

**ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES**

MSc THESIS

Erhan ÖZTÜRK

**CYBERBULLYING DETECTION USING TEXT
CLASSIFICATION FOR TURKISH LANGUAGE**

DEPARTMENT OF COMPUTER ENGINEERING

ADANA-2019

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES


CYBERBULLYING DETECTION USING TEXT CLASSIFICATION
FOR TURKISH LANGUAGE

Erhan ÖZTÜRK


MSc THESIS

DEPARTMENT OF COMPUTER ENGINEERING

We certify that the thesis titled above was reviewed and approved for the award of degree of the Master of Science by the board of jury on 18/07/2019


.....
Prof. Dr. Selma Ayşe ÖZEL
SUPERVISOR


.....
Asst. Prof. Dr. Buse Melis ÖZYILDIRIM
MEMBER


.....
Asst. Prof. Dr. Onur ÜLGEN
MEMBER

This MSc Thesis is written at the Computer Engineering Department of Institute of Natural and Applied Sciences of Çukurova University.

Registration Number:

Prof. Dr. Mustafa GÖK
Director
Institute of Natural and Applied Sciences

Note: The usage of the presented specific declarations, tables, figures, and photographs either in this thesis or in any other reference without citation is subject to "The law of Arts and Intellectual Products" number of 5846 of Turkish Republic

ABSTRACT

MSc THESIS

CYBERBULLYING DETECTION USING TEXT CLASSIFICATION FOR TURKISH LANGUAGE

Erhan ÖZTÜRK

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF COMPUTER ENGINEERING

Supervisor : Prof. Dr. Selma Ayşe ÖZEL
II. Supervisor : Asst. Prof. Dr Esra SARAÇ EŞSİZ
Year: 2019, Pages: 83
Jury : Prof. Dr. Selma Ayşe ÖZEL
: Asst. Prof. Dr. Buse Melis ÖZYILDIRIM
: Asst. Prof. Dr. Onur ÜLGEN

Cyberbullying is an electronic form of peer harassment. It includes relational attack behaviors such as harassing people, mocking people, threatening, spreading gossip, and insulting people on the internet by using information and communication technologies. In Turkey and many European countries, the cyberbullying is considered as a serious problem after the cyberbullying related suicides occurred. In recent years, researches are being carried out and solutions are tried to be found by experts, especially with educational scientists and psychologists, about cyberbullying.

The aim of this study is to create the largest Turkish dataset so far for the detection of cyberbullying texts and to show the effects of preprocessing, feature selection and classifiers for the detection of cyberbullying from texts.

In this study, a number of preprocessing steps are applied, and two well-known filter-based methods that are information gain and chi square are used for feature selection. Among the classifiers tested, Naive Bayes Multinomial is determined to be the most successful method for detecting cyberbullying from texts written in Turkish language. In addition, a filter-based classifier is proposed, and its performance is tested on the collected dataset. The proposed method has promising accuracy and can be used for labeling any Turkish text document without re-training the classifier.

Key Words: Cyberbullying, Classification, Preprocessing, Turkish Dataset, Filter Based Classifier

ÖZ

YÜKSEK LİSANS TEZİ

TÜRKÇE DİLİ İÇİN METİN SINIFLANDIRMA KULLANARAK SİBER ZORBALIK TESPİTİ

ERHAN ÖZTÜRK

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Danışman : Prof. Dr. Selma Ayşe ÖZEL
II. Danışman : Dr. Öğr. Üyesi Esra SARAÇ EŞSİZ
Year: 2019, Pages: 83
Jüri : Prof. Dr. Selma Ayşe ÖZEL
: Dr. Öğr. Üyesi Buse Melis ÖZYILDIRIM
: Dr. Öğr. Üyesi Onur ÜLGEN

Siber zorbalık ekran tacizinin elektronik bir formudur. Bilgi ve iletişim teknolojilerini kullanarak kişileri sürekli rahatsız etme, kişilerle alay etme, tehdit, dedikodu yayma, internet üzerinden kişiye hakaret etme gibi ilişkisel saldırı davranışlarını içerir. Türkiye ve pek çok Avrupa ülkesi için intiharla sonuçlanan olaylardan sonra ciddi bir konu olarak ele alınmıştır. Siber zorbalık ile ilgili özellikle son yıllarda eğitim bilimciler ve psikologlar başta olmak üzere, uzmanlar tarafından araştırmalar yapılmakta ve çözüm yöntemleri aranmaktadır.

Bu çalışmanın amacı Türkçe içerikli siber zorbalık metinlerinin tespiti için şu ana kadar yapılmış en büyük Türkçe veri kümesini oluşturmak ve siber zorbalık metinlerinin tespiti için önışleme, nitelik seçimi ve sınıflandırıcıların etkilerini göstermektir.

Bu çalışmada birçok önışleme adımı uygulanmış olup, nitelik seçimi için iki adet çok bilinen filtre tabanlı nitelik seçim yöntemi (bilgi kazancı ve ki-kare yöntemleri) uygulanmıştır. Test edilen sınıflandırıcılar arasından Naive Bayes Multinomial Türkçe içerikli siber zorbalık metinlerini sınıflandırmada en etkili yöntem olarak belirlenmiştir. Ayrıca toplanan veri kümesi üzerinden filtre tabanlı bir sınıflandırıcı önerilmiş olup, elde edilen veri kümesi üzerinde doğruluk analizi yapılmıştır. Önerilen yöntemin tatmin edici sınıflama başarısı olduğu görülmüş olup, herhangi bir Türkçe metni sınıflayıcıyı tekrar eğitmeden sınıflayabilecek yapıdadır.

Anahtar Kelimeler: Siber Zorbalık, Sınıflandırma, Ön İşleme, Türkçe Veri Kümesi, Filtre Tabanlı Sınıflayıcı

EXTENDED ABSTRACT

Technology is inevitably involved in every aspect of our lives. Especially with the widespread use of computers and the increase in internet usage, access to information has become very easy. While there are many positive aspects of technology that make human life easier, there are some negative aspects. One of the most important of these negative side-effects of the technology is cyberbullying.

Cyberbullying is an electronic form of peer harassment. Cyberbullying is an important problem that can have negative psychological effects on individuals if precautions are not taken. Individuals exposed to cyberbullying are adversely affected by these events. Although the people who exhibit cyberbullying behaviors think they do not harm anyone, those who are exposed to these events can end their lives due to these negative events. In spite of the fact that each individual does not attempt suicide as a result of these negative events, some individuals are affected adversely due to these events. Especially in recent years, cyberbullying messages are frequently seen in social media applications. In this context, it is important to detect cyberbullying in messages sent from the electronic environment and prevent them from harming the victim.

When the past studies are examined, it is seen that the number of samples of dataset collected in the Turkish language for the detection of cyberbullying texts is not very large and the number of messages collected for cyberbullying texts is around 3000-5000. In this study, it is aimed to create the largest dataset in Turkish language to detect the cyberbullying texts and to examine the effects of preprocessing, feature selection and classifiers on this dataset. The dataset created is aimed to be open source and will be available to everyone in the near future.

The dataset used in this study was collected from 4 different web sites one by one, or automatically with the help of written code snippets. Cyberbullying texts

are collected by using text content in Turkish messages obtained from social media applications such as Facebook, Twitter, Youtube and Instagram. Comments containing swearing, insults, and sexual assault texts are selected and collected from Facebook, Twitter and Youtube sites one by one, while the Instagram data is automatically downloaded with a written javascript code from the website. All collected text messages were classified with two labels as positive or negative. Groups containing cyberbullying texts are labeled as negative, whereas others are labeled as positive.

A total of 15658 Turkish texts, 7995 of which were positive samples without cyber-bullying text, and 7663 of negative instances containing cyber-bullying text, are obtained. The size of the dataset with respect to number of labeled samples is the most comprehensive dataset for the Turkish cyber-bullying dataset.

After the creation of the dataset containing a sufficient number of cyberbullying texts, preprocessing and classification steps are applied to show the effectiveness of the collected dataset and text classification methods on detection of cyberbullying from Turkish text contents. To perform preprocessing and classification tasks WEKA software is used. WEKA is a java-based software that contains many machine learning libraries such as preprocessing, classification, clustering, feature selection and feature extraction. Taking advantage of these features, the collected dataset in this study is classified by using the WEKA, also the effects of several preprocessing, feature selection, and classification methods are compared.

In the preprocessing step, all special characters in the text contents are removed at first. All texts are translated into ARFF format so that these texts can be read by the WEKA software. By default, WEKA has a structure that cannot read Turkish characters in ARFF files. To prevent this, the Arabic Light Stemmer package developed by Motaz K. Saad (2010), which can read Turkish words, is

included in the WEKA program. As a result, utf-8 format characters which is also included in the Turkish texts in the ARFF data added to the system could be read easily. This information, read with WEKA, is basically labeled as two classes, positive and negative. Many preprocessing steps such as TF-IDF weighting, stemming, and minTermFreq filtering in the WEKA software have been used to observe the effects of these methods on the classification performance.

In the feature selection stage, two commonly used methods, that are chi-square and information gain methods, are used and their performance on cyberbullying detection are compared. Chi-square feature selection method is found to be slightly more successful than Information Gain. In the feature selection step, a number of features having the highest scores are selected, and also features having scores below 0.001 are removed from the feature set. In order to test the success of the applied methods, k-fold cross validation is chosen as the evaluation method. Therefore, the dataset is appropriately divided into training and test datasets.

In the classification section, Naive Bayes Multinomial, Support Vector Machines (libSVM classifier of WEKA data mining tool), Decision Trees – J48, Random Forest, and the proposed filter-based classifiers are used and compared. All the classifiers except the proposed method are realized with the help of WEKA software. For the proposed filter-based classifier, the dataset is first preprocessed with the help of Zemberek tool so that the misspelled words are corrected and then the corrected words are stemmed. The words from the positive and negative messages are listed. After that, set of positive words are subtracted from the set of negative words, and the remaining words are sorted with respect to their frequency in the dataset. The most frequent 5000 words are chosen as bad words and the list of bad words is obtained. Then, a new text message is classified by counting the number of bad words in it. If this value is greater than a threshold value then, the text message is classified as negative, otherwise it is labeled as positive. This threshold value is determined by experimentally and it is equal to 3. All

classification performances are measured with F-measure. Performance evaluations are done separately for preprocessing, feature selection and classification methods selected in this study.

When the performance results are examined, it is observed that TF*IDF weighting method gives the best result as weighting method for preprocessing stage. Also, stemming has positive effect on the classification process. Naive Bayes classification is used in each step of preprocessing and feature selection. Chi-square algorithm is found to be more successful than information gain algorithm for feature selection. According to the classification performance comparison, the best performance belongs to Naive Bayes algorithm and the worst performance is obtained from Random Forest algorithm. Our proposed classifier has the second best performance among the classifiers used.

As a result, in this thesis, a dataset in Turkish language, which can be used for cyberbullying detection from text messages, is created. It is seen that TF*IDF weighting and stemming have a positive effect on the accuracy of the classification process. Chi-square feature selection method is found to be more successful than Information Gain for the feature selection. Naive Bayes Multinomial is the most suitable classifier in terms of classification time and classification accuracy. In addition, the proposed classifier is found to be close to other classifiers in terms of performance, which shows that the dataset is large enough to extract bad words list for Turkish.

As future work, it is planned to further develop the dataset with Turkish content and to collect more text content. The dataset prepared for this purpose is aimed to be shared on the internet as an open source. With the help of this dataset, it is expected that it will be a source for further studies to detect cyberbullying in Turkish texts.

GENİŞLETİLMİŞ ÖZET

Teknoloji kaçınılmaz olarak hayatımızın her alanında yer almaktadır. Özellikle bilgisayarların yaygın kullanılması ve internet kullanımının artışı ile bilgiye erişim oldukça kolay hale gelmiştir. Her ne kadar teknolojinin insan hayatını kolaylaştıran yanları olsa da, bazı olumsuz yanları vardır. Teknolojinin bu olumsuz yan etkilerinin en önemlilerinden birisi siber zorbalıktır. Siber zorbalık akran tacizinin elektronik biçimidir. Siber zorbalık eğer önlem alınmaz ise bireyler üzerinde olumsuz psikolojik etkileri olan önemli bir problemdir.

Siber zorbalığa maruz kalan bireyler bu olaylardan olumsuz şekilde etkilenmektedir. Siber zorbalık davranışları sergileyenler bu davranışları yaptıkları kişilere herhangi bir zarar vermediklerini düşünseler de; bu olaylara maruz kalanlar, yaşadıkları olumsuz olayın etkisiyle hayatlarına son verebilmektedirler. Yaşanan bu olumsuz olaylar sonucunda her birey intihara kalkışmasa da bazı bireylerde olumsuz sonuçlar doğurmaktadır. Özellikle son zamanlarda, siber zorbalık mesajları sosyal medya uygulamalarında sıklıkla görülmektedir. Bu bağlamda, elektronik ortamda gönderilen mesajlardaki siber zorbalıkları tespit etmek ve mağdura zarar vermesini önlemek önemlidir.

Geçmiş çalışmalar incelendiğinde Türkçe dilinde siber zorbalık metnlerinin tespiti için yapılan çalışmalarda, toplanan veri kümelerinin örnek sayısının çok geniş olmadığı, siber zorbalık metinleri için toplanan mesajların sayısının en fazla 3000-5000 civarlarında olduğu görülmüştür. Bu çalışmada, siber zorbalık metnlerini tespit etmek için Türkçe dilindeki en büyük veri kümesinin oluşturulması ve önışleme, nitelik seçimi ve sınıflandırıcıların bu veri kümesi üzerindeki etkilerinin incelenmesi amaçlanmıştır. Oluşturulan veri kümesinin açık kaynak olması ve yakın gelecekte herkesin kullanımına açık olması hedeflenmektedir.

Bu çalışmada kullanılan veri kümesi 4 farklı web sitesinden tek tek ya da yazılan kod parçacıklarıyla otomatik olarak toplanmıştır. Türkçe mesaj içeren sosyal medya uygulamaları olan; Facebook, Twitter, Youtube ve Instagram'dan

elde edilen metin içerikleri kullanılarak siber zorbalık metinleri toplanmaya çalışılmıştır. Bu yorumlar toplanırken özellikle “Siber Zorbalık” başlığı altında küfür, hakaret, cinsel içerikli saldırı metinleri Facebook, Twitter ve Youtube sitelerinden tek tek seçilip el ile toplanırken, Instagram verileri yazılan bir javascript koduyla web sitesi üzerinden otomatik olarak çekilmiştir. Toplanan tüm veri kümeleri pozitif veya negatif olmak üzere iki etiket ile sınıflandırılmıştır. Siber zorbalık metinleri içeren örnekler negatif, bu metinlere dâhil olmayanlar pozitif metin olarak etiketlenmiştir.

