L., Guckin, C. M., & Prentice, G. (2015). Cyberbullying or cyber aggression?: A review of existing definitions of cyber-based peer-to-peer aggression. *Societies*, *5*(2), 245–255. https://doi.org/10.3390/soc5020245

Cutler, A., & Cutler, D. R. (2012). Random Forests. *Ensemble Machine Learning*, *January*. https://doi.org/10.1007/978-1-4419-9326-7

Dadvar, M., Trieschnigg, D., & De Jong, F. (2013). Expert knowledge for automatic detection of bullies in social networks. *Belgian/Netherlands Artificial Intelligence Conference*, 57–63.

David-Ferdon, C., & Hertz, M. F. (2007). Electronic Media, Violence, and Adolescents: An Emerging Public Health Problem. *Journal of Adolescent Health*, *41*(6 SUPPL.), 1–5. https://doi.org/10.1016/j.jadohealth.2007.08.020

Dehue, F., Bolman, C., & Völlink, T. (2008). Cyberbullying: Youngsters' experiences and parental perception. *Cyberpsychology and Behavior*, *11*(2), 217–223. https://doi.org/10.1089/cpb.2007.0008

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*(Mlm), 4171–4186.

Dharma, E. M., Gaol, F. L., Warnars, H. L. H. S., & Soewito, B. (2022). the Accuracy Comparison Among Word2Vec, Glove, and Fasttext Towards Convolution Neural Network (Cnn) Text Classification. *Journal of Theoretical and Applied Information Technology*, *100*(2), 349–359.

Dinakar, K., Reichart, R., & Lieberman, H. (2011). Modeling the detection of textual cyberbullying. *Proceedings of the International AAAI Conference on Web and Social Media*, *WS-11-02*, 11–17.

DistilBERTurk. (n.d.). dbmdz/distilbert-base-turkish-cased. *Available Online*. https://huggingface.co/dbmdz/distilbert-base-turkish-cased

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87. https://doi.org/10.1145/2347736.2347755

Dracic, S. (2009). *Bullying And Peer Victimization*. *21*(4), 216–219.

Edosomwan, S. (2011). SocialMedia-JAME. *The Journal of Applied Management and Entrepreneurship*, *16*(3).

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211. https://doi.org/10.1016/0364-0213(90)90002-E

Elsafoury, F. (2020). *Cyberbullying datasets*. https://doi.org/10.17632/jf4pzyvnpj.1

Emmery, C., Verhoeven, B., De Pauw, G., Jacobs, G., Van Hee, C., Lefever, E., Desmet, B., Hoste, V., & Daelemans, W. (2021). Current limitations in cyberbullying detection: On evaluation criteria, reproducibility, and data scarcity. *Language Resources and Evaluation*, *55*(3), 597–633. https://doi.org/10.1007/s10579-020-09509-1

Farha, I. A., & Magdy, W. (2019). Mazajak: An online arabic sentiment analyser. *ACL 2019 - 4th Arabic Natural Language Processing Workshop, WANLP 2019 - Proceedings of the Workshop*, 192–198. https://doi.org/10.18653/v1/w19-4621

Fatima, M., & Pasha, M. (2017). Survey of Machine Learning Algorithms for Disease Diagnostic. *Journal of Intelligent Learning Systems and Applications*, *09*(01), 1–16. https://doi.org/10.4236/jilsa.2017.91001

Fetchenhauer, D., & Belschak, F. D. (2009). *Cyberbullying : Who Are the Victims ? A Comparison of Victimization in Internet Chatrooms and Victimization in School*. *November 2015*. https://doi.org/10.1027/1864-1105.21.1.25

François Chollet. (2017). Deep Learning with Python & Keras. *Manning Publications*, *80*(1), 453. http://www.ncbi.nlm.nih.gov/pubmed/20608803

Garbin, C., Zhu, X., & Marques, O. (2020). Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia Tools and Applications*, *79*(19–20), 12777–12815. https://doi.org/10.1007/s11042-019-08453-9

Goldberg, A. (2010). New Directions in Semi-Supervised Learning. *Learning*. http://pages.cs.wisc.edu/~jerryzhu/pub/andrew-thesis.pdf

Guven, Z. A. (2022). The Comparison of Language Models with a Novel Text Filtering Approach for Turkish Sentiment Analysis. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(2). https://doi.org/10.1145/3557892

Haber, C. (2016). Ending the torment: tackling bullying from the schoolyard to cyberspace. In *Violence against children* (Issue October). https://sustainabledevelopment.un.org/content/documents/2577tackling_bullying_from_schoolyard_to_cyberspace_low_res_fa.pdf#page=136

Haffari, G. (2006). *A Survey on Inductive Semi-supervised Learning*. 1–51.

Hamdan, H., Bellot, P., & Bechet, F. (2016). Supervised Term Weighting Metrics for Sentiment Analysis in Short Text. *ArXiv Preprint ArXiv:1610.03106.*

Han, J. K., & Kamber, M. et al. (2001). *Data Mining: Concepts and Techniques.* (Morgan-

Kaufman & S. of D. M. Systems (Eds.)). San Diego: Academic Press.

Han, W., Cai, Z., & Chen, X. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. *Communications in Computer and Information Science*, *1944 CCIS*, 151–160. https://doi.org/10.1007/978-981-99-7743-7_9

Hani, J., Nashaat, M., Ahmed, M., Emad, Z., Amer, E., & Mohammed, A. (2019). Social media cyberbullying detection using machine learning. *International Journal of Advanced Computer Science and Applications*, *10*(5), 703–707. https://doi.org/10.14569/ijacsa.2019.0100587

Harris, C. R., Millman, K. J., & van der Walt, S. J. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hart, P. (1968). The Condensed Neighbour Rule (corresp.). *IEEE Transactions on Information Theory*, *14*(3), 515–516.

Hasan, M. T., Hossain, M. A. E., Mukta, M. S. H., Akter, A., Ahmed, M., & Islam, S. (2023). A Review on Deep-Learning-Based Cyberbullying Detection. *Future Internet*, *15*(5), 1–47. https://doi.org/10.3390/fi15050179

Hay, C., & Meldrum, R. (2010). Bullying victimization and adolescent self-harm: Testing hypotheses from general strain theory. *Journal of Youth and Adolescence*, *39*(5), 446–459. https://doi.org/10.1007/s10964-009-9502-0

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *Proceedings of the International Joint Conference on Neural Networks*, *3*, 1322–1328. https://doi.org/10.1109/IJCNN.2008.4633969

Hinduja, S., & Patchin, J. W. (2007). Offline consequences of online victimization: School violence and delinquency. *Journal of School Violence*, *6*(3), 89–112. https://doi.org/10.1300/J202v06n03_06

Hinduja, S., & Patchin, J. W. (2010). Bullying, cyberbullying, and suicide. *Archives of Suicide Research*, *14*(3), 206–221. https://doi.org/10.1080/13811118.2010.494133

Hmeidi, I., Al-Ayyoub, M., Abdulla, N. A., Almodawar, A. A., Abooraig, R., & Mahyoub, N. A. (2015). Automatic Arabic text categorization: A comprehensive comparative study. *Journal of Information Science*, *41*(1), 114–124. https://doi.org/10.1177/0165551514558172

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Holmström, L., & Koistinen, P. (2010). Pattern recognition. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(4), 404–413. https://doi.org/10.1002/wics.99

Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). Lora: Low-Rank Adaptation of Large Language Models. *ICLR 2022 - 10th International Conference on Learning Representations*, 1–26.

Ian, G., Yoshua, B., & Aaron, C. (2016). Deep learning. In *Adapt. Comput*. https://doi.org/10.1007/978-981-13-9113-2_16

Islam, M. M., Uddin, M. A., Islam, L., Akter, A., Sharmin, S., & Acharjee, U. K. (2020). Cyberbullying Detection on Social Networks Using Machine Learning Approaches. *2020 IEEE Asia-Pacific Conference on Computer Science and Data Engineering, CSDE 2020*, 6–11. https://doi.org/10.1109/CSDE50874.2020.9411601

Iwendi, C., Srivastava, G., Khan, S., & Maddikunta, P. K. R. (2020). Cyberbullying detection solutions based on deep learning architectures. *Multimedia Systems*, *29*(3), 1839–1852. https://doi.org/10.1007/s00530-020-00701-5

Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2016). Reading Text in the Wild with Convolutional Neural Networks. *International Journal of Computer Vision*, *116*(1), 1–20. https://doi.org/10.1007/s11263-015-0823-z

Jain, V., Kumar, V., Pal, V., & Vishwakarma, D. K. (2021). Detection of Cyberbullying on Social Media Using Machine learning. *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, *Iccmc*, 1091–1096. https://doi.org/10.1109/ICCMC51019.2021.9418254

Jeffrey Pennington, Richard Socher, C. D. M. (2014). GloVe: Global Vectors for Word Representation. *AES: Journal of the Audio Engineering Society*.

Jurafsky, D. (2019). Speech and Language Processing: An introduction to natural language processing,. *SPEECH and LANGUAGE PROCESSING An Introduction to Natural Language Processing Computational Linguistics and Speech Recognition*, *May*, 1–18. http://www.cs.colorado.edu/~martin/slp.html

Juvonen, J., & Graham, S. (2014). Bullying in schools: The power of bullies and the plight of victims. *Annual Review of Psychology*, *65*, 159–185. https://doi.org/10.1146/annurev-psych-010213-115030

Kanaris, I., Kanaris, K., Houvardas, I., & Stamatatos, E. (2007). *Words vs. character n-grams for anti-spam filtering*. *XX*(X), 1–20.

Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! The challenges and opportunities of Social Media. *Business Horizons*, *53*(1), 59–68. https://doi.org/10.1016/j.bushor.2009.09.003

Karayiğit, H., İnan Acı, Ç., & Akdağlı, A. (2021). Detecting abusive Instagram comments in Turkish using convolutional Neural network and machine learning methods. *Expert Systems with Applications*, *174*(January). https://doi.org/10.1016/j.eswa.2021.114802

Karpathy, A., Johnson, J., & Fei-Fei, L. (2015). *Visualizing and Understanding Recurrent Networks*. 1–12. http://arxiv.org/abs/1506.02078

Kasneci, E., & Sessler, K. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences*, *103*(February).

https://doi.org/10.1016/j.lindif.2023.102274

Kaur, G. (2014). Usage of Regular Expressions in Nlp. *International Journal of Research in Engineering and Technology*, *03*(01), 168–174. https://doi.org/10.15623/ijret.2014.0301026

Khan, A. (2007). A Review of Machine Learning Algorithms for Text-Documents Classification. *European Conference on Machine Learning*, *13*(2).

Khoshgoftaar, T. M., Golawala, M., & Van Hulse, J. (2007). An empirical study of learning from imbalanced data using random forest. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, *2*, 310–317. https://doi.org/10.1109/ICTAI.2007.46

Kibriya, A. M., Frank, E., Pfahringer, B., & Holmes, G. (2005). Multinomial naive bayes for text categorization revisited. *Australian Joint Conference*, *3339*, 488–499. https://doi.org/10.1007/978-3-540-30549-1_43

Kilimci, Z. H., & Akyokus, S. (2019). The Evaluation of Word Embedding Models and Deep Learning Algorithms for Turkish Text Classification. *UBMK 2019 - Proceedings, 4th International Conference on Computer Science and Engineering*, 548–553. https://doi.org/10.1109/UBMK.2019.8907027

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference* (pp. 1746–1751). https://doi.org/10.3115/v1/d14-1181

Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.

Kiritchenko, S., & Matwin, S. (2001). Email Classification with Co-Training. *Proceedings of the 2001 Conference of the Centre for Advanced Studies on Collaborative Research, April 2002*, 301–312. http://portal.acm.org/citation.cfm?id=782096.782104

Kistner, J. A., David-Ferdon, C. F., Repper, K. K., & Joiner, T. E. (2006). Bias and accuracy of children's perceptions of peer acceptance: Prospective associations with depressive symptoms. *Journal of Abnormal Child Psychology*, *34*(3), 349–361. https://doi.org/10.1007/s10802-006-9030-2

Ko, B., Kim, H. G., Oh, K. J., & Choi, H. J. (2017). Controlled dropout: A different approach to using dropout on deep neural network. *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017*, 358–362. https://doi.org/10.1109/BIGCOMP.2017.7881693

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *IJCAI International Joint Conference on Artificial Intelligence*, *2*(June), 1137–1143.

Koksal, O., & Akgul, O. (2022). A Comparative Text Classification Study with Deep Learning-

Based Algorithms. *2022 9th International Conference on Electrical and Electronics Engineering, ICEEE 2022,* 387–391. https://doi.org/10.1109/ICEEE55327.2022.9772587

Kowalski, R. M., & Limber, S. P. (2007). Electronic Bullying Among Middle School Students. *Journal of Adolescent Health*, *41*(6 SUPPL.), 22–30. https://doi.org/10.1016/j.jadohealth.2007.08.017

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information (Switzerland)*, *10*(4), 1–68. https://doi.org/10.3390/info10040150

Krawczyk, B. (2016). Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, *5*(4), 221–232. https://doi.org/10.1007/s13748-016-0094-0

Krutka, D. G., Manca, S., Galvin, S. M., Greenshow, C., Kohler, M. J., & Askari, E. (2020). Teaching "against" social media: Confronting problems of profit in the curriculum. *Teachers College Record*, *121*(14), 1–42. https://doi.org/10.1177/016146811912101410

Kyle, K. (2021). Natural language processing for learner corpus research. *International Journal of Learner Corpus Research*, *7*(1), 1–16. https://doi.org/10.1075/ijlcr.00019.int

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: a Lite Bert for Self-Supervised Learning of Language Representations. *8th International Conference on Learning Representations, ICLR 2020*, 1–17.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2323. https://doi.org/10.1109/5.726791

Lei, J. (2020). Cross-Validation With Confidence. *Journal of the American Statistical Association*, *115*(532), 1978–1997. https://doi.org/10.1080/01621459.2019.1672556

Lemaˆıtre, G., Nogueira, F. N., & Aridas, C. (2017). *Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning Guillaume*. http://arxiv.org/abs/1512.03502

Li, M., & Zhou, Z. (2005). S ETRED : Self-Training with Editing. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

Li, Pengpeng, Luo, A., Liu, J., Wang, Y., Zhu, J., Deng, Y., & Zhang, J. (2020). Bidirectional gated recurrent unit neural network for Chinese address element segmentation. *ISPRS International Journal of Geo-Information*, *9*(11). https://doi.org/10.3390/ijgi9110635

Li, Q. (2006). Cyberbullying in schools: A research of gender differences. *School Psychology International*, *27*(2), 157–170. https://doi.org/10.1177/0143034306064547

Li, Y., Peng, Y., Ji, W., Zhang, Z., & Xu, Q. (2017). User Identification Based on Display Names Across Online Social Networks. *IEEE Access*, *5*, 17342–17353.

https://doi.org/10.1109/ACCESS.2017.2744646

Liciotti, D., Bernardini, M., Romeo, L., & Frontoni, E. (2020). A sequential deep learning application for recognising human activities in smart homes. *Neurocomputing*, *396*, 501–513. https://doi.org/10.1016/j.neucom.2018.10.104

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). *A Critical Review of Recurrent Neural Networks for Sequence Learning*. 1–38. http://arxiv.org/abs/1506.00019

Liu, L., Zuo, W., Han, J., Xu, Y., & Peng, T. (2003). Building text classifiers using positive, unlabeled and 'outdated' examples. *IEEE International Conference on Data Mining*, *28*(13), 3691–3706. https://doi.org/10.1002/cpe.3879

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. *1*. http://arxiv.org/abs/1907.11692

Lobur, M., Romanyuk, A., & Romanyshyn, M. (2011). Using NLTK for educational and scientific purposes. *2011 11th International Conference - The Experience of Designing and Application of CAD Systems in Microelectronics, CADSM 2011*, 426–428.

Lopez, P. (2019). *RNN, LSTM & GRU*. Available Online. http://dprogrammer.org/tag/neural-network

Mahesh, B. (2018). Machine Learning Algorithms - A Review | Enhanced Reader. *International Journal of Science and Research*, *9*(1), 381–386. https://doi.org/10.21275/ART20203995

Mahmud, M. I., Mamun, M., & Abdelgawad, A. (2022). A Deep Analysis of Textual Features Based Cyberbullying Detection Using Machine Learning. *2022 IEEE Global Conference on Artificial Intelligence and Internet of Things, GCAIoT 2022*, 166–170. https://doi.org/10.1109/GCAIoT57150.2022.10019058

Marzieh, M., Farahbakhsh, R., & Crespi, N. (2020). A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. *Studies in Computational Intelligence*, *881 SCI*, 928–940. https://doi.org/10.1007/978-3-030-36687-2_77

Maslej-Krešňáková, V., Sarnovský, M., Butka, P., & Machová, K. (2020). Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification. *Applied Sciences (Switzerland)*, *10*(23), 1–26. https://doi.org/10.3390/app10238631

Mazari, A. C., Boudoukhani, N., & Djeffal, A. (2023). BERT-based ensemble learning for multi-aspect hate speech detection. *Cluster Computing*, *0123456789*. https://doi.org/10.1007/s10586-022-03956-x

McKinney, W. (2011). Pandas: a Foundational Python Library for Data Analysis and Statistics. *Python for High Performance and Scientific Computing*, *14*(9), 1–9. https://doi.org/10.1002/mmce.20381

Meurers, D. (2012). Natural Language Processing and Language Learning. *The Encyclopedia of Applied Linguistics*, 1–15. https://doi.org/10.1002/9781405198431.wbeal0858.pub2

Mfenyana, S. I., Moorosi, N., & Thinyane, M. (2013). *Facebook Crawler Architecture for Opinion Monitoring and Trend Analysis Purposes*. *April*.

Michel, T. M. (1997). Machine Leraning Really Work ? *AI Magazine*, *18*(3), 71–83. http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/1303

Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. 1–12.

Milosevic, T. (2016). Social media companies' cyberbullying policies. *International Journal of Communication*, *10*(2016), 5164–5185.

Min, B., Ross, H., Sulem, E., Veyseh, A. P. Ben, Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., & Roth, D. (2023). Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey. *ACM Computing Surveys*, *56*(2). https://doi.org/10.1145/3605943

Minsky, M. (1961). Steps Toward Artificial Intelligence. *Proceedings of the IRE*, *49*(1), 8–30. https://doi.org/10.1109/JRPROC.1961.287775

Mitchell, T. M. (2006). The Discipline of Machine Learning. *Machine Learning*, *17*(July), 1–7. http://www-cgi.cs.cmu.edu/~tom/pubs/MachineLearningTR.pdf

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, 243–248. https://doi.org/10.1109/ICICS49469.2020.239556

Mossie, Z., & Wang, J. H. (2020). Vulnerable community identification using hate speech detection on social media. In *Information Processing and Management* (Vol. 57, Issue 3). https://doi.org/10.1016/j.ipm.2019.102087

Muneer, A., & Fati, S. M. (2020). A comparative analysis of machine learning techniques for cyberbullying detection on twitter. *Future Internet*, *12*(11), 1–21. https://doi.org/10.3390/fi12110187

Munezero, M., Mozgovoy, M., Kakkonen, T., Klyuev, V., & Sutinen, E. (2013). Antisocial Behavior Corpus for Harmful Language Detection. *2013 Federated Conference on Computer Science and Information Systems*, 261–265.