Toplamda 7995 adet siber zorbalık metni içermeyen pozitif etiketli örnek, 7663 adet siber zorbalık metni içeren negatif etiketli örnek olmak üzere; 15658 adet Türkçe metinden oluşan etiketli örnek elde edilmiştir. Elde edilen bu veri kümesi, örnek sayısı dikkate alındığında Türkçe içerikli siber zorbalık veri kümesi adına en geniş kapsamlı veri kümesidir.

Yeterli sayıda siber zorbalık metni içeren bir veri kümesi oluşturulduktan sonra, Türkçe metinli mesajlardaki siber zorbalığın tespiti için önışleme, nitelik seçimi ve metin sınıflandırma yöntemleri uygulanarak oluşturulan veri kümesinin etkinliği gösterilmiştir.

Önışleme ve sınıflandırma işlemlerini yapmak için WEKA yazılımı kullanılmıştır. WEKA java tabanlı bir yazılım olup önışleme, sınıflandırma, kümeleme, nitelik seçimi ve nitelik çıkarımı gibi birçok makina öğrenme kütüphanesi içermektedir. Bu özelliklerden yararlanılarak, bu çalışmada oluşturulan veri kümesi WEKA kullanılarak sınıflandırılmış ve aynı zamanda birçok önışleme, nitelik seçimi ve sınıflandırma yöntemlerinin etkileri karşılaştırılmıştır.

Önışleme adımında, ilk olarak veri kümesindeki metinlerde yer alan tüm özel karakterler kaldırılmıştır. Tüm metinler, WEKA programı tarafından okunabilmesi için ARFF formatına dönüştürülmüştür. WEKA default olarak ARFF dosyalarında geçen Türkçe karakterleri okuyamayan bir yapıya sahiptir. Bunun önüne geçmek için, Motaz K. Saad (2010) tarafından geliştirilen ve Türkçe dilindeki kelimeleri okuyabilen “Arabic Light Stemmer” paketi WEKA programına dâhil edilmiş olup, sisteme eklenen ARFF verilerinin içindeki Türkçe’nin de dâhil

olduđu “utf-8” formatlı karakterler sorunsuzca okunabilmiştir. WEKA ile okunan bu bilgiler temel olarak pozitif ve negatif olmak üzere iki sınıf olarak etiketlenmiştir. WEKA yazılımında yer alan TF*IDF ağırlıklandırma, kök bulma ve minTermFreq filtreleri gibi birçok önışleme adımları, bu adımların sınıflandırma performansı üzerindeki etkilerini gözlemlemek için kullanılmıştır.

Önışleme adımından sonra veri kümesi, nitelik seçimi yöntemlerini uygulanarak öznelik uzayı küçültülmüştür. Nitelik seçimi kısmında, yaygın kullanılan iki yöntem olan ki-kare ve bilgi kazancı yöntemleri kullanılmış ve bu yöntemlerin siber zorbalık tespitindeki performansları karşılaştırılmıştır. Bu amaçla ki-kare nitelik seçimi yönteminin bilgi kazancı yöntemine göre daha başarılı sonuç verdiği tespit edilmiştir. Nitelik seçimi sonrasında ki-kare ya da bilgi kazancı değeri 0.001’in altında kalan nitelikler, nitelik kümesinden çıkartılmıştır.

Bu çalışmada uygulanan yöntemlerin başarısını test etmek için, k-katlamalı çapraz doğrulama, değerlendirme yöntemi olarak seçilmiştir. K değeri 10 olarak belirlenmiş ve veri kümesi uygun şekilde eğitim ve test veri setlerine bölünmüştür.

Sınıflandırma kısmında, Naive Bayes Multinomial, Destek Vektör Makinaları, Karar Ağaçları – J48, Rastgele Orman ve bu tez çalışmasında önerilen filtre tabanlı sınıflandırıcı kullanılmış ve karşılaştırılmıştır. Önerilen sınıflandırıcı hariç tüm sınıflandırıcılar WEKA programı yardımı ile gerçekleştirilmiştir.

Önerilen filtre tabanlı sınıflandırıcı için, veri kümesi ilk olarak Zemberek programı yardımı ile önışlemeden geçirilmiş ve yanlış yazılmış kelimeler düzeltilmiştir. Ardından düzeltilmiş kelimelerin kökleri alınmıştır. Veri kümesindeki cümleler kelimelere ayrılmıştır. Daha sonra pozitif cümlelerdeki kelimeler bir gruba, negatif cümlelerdeki kelimeler ise diđer gruba alınmıştır. Negatif kelime grubundan, pozitif cümlelerden elde edilen kelimeler çıkarılarak kötü kelimeler listesi elde edilmiştir. Sonraki aşamada; kelimelerin cümlelerde geçtiđi toplam tekrar sayısına bakılarak bir sıralama yapılmıştır. Bu sıralamada en yüksek tekrara sahip 5000 kelime; kötü kelime listesine dâhil edilmiştir. Sonuç olarak sınıflandırıcı çalıştıđında; eđer gelen cümle kötü kelime listesinden deneysel olarak belirlenmiş bir eşik değeri kadar veya daha fazla sayıda kötü kelime içeriyorsa bu yorum siber zorbalık metni olarak işaretlenip sınıflandırılmış, deđilse

pozitif olarak sınıflandırılmıştır. Eşik değeri 3 olarak alınmıştır. Tüm sınıflandırma performansları F-ölçeği ile ölçülmüştür. Performans değerlendirmeleri bu çalışma için seçilmiş olan önerleme, nitelik seçimi ve sınıflandırma yöntemleri için ayrı olarak değerlendirilmiştir.

Performans sonuçlarına bakıldığında, önerleme adımı için TF*IDF ağırlıklandırma yöntemi, ağırlıklandırma yöntemi olarak en iyi sonucu vermiştir. Kök bulma algoritmasının sınıflama üzerinde olumlu etkisi olduğu görülmüştür. Naive Bayes sınıflandırma, önerleme ve nitelik seçimi deneylerinde kullanılmış ve ki-kare algoritmasının bilgi kazancı algoritmasından küçük farkla daha başarılı olduğu görülmüştür. Sınıflandırma performanslarında hem modelleme süresi hem de sınıflandırma doğruluğu açısından en uygun sınıflandırıcının Naive Bayes Multinomial olduğu tespit edilmiştir. Sınıflandırma performansı karşılaştırmasına göre, en iyi performans Naive Bayes Multinomial algoritmasına ait olup, en kötü performansa sahip algoritma ise Rastgele Orman algoritması olmuştur. Bu tezde geliştirilen filtre tabanlı sınıflandırıcı, performans karşılaştırmasında ikinci sırada yer almıştır.

Sonuç olarak bu tezde, siber zorbalık metin tespiti çalışmaları için kullanılacak Türkçe dilinde en geniş kapsamlı veri kümesi oluşturulmuştur. Önerleme adımlarında TF*IDF ağırlıklandırma ve kelime kökü bulmanın sınıflama doğruluğu üzerinde olumlu etkilerinin olduğu görülmüştür. Nitelik seçim yöntemi olarak ki-kare nitelik seçim yönteminin bilgi kazancı yönteminden daha başarılı olduğu gözlenmiştir. Naive Bayes Multinomial, sınıflandırma süresi ve sınıflandırma doğruluğu açısından en uygun sınıflandırıcı olarak değerlendirilmiştir. Ayrıca, bu tezde önerilen sınıflandırıcının performans olarak diğer sınıflandırıcılara yakın olduğu görülmüş ve bu da veri kümesinin oldukça kapsamlı olduğunu göstermiştir. Gelecekteki çalışmalar için, hazırlanan Türkçe içerikli veri kümesini daha da geliştirmek ve daha fazla metin içeriği toplanması planlanmaktadır. Bu amaçla hazırlanan veri kümesinin, açık kaynak olarak internet ortamında paylaşılması hedeflenmektedir. İnternet ortamında açık kaynak olarak kullanılacak bu veri kümesi ile Türkçe metinlerde geçen siber zorbalık tespiti için yeni çalışmaların yapılmasına bir kaynak olması beklenmektedir.

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to head of Computer Engineering Department and my thesis supervisor Prof. Dr. Selma Ayşe ÖZEL, for her supervision guidance, encouragements, patience, motivation, constructive and useful suggestions. I am so grateful to her for all the things that I have learned under her supervision.

I would like to thank members of MSc thesis jury, Asst. Prof. Dr. Buse Melis ÖZYILDIRIM and Asst. Prof. Dr. Onur ÜLGEN, for their suggestions and corrections. I would also like to thank my other supervisor Asst. Prof. Dr. Esra SARAÇ EŞSİZ for her useful suggestions.

My sincere thanks go to my brother Serkan ÖZTÜRK for his support, patience, motivation, useful suggestions and his valuable time for this work. I'm grateful to him.

Last but not least, I would like to thank my family: my parents Mehmet, Perihan and my sister Anka Fatma for their endless support and encouragements for my life and career and also special thanks to my beloved fiancée Şahika Topcu, for her support and patience. I would also like to thank my dear friend Tahir Erşan Şanlı for his support.

CONTENTS	PAGE
ABSTRACT.....	I
ÖZ	II
EXTENDED ABSTRACT	III
GENİŞLETİLMİŞ ÖZET	VII
ACKNOWLEDGEMENTS	XI
CONTENTS.....	XII
LIST OF TABLES	XVI
LIST OF FIGURES	XVIII
1. INTRODUCTION	1
1.1. Cyberbullying Types.....	2
1.1.1. Denigration	3
1.1.2. Impersonation	3
1.1.3. Outing	4
1.1.4. Trickery.....	4
1.1.5. Exclusion	4
1.1.6. Cybertalking.....	4
1.1.7. Harassment.....	4
1.1.8. Flaming	5
1.2. Cyberbullying Tools	5
1.3. Negative Effects of Cyberbullying	6
1.4. Strategies to Cope with Cyberbullying	7
1.5. Machine Learning	8
1.6. Text Mining	9
1.7. Feature Selection.....	10
1.7.1. Wrapper Methods	10
1.7.2. Filter Method	10
1.7.3. Embedded Methods	11

1.8. Aim and Contribution of this Thesis.....	11
2. RELATED WORKS	13
3. MATERIALS AND METHODS.....	19
3.1. Dataset	19
3.1.1. Data Collection from Facebook, Twitter and Youtube.....	19
3.1.2. Data Collection from Instagram.....	20
3.2. WEKA (Waikato Environment for Knowledge Analysis) Tool.....	21
3.3. Methods	25
3.3.1. Preprocessing for Useless Character Removal	26
3.3.2. ARFF (Attribute Relationship File Format)	27
3.3.3. Conversion of Data to ARFF Format.....	30
3.4. Preprocessing Operations with WEKA	32
3.4.1. Main Filtering Properties	33
3.4.2. Separation of Sentences into Words, Weighting and Stemming	35
3.5. Feature Selection.....	38
3.5.1. Information Gain (IG).....	38
3.5.2. Chi-Square (CHI2) Method	40
3.6. Evaluation Method (K-Fold Cross Validation).....	41
3.7. Classification	42
3.7.1. Naive Bayes Classifier.....	43
3.7.2. Support Vector Machines	45
3.7.3. Decision Trees – J48.....	47
3.7.4. Random Forest.....	48
3.7.5. The Proposed Classifier	49
3.7.6. Classification Performance Metric	50
4. RESULTS AND DISCUSSIONS.....	53
4.1. Results and Comparison of Preprocessing Steps	54
4.1.1. Effect of TF*IDF Weighting	54
4.1.2. Effect of Stemming Function.....	55

4.1.3. The Effect of Stopwords Removal.....	56
4.1.4. The Performance Comparison of Preprocessing Steps.....	58
4.2. Results and Comparison of Feature Selection Functions.....	59
4.3. The Results of the Classification Methods.....	60
4.3.1. The Results of Naïve Bayes Classifier.....	61
4.3.2. The Results of the Support Vector Machine Classifier.....	62
4.3.3. The Results of the Decision Tree-J48.....	63
4.3.4. The Results of Random Forest Classifier.....	64
4.3.5. Comparison of Classification Results and Discussion.....	65
4.3.6. The Proposed Classifier and Results.....	68
5. CONCLUSION.....	71
REFERENCES.....	73
CURRICULUM VITAE.....	79
APPENDIX.....	80



LIST OF TABLES	PAGE
Table 2.1. Comparison of The Methods Used in This Thesis with the Previous Studies.....	18
Table 4.1. The General Information About the Dataset Collected	53
Table 4.2. Comparison of TF*IDF Methods for All Dataset (with F Measure Value).....	55
Table 4.3. The Effect of Stemmer Function on the Dataset.....	56
Table 4.4. The Effect of Deleting Stopwords	57
Table 4.5. The Effects of Stopwords Removal on Negative and Positive Labeled Comments	58
Table 4.6. Performance Comparison of the Preprocessing Steps	59
Table 4.7. The Effect of Feature Selection Algorithms on Classification	60
Table 4.8. Results of the Naive Bayes Classifier.....	61
Table 4.9. Results of the Support Vector Machine Classifier.....	62
Table 4.10. Results of the J48 Classifier.....	63
Table 4.11. Results of Random Forest Classifier	65
Table 4.12. Results of the Proposed Classifier	68



LIST OF FIGURES	PAGE
Figure 3.1. The Initial Graphical User Interface of Weka	22
Figure 3.2. Explorer Environment of Weka.....	23
Figure 3.3. Classify Panel of Weka.....	24
Figure 3.4. Main Steps and Methods Used in Classification	26
Figure 3.5. Dataset Before Any Character Removal.....	27
Figure 3.6. The Same Dataset After Useless Characters Removal	27
Figure 3.7. Arff Header Section Example.....	28
Figure 3.8. Data Section Example.....	29
Figure 3.9. Dataset Example Before Converting to ARFF File.....	30
Figure 3.10. Example of Attribute Information	31
Figure 3.11. Positive Labeled Data Example.....	32
Figure 3.12. Negative Labeled Data Example	32
Figure 3.13. WEKA Filter Window	34
Figure 3.14. WEKA StringToWordVector Window	35
Figure 3.15. K-Fold Cross Validation Method	42
Figure 3.16. SVM Classifier	45
Figure 4.1. Comparison of Precision, Recall and F-Measure Values	66
Figure 4.2. Accuracy Comparison of Classifier Results	67
Figure 4.3. Time Taken to Build Model	67
Figure 4.4. Comparison of Accuracies of Classifiers	69
Figure 4.5. Time Taken to Build Model	70



1. INTRODUCTION

In our age, technology is used in almost all areas of our lives. It can be said that, the daily life is affected highly positive with the introduction of the internet and computer into our lives. The usage of computer and internet together has been made easier a person's communication, transaction, quick access to information, and many other activities. Emerging technologies and electronic communication networks bring new problems as well as facilitating human life. At the top of these problems is the cyberbullying, which threatens young people and their families.

Cyberbullying can be defined as the harmful behavior in a technical or relational way against a private or legal person, using information and communication technologies (Arıcak, 2011). It involves relational assault behaviors such as constantly harassing people (cyber-stalking) by using information and communication technologies, mocking people, threatening them, spreading gossip, and insulting people over the internet.

This concept has begun to draw attention of the researchers since the 2000s and has been named in various ways by focusing on their different features by the researchers. According to this, the cyberbullying has been mentioned with various names in the literature as electronic bullying, online bullying, internet bullying, digital bullying, and online harming (Kowalksi and Limber, 2007).

Cyberbullying is the intentional negative behaviors made occasionally or several times against to a victim who cannot protect himself, by an individual or a group who are using information and communication technology (Olweus, 1993). Belsey (2017) has defined the cyberbullying as a repetitive behavior disorder with the aim of harming. On the other hand, Arıcak (2009) has collected all the acts under the same heading of cyberbullying such as anonymous calls, identity concealed sent insult/threat, sent e-mails, texts, videos or images to denigrate an individual or a group, published video or image, and infected e-mails. Hinduja and Patchin (2009) have described cyberbullying as a repetitive act of intentional

harming by using the computer, cell phones, and other technological tools. Mason (2008) has defined the cyberbullying as a behavioral disorder which involves repeated or intentional harassment or threats to an individual or group, and included verbal violence statements or their records.

Cyberbullying is an electronic form of peer harassment. The issue of cyberbullying, which is a new issue for Turkey and even many European countries, has been investigated by other specialists, especially education scientists and psychologists in USA and Canada for the last 10 years. In order to prevent the threats of cyberbullying, a series of national and international child protection initiatives such as The Suicide Prevention Center and Child Focus have been initiated. Despite this efforts, many messages and unsolicited threats remain online (Nahar et al. 2012, 2013). Parents cannot provide their children with full control even if they try to take some precautions to protect their children from electronic communication tools, especially the Internet, and as a result, cyberbullying leads to many personal problems. These problems cause the person to exhibit psychosocial problems, academic problems and aggressive behavior. The cyberbullying messages that are frequently encountered in social sharing platforms which have developed in the last years such as Facebook, Twitter, Youtube, Instagram, Tumblr and WhatsApp are deeply affecting the psychology of individuals. At this point, detecting and filtering the textual items that cause the cyberbullying in the electronic environment and taking precautions before reaching to the individual will be an effective method to find a main solution to the problem.

1.1. Cyberbullying Types

Through the information and communication technology cyberbullying events in intentional form may occur occasionally or continually. Cyberbullying can be realizable in different ways either sharing content, communicating or sending insulting electronic messages. It is observed that various classifications are done about this topic in the literature. However, one of the most accepted

classification method was realized by Willard (2006). According to this, different kinds of cyberbullying events are collected in 8 different categories (Willard, 2006) these are;

- Denigration
- Impersonation
- Outing
- Trickery
- Exclusion
- Cybertalking
- Harassment
- Flaming

1.1.1. Denigration

It can be specified as one of the most common types of cyberbullying. It usually occurs as a result of the use of communication styles of problematic individuals at adolescent ages. It is defined as sharing false news or sending electronic messages about a person or a group. Especially in the last period, it is a frequently encountered method. Main cause of this case is stated as the increasing social media usage among the young people. This type of cyberbullying can be used by students against to school teachers (Mason, 2006).

1.1.2. Impersonation

It is seen mostly on social media platforms such as Facebook, Twitter and Instagram. The cyberbully can make sharing by acting as a victim of cybercrime by creating a fake account of the person he is going to harm. In this way, the cyberbully can leave the victim in a difficult situation and send inappropriate messages to other people.

1.1.3. Outing

It is the type of cyberbullying that is carried out online by sharing the information and materials that will harm the victim.

1.1.4. Trickery

It has some common features with Outing. However, unlike the Outing, there is a situation of gaining trust and manipulating of that trust. A person trusts someone he meets on the internet and shares the information or images he may be ashamed of when revealed. The person that he trusts shares confidential information of the victim by abusing his trust.

1.1.5. Exclusion

This type of cyberbullying is defined as preclusion, being unwanted, and exclusion of the victim from the social media platforms, forums, electronic message sending groups and online gaming. The aim is to make the victim feel bad by excluding and isolating him from the relevant places.

1.1.6. Cybertalking

It is the type of cyberbullying that makes the victim frightened by humiliating the victim with the electronic messages including insulting materials, or sound/video records, and images. The cyberbully frightens the victim by saying that he will hurt, kill or beat him by demanding the address information.

1.1.7. Harassment

Threat and harassment are similar cases. In harassment, the cyberbullying is described as swearing, sending obscene video image or insulting text messages. The difference of the harassment is the persistence of this bullying.

1.1.8. Flaming

It is the conversation made with the victim in an angry, nervous, insulting and irritable way in the online environment. It is the suppression of the victim by swearing or threatening when discussing.

1.2. Cyberbullying Tools

One of the focus points of the researchers trying to explain the nature of cyberbullying is to determine which tools are being used for this negative behavior to be carried out. The most common tools that are used for cyberbullying are listed as follows (Shariff and Gouin, 2005):

- E-mail,
- Discussion groups,
- Mobile phones or web cams,
- SMS or instant messaging tools,
- Social networking sites,
- Chat rooms,
- Blogs,
- Video clips,
- MUDs (MUDs are virtual environments that enable individuals to get different identities).

Especially after the year 2010, blogs and social networking sites that created more free space on the internet have become cyberbullying materials. These virtual environments mean free usage area for everyone. Individuals can create and share blogs in these environments without any restrictions. On the other hand, these environments can be used to embarrass, mock and attack to other groups. According to a research study made by Bahat (2008) cyberbullies publish

the comments of the victim related with their appearance, intelligence, health and sexual orientation in the blogs. In another study, Slonje and Smith (2008) reported that the most common known type of cyberbullying which is around 46% among the participants is the cyberbullying carried out by comments, videos and pictures, it is followed by telephone calls with 37%, and then by text messages with 29%.

1.3. Negative Effects of Cyberbullying

Individuals exposed to cyberbullying are adversely affected by these events. While the cyberbullies who exhibit cyberbullying behaviors think that they do not do any harm to the people, the victims that exposed to these cyberbullying acts can end their lives due to the negative impact of these events. Although every individual does not end their lives as a result of these negative events, it can cause negative consequences in some individuals. It is seen on the conducted studies that the cyberbullying is a common problem. In this kind of bullying, although there is no physical contact between the bully and the victim, there are psychological symptoms such as low self-esteem, sadness, disappointment, school fear, academic failure, loneliness, anxiety, depression and suicide (Bargh and Mckenna, 2004).

It has been stated that cyberbullying behaviors negatively affect the social communication of the victims, make it difficult to adapt to the social environment, cause them to have difficulty in establishing friendships, reduce their self-esteem and make them feel worthless, have difficulties in communicating with their classmates, and make them feel excluded and helpless (Hinduja and Parchin, 2008). 58% of the cyberbullying victims experienced depressive feelings and stated that if long-term cyberbullying behaviors persisted, the victims would feel themselves worthless (Ybarra and Mitchell, 2007).

Some individuals who exhibits cyberbullying behavior can create fake personalities on the internet by hiding their real identity and these fake personalities can be the exact opposite of the individual's self-personality (Sayar,

2006). Men can act as a woman; women can act as a man and an introvert person can act in the opposite way of its personality.

1.4. Strategies to Cope with Cyberbullying

Strategies to cope with cyberbullying can be expressed as the measures and actions taken by an individual when exposed to cyberbullying. There are methods to be used to cope with the cyberbullying by the individual, his friends and his family who are exposed to the cyberbullying. The prominent one among these strategies is the personal coping strategies for the cyberbullying exposed individual.

For individuals exposed to cyberbullying, strategies to cope with individual cyberbullying can be examined in three topics (Parris et al., 2012) that are actively reacting, preventing, and ignoring. Actively reaction issue can be addressed in four ways as avoidance, acceptance, justification, and social support search. Strategies to cope with preventive cyberbullying include speaking and raising awareness with a person (Akbaba and Şahin, 2018).

Studies investigating strategies to cope with cyberbullying include different findings. It is generally divided into four subjects. These are; search for help, avoidance, ignoring, cognitive security, and privacy (Koç et al., 2016).

In their help seeking behaviors the individuals generally get help from their families, friends, other adults and teachers, and especially from security forces.

Three main factors are prominent for the information search point. These are; obtaining technical information, confrontation with bully, retaliate to bully, or threaten the bully. Whichever behaviour the individual chooses; he can try to manage the process by searching the unknown information related to these behaviors. The individuals who internalize the cyberbullying instead of coping with it, may develop information search behaviour for the psychological negativities such as anxiety, stress, and depression which are the results of cyberbullying. Those who express their reactions instead of internalizing the

process, may seek information about this because they are more aggressive, or use verbal or physical violence.

On the other hand, in terms of increasing cognitive security and confidentiality, individuals can try to obtain information that will increase knowledge, skills and awareness about technological tools and internet. Here, they try to make the privacy and security features of the internet and internet related equipment's more functional.

Avoidance and ignorance can be seen as a way of coping with the cyberbullying. In this process, the individual is insensitive to the cyberbullying events. To avoid from cyberbullying, machine learning approaches have been used to automatically detect cyberbullying in the contents sent to the victim and then these contents can be blocked automatically before they reach to the victim.

1.5. Machine Learning

Machine learning is the common name of computer algorithms that model a given problem according to the data obtained from the environment of the problem. Machine learning allows the computer to learn the experiences gained from previous examples. Therefore, this event can be described as learning from experience (Öztemel, 2003). Many approaches and algorithms have been proposed about this topic. Some of these approaches have the capability of prediction and estimation, and some have classification. There is a direct relationship between machine learning and data mining. Application of machine learning techniques to large databases is data mining. This feature can be used to classify text or documents.

Machine learning algorithms are divided into two main categories as supervised and unsupervised learning. In the case of supervised learning, it can be mentioned that an expert can provide information to the system. Classifiers such as Naive Bayes, decision trees, random forest, neural networks, and nearest neighbor algorithms are examples of this type of learning. Unsupervised learning is the

access to information as a result of self-training of the system. In this learning, the target variable is unknown or only a very limited number of target data is recorded. Clustering algorithms are examples of this type of learning.

Automatic cyberbullying detection from text content can be done by applying machine learning techniques. Especially text mining is used for this purpose.

1.6. Text Mining

Text mining studies have gained importance in recent years due to the increasing number of sources and electronic documents. These documents usually contain unstructured or semi-structured information. Text mining is a data mining study that considers the text as a data source. In other words, it aims to obtain structured data over the text (Şeker, 2014). The main purpose of text mining is to enable users to get information from text sources and to automatically classify and discover different types of documents using various algorithms (Korde and Mahender, 2012).

Text mining studies often work together with natural language processing, which is another area of study in text-related literature. Natural language processing mainly involves studies based on linguistics knowledge under artificial intelligence. On the other hand, text mining studies aim to reach to the results statistically over the text.

Text documents are a set of terms that are difficult to interpret by a classifier. Therefore, unstructured text data must be converted to a form that the machine can understand (Aggarwal and Zhai, 2012). Text classification was first used as automatic text indexing systems in the 1970s (Salton, 1968). It was later developed with the help of machine learning systems. At this point, the basic problem of text classification is that the set of features in the documents is of very high dimensions (Zhu et al., 2007). In order to reduce these high dimensions and improve the performance of the classifier used, it is necessary to select the

appropriate sub-sets from the high-dimensional feature set. There are several approaches to choose appropriate features (Yu et al., 1999). These methods include: Document Frequency, Information Gain, Mutual Information and chi-square tests, etc. (Yang and Pedersen, 1997).

1.7. Feature Selection

In data mining, feature selection is the name given to piecemeal evaluation in order to determine which features are more effective on the results in the dataset. From this point of view, the feature selection process is a feature size reduction task. Accordingly, a complex data is reduced to a simpler form by reducing its dimensions (Şeker, 2008).

The selection of features can be made in 3 different ways according to the search size. These are: wrapper methods, filter methods, and embedded methods (Guyon and Elisseeff, 2003).

1.7.1. Wrapper Methods

The wrapper methods use a model that scores feature subsets in feature selection. These subsets express different combinations of attributes. Each new subset created, in other words combinations, is used to train the tested models. At this stage, an error rate is given to each model tested. Because each created subset trains a different model, it contains intensive calculations and is slow. As the search methods, some algorithms such as best first search, heuristic methods, and back and forth transitions to add and remove features can be used. The recursive feature elimination (RFE) algorithm is an example of wrapper method.

1.7.2. Filter Method

The filter method performs a statistical calculation instead of the error rate scoring in subsets that occur in the feature selections. As a result of this calculation,

features are scored and ranked. After this process, the subset of features is preserved or deleted according to a specified score threshold. The filter method is a rapid feature selection method, but it can fail if the determined threshold value is not chosen properly.

1.7.3. Embedded Methods

Embedded methods operate by learning the best features that will contribute to that model in model selection. Unlike filter and wrapper methods, the learning section and feature selection section cannot be separated in embedded methods - the structure of the considered model class plays an important role. Examples of embedded feature selection algorithms are LASSO, Elastic Net, and Ridge Regression.