Murshed, B. A. H., Abawajy, J., Mallappa, S., Saif, M. A. N., & Al-Ariki, H. D. E. (2022). DEA-RNN: A Hybrid Deep Learning Approach for Cyberbullying Detection in Twitter Social Media Platform. *IEEE Access*, *10*, 25857–25871. https://doi.org/10.1109/ACCESS.2022.3153675

Nahar, V., Al-maskari, S., Li, X., & Pang, C. (2014). *Semi-supervised Learning for Cyberbullying*. 160–171.

Najib, A., Ozel, S. A., & Coban, O. (2023). Comparative Analysis of Cyberbullying Detection: A case study for Turkish and English. *2023 Innovations in Intelligent Systems and Applications Conference, ASYU 2023*, 1–6. https://doi.org/10.1109/ASYU58738.2023.10296722

Naqa, I. El, & Murphy, M. J. (2015). Machine Learning in Radiation Oncology. *What Is Machine Learning*, 3–11. https://doi.org/10.1007/978-3-319-18305-3

Narkhede, S. (2018). *Matrix, Understanding Confusion*. [Onlinr]. https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Nartey, O. T., Yang, G., Wu, J., & Asare, S. K. (2020). Semi-Supervised Learning for Fine-Grained Classification with Self-Training. *IEEE Access*, *8*, 2109–2121. https://doi.org/10.1109/ACCESS.2019.2962258

Naseem, U., Razzak, I., Khan, S. K., & Prasad, M. (2021). A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *20*(5). https://doi.org/10.1145/3434237

Nguyen, H. M., Cooper, E. W., & Kamei, K. (2011). Borderline over-sampling for imbalanced data classification. *International Journal of Knowledge Engineering and Soft Data Paradigms*, *3*(1), 4. https://doi.org/10.1504/ijkesdp.2011.039875

On, E. P., & Yeniterzi, R. (2020). Cyberbullying Detection using Deep Learning and Word Embedding Analysis. *2020 28th Signal Processing and Communications Applications Conference, SIU 2020 - Proceedings*, 1–4. https://doi.org/10.1109/SIU49456.2020.9302297

Ortega, R., Elipe, P., Mora-Merchán, J. A., Calmaestra, J., & Vega, E. (2009). The emotional impact on victims of traditional bullying and cyberbullying: A study of Spanish adolescents. *Journal of Psychology*, *217*(4), 197–204. https://doi.org/10.1027/0044-3409.217.4.197

Özberk, A., & Çiçekli, I. (2021). Offensive Language Detection in Turkish Tweets with Bert Models. *Proceedings - 6th International Conference on Computer Science and Engineering, UBMK 2021*, 517–521. https://doi.org/10.1109/UBMK52708.2021.9559000

Özel, S. A., Akdemir, S., Saraç, E., & Aksu, H. (2017). Detection of cyberbullying on social media messages in Turkish. *2nd International Conference on Computer Science and Engineering, UBMK 2017*, 366–370. https://doi.org/10.1109/UBMK.2017.8093411

Öztürk, E. (2019). Cyberbullying detection using text classification for turkish language [Cukurova University]. In *tez.yok.gov.tr* (Vol. 3, Issue 1). https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp

Patchin, J. W., & Hinduja, S. (2006). Bullies Move Beyond the Schoolyard: A Preliminary Look at

Cyberbullying. *Youth Violence and Juvenile Justice*, *4*(2), 148–169. https://doi.org/10.1177/1541204006286288

Paul, S., & Saha, S. (2022). CyberBERT: BERT for cyberbullying identification. *Multimedia Systems*, *28*(6), 1897–1904. https://doi.org/10.1007/s00530-020-00710-4

Pedregosa. (2011). Scikit-learn: Machine Learning in Python Fabian. *Journal of Machine Learning Research*. https://doi.org/10.1289/EHP4713

Pericherla, S., & Ilavarasan, E. (2021). Performance analysis of Word Embeddings for Cyberbullying Detection. *IOP Conference Series: Materials Science and Engineering*, *1085*(1), 012008. https://doi.org/10.1088/1757-899x/1085/1/012008

Ponnappa, R. (2019). *Google Colab: Using GPU for Deep Learning*. https://python.gotrained.com/google-colab-gpu-deep-learning/

Privitera, C., & Campbell, M. A. (2009). Cyberbullying: The new face of workplace bullying? *Cyberpsychology and Behavior*, *12*(4), 395–400. https://doi.org/10.1089/cpb.2009.0025

Provost, F., & Kohavi, R. (1998). On Applied Research in Machine Learning. *Machine Learning*, *30*(2/3), 127–132. http://link.springer.com/10.1023/A:1007442505281

Qi, X., & Science, C. (2007). Web Page Classi cation: Features and Algorithms ∗. *Science*, *June*, 1–31.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1007/bf00116251

Quoc, L., & Tomas, M. (2014). Distributed Representations of Sentences and Documents Quoc. *International Conference on Machine Learning*, *9*(2), 42.

Raj, C., Agarwal, A., Bharathy, G., Narayan, B., & Prasad, M. (2021). Cyberbullying detection: Hybrid models based on machine learning and natural language processing techniques. *Electronics (Switzerland)*, *10*(22), 1–20. https://doi.org/10.3390/electronics10222810

Raskauskas, J. (2009). Text-bullying: Associations with traditional bullying and depression among New Zealand adolescents. *Journal of School Violence*, *9*(1), 74–97. https://doi.org/10.1080/15388220903185605

Raskauskas, J., & Stoltz, A. D. (2007). Involvement in traditional and electronic bullying among adolescents. *Developmental Psychology*, *43*(3), 564–575. https://doi.org/10.1037/0012-1649.43.3.564

Raut, V., & Patil, P. (2016). International Journal on Recent and Innovation Trends in Computing and Communication Use of Social Media in Education: Positive and Negative impact on the students. *International Journal on Recent and Innovation Trends in Computing and Communication*, *4*(1), 281–285. http://www.ijritcc.org

Rehurek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50.

Renjith, S., Sreekumar, A., & Jathavedan, M. (2020). An extensive study on the evolution of context-aware personalized travel recommender systems. *Information Processing and Management*, *57*(1), 102078. https://doi.org/10.1016/j.ipm.2019.102078

Rennie, J., .Lawrence, Shih, K., Jaime, T., & Dean, J. (2003). Tackling the Poor Assumptions of Naive Bayes Text Classifiers Jason. *Proceedings of the 20th International Conference on Machine Learning*, *8*(1973). https://doi.org/10.1186/1477-3155-8-16

Reynolds, K., Kontostathis, A., & Edwards, L. (2011). Using machine learning to detect cyberbullying. *Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011*, *2*, 241–244. https://doi.org/10.1109/ICMLA.2011.152

Rivers, I., & Noret, N. (2010). Findings from a Five-Year Study of Text and E-mail Bullying. *British Educational Research Journal*, *36*, 643–671.

Rivers, I., & Noret, N. (2010). "I h8 u": Findings from a five-year study of text and email bullying. *British Educational Research Journal*, *36*(4), 643–671. https://doi.org/10.1080/01411920903071918

Rysbek, D. (2019). Sentiment analysis with recurrent neural networks on Turkish reviews domain. In *PhD Thesis. Middle East Technical University* (Vol. 1, Issue 1).

Sadarangani, A., & Jivani, A. (2016). *A SURVEY OF SEMI-SUPERVISED LEARNING*. *7*(4), 217–223.

Samara, M., Burbidge, V., El Asam, A., Foody, M., Smith, P. K., & Morsi, H. (2017). Bullying and cyberbullying: Their legal status and use in psychological assessment. *International Journal of Environmental Research and Public Health*, *14*(12). https://doi.org/10.3390/ijerph14121449

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2–6. http://arxiv.org/abs/1910.01108

Santur, Y. (2019). Sentiment analysis based on gated recurrent unit. *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*, 1–5. https://doi.org/10.1109/IDAP.2019.8875985

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681. https://doi.org/10.1109/78.650093

Segura-Bedmar, I., Colón-Ruíz, C., Tejedor-Alonso, M. Á., & Moro-Moro, M. (2018). Predicting of anaphylaxis in big data EMR by exploring machine learning approaches. *Journal of Biomedical Informatics*, *87*(August), 50–59. https://doi.org/10.1016/j.jbi.2018.09.012

Shah, R., Aparajit, S., Chopdekar, R., & Patil, R. (2020). Machine Learning based Approach for Detection of Cyberbullying Tweets. *International Journal of Computer Applications*, *175*(37), 51–56. https://doi.org/10.5120/ijca2020920946

Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). *A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU*.

http://arxiv.org/abs/2305.17473

Singh, S. (2023). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. [Online]. https://www.labellerr.com/blog/roberta-a-robustly-optimized-bert-pretraining-approach/

Singhs, N. K., Singh, P., & Chand, S. (2022). Deep Learning based Methods for Cyberbullying Detection on Social Media. *3rd IEEE 2022 International Conference on Computing, Communication, and Intelligent Systems, ICCCIS 2022*, 521–525. https://doi.org/10.1109/ICCCIS56430.2022.10037729

Siwag, T., Sirohi, P., & Singhal, N. (2014). *Novel Architecture of a focused crawler for social websites*. *VII*(Iii), 132–144.