1.8. Aim and Contribution of this Thesis

In this thesis it is aimed to make one of the pioneer works in this field in order to determine and filter cyberbullying texts in Turkish language. To reach this goal, a dataset compiled from Turkish texts has been created that can be used by everyone. The dataset has been collected from 4 major social networks that are Facebook, Youtube, Instagram, and Twitter, without depending on to a single source. It is aimed to maximize efficiency in machine learning with this comprehensive dataset. The collected cyberbullying dataset which has Turkish text contents is the most extensive dataset that can be used by everyone in this field. After collecting the dataset, our second aim is to show the effectiveness of this dataset by applying traditional text classification processes to detect cyberbullying for Turkish texts. Then, we try to develop a filter-based classifier to detect cyberbullying for text messages written in Turkish.

The main contributions of this thesis study are the dataset collected, and the filter-based classifier proposed. As there are only few studies on Turkish to detect cyberbullying, the dataset prepared in this study will help researchers to

develop more efficient and effective methods to detect and prevent from cyberbullying.



2. RELATED WORKS

Most of the previous studies made for automatically detecting cyberbullying belongs to English language. There are only a few studies done for Turkish. In this section we first summarize the studies for English, after that we give details of studies for Turkish and compare with this thesis.

The first study on automatic detection of cyberbullying belongs to Kontostathis et al. (2009) who have developed the first dataset by downloading 288 chat logs that were available from the Perverted Justice (PJ) website as of August 2008. They used this dataset to categorize internet predators. The dataset used in this study belongs to the English language. They have performed two sets of text mining experiments using this dataset. The first experiment attempts to categorize communication strategies; and tries to distinguish between predator and victim, or predatory and normal chat. In the second experiment, clustering is used to determine whether or not different communicative strategies are used for luring children.

Yin et al. (2009) use Kongregate, Slashdot and MySpace web site posts as the dataset to detect harassment. Dataset contains 1,946 posts in total, and it belongs to the English language. The collected dataset has been used for detecting harassment and classical text classification methods are applied by representing terms with TF*IDF weighting. They employ libSVM algorithm for classification.

Kontostathis et al. (2010) in their second study, take their corpus as a collection of posts from Formspring.me. They used this dataset to detect cyberbullying. Queries are expanded with bullying terms. Each post is labelled by Amazon Mechanical Turk. Kontostathis et al. (2010) used Latent Semantic Indexing and Singular Value Decomposition to find bullying terms. They achieved a success rate of 91.25% for cyberbully detection.

Chen et al. (2011) proposed a lexical semantic approach (LSA) to predict online user's offensiveness levels. They consider the typing characteristic of the

users to detect potential offensive users. According to the experimental results, it is observed that using LSA framework improves performance with respect to the existing studies in this area. They achieved an average precision of 98.24% for offensive sentence detection. They also achieved 77.9% precision for offensive user detection.

Dinakar et al. (2011) used a corpus of 4500 YouTube comments, applying a range of binary and multiclass classifiers to detect textual cyberbullying. They manually labeled YouTube comments and use Naive Bayes, Rule-based JRip, Tree-based J48, and SVM algorithms to classify documents. JRip gives the best performance in terms of accuracy, whereas SVM is the most reliable classifier as measured by the kappa statistic, and 66.7% accuracy is achieved for detecting cyberbullying. Their study shows that building binary classifiers are more effective than multiclass classifiers at detecting such sensitive messages.

Reynolds et al. (2011), have proposed to use NUM and NORM features for cyberbully detection. These features are devised by assigning a severity level to the bad words obtained from nosewaring.com Web site. NUM is a count, and NORM is a normalization of the bad word, respectively. They used C4.5 classifiers and an instance base learner, from Weka data mining tool for text classification. Positive examples are replicated up to ten times to balance the dataset, and accuracy of the classifiers are reported. Their findings showed that the C4.5 decision tree and an instance based learner are able to identify the true positives with 78.5% accuracy.

Sanchez and Kumar (2011) used Twitter comments to detect cyberbullying with Naïve Bayes classifier. A gender specific bullying detection on twitter dataset is performed for English language. 67.3% accuracy values with Naïve Bayes classifier are obtained.

Dadvar et al. (2012), demonstrated that taking gender-specific language features are preferred and users are categorized into male and female groups. YouTube comments are used as the dataset and SVM is applied as the classifier.

This study showed that when user based context is taken into account, the classification accuracy increases.

Xu et al. (2012), identified several problems in using social media to study bullying and formulated them as familiar NLP tasks. Their study describes seven frequent emotions, some of which have been previously well-studied, and some are non-standard in bullying. Twitter dataset is used to detect cyberbully posts with SVM classifier. The overall success of this experiment reaches to 85% accuracy.

Dadvar et al. (2013) used a multi-criteria evaluation system to obtain a better understanding of YouTube users' behavior and their characteristics through expert knowledge. Scores are assigned to all users, which are given by the system, based on their previous activities. These scores show their cyberbully level. It is found that the scores are helpful to decide if a user is bullying or not. The scores can be used to discriminate among users with a bullying history and those who have not engaged in hurtful acts and helpful to decide if a user is bullying or not.

Munezero et al. (2013) used individual words as features without any additional syntactic or semantic knowledge. They used a public dataset for harmful language detection. Their study achieves high accuracy using Naïve Bayes Multinomial and SMO classifiers from Weka.

Nahar et al. (2012) proposed an effective approach to detect cyberbullying messages from social media through a weighting scheme of feature selection. They presented a graph model to show most active cyberbullying predators and victims through ranking algorithms. They used Kongregate, Slashdot and MySpace web site posts as the dataset and weighted TF*IDF term weighting. They used LibSVM for classification and obtained 0.31 and 0.92 F- measure values for baseline and weighted TF*IDF approaches. Until 2016, similar studies were made to these studies.

Saraç (2016) showed the effects of feature extraction, feature selection and classifier used on the performance of cyberbully detection. She proposes a new feature selection method based on Ant Colony Optimization and Chi-Square

statistic. Formspring.me, MySpace, YouTube, Twitter, Web blog comments are used for dataset in this study. The results of this study proved that, Ant Colony Optimization is an acceptable optimization algorithm for feature selection to detect cyberbullying, and applying feature selection reduces the number of features to be used during the classification process and improves runtime and classification performance.

All related studies summarized above were conducted for English language. The first study which makes cyberbully detection on Turkish texts was published by Özel et al. (2017). They prepared a small dataset having 900 comments from Instagram and Twitter messages written in Turkish and then applied machine learning techniques that are Support Vector Machines, Decision Tree (C4.5), Naïve Bayes Multinomial, and k Nearest Neighbors classifiers to detect cyberbullying. The study results show that Naïve Bayes Multinomial classifier is the most successful one in terms of both classification accuracy and running time. Also, they used information gain and chi-square feature selection methods. When these feature selection methods are applied, classification accuracy improves up to 84% for the dataset used.

Bozyiğit et al. (2018) aimed to detect Turkish cyberbullying messages on social media. In this direction, a dataset is created and published on the internet, since there is no publicly available dataset for Turkish cyberbullying contents. Dataset were collected from Twitter messages with an application and contains 3000 messages. This study shows that Naïve Bayes Multinomial and Support Vector Machines are the most successful classifiers for detection of Turkish cyberbullying contents. The observed classification F-measure scores are between 0.86 and 0.91. In addition, C4.5, bagging and random forest methods have poor performance in terms of running time.

The dataset used in this thesis contains 15658 Turkish text messages and has more content than the previous ones. In addition, the dataset in this thesis was collected not only from Twitter but also from Facebook, Youtube and Instagram.

Therefore, it is not social media dependent dataset, and it is the most comprehensive dataset in Turkish language for detecting cyberbullying. In this study, classifiers such as Naive Bayes Multinomial, Support Vector Machines, Decision Trees - J48 and Random Forest are tested, and the Naive Bayes Multinomial classifier is found to be the most successful algorithm as in the previous studies that are Özel et al. (2017) and Bozyiğit (2018). Apart from these classifiers, a filter-based classifier is proposed in this thesis to show the effectiveness of the developed dataset.



Table 2.1. Comparison of The Methods Used in This Thesis with the Previous Studies

	Tokenization	Stop Words	Stemming	Classifier	Feature Sel.	TF*IDF
Kontostathis et al. (2009)	✓			C4.5	✓	✓
Yin et al. (2009)				SVM		✓
Chen et al. (2011)	✓			NB, SVM		✓
Dinakar et al. (2011)		✓	✓	C4.5, JRip, NB, SVM		
Reynolds et al. (2011)				C4.5, JRip, kNN, SVM		
Sanchez and Kumar (2011)		✓		NB		✓
Dadvar and Jong (2012)				SVM		
Xu et al. (2012)	✓	✓	✓	SVM, NB, MaxEnt		✓
Dadvar et al. (2013)	✓	✓	✓	SVM		
Munezero et al. (2013)	✓	✓	✓	C4.5, NBM, SVM		
Nahar et al. (2012)	✓			SVM	✓	✓
Saraç (2016)	✓	✓	✓	C4.5, kNN, NBM, SVM	✓	✓
Özel et al. (2017)	✓	✓	✓	J48, NBM, IBk, SVM	✓	✓
Bozyiğit et al. (2018)	✓			C4.5, NBM, K-NN, RF		✓
Erhan Öztürk (2019)	✓	✓	✓	NBM, SVM, j48, RF, Proposed	✓	✓

3. MATERIALS AND METHODS

The first step of this thesis study is to prepare a dataset in Turkish language for automatic detection of cyberbullying owing to the absence of an open source large enough dataset in Turkish language. In this section, it is explained in detail that how the dataset is prepared, which stages are passed, and which methods are used to arrange it.

3.1. Dataset

The dataset used in this study has been collected manually from four social media platforms that are Facebook, Twitter, Youtube, and Instagram by browsing the four websites manually or automatically. To collect the dataset, Facebook, Twitter, Youtube and Instagram social media platforms are crawled, and text messages written in Turkish that contains cyberbullying content are stored. While these comments are collected, the texts containing cyberbullying expressions such as insult, swearing, and sexual assaults are taken manually from Facebook, Twitter and Youtube websites, text content from Instagram are collected automatically from the website by using an implemented javascript code. All collected text contents are manually checked and labeled as positive or negative such that if a text content has cyberbullying then it is labeled as negative; otherwise it is labeled as positive text. A total of 15658 Turkish text content comments have been obtained, of which 7995 text messages are positive and 7663 text messages are negative. So, we have almost a class balanced dataset.

3.1.1. Data Collection from Facebook, Twitter and Youtube

When collecting datasets from these platforms, positive or negative comments made for public sharing are manually selected and processed. While selecting these texts, hate speech, sexual assault, and insults made under public

sharing have been accepted as cyberbullying texts, and they are evaluated and labeled as “negative” in the dataset. Apart from these, the texts which are as different as from general comments have been evaluated and labeled as “positive”. As a result, a total of 6229 Turkish messages were taken manually from the Facebook, Twitter and Youtube websites. While 3102 of these messages were evaluated and labeled as positive, the remaining 3127 messages were labeled negative, assuming they contain cyberbullying texts. This data is stored in an excel table for further processing. The preprocessing steps of the dataset will be discussed later.

3.1.2. Data Collection from Instagram

Instagram is one of the applications that allow free sharing of photos and videos on social media. While it was a small application that users shared their photos when it was founded in October 2010, it has now become one of the most visited social media applications in the world. In this social network with 800 million active users per day, cyberbullying texts are frequently encountered due to the comments made by the users by using the user names they created (<https://www.wikiwand.com/en/Instagram>). This platform was mainly used in this thesis to create the dataset which will be used to determine the cyberbullying from texts written in Turkish language.

Manual data collection from Instagram causes many different problems. Some of these problems which avoid a proper data acquisition are copying of small text snippets, user names, and unrecognized characters. Also, since the Instagram limits the comment preview area, reviewing and copying all comments creates an additional cost in terms of time required. To avoid the aforementioned problems, a javascript code was written to gather the comments from the website automatically. In total, 9429 comments were determined to be included into the dataset, and 4536 of them that contains cyberbullying texts are labeled as negative, and the remaining

4893 text contents are labeled as positive. The labeled text contents are then recorded in an excel table for the preprocessing.

3.2. WEKA (Waikato Environment for Knowledge Analysis) Tool

WEKA (<https://www.cs.waikato.ac.nz/ml/weka>) is the name of one of the packages used in machine learning, which is one of the important subjects of computer science. It was developed as open source in JAVA language at Waikato University and distributed under GPL license. The name comes from the initials of the words Waikato Environment for Knowledge Analysis.

WEKA reads data from a simple arff file and agrees that these data consist of numerical or nominal values. At the same time, it can take and process the data through the database. There are many libraries available on WEKA for machine learning and statistics. Some of these are, data preprocessing, regression, classification, clustering, feature selection or feature extraction. There are also visualization tools that allow the output of these processes to be displayed.

The initial graphical user interface of Weka is shown in Figure 3.1. It has five different operating modes: Explorer, Experimenter, KnowledgeFlow, Workbench, and Simple CLI.

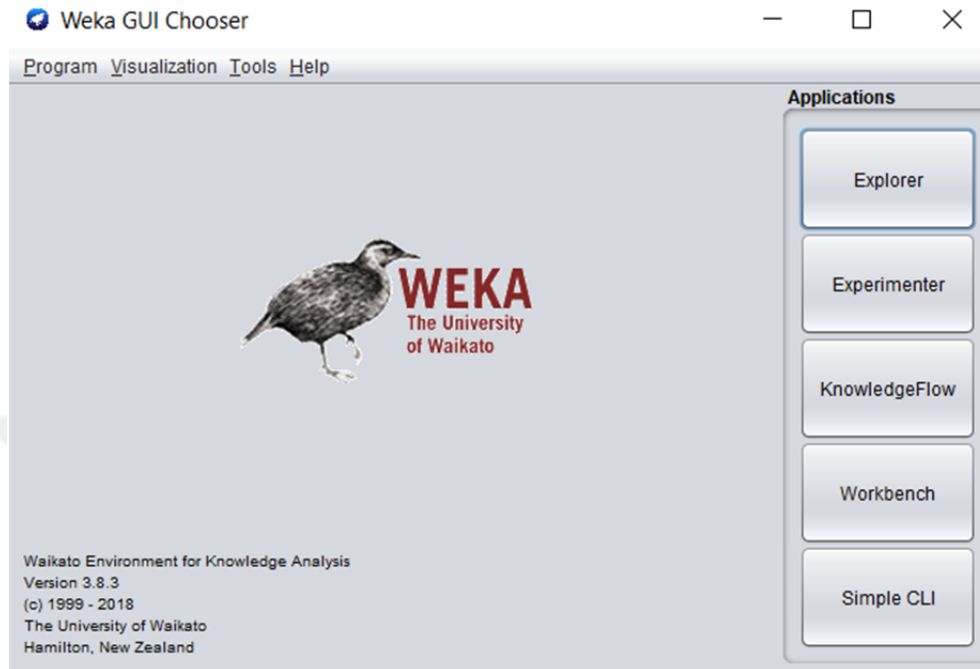


Figure 3.1. The Initial Graphical User Interface of Weka

Explorer is the most widely used media of WEKA. This interface is shown in Figure 3.2. Many stages of this study have been done by using this environment, and they are explained in detail in the below paragraphs. Experimenter is an environment for conducting experiments and statistical tests between learning schemes. KnowledgeFlow is a Java-Beans based interface for tuning and machine learning experiments and also drag-and-drop interface of the Experimenter. The Weka Workbench is an environment that combines all of the GUI interfaces into a single interface. It is useful if you find yourself jumping a lot between two or more different interfaces, such as between the Explorer and the Experimenter environment. Simple CLI provides a simple command-line interface that allows users to run Weka commands directly from the operating system.

Explorer is the most popular WEKA environment. Under this environment, many operations such as Preprocess, Classify, Clustering, Associate, Attributes Selection, and Visualize can be done as shown in Figure 3.2.

The Preprocess Panel is where the preprocessing is performed. In this panel, datasets can be loaded and preprocessed by using the filters in WEKA. The data is processed as a .arff file in this field. Stemming, stopwords removal, TF*IDF weighting, and lowercase conversion preprocessing steps are done at this panel.

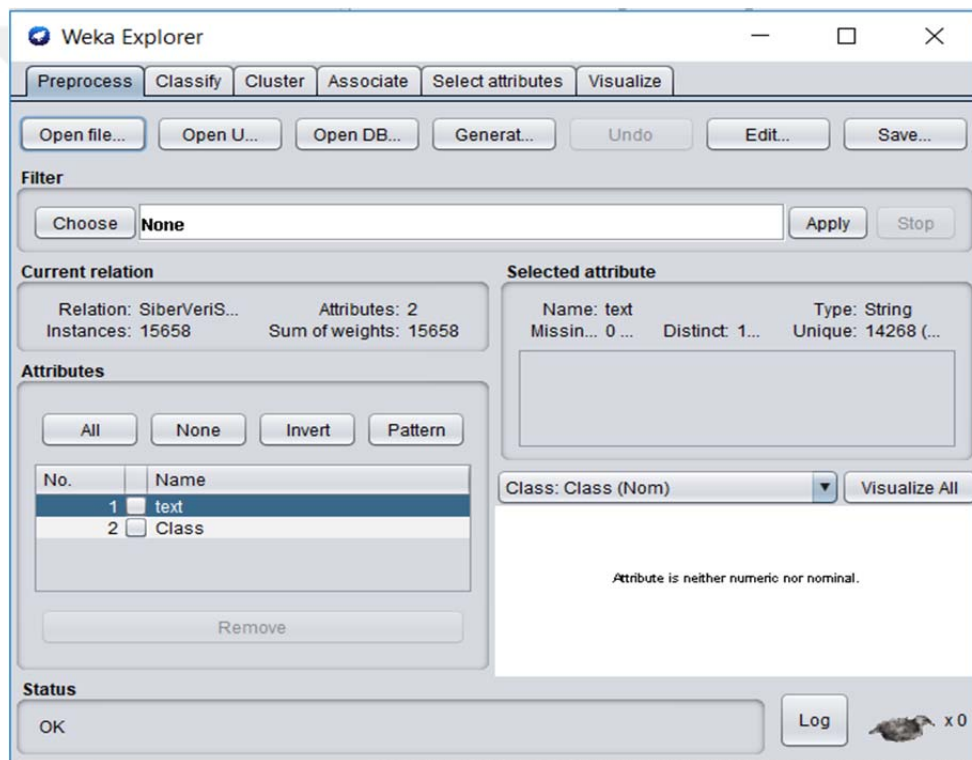


Figure 3.2. Explorer Environment of Weka

The Classify panel (see Figure 3.3) is the panel on which the classification is made on the existing dataset using any of the classification algorithms installed in WEKA. The 71 algorithms available in Classify tab of Weka are grouped into 6 categories, namely, Bayes (Bayesian classifiers such as Bayes Net, Naïve Bayes,

Naïve Bayes Multinomial, etc.), Functions (function algorithms such as Logistic, SimpleLogistic, LibLINEAR, LibSVM, RBFNetwork, SMO, etc.), Lazy (lazy algorithms or instance based learners such as IB1, IBk, KStar, etc.), Meta (algorithms that combine several models and in some cases models from different algorithms such as AdaBoostM1, Dagging, Bagging, etc.), Trees (classification/regression tree algorithms such as J48, BFTree, ADTree, etc.) and Rules (rule based algorithms such as JRip, OneR, ZeroR, etc.) (Saraç, 2016). It is also possible to use separate sets for testing and validation on this screen. Classification errors are displayed on a separate screen, and if the classification algorithm creates a decision tree, it is also displayed on a separate screen.

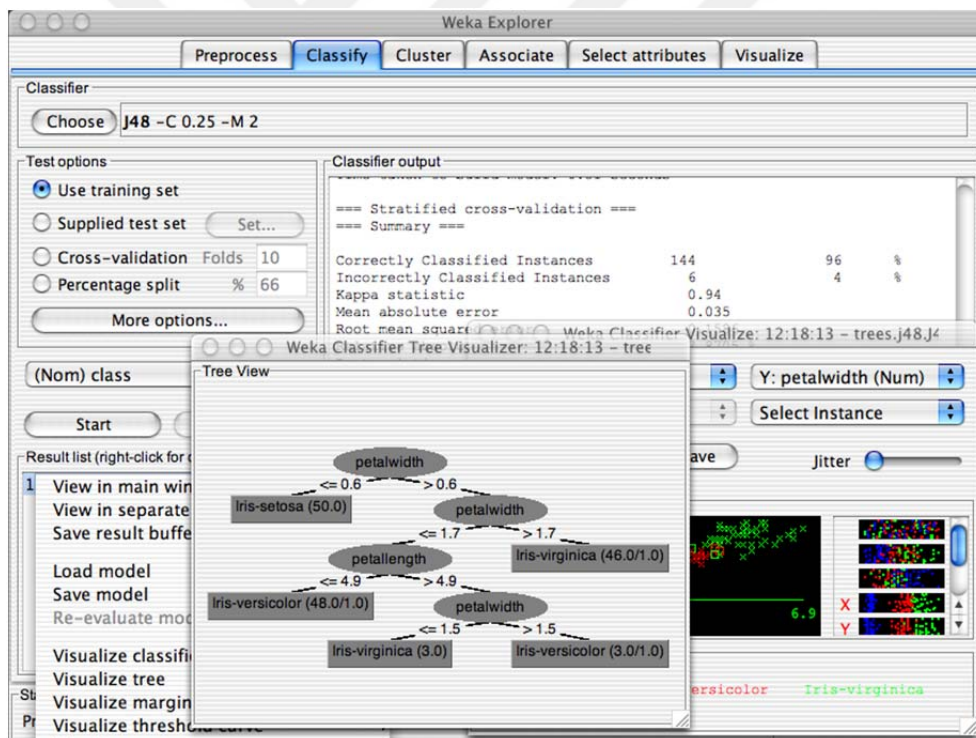


Figure 3.3. Classify Panel of Weka

Cluster Panel, similar to classify panel, is used for grouping the data objects and has a visualization interface. The Associate Panel enables association rule mining to be performed on the selected dataset. Select attributes tab includes attribute selection methods. Finally, with Visualize tab, 2D plot of the dataset can be viewed.

Select Attributes Panel is used to set the selection and processing properties of the dataset. If one of the selection schemes transforms the data, the transformed data can be seen in the visualization screen.

The Visualize panel can show a drawing over the dataset. The dimensions of the cells and points can be adjusted from the panel at the bottom of the screen. From the selection properties screen, the number of cells on the matrix can be changed. In addition, when working with very large datasets, it is also possible to use only the lower sample space for ease of operation.

3.3. Methods

In this section, the preprocessing steps applied to the dataset and the methods used for feature selection and classification are explained in detail.

The applied method in this thesis study consists of 3 main steps: Preprocessing, Feature Selection, and Classification. The representative flow chart of the methods applied is given in Figure 3.4. The datasets used in this study were collected manually as described in sections 3.1.1, 3.1.2, and 3.1.3. The data collected in the excel table is passed through a preparation phase for WEKA and converted into a format that WEKA can use. After this step, the data is processed by applying various weighting methods for features and then, the feature selection step begins. In order to determine the best feature subsets of the datasets in the feature selection step, the results are compared with each other using Chi-square (Chi²), and Information Gain (IG) feature selection algorithms and the feature spaces are reduced. Finally, the classification of the dataset is performed by using the selected features. In the classification phase; Naive Bayes Multinomial, Support

Vector Machines, Decision Trees - J48, Random Forest algorithms, and finally the proposed classifier are used. These main steps used in the studies are explained in detail in the below subsections.

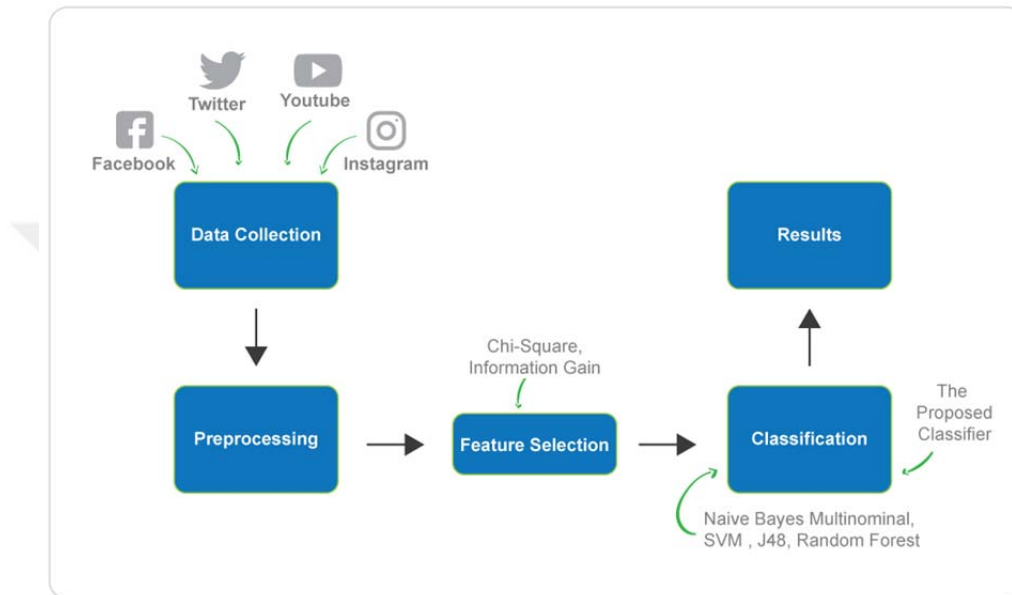


Figure 3.4. Main Steps and Methods Used in Classification

3.3.1. Preprocessing for Useless Character Removal

In this study, 15658 comments are stored in the excel table together with their labels (without any text editing). In the first stage, it is aimed to remove special characters which are not textual elements in the data. For this study, user names beginning with the “@” mark in the texts are completely removed. In order to do this, find-and-replace feature of excel is used, and all data have been processed by finding and deleting the words that start with the character @ in the dataset. As a result of this process, 7307 user names in the texts have been deleted. Afterwards, the emoticons, special characters, and the numbers that are assumed to have no effect over the cyberbullying are determined manually and these characters are deleted. Find-and-replace method has been applied again for this process. An

example which shows a small subset of the dataset before and after the useless character removal step is presented in Figure 3.5, and Figure 3.6.

	A
1	@erhanozturk_ harika bir iş başarmışsınız!!! tebrik ediyorum
2	ya niye zahmet ettiniz ki :)))
3	merhaba bu konuda katılıyorum,, çok doğru!
4	hava yine mükemmel @sebnem_1554

Figure 3.5. Dataset Before Any Character Removal

	A	B	C	D
1	harika bir iş başarmışsınız tebrik ediyorum			
2	ya niye zahmet ettiniz ki			
3	merhaba bu konuda katılıyorum çok doğru			
4	hava yine mükemmel			

Figure 3.6. The Same Dataset After Useless Characters Removal

These manually collected dataset has been prepared to be sent to the WEKA program for further preprocessing, weighting, feature selection and classification after the user name, unwanted special characters, emoticons, and numbers are cleaned.

3.3.2. ARFF (Attribute Relationship File Format)

In order to read the datasets in WEKA software, we need to convert it to the ARFF format. The ARFF format is a data format used worldwide for scientific purposes and the most important advantage of it is that it can be used with WEKA. The ARFF format is also a format for developing machine learning applications with python.

ARFF files have two different sections. The first section is called as the Header, and the second section is named as Data. The Header part of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data),

and their data types. An example header on the standard IRIS dataset is shown in Figure 3.7.

```

% 1. Title: Iris Plants Database
%
% 2. Sources:
%   (a) Creator: R.A. Fisher
%   (b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
%   (c) Date: July, 1988
%
@RELATION iris

@ATTRIBUTE sepallength NUMERIC
@ATTRIBUTE sepalwidth NUMERIC
@ATTRIBUTE petallength NUMERIC
@ATTRIBUTE petalwidth NUMERIC
@ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica}

```

Figure 3.7. Arff Header Section Example

Lines that start with % character in the header area are those that are not considered by WEKA and are for information purposes only. These lines are also called as comment lines. The details about the data or about the people that prepared the dataset can be written as a comment in this part.