Siwei, L., Liheng, X., Kang, L., & Jun, Z. (2015). Convolutional recurrent neural networks for text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(4), 65–82. https://doi.org/10.4018/JDM.2021100105

Slonje, R., & Smith, P. K. (2008). Cyberbullying: Another main type of bullying?: Personality and Social Sciences. *Scandinavian Journal of Psychology*, *49*(2), 147–154. https://doi.org/10.1111/j.1467-9450.2007.00611.x

Smith, P. K., Mahdavi, J., Carvalho, M., Fisher, S., Russell, S., & Tippett, N. (2008). Cyberbullying: Its nature and impact in secondary school pupils. *Journal of Child Psychology and Psychiatry and Allied Disciplines*, *49*(4), 376–385. https://doi.org/10.1111/j.1469-7610.2007.01846.x

Smits, M. (2013). The Impact of Social Media on Business Performance. *Proceedings of the 21st European Conference on Information Systems THE*.

Smokowski, P. R., & Kopasz, K. H. (2005). Bullying in school: An overview of types, effects, family characteristics, and intervention strategies. *Children and Schools*, *27*(2), 101–109. https://doi.org/10.1093/cs/27.2.101

Sosa, P. M. (2017). Twitter Sentiment Analysis using combined LSTM-CNN Models. *Eprint Arxiv*, 1–9.

Sowmya, V., Bodhisattwa, M., Anuj, G., & Harshit, S. (2020). *Practical Natural Language Processing*.

Spears, B. (2009). Behind the scenes and screens: Insights into the human dimension of covert and cyberbullying. *Zeitschrift Für Psychologie/Journal of Psychology*.

Spears, Barbara, Slee, P., Owens, L., & Johnson, B. (2009). Behind the scenes and screens insights into the human dimension of covert and cyberbullying. *Journal of Psychology*, *217*(4), 189–196. https://doi.org/10.1027/0044-3409.217.4.189

Sreelakshmi, K., Premjith, B., Chakravarthi, B. R., & Soman, K. P. (2023). Detection of Hate Speech and Offensive Language CodeMix Text in Dravidian Languages using Cost-Sensitive Learning Approach. *IEEE Access*, *12*(January), 20064–20090. https://doi.org/10.1109/ACCESS.2024.3358811

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Sticca, F., & Perren, S. (2013). Is Cyberbullying Worse than Traditional Bullying? Examining the Differential Roles of Medium, Publicity, and Anonymity for the Perceived Severity of Bullying. *Journal of Youth and Adolescence*, *42*(5), 739–750. https://doi.org/10.1007/s10964-012-9867-3

Stojanovski, D., Strezoski, G., Madjarov, G., & Dimitrovski, I. (2016). Finki at SemEval-2016 task 4: Deep learning architecture for Twitter sentiment analysis. *SemEval 2016 - 10th International Workshop on Semantic Evaluation, Proceedings*, 149–154. https://doi.org/10.18653/v1/s16-1022

Su, J., Sayyad-Shirabad, J., & Matwin, S. (2011). Large scale text classification using semi-supervised multinomial naive bayes. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 97–104.

Sulzmann, J. N., Fürnkranz, J., & Hüllermeier, E. (2007). On pairwise naive bayes classifiers. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4701 LNAI*, 371–381. https://doi.org/10.1007/978-3-540-74958-5_35

Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient Transformers: A Survey. *ACM Computing Surveys*, *55*(6). https://doi.org/10.1145/3530811

Teng, T. H., & Varathan, K. D. (2023). Cyberbullying Detection in Social Networks: A Comparison Between Machine Learning and Transfer Learning Approaches. *IEEE Access*, *11*(June), 55533–55560. https://doi.org/10.1109/ACCESS.2023.3275130

Terrana, D., Augello, A., & Pilato, G. (2014). Facebook users relationships analysis based on sentiment classification. *Proceedings - 2014 IEEE International Conference on Semantic Computing, ICSC 2014*, 290–296. https://doi.org/10.1109/ICSC.2014.59

Theguardian. (2008). *No Title*. https://www.theguardian.com/world/2008/jun/17/usa.news [Accessed: December 2023]

Tjavanga, H. (2012). *SCHOOL BULLIES AND EDUCATION IN BOTSWANA : IMPACT ON*. *2*(1), 641–650.

Tokunaga, R. S. (2010). Following you home from school: A critical review and synthesis of research on cyberbullying victimization. *Computers in Human Behavior*, *26*(3), 277–287. https://doi.org/10.1016/j.chb.2009.11.014

Tomek, I. (2010). Two modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics*, *6(11): 769*.

Triguero, I., García, S., & Herrera, F. (2015). Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*, *42*(2),

245–284. https://doi.org/10.1007/s10115-013-0706-y

Ünal, H. E., & Özel, S. A. (2023). A Novel Approach for Semi-supervised Learning: Incremental Parallel Training with Cross-Validation (IPT-CV). *Arabian Journal for Science and Engineering*, *48*(8), 10457–10477. https://doi.org/10.1007/s13369-022-07433-w

Uysal, A. K., & Murphey, Y. L. (2017). Sentiment Classification: Feature Selection Based Approaches Versus Deep Learning. *IEEE CIT 2017 - 17th IEEE International Conference on Computer and Information Technology*, 23–30. https://doi.org/10.1109/CIT.2017.53

Vaez, M., Ekberg, K., & Laflamme, L. (2004). *Abusive Events at Work among Young Working Adults Magnitude of the Problem and its Effect on Self-Rated Health La violence au travail chez les jeunes adultes : l ' importance du problème et ses effets sur la santé Casos de abuso en*.

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373–440. https://doi.org/10.1007/s10994-019-05855-6

Van Hee, C., Jacobs, G., Emmery, C., Desmet, B., Lefever, E., Verhoeven, B., De Pauw, G., Daelemans, W., & Hoste, V. (2018). Automatic detection of cyberbullying in social media text. *ArXiv*, 1–22. https://doi.org/10.17605/OSF.IO/RGQW8.

Van Hee, C., Jacobs, G., Emmery, C., DeSmet, B., Lefever, E., Verhoeven, B., De Pauw, G., Daelemans, W., & Hoste, V. (2018). Automatic detection of cyberbullying in social media text. *PLoS ONE*, *13*(10), 1–23. https://doi.org/10.1371/journal.pone.0203794

Vandebosch, H., & van Cleemput, K. (2009). Cyberbullying among youngsters: Profiles of bullies and victims. *New Media and Society*, *11*(8), 1349–1371. https://doi.org/10.1177/1461444809341263

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, *10*(5), 988–999. https://doi.org/10.1109/72.788640

Vardhini, V., Raja, P., & Devi, K. (2019). Social media as the next trend in social business marketing social media as the next trend in solar business marketing. *International Journal of Innovative Technology and Exploring Engineering*, *8*(11 Special Issue), 760–763. https://doi.org/10.35940/ijitee.K1133.09811S19

Wang, Z., & Wang, H. (2021). A neural joint model with BERT for Burmese syllable segmentation, word segmentation, and POS tagging. *Transactions on Asian and Low-Resource Language Information Processing*. https://dl.acm.org/doi/abs/10.1145/3436818

Wilson, D. L. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE*

Transactions on Systems, Man and Cybernetics, *2*(3), 408–421. https://doi.org/10.1109/TSMC.1972.4309137

Witten, I. H., Hall, M. A., & Frank, E. (2005). Data Mining: Practical Machine Learning Third Edition. *(Morgan Kaufmann Series in Data Management Systems). Morgan Kaufmann*, *104*(June), 113.

Wolf, T. et al. (2020). State-of-the-Art Natural Language Processing. *Deep Learning Approach for Natural Language Processing, Speech, and Computer Vision*, 49–75. https://doi.org/10.1201/9781003348689-3

Wong, C. I., Wong, K. Y., Ng, K. W., Fan, W., & Yeung, K. H. (2014). Design of a crawler for online social networks analysis. *WSEAS Transactions on Communications*, *13*(March), 263–274.

Wright, V. H., Burnham, J. J., Inman, C. T., & Ogorchock, H. N. (2009). Cyberbullying: Using Virtual Scenarios to Educate and Raise Awareness. *Journal of Computing in Teacher Education*, *26*(1), 35–42.

Xia, R., Wang, C., Dai, X., & Li, T. (2015). Co-Training for semi-supervised sentiment classification based on dual-view bags-of-words representation. *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, *1*, 1054–1063. https://doi.org/10.3115/v1/p15-1102

Xiao, Z., Liu, B., Hu, H., & Zhang, T. (2012). Design and implementation of facebook crawler based on interaction simulation. *Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012*, 1109–1112. https://doi.org/10.1109/TrustCom.2012.121

Xu, J. M., Jun, K. S., Zhu, X., & Bellmore, A. (2012). Learning from bullying traces in social media. *NAACL HLT 2012 - 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 656–666.

Yang, A., & Salmivalli, C. (2013). Different forms of bullying and victimization: Bully-victims versus bullies and victims. In *European Journal of Developmental Psychology* (Vol. 10, Issue 6, pp. 723–738). Taylor & Francis. https://doi.org/10.1080/17405629.2013.793596

Ybarra, M. L., Ph, D., Mitchell, K. J., & Ph, D. (2007). *Prevalence and Frequency of Internet Harassment Instigation : Implications for Adolescent Health*. *41*, 189–195. https://doi.org/10.1016/j.jadohealth.2007.03.005

Yin, D., Xue, Z., Hong, L., Davison, B. D., Kontostathis, A., & Edwards, L. (2009). Detection of

Harassment on Web 2.0. *Proceedings of the Content Analysis in the WEB ., 2*, 1–7.

Young, B., & Nelson, D. (1999). Relational Aggression in Schools : Information for Educators. *Helping Children at Home and School IIi*, 1999–2002.

Yu, N. (2013). Domain Adaptation for Opinion Classification: A Self-Training Approach. *Journal of Information Science Theory and Practice*, *1*(1), 10–26. https://doi.org/10.1633/jistap.2013.1.1.1

Zafarani, R., Abbasi, M. A., & Liu, H. (2014). Social media mining: An introduction. *Social Media Mining: An Introduction*, *9781107018*, 1–320. https://doi.org/10.1017/CBO9781139088510

Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z., & Çöltekin, Ç. (2020). SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). *14th International Workshops on Semantic Evaluation, SemEval 2020 - Co-Located 28th International Conference on Computational Linguistics, COLING 2020, Proceedings*, *OffensEval*, 1425–1447. https://doi.org/10.18653/v1/2020.semeval-1.188

Zhang, Dejun, Tian, L., Hong, M., Han, F., Ren, Y., & Chen, Y. (2018). Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification. *IEEE Access*, *6*, 73750–73759. https://doi.org/10.1109/ACCESS.2018.2882878

Zhang, & Liu, B. (2018). Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(4), 1–25. https://doi.org/10.1002/widm.1253

Zhang, Rahman, M. M., Braylan, A., Dang, B., Chang, H.-L., Kim, H., McNamara, Q., Angert, A., Banner, E., Khetan, V., McDonnell, T., Nguyen, A. T., Xu, D., Wallace, B. C., & Lease, M. (2016). *Neural Information Retrieval: A Literature Review*. http://arxiv.org/abs/1611.06792

Zhang, X., Tong, J., Vishwamitra, N., Whittaker, E., Mazer, J. P., Kowalski, R., Hu, H., Luo, F., Macbeth, J., & Dillon, E. (2017). Cyberbullying Detection with a Pronunciation Based Convolutional Neural Network. *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 740–745. https://doi.org/10.1109/icmla.2016.0132

Zhang, Z. (2019). Improved Adam Optimizer for Deep Neural Networks. *2018 IEEE/ACM 26th International Symposium on Quality of Service, IWQoS 2018*, 1–2. https://doi.org/10.1109/IWQoS.2018.8624183

Zhou, Z. H., & Li, M. (2005). Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, *17*(11), 1529–1541. https://doi.org/10.1109/TKDE.2005.186

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015).

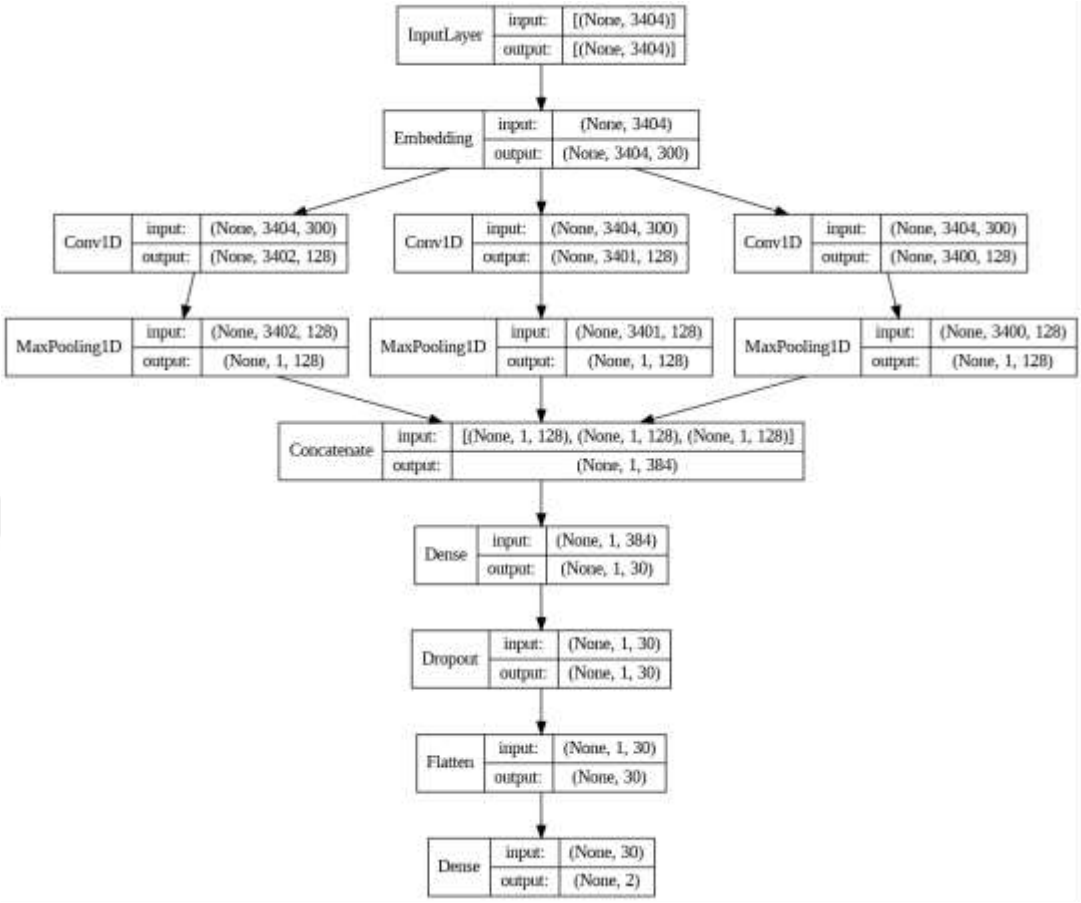
Aligning Books and Movies : Towards Story-like Visual Explanations by Watching Movies and Reading Books. 19–27.

CURRICULUM VITAE

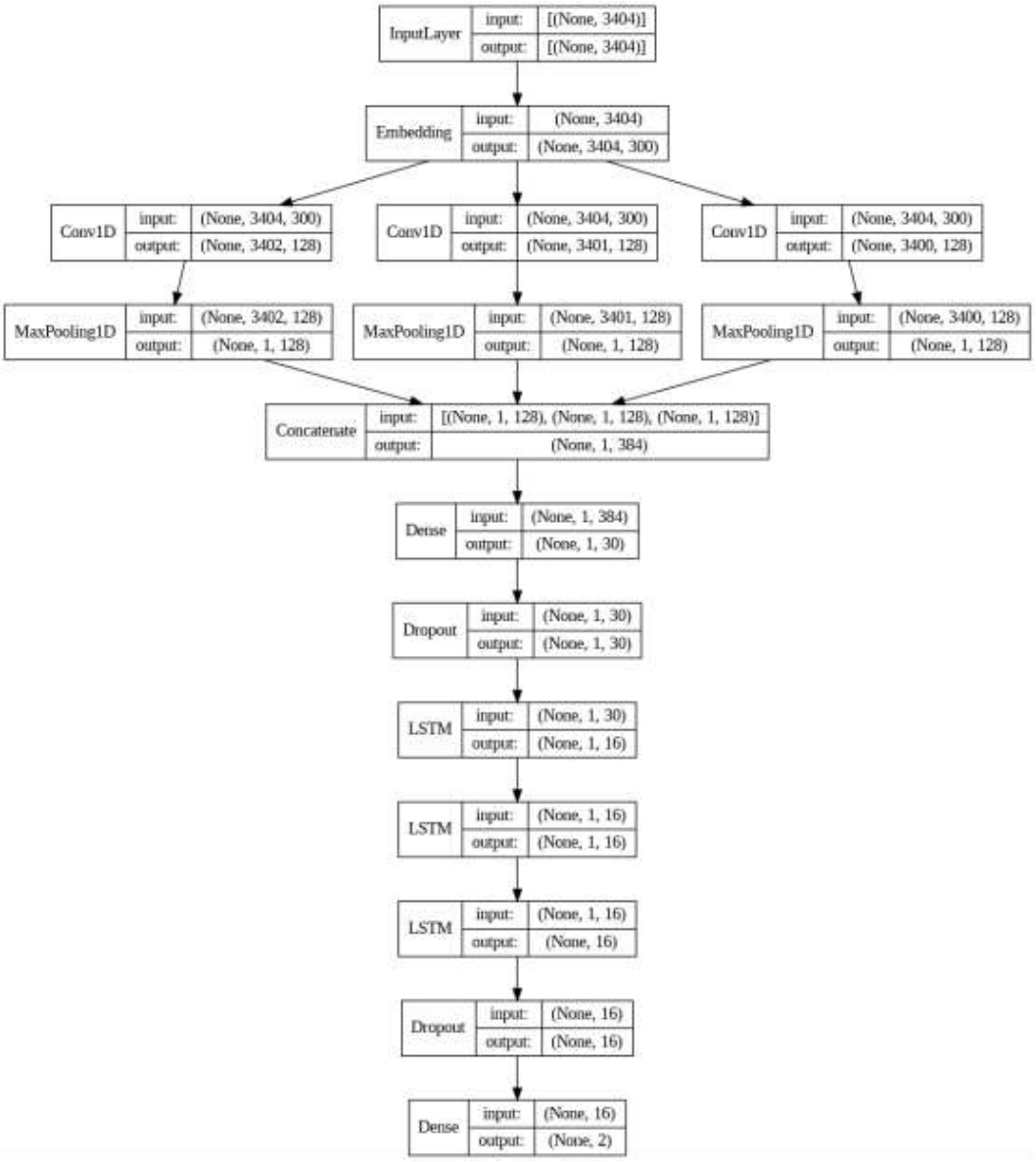
Ali NAJIB is an academic scholar from Uganda. Najib obtained Bachelor's degree in Information Technology from the Islamic University in Uganda (IUIU), followed by a Master's degree in Computing from University Brunei Darussalam (UBD). Subsequently, Najib completed a postgraduate diploma in Monitoring and Evaluation from Uganda Management Institute (UMI). Until recently, Najib works as a lecturer at the department of Computer Science IUIU.