@RELATION is one of the main and important expressions in ARFF. The name to be specified here is the name of the relation on the WEKA software under the choose button on the preprocess page. Just below is the @ATTRIBUTE expression. The features defined here create the columns of the dataset. When defining the feature, it must be expressed as follows: Any desired name can be given to the feature. However, only certain types can be defined as the data type. Numeric data is defined in two different ways: REAL and INTEGER. INTEGER defines the integer numbers, and REAL defines all real numbers.

Date type variables are defined by “DATE” data type. Text type variables are defined by “STRING” data type. Data in the form of a cluster is defined as NOMINAL. As shown in Figure 3.7, IRIS file has 5 attributes; the first four

attributes that are sepalwidth, sepalwidth, petalwidth, and petalwidth are numeric attributes while the last attribute which is class is a nominal attribute and has values from the set given as {Iris-setosa, Iris-versicolor, Iris-virginica} and shows the class labels for the dataset.

The data part of the file begins with the expression @DATA. After typing this expression, the values of the dataset are generated based on the order of features defined in the @ATTRIBUTE section. The @DATA part of the ARFF file whose header section is given in Figure 3.7 is shown in Figure 3.8:

```
@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

Figure 3.8. Data Section Example

The @DATA declaration is a single line denoting the start of the data segment in the file as in Figure 3.8. Each instance is represented on a single line, with carriage return denoting the end of the instance. Attribute values for each instance can be delimited by commas or tabs. A comma/tab may be followed by zero or more spaces. Attribute values must appear in the order in which they were declared in the header section.

3.3.3. Conversion of Data to ARFF Format

All data prepared in this study were collected in an excel table, so as to include a different comment for each row in the table. There are 2 different excel records, including comments that contain cyberbullying texts and comments that do not contain cyberbullying texts. Before this dataset has been converted to the ARFF file, some operations have been performed on it as described in Section 3.3.1. Figure 3.9. shows the sample data of the positive class in the excel table before it is converted to an ARFF file. Each row in this figure will create a line in the “@DATA” part in our ARFF file.

	A	B	C	D
1	harika bir iş başarmışsınız tebrik ediyorum			
2	ya niye zahmet ettiniz ki			
3	merhaba bu konuda katılıyorum çok doğru			
4	hava yine mükemmel			
5	muhteşem görünüyorsun cidden hayran kaldım			
6	resimleriniz içimi açıyor			
7	seni öyle çok seviyorum ki anlatamam			
8	HEPİMİZİN KALBİNDE YER ALDIN			
9	melek misin nesen yaaaa			
10	helal OLSUN kardeşim GURURUMUZSUN			

Figure 3.9. Dataset Example Before Converting to ARFF File

In this study, when creating an ARFF file, it was aimed firstly to organize the @RELATION, @ATTRIBUTE and @DATA parts which are the basic descriptive expressions of the ARFF. Editing has been done with the Notepad++ text editor.

For the dataset to be classified as cyberbullying or not, @RELATION is defined in the first line of the text editor and the relationship name of it has been determined as SiberVeriSeti. In the next step @ATTRIBUTE section is processed. There are 2 types of class labels which are positive and negative. Since all the data

in this class are textual elements and are free of numbers, the data type is defined as string.

```
1 @RELATION SiberVeriSeti
2
3 @ATTRIBUTE text string
4 @ATTRIBUTE Class {positive,negative}
```

Figure 3.10. Example of Attribute Information

Then, the data that is stored in a single row in the excel table and that have the positive class label have been copied to the bottom of the @DATA statement in the text editor. After 7995 positive comments are pasted to the bottom of the @DATA expression, a single quotation mark is placed at the beginning and end of each sentence so that the ARFF file could see the data as text. To be able to do this, “\$.*” and “^.*” expressions of the Notepad++ editor was used that can find the beginning and end of each sentence. To find the head of the sentence, search function of the text editor has been used, and a single quotation mark is added to the head of each row in the text by writing “\$.*” expression into the search tab. In the same way, by writing “^.*” expression into the search tab of the text editor, a single quotation mark is added to the end of each text, and finally all the text statements are arranged so as to be between single quotation marks. After adding the single quotation mark at the end of the text, in order to identify the class labels, a comma is placed and then a class tag is written. The same operations are done in a separate file for classes with negative tags and added at the end of this file. ARFF file that has been created as a result of these operations can be seen in Figure 3.11 and Figure 3.12.

```

1 @RELATION SiberVeriSeti
2
3 @ATTRIBUTE text string
4 @ATTRIBUTE Class {positive,negative}
5
6 @DATA
7 'harika bir iş başarmışsınız tebrik ediyorum',positive
8 'ya niye zahmet ettiniz ki ',positive
9 'merhaba bu konuda katılıyorum çok doğru ',positive
10 'hava yine mükemmel ',positive
11 'muhteşem görünüyorsun cidden hayran kaldım',positive
12 'resimleriniz içimi açıyor',positive
13 'seni öyle çok seviyorum ki anlatamam',positive
14 'HEPİMİZİN KALBİNDE YER ALDIN',positive
15 'melek misin nesen yaaaa',positive
16 'helal OLSUN kardeşim GURURUMUZSUN',positive

```

Figure 3.11. Positive Labeled Data Example