APPENDICES

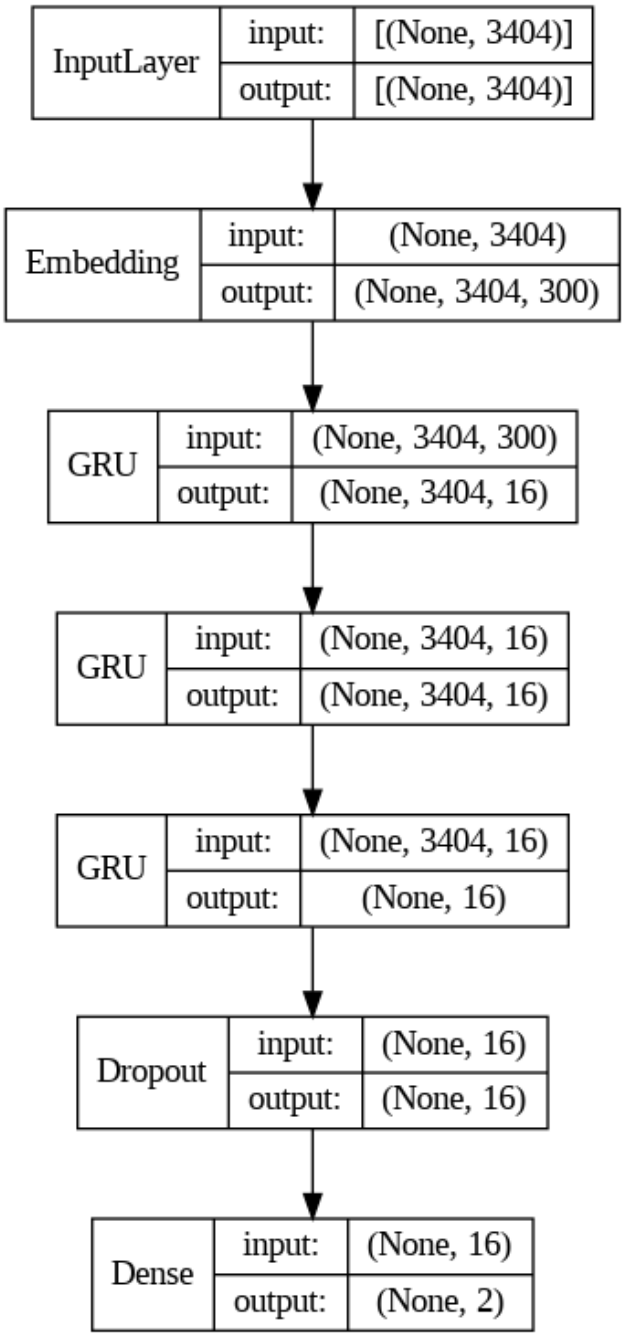
APPENDIX A:



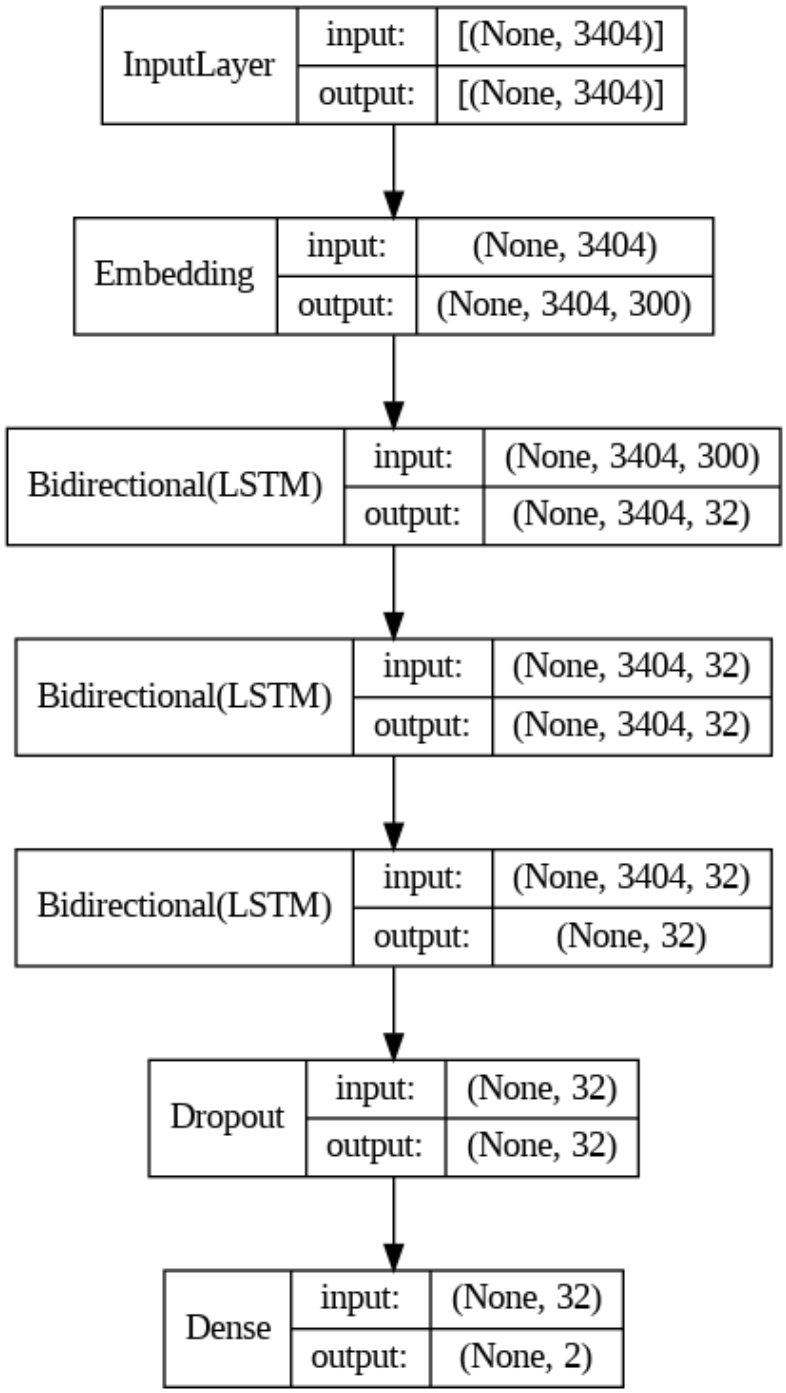
Appendix A.1. summary of CNN structure



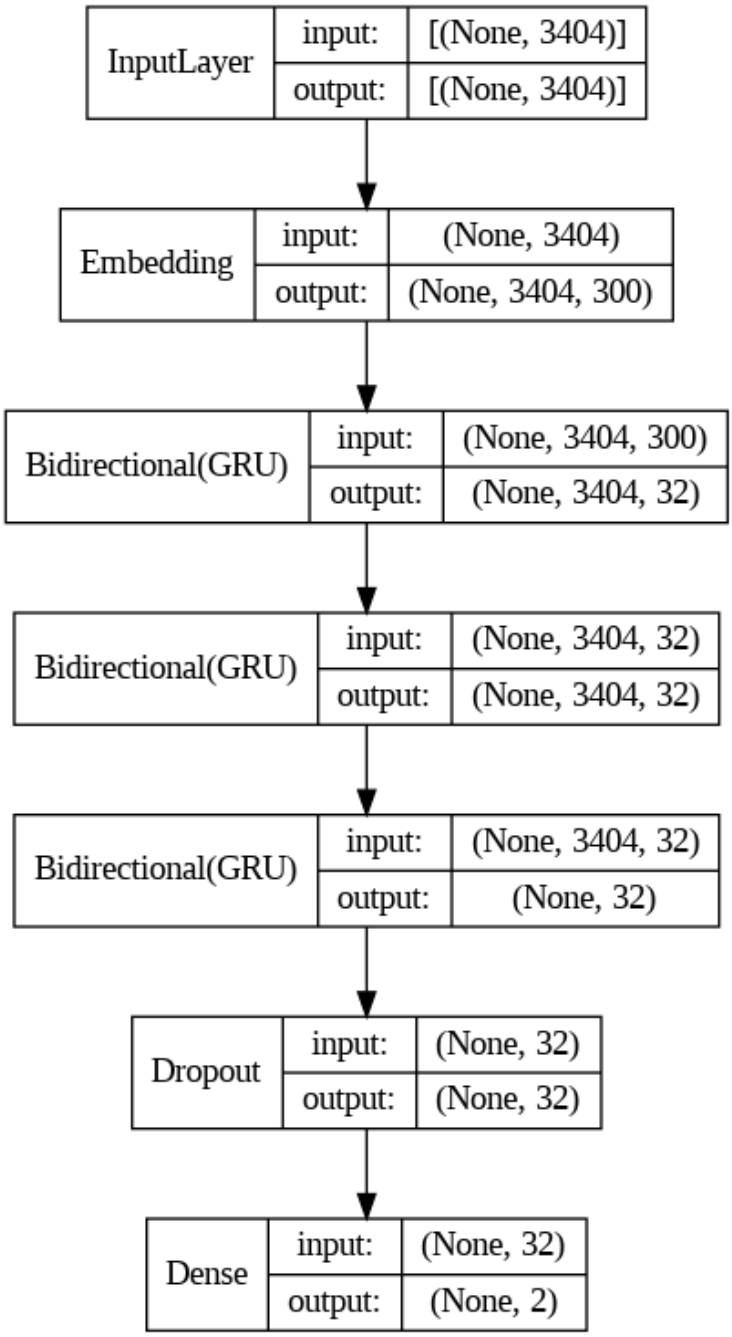
Appendix A.2. summary of LSTM structure



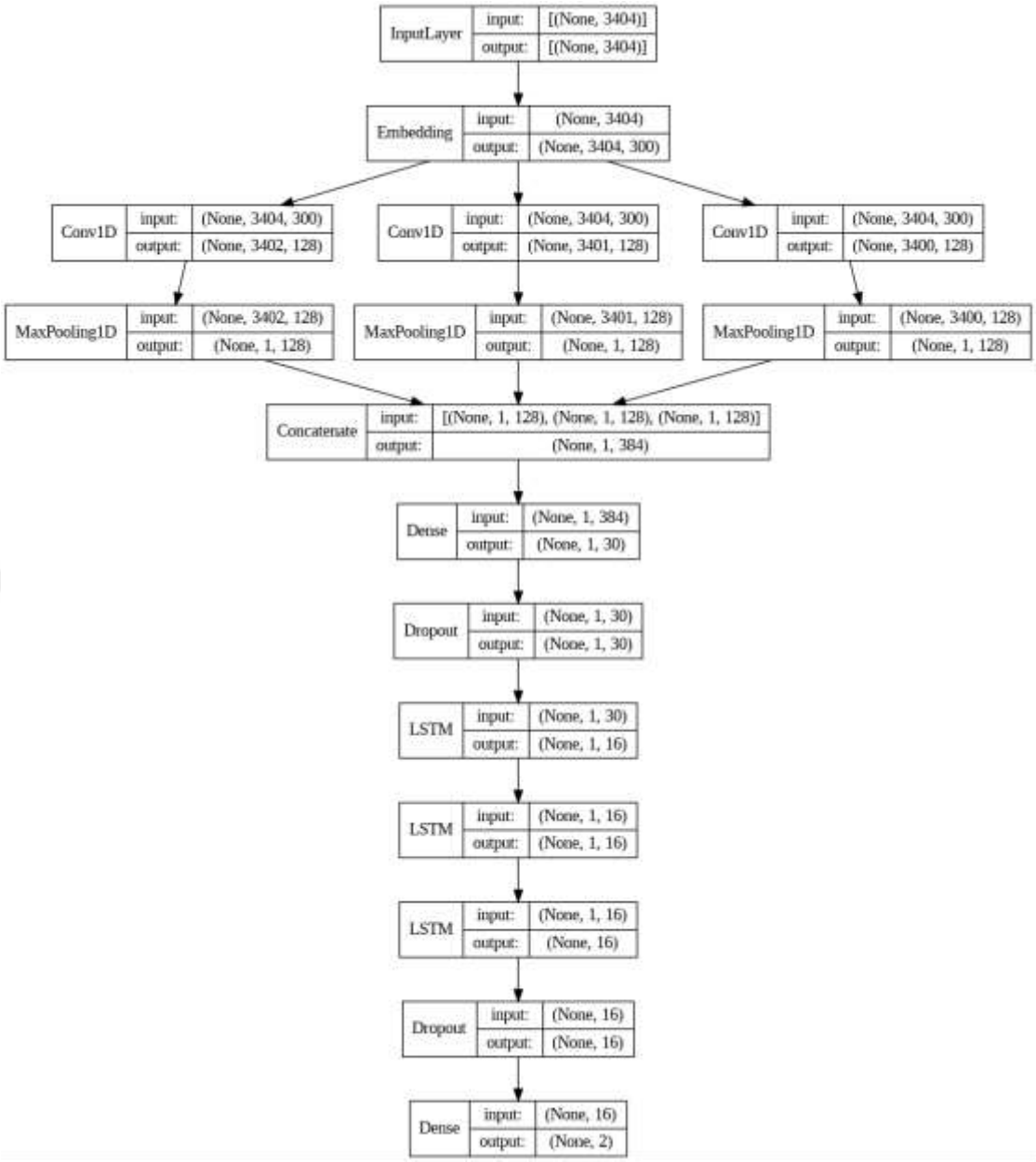
Appendix A.3. summary of GRU structure



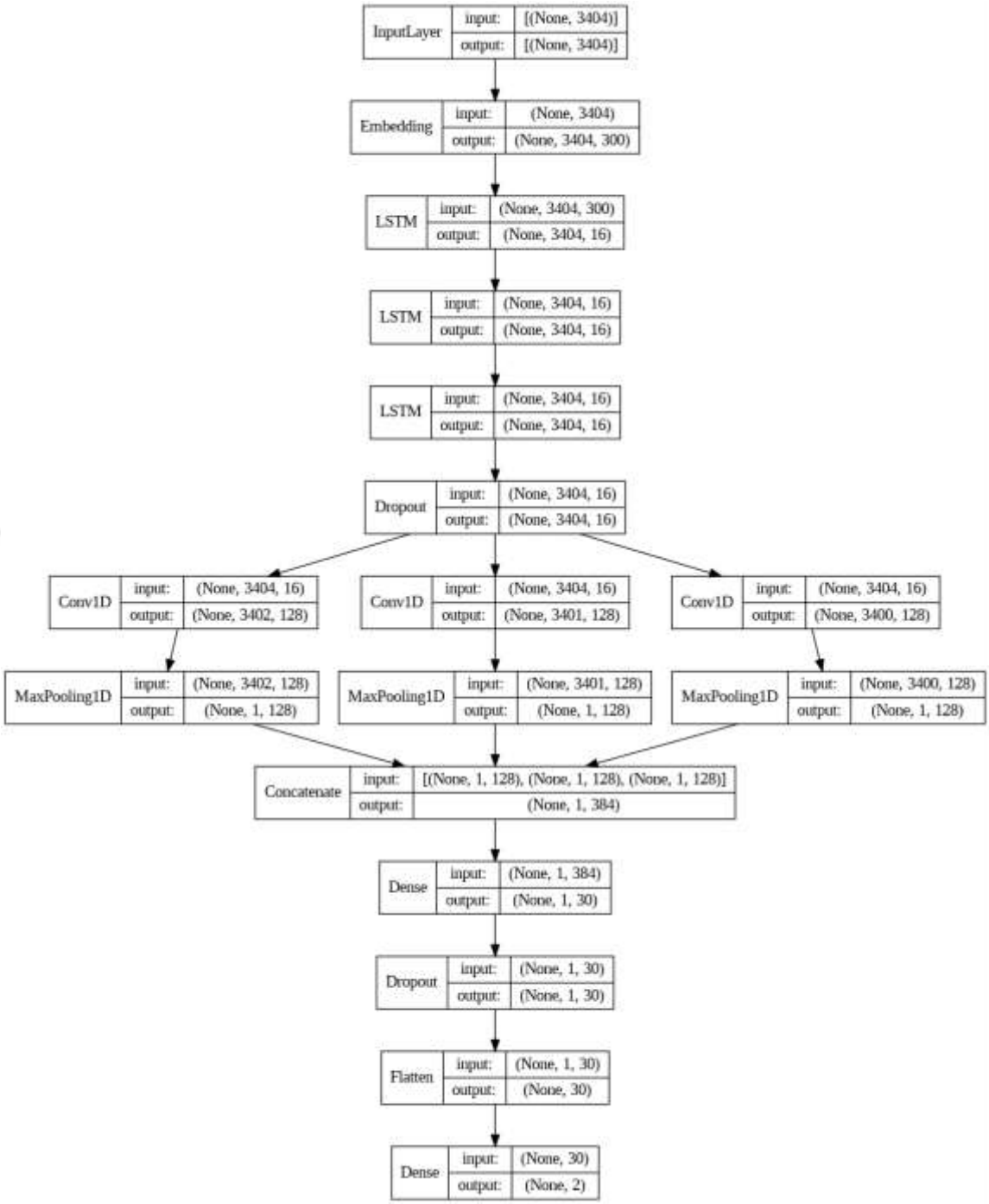
Appendix A.4. summary of BiLSTM structure



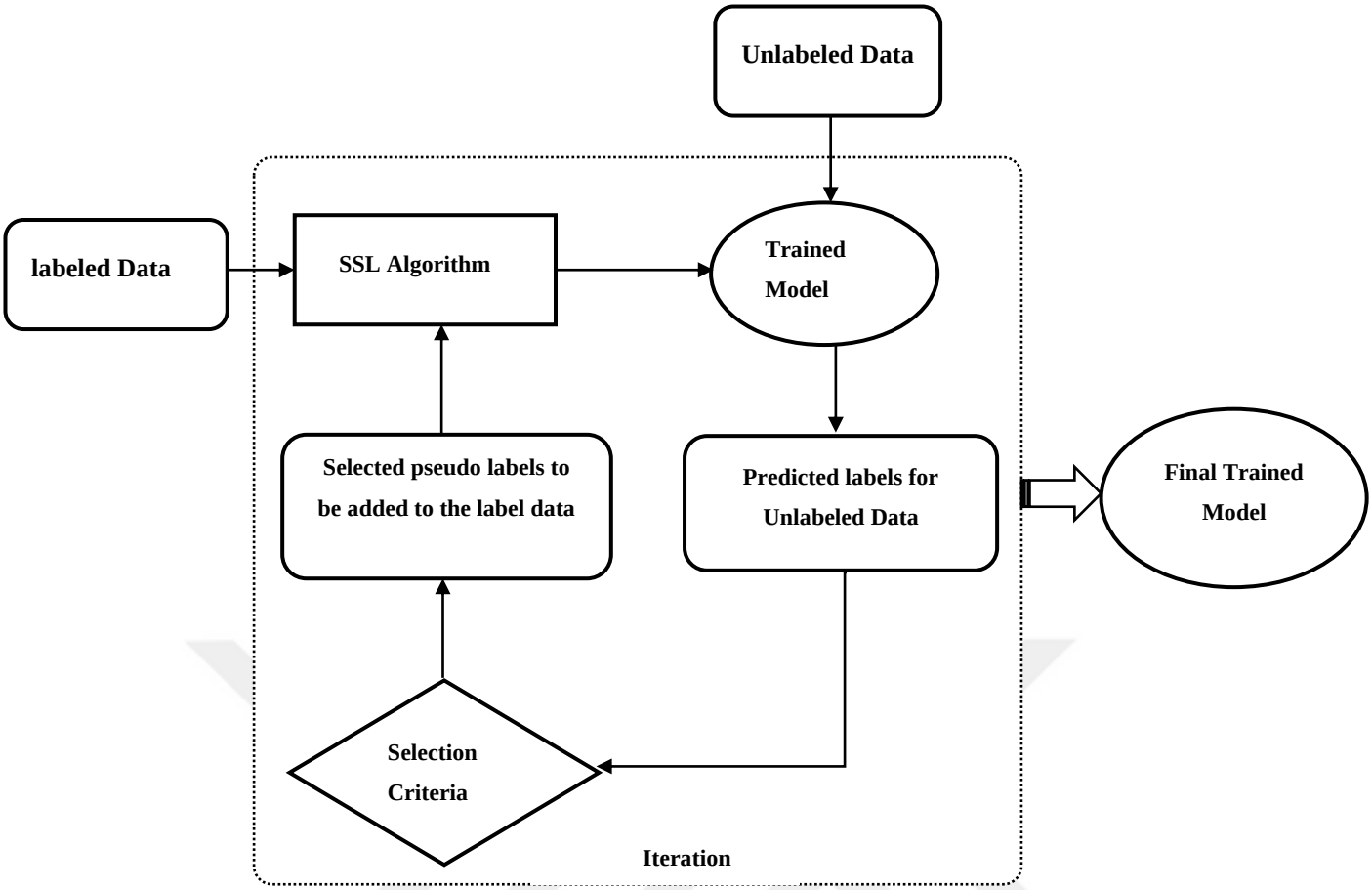
Appendix A.5. summary of BiGRU structure



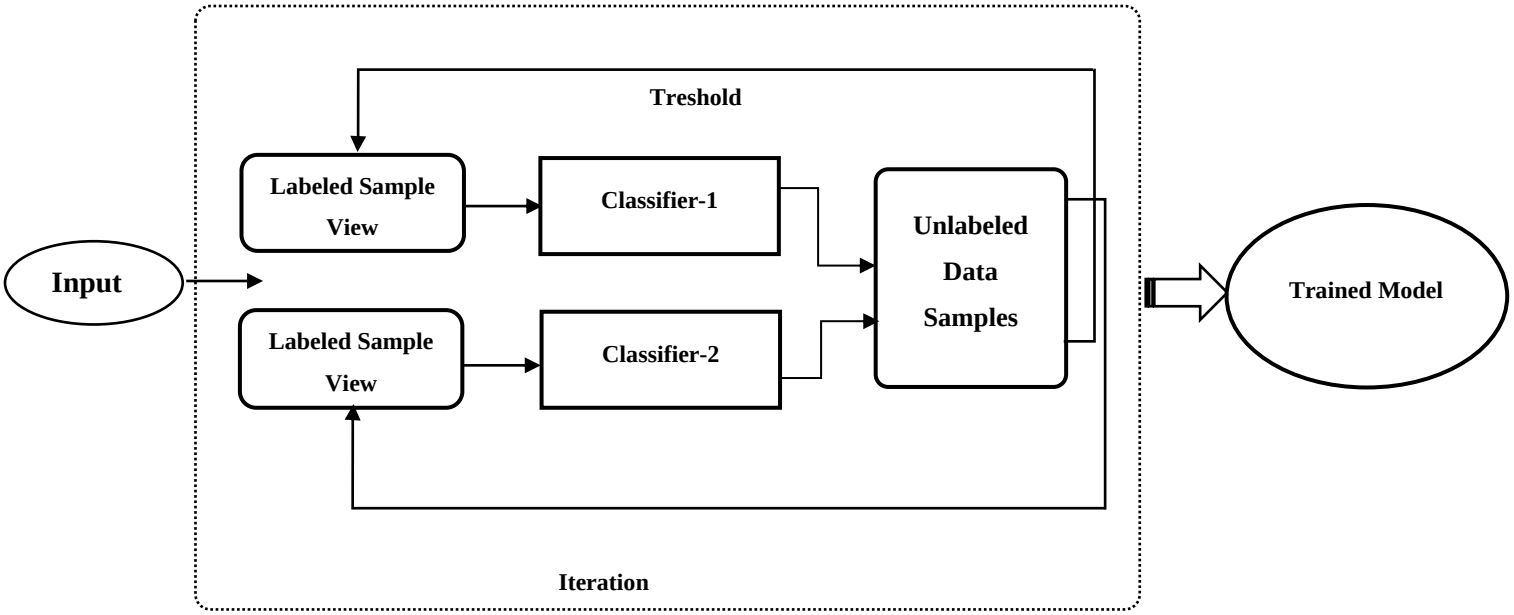
Appendix A.6. summary of CNN-LSTM structure



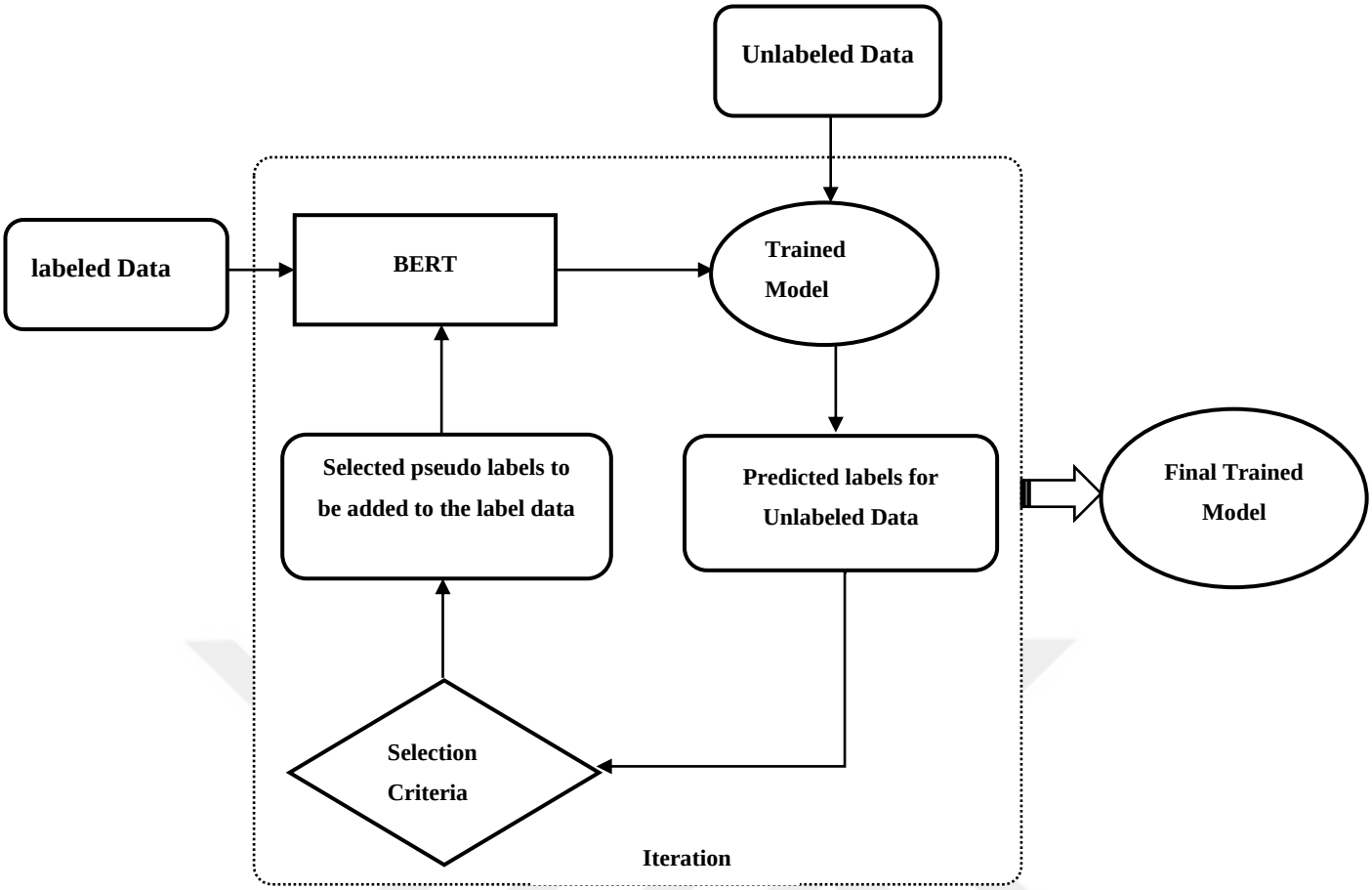
Appendix A.7. summary of LSTM-CNN structure



Appendix A.8. Self-training



Appendix A.9. Co-training



Appendix A. 10. Self-training with Bert

APPENDIX C: Research Publications from this Thesis

Title	Status	Year
Comparative Analysis of Cyberbullying Detection: A case study for Turkish and English **2023 IEEE Innovations in Intelligent Systems and Applications**	Published	2023
Cyberbullying Detection Using Machine Learning and Deep Learning for Turkish Text **UDEF 9th International Student Symposium**	Published	2024