```

14427 'şerefsiz defol git',negative
14428 'senin kadar namussuzunu bu topraklar görmedi',negative
14429 'aptal aptal ekrana bakıyorsun',negative
14430 'bu kadar çirkin olmayı nasıl başarıyorsun',negative
14431 'geberdiğin günü görürsem parti yapacağım',negative

```

Figure 3.12. Negative Labeled Data Example

After these operations, all the data in excel table is converted to ARFF format to be read by WEKA program.

3.4. Preprocessing Operations with WEKA

In this section, all preprocessing steps that are applied by using WEKA on the prepared ARFF file are explained in detail.

By default, WEKA has a structure that cannot read Turkish characters in ARFF files. To overcome this issue, an Arabic Light Stemmer software package developed by Motaz K. Saad (2010) was integrated into the WEKA program to be able to read the Turkish characters. As a result of that, utf-8 formatted Turkish characters in the dataset have been read without any problem. After the successful

transfer of the ARFF file to the system, WEKA preprocess window appears as in Figure 3.2.

At the bottom of the Attributes window, text and Class features appear. The text feature contains 15658 data instances, which have the positive and negative tags that are class labels. The process that is done in this step is to divide all the text content in the data instances into words and make all words available as features to be used in the classification. WEKA's Filter feature is used for this purpose.

3.4.1. Main Filtering Properties

Another property in the preprocessing window of the WEKA program is the filters. Filters are very important for data preprocessing. They consist of two parts as supervised and unsupervised methods. Filters can be used on attributes and instances. Some filters are explained briefly in the below paragraphs:

Remove: Any attribute can be deleted by using this filter. To delete an attribute, simply this filter is selected, then the index of the attribute to be deleted is given, and the filter is applied. In fact, this operation can be done manually by choosing the attributes to be removed and clicking the remove button, however if the Java will be used for the operations in WEKA this filter has to be executed from the Java source codes.

RemoveByName: This filter can be used when you want to delete an attribute by specifying its name.

NumericToNominal: This filter is used to collect numeric values in a cluster.

NominalToString: This filter is used to convert the data in a cluster to a string type.

StringToVector: It is a very frequently used filter especially for text mining. For example, when the Reuters file, which is one of the datasets in WEKA, is opened, a list of news agencies will appear. When this list is passed through the

StringToVector filter, all string expressions are separated into individual features. These features can then be used as needed. This filter is used to separate the sentences in our dataset into words to get features and these operations will be explained in detail in section 3.4.2.

RemoveDuplicates: This filter is used for removing repeating instances from the dataset.

RemoveRange: It is used to delete instances in the defined range. The purpose of this filter is to remove the redundant data that will not be used in the dataset and to make the data ready for processing according to need. There are many more filters than mentioned above. Generally, these filters are examined and used when they are needed. The list of the filters is given in Figure 3.13.

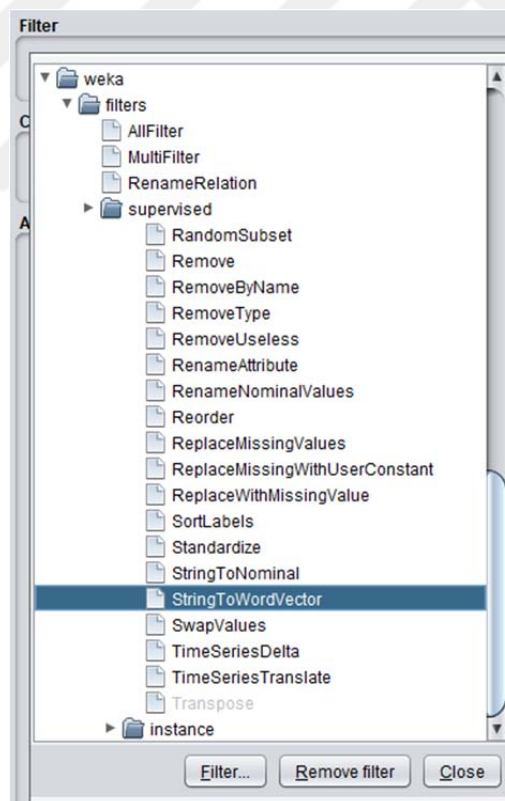


Figure 3.13. WEKA Filter Window

3.4.2. Separation of Sentences into Words, Weighting and Stemming

StringToWordVector filter is used to separate 15658 sentences into set of words in our dataset. By using this filter all the strings under Text attribute are separated into words and each unique word is used as an attribute. When the StringToWordVector filter is selected the window in Figure 3.14 is opened.

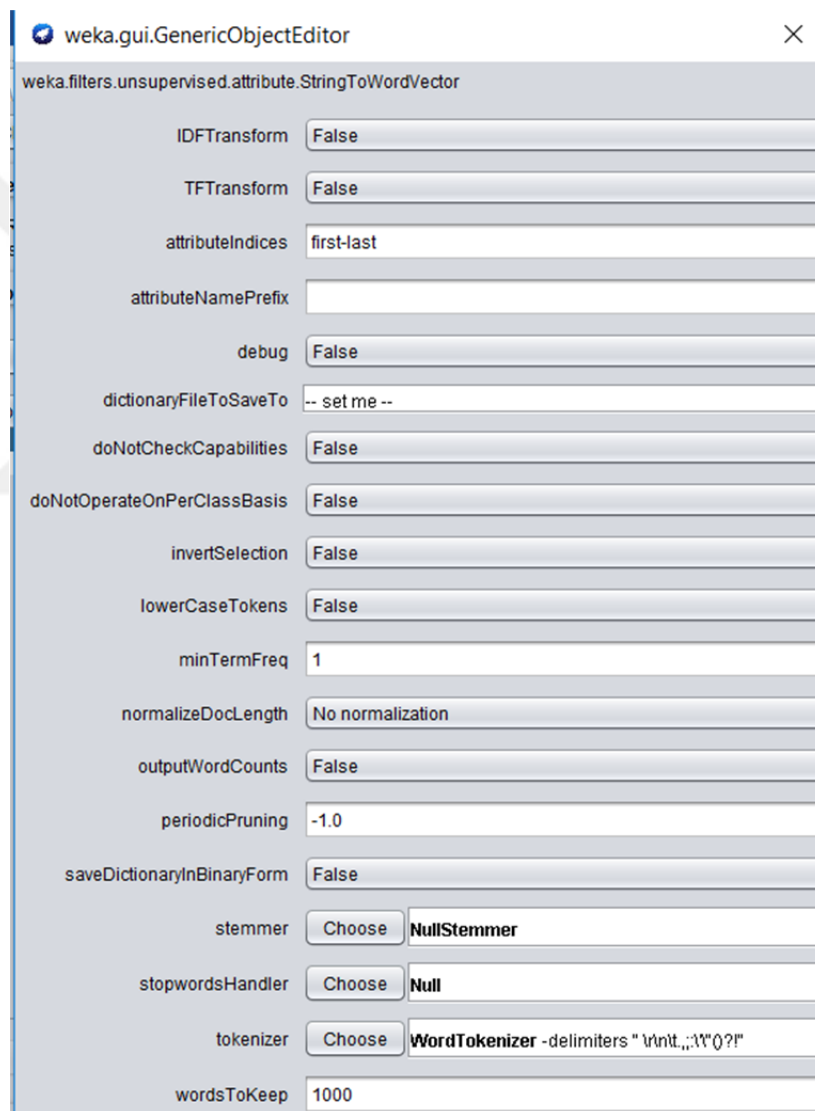


Figure 3.14. WEKA StringToWordVector Window

TF*IDF Calculation: After separating text contents into words, a numeric weight value is assigned to each word in each text message in the dataset. To assign weights to words in the dataset Term Frequency * Inverse Document Frequency (TF*IDF) term weighting method is used where TF is the frequency of the word in the given text message, and IDF is the inverse document frequency of the word for the whole dataset. Classically, it gives better result than using only TF as term weighting. In this study, TF*IDF values of words are computed by setting the parameters in Figure 3.14 as follows:

IDF Transform: is set to True. When it is selected as true, the frequency of the word i in a text message j is multiplied by the inverse document frequency. Therefore, weight of word i for a text message j is computed as in equation 3.1.

$$f_{ij} = f_{ij} \times \log\left(\frac{\text{Total number of documents}}{\text{Number of documents with term } i \text{ in it}}\right) \quad (3.1)$$

where f_{ij} is the frequency of word i in the text message j , and the value multiplied by f_{ij} is the inverse document frequency.

TF Transform: is set to True. When it is selected as true, the frequencies of the word i in a text message j is computed as in the equation 3.2:

$$f_{ij} = \log(1 + f_{ij}) \quad (3.2)$$

where f_{ij} is the number of occurrences of word i in the text message j .

attributeIndices: is set to first-last. It determines which attributes are used and which are not used. All attributes are used if the first-last is set.

doNotOperateOnPerClassBasis: is set to True. This indicates that the specified maximum number of words and the minimum term frequency will be applied to the entire dataset and not to each class. For example, if the maximum

number of words is set to 5000, and this value is equal to True, the most common 5000 words from the entire dataset is specified as attributes.

lowercaseTokens: is set to True. In that case all words are converted to lowercase before they are added to the attribute dictionary.

minTermFreq: is equal to 1 which indicates the number of times a word must be contained in the dataset, so that it can be evaluated as an attribute.

normalizeDocLength: is set to normalize all data. In that case, the word frequencies are normalized according to the document size. Separate options are available for test and training datasets.

outputWordCounts: is set to true. When it is selected as true, features are calculated based on the frequency of presence of the words in the documents. When it is selected as false, the existence of the words in the document is indicated by 0 or 1, and the frequency of the words are not shown.

stemmer: is selected as Snowball Stemmer. It has automatic functions that finds the root of the words. The Snowball Stemmer algorithm which is the most accurate version for the Turkish language was integrated into the WEKA, and by this way the roots of the words were found. Results were obtained both with stemmer and without stemmer for the efficiency tests in this thesis.

stopwords: The list of ineffective words is excluded from the features. By using an ineffective words list which contains the most used Turkish words and compiled manually, the words which are redundant and frequent, and will not be an indicator for the classes were eliminated. Therefore, number of attributes to be used will be reduced.

tokenizer: is set to WordTokenizer. It determines how to divide each text message into the words. It accepts the character group as a word in the interval which ends with space character.

useStoptlist: is set to true. It indicates whether the ineffective word list is used or not.

wordstoKeep: is equal to 5000 which is equal to the total numbers of features. Therefore, the top 5000 words with the highest frequency are determined as features and the feature space is reduced by eliminating the other words.

All of the words obtained with the use of StringToWordVector filter are accepted as attributes and sent to the next step named Feature Selection in the WEKA window.

3.5. Feature Selection

After the preprocessing operations, we apply two well-known feature selection methods that are chi-square (CHI2) and Information Gain (IG), then we compare their effects on the performance of cyberbully detection.

3.5.1. Information Gain (IG)

Information Gain (IG) is inspired from Shannon's Information Theory and is based on thermodynamics. IG is frequently used in the field of machine learning as an entropy-based method of feature evaluation. IG computes the level of data in bits for the class prediction. IG is used if the only data available is the presence of a feature and the corresponding class distribution (Mitchell, 1997).

When calculating the IG, all the data in the dataset and the attribute for which IG is to be calculated are used. The process starts by finding the entropy of the dataset. Entropy of a dataset D which is represented by $Info(D)$ is computed as in equation 3.3.

$$Info(D) = - \sum_{i=1}^m p_i \log_2 p_i \quad (3.3)$$

where p_i is the probability of class i in the dataset, and m is the number of classes. Then the information value for each attribute A is calculated according to equation 3.4.

$$Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j) \quad (3.4)$$

where v is the number of distinct values that the attribute A can take, $|D|$ is the number of data instances in the whole dataset D , D_j is the subset of D where attribute A takes value equal to a_j , and $|D_j|$ is the number of data instances in D_j . In the above equation 3.4 the information calculation is done for each attribute value a_j . Finally, the gain is calculated according to the calculated entropy and information value. The gain of attribute A is calculated as given in equation 3.5:

$$Gain(A) = Info(D) - Info_A(D) \quad (3.5)$$

Consequently, the gain for an attribute A is equal to the difference between the entropy of the whole dataset and the entropy of the attribute. Gain value is computed for all attributes in the dataset. Then attributes having the highest Gain values are chosen in the IG feature selection method. In other words, $Gain(A)$ tells us what would be acquired by branching on attribute A . This is the anticipated reduction in the information needs caused by knowing the value of A . The attributes having gain values that are above of some threshold value are selected as a feature subset (Saraç, 2016).

In WEKA, Information Gain is used as the feature selection filter, which gives good results in text classification according to our tests. With this method, the features which have high information gain are selected, and the others are eliminated. In the filter parameters window, InfoGainAttributeEval was selected as evaluator, ranker was selected as search method, and the threshold was set to 1 (one) to be able to perform these steps.

3.5.2. Chi-Square (CHI2) Method

Chi-square test is a hypothesis test method used for discrete data which provides to determine the relationship between the two variables whether they are dependent or not (Yates, 1934). Feature selection method based on Chi-square statistic includes two steps. In the first part of the method, chi-square statistics of the features with respect to classes are calculated. On the other hand, in the second part, chi-square values are analyzed, and the features are parsed repeatedly until inconsistent properties are found in the dataset (Kavzoğlu, 2014). The calculated chi-square value for a feature included in the dataset, measures its dependency in the class. A feature that has a nearly zero value indicates that feature is independent. A feature that has a high chi-square value is more important for the dataset. The equations used to calculate the chi-square value are given below (Kavzoğlu, 2014).

$$\chi^2 = \sum_{i=1}^2 \sum_{j=1}^k \frac{(A_{ij}-E_{ij})^2}{E_{ij}} \quad (3.6)$$

$$E_{ij} = \frac{(R_i * C_j)}{N} \quad (3.7)$$

In equation 3.6, k indicates the number of classes in the dataset, A_{ij} is the observed frequency value of attribute A when it is equal to a_i for class j (i row, j column) and E_{ij} denotes the expected frequency value of the A_{ij} . In equation 3.7, R_i is the number of data instances where $A = a_i$, C_j is the number of observations in the j^{th} class, N is the sum of the observations in the class. This method is used within the scope of intervals for the numerical values.

ChiSquaredAttributeEval function is chosen under the Select Attributes window for the Chi-square test in WEKA and the ranker is selected as search method on the filter parameter setting.

3.6. Evaluation Method (K-Fold Cross Validation)

In data mining studies, it is needed to separate the dataset as training set and test set to evaluate the success of the learned method. This separation process can be done in various ways. For example, dividing the dataset into two parts such that 66% as training and 33% as test sets, then evaluating the success of the learned model with the test set after learning the classification model by using the training set is one of the possible methods to be used. The random assignment of these training and test sets is another method. But the studies show that the K-fold Cross Validation method is the most efficient one to evaluate the performance of the learned model.

K-fold cross validation divides the whole dataset into random k disjoint parts, then in the 1st fold, the k^{th} part is used as the test set, and the remaining $k-1$ parts are combined and used for the training set. In the 2nd fold, the $k-1^{\text{st}}$ part is used as the test set, and the remaining $k-1$ parts are combined and used as the training set, and this process goes on k times. The values obtained in each iteration are averaged, and consequently the performance of the model is evaluated. This process is summarized in Figure 3.15, where k is equal to 10.



Figure 3.15. K-Fold Cross Validation Method

The dataset shown in Figure 3.15 is divided into 10 parts. In each round the painted area is reserved for the test set, the other parts are reserved for training. At the end of each round, the performance scores of the classifier are recorded to the E variable. When all the tours are over, the arithmetic average of E shows the performance of the learned model.

3.7. Classification

In this study, Naive Bayes Multinomial, Support Vector Machines (libSVM classifier of WEKA data mining tool), Decision Trees – J48, Random Forest and finally the proposed filter based classifier are used for classification processes. All experiments except for the proposed classifier are performed in WEKA environment.

3.7.1. Naive Bayes Classifier

Naive Bayes is a classification technique based on Bayes theorem. It takes into account the independence of features from each other when the classification is made. For instance, Naive Bayes classifier increases one unit of the possibility of a fruit to be an apple, if the apple is red or the width of the apple is lower than 5 cm. The combination of these two features at the same time does not provide any added value for Naive Bayes, and the Naive name comes from as a result of that pure behavior.

Bayes theorem is widely used in the conditional probability and very popular among the statisticians. It can be summarized with an example as follows:

Assume that we have some bicycles produced by factories A and B. Let factory A make 70% of production and factory B make 30%. In this case if we choose any of our bikes, probability that bicycle is produced by factory A is 0.7 and probability that the bicycle is from factory B is 0.3. In other words, probability of $A = P(A) = 0.7$, and probability of $B = P(B) = 0.3$

Now, we have a new information that the factory A produces 5% of the bicycles as defective and factory B produces only 3% defective bicycle. Therefore, the conditional probability that a defective bicycle is produced by factory A and B are as follows:

$$P(\text{Defective} | A) = 0.05 \text{ and } P(\text{Defective} | B) = 0.03$$

The first term in the parenthesis indicates that given probability (being defective), and the term after the | sign indicates the condition (from which factory). So we are writing a conditional probability.

In this case, what is the probability that the randomly selected bicycle that appears to be defective is from factory A? In other words what is $P(A | \text{Defective}) = ?$ This is where the Bayes theorem comes into play. According to Bayes theorem, $P(A|B) = P(B|A) \times P(A) / P(B)$

If we apply this to our example: Probability of defective bicycle from factory A is equal to the multiplication of probability of defective bicycle from

factory A with probability of all bicycles coming from A divided by probability of defective bikes. Therefore, it is calculated as follows:

$$P(A | \text{Defective}) = P(\text{Defective} | A) \times P(A) / P(\text{Defective})$$

$$P(A | \text{Defective}) = 0.05 \times 0.7 / (0.05 \times 0.7 + 0.03 \times 0.3)$$

$$P(A | \text{Defective}) = 0.79$$

In this case we can obtain the probability of a bicycle from B by using the information of both defective and not from A, as follows:

$$P(B | \text{Defective}) = 1 - 0.79 = 0.21$$

The probability of a randomly selected defective bicycle produced in the factory A is about 79%. This result which is consistent with our data is belong to the Bayes theorem.

Let's see now how Naive Bayes classification works. Naive Bayes is a supervised machine learning method (McCallum and Nigam, 1998). Naive Bayes cannot make classification by itself therefore, the data will be used should have been classified before.

Let's say we have a lot of email data as training data and they are classified as spam and normal. We assume that the algorithm in our mail server that we trained with this data will mark the incoming mails as spam or not.

Naive Bayes acts as follows: By taking each word in each email in the training set, it determines a probability whether it is a spam or not by looking at the related spam status of the mail which contains the word. As mentioned earlier, this rate is completely independent of the other words in the mail.

We have many words that are likely to be included in a spam or non-spam mail group. The probability of spam is calculated and classified according to the words in the new incoming mail.

Finally, Multinomial Naive Bayes classifier is a variance of a Naive Bayes classifier which uses a multinomial distribution for each individual feature. Naive Bayes Multinomial is generally used for text classification and it has a comparable performance with support vector machines (Rennie et al., 2003).

3.7.2. Support Vector Machines

Support Vector Machines (Cortes, 1995) (SVM) is a supervised learning method which can be used for regression analysis and classification. It is one of the effective and accurate methods used in classification.

For classification, it is possible to separate the two groups by drawing a boundary between the two groups in one plane. Where this boundary is drawn should be the farthest from the members of both groups. Here SVM determines how to draw this boundary. In order to carry out this process, two boundary lines near and parallel to each other are drawn and these boundary lines are brought closer to each other to produce a common boundary line. For example, consider data instances from two groups as shown in Figure 3.16:

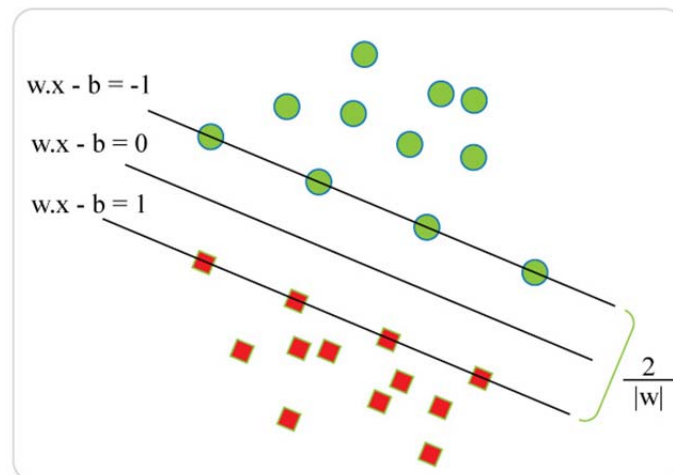


Figure 3.16. SVM Classifier

As shown in Figure 3.16, the two groups are shown on a two-dimensional plane. It is possible to consider this plane and dimensions as features. In other words, a feature extraction of each input entered into the system is made in a simple sense and as a result a different point is obtained in this two-dimensional plane showing each input. Classification of these points means the classification of inputs according to the extracted features.

The interval between the two classes in Figure 3.16 is called offset. The definition of each point in this plane can be made in the following notation:

$$D = \left\{ (x_i, c_i) \mid x_i \in \mathbb{R}^p, c_i \in \{-1, 1\} \right\}_{i=1}^n \quad (3.8)$$

It is possible to interpret the equation 3.8 as follows: for each data point x , and its class label c , x is a point in our vector space and c is the class value indicating that this point is -1 or +1. There are i data instances in the dataset D for $i = 1$ to n . In other words, this representation refers to the data points shown in Figure 3.16. Considering that this representation is on a hyperplane, every point in this representation is expressed as:

$$wx - b = 0 \quad (3.9)$$

where, w is the normal vector perpendicular to the extreme plane and x is the variable of the data point and b is the shear rate. It is possible to compare this equation to the classical $ax + b$ line equation.

Again according to the above equation $b / \| w \|$ value gives us the difference in distance between two groups. We have also called this distance difference offset before. In order to maximize the distance according to this distance difference equation, $2 / \| w \|$ formula is used in the equation that gives 3

lines with 0, -1 and +1 values shown in Figure 3.16. So, the distance between the lines is determined as 2 units. Obtained two line equations according to this equation are:

$$wx - b = -1 \quad (3.10)$$

$$wx + b = 1 \quad (3.11)$$

In fact, these equations are the result of finding the highest values obtained by shifting the lines. At the same time, these equations are assumed to be linearly separable.

The general idea of the algorithm is as explained above, and to implement the SVM algorithm using WEKA, we need to install a package called LibSVM (Chang and Lin, 2011). After the package is loaded, it can be placed among the classifiers and necessary procedures can be performed for the classification.

3.7.3. Decision Trees – J48

The decision tree algorithm used in this thesis is an algorithm referred to as j48 in WEKA and classifies data by learning a decision tree in top-down fashion. The algorithm used is known as C4.5 (Quinlan, 1993).

This algorithm aims to optimize the decision tree by utilizing Shannon's Information Theory (1948) hypothesis. This is based on the entropy values of the variables. C4.5 first calculates the entropy value for the target variable/class. It then calculates the information value for each estimator variable/class. It then calculates the information gain of each estimator variable/class. The purpose of these calculations is to determine the class of estimators that provides the highest information gain. Information Gain formulas used in these calculations are described in section 3.5.1.

The estimator variable that provides the highest information gain is determined and the tree starts branching from this variable. Thus, the data will be distributed evenly under each branch. After the first estimator variable is determined, the same process is repeated this time not on the total entropy, but on the information value of this determined estimator variable. It is calculated by which of the remaining estimating variables that the division of this determined variable will provide more information gain. This process continues until all the predictive variables are inserted into the tree, or all the data instances on the node belongs to the same class.

3.7.4. Random Forest

Instead of branching selected nodes from the best attributes, in the Random Forest (RF) decision tree set, it randomizes all nodes into branches by selecting the best of randomly acquired features from each node. Each dataset is generated displaceable from the original dataset. Trees are developed using random feature selection and no pruning (Breiman, 2001). This is the reason why random forest algorithm is faster and more accurate than other algorithms.

The RF model is based on 2 parameters. These parameters are the number of trees to be created (B) and the number of estimators (m) to be randomly selected in each node separation. When each decision tree is created, a sample is created such that the number of observations (n) in the original dataset is the same. $2/3$ of the examples are used as the training dataset (inBag) that are used to construct the tree; and the remaining $1/3$ of the dataset is used as the test set (out of bag or OOB) to test the internal error rate of the learned model.

The RF algorithm is set up as follows:

- 1) A sample of the dataset having n instances is selected by the Bootstrap method. This dataset is divided into training dataset (inBag) and test dataset (OOB).

- 2) A decision tree (CART) of the largest width is created with the training dataset (inBag) and the resulting decision tree is not pruned. In the creation of this tree, each of the m nodes is randomly selected among the p predictor variables. The condition $m < p$ must be satisfied here. Because overfitting of the tree to the training dataset is not desired. From these selected m estimators, branching occurs with the one which has the highest information gain. The Gini index is used to determine the difference between these m variables. This process is repeated for each node until there are no more branches to create.
- 3) Each leaf node is assigned a class label. The test dataset (OOB) is then dropped from the top of the tree and the class assigned to each observation in this dataset is recorded.
- 4) All steps from 1 to 3 are repeated B times.
- 5) An evaluation is made with unused observations (OOB) when creating trees. The number of times an observation is categorized in classes is counted.
- 6) A class is assigned to each observation with a majority of votes determined on the tree sets. For example, in a classification model of 2 categories, an observation carries the label of the class from which it receives a majority of at least 51% of the vote, and this class becomes its estimated class value.

3.7.5. The Proposed Classifier

In this study, a filter-based classifier is proposed and its performance is evaluated over the Turkish dataset collected. Performance of the proposed classifier is also compared with the well-known classifiers that are Multinomial Naïve Bayes, decision tree, support vector machines, and random forest.

In the first stage of the classification, words are extracted from the dataset collected. Zemberek is used for stemming and spell checking of the words extracted. Zemberek is an open source Turkish natural language processing library

(Akin et al., 2007). By using Zemberek, root of each word is found and if the word is misspelled, it is also corrected. Then the corrected and stemmed word list is used to create a list of bad words to be used to detect cyberbullying by our classifier. To create the bad words list, all the stemmed and corrected words from the positive comments are taken. The same process is applied to the negative comments also. So, there are two words lists: one list that is generated from the positive comments, and another list that is formed from the negative comments. As the positive word list does not contain any cyberbullying word, words in the positive words list are subtracted from the negative words list, therefore a bad words list is created.

The proposed classifier uses the generated bad words list to determine whether there is cyberbullying or not on the given text content as follows: The classifier takes an input text content, then it is tokenized and words in the text content are extracted. Then, each extracted word is stemmed and corrected by using the Zemberek. After that, the processed words are searched from the bad words list. If the text content has at least 3 bad words, our classifier labels it as negative that is the text contains cyberbullying. Otherwise it is labeled as positive meaning that there is no cyberbullying. The number of bad words threshold value which is equal to 3 is determined experimentally.

3.7.6. Classification Performance Metric

Classification performance is measured with F-measure (Han and Kamber, 2006) value which is given in equation 3.12. F-measure is a harmonic mean of precision and recall values.

$$F - measure = \frac{2 * recall * precision}{recall + precision} \quad (3.12)$$

Recall is the ratio of true positives to the number of samples that are positives, as in equation 3.13.

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives} \quad (3.13)$$

Precision is the ratio of the true positives to the number of samples labeled as positives, as in equation 3.14.

$$precision = \frac{TruePositives}{TruePositives + FalsePositives} \quad (3.14)$$





4. RESULTS AND DISCUSSIONS

In this section, the experimental results of the methods used, and the proposed classifier are presented and compared. In this study, WEKA software was preferred for preprocessing, feature selection and classification; all operations were performed in WEKA environment and results were obtained.

The general information about the dataset collected before proceeding to the preprocessing step is shown in the table below.

Table 4.1. The General Information About the Dataset Collected

Dataset	Total number of comments	Total number of comments labeled as negative	Total number of comments labeled as positive	Total number of words	Total number of unique words
Facebook, Twitter, Youtube	6230	2326	3904	58854	17382
Instagram	9428	5338	4090	91712	25712
Whole Dataset	15658	7664	7994	150566	35618

As seen in Table 4.1, the comments collected from Facebook, Twitter and Youtube sites are 6230 in total. 2326 of these comments are labeled as negative because they contain cyberbullying texts, and 3904 comments are labeled as positive. Comments collected from Facebook, Twitter and Youtube sites contain a total of 58854 words and 17382 unique words. The total number of comments from

Instagram site is 9428. 5338 of these comments are labeled as negative because they contain cyberbullying texts, while 4090 are labeled as positive. The comments collected from Instagram contain 91712 words and 25712 unique words in total. Considering all datasets, a total of 15658 comments are collected, of which 7664 comments are marked as negative and 7994 are marked as positive. In addition, there are 150566 words and 35618 unique words in our dataset.

4.1. Results and Comparison of Preprocessing Steps

In this section, the results of preprocessing methods performed in WEKA environment are compared on the dataset collected. Since preprocessing methods directly affect the classification performance, the best method for our Turkish dataset is tried to be determined. The preprocessing methods are evaluated with F-Measure value and Naive Bayes classification is used in all preprocessing steps for comparison.

4.1.1. Effect of TF*IDF Weighting

In this section, the effect of TF*IDF weighting method on our dataset and classification is evaluated. The StringToWordVector algorithm described in section 2.2.2 is used for this operation in WEKA environment and TFTransform and IDFTransform options under this algorithm functions are marked as True or False and top 1000, 3000, 5000, 7000, and 10000 words with the highest frequency are chosen as attributes. The results of the classification are shown in Table 4.2.

Table 4.2. Comparison of TF*IDF Methods for All Dataset (with F Measure Value)

	TFTransform: False	TFTransform: True	TFTransform: False	TFTransform: True
	IDFTransform: False	IDFTransform: False	IDFTransform: True	IDFTransform: True
Top 1000 Words	0.844	0.843	0.846	0.847
Top 3000 Words	0.877	0.876	0.880	0.882
Top 5000 Words	0.890	0.888	0.894	<u>0.895</u>
Top 7000 Words	0.862	0.870	0.878	0.881
Top 10000 Words	0.860	0.859	0.865	0.866

As can be seen from the table, when 5000 features are used, the best F-measure values are observed for all TF and IDF combinations. Among all four TF and IDF combinations, in the TF*IDF weighting method, the option where both are set to True has the best F-measure value. In this study, TF*IDF weighting method is used in the subsequent experiments for term weighting, and feature size is chosen as 5000.

4.1.2. Effect of Stemming Function

In this section, effects of a stemming algorithm which takes the roots of words in the whole dataset is investigated in WEKA environment. For stemming,

Snowball Stemmer function is used. The experimental results before and after using the stemmer are shown in Table 4.3.

Table 4.3. The Effect of Stemmer Function on the Dataset

	Total Unique Words Before Stemming	Total Unique Words After Stemming	Classification F Measure Value Before Stemming	Classification F Measure Value After Stemming
Whole Dataset	35618	24804	0.895	0.914

As can be seen from Table 4.3, there are a total of 35618 unique words before the stemmer function is applied, but these words are directly assumed to be annexed without being separated to its roots. For example, while the words *kalem* (pencil), *kalemler* (pencils), *kalemim* (my pencil), and *kaleminden* (from pencil) are considered as four distinct words, these words are taken as *kalem* (pencil) after the stemmer function is applied and they are assumed as the same one word. As a result of the stemming, a total of 10814 words are excluded. As seen from the F-Measure values, applying stemming increases the classification F-measure about ~ 2.12%. F-Measure value increased from 0.895 to 0.914 after stemming is applied. Therefore, in this study, stemming is applied for the subsequent experiments.

4.1.3. The Effect of Stopwords Removal

In this section, stopwords that have no effect on the classification in Turkish language are deleted from the words list that are used as attribute in the classification process and the experimental results are obtained. The list of ineffective words is compiled manually, and this list contains the most frequently

used words in Turkish. In our dataset 284 unnecessary and frequent words are identified and they are eliminated from the attribute list. The experimental results for stopwords removal step are shown in Table 4.4.

As seen in Table 4.4, after stopwords removal, we have 24628 unique words left out of 24804 unique words. This means that 176 of the 284 stopwords exist in our dataset. By eliminating these words, the F-Measure value which is 0.914 decreases to 0.910.

Table 4.4. The Effect of Deleting Stopwords

	Total Unique Words Before Stopwords Removal	Total Unique Words After Stopwords Removal	Classification F Measure Value Before Stopwords Removal	Classification F Measure Value After Stopwords Removal
Whole Dataset	24804	24628	0.914	0.910

This table shows us that the elimination of stopwords has a negative effect on the classification accuracy. A new comparison is also considered to investigate the cause of this decline. In the next experiment we try to find the answer of the question “does the removal of the stopwords cause accuracy decrease in negative comments or positive comments?”. This comparison is shown in Table 4.5.

Table 4.5. The Effects of Stopwords Removal on Negative and Positive Labeled Comments

Dataset	Classification F-Measure Value Before Stopwords Removal	Classification F-Measure Value After Stopwords Removal
Negative Comments	0.907	0.906
Positive Comments	0.915	0.912

As can be seen in Table 4.5, there is a slight decline in the F-Measure value on negative comments, whereas there is a greater decrease in the classification accuracy on the positive labeled comments with respect to negative labeled comments. This decrease is also the reason for the decrease in the overall classification accuracy. This table shows us that the stopwords in the Turkish language are not frequently used in the negative texts that contain cyberbullying however, stopwords are more frequently used in the positive comments. The use of stopwords removal in the dataset is not recommended in this study as it reduces the accuracy of the detection of cyberbullying texts.

4.1.4. The Performance Comparison of Preprocessing Steps

This section contains the comparison of all preprocessing steps on the classification accuracy. The best and the worst methods are determined, and the most suitable methods are suggested before proceeding to the feature selection step. The best and worst methods with respect to classification success are underlined. Table 4.6. shows all combinations of the methods used.

Table 4.6. Performance Comparison of the Preprocessing Steps

Stemmer and Stop Words Removal	TFTransform: False IDFTransform: False	TFTransform: True IDFTransform: False	TFTransform: False IDFTransform: True	TFTransform: True IDFTransform: True
Without Stopwords Removal, Without Stemmer	0.887	0.888	0.894	0.895
Without Stopwords Removal, With Stemmer	0.910	0.908	0.912	0.914
With Stopwords Removal, Without Stemmer	0.892	0.891	0.892	0.893
With Stopwords Removal, With Stemmer	0.906	0.907	0.908	0.910

As can be seen from Table 4.6 the classification F-Measure has the worst value which is equal to 0.887 when no stopwords removal, and no stemmer functions are used and no TF*IDF weighting is performed. The classification with no stopwords removal, but with the stemmer algorithm and TF*IDF weighting shows the best performance with 0.914 F Measure. These values in our Turkish dataset show us that the best choice among the processes applied in the preprocessing steps is the application of stemmer algorithm and the selection of TF*IDF calculation as weighting method without stopword removal. In the next step, selection of the best performance options, which are mentioned above, are accepted and the operations are performed accordingly.

4.2. Results and Comparison of Feature Selection Functions

In this section, the effects of Chi-square and Information Gain feature selection functions on the classification performance are evaluated and the results

are presented. Naive Bayes Multinomial classification is used in all feature selection steps, as in the previous experiment.

In Table 4.7, 1000, 3000 and 5000 words with the highest frequency are selected and compared with the 1000, 3000, and 5000 words selected by using the Chi-square and Information Gain feature selection method and their effects on the classification are evaluated.

Table 4.7. The Effect of Feature Selection Algorithms on Classification

Word Count	Most Frequent Words	Chi-Square	Information Gain
1000	0.873	0.877	0.878
3000	0.899	0.902	0.903
5000	0.914	<u>0.917</u>	0.915

Feature selection methods are compared on the basis of F-Measure values. The numbers of selected features are determined as 1000, 3000 and 5000. After feature selection, features having scores that are lower than 0.001 with respect to the applied feature selection method were also removed from the feature set to further reduce the size of the feature set. As can be seen in Table 4.7, classification success has increased as the number of features increases. As seen in the table, Chi-square achieves the highest performance with 0.917 F-measure in the selection of 5000 words. Therefore, Chi-square feature selection method is preferred when comparing classification methods in the next subsections.

4.3. The Results of the Classification Methods

In this section, the results of Naive Bayes Multinomial, Support Vector Machines, Decision Trees - J48, Random Forest algorithms and finally the proposed classifier are presented, and the accuracy rates are compared. For each classifier, the time elapsed in the classification is determined and these values are

compared graphically. At the end of the study, the most efficient method is determined.

4.3.1. The Results of Naïve Bayes Classifier

After preprocessing and feature selections, the Naive Bayes Multinomial classifier is called in the WEKA environment to use the Naive Bayes classifier. 5000 words selected with the Chi-square method after stemming, and TF*IDF weighting are used as preprocessing of the dataset for this classifier. The results obtained are as follows:

Table 4.8. Results of the Naive Bayes Classifier

Classifier	Naive Bayes Multinomial (with WEKA)
Instances	15658
Attributes	5000
Test mode	10-fold cross-validation
Stemmer	Snow Ball (Turkish)
Feature Selection	Chi-Square
Time Taken to Build model	0.01 seconds
Precision	0.915
Recall	0.916
F-Measure	0.917
Correctly Classified Instances	14311 (91.3974 %)
Confusion Matrix	<pre> a b <-- classified as 7478 517 a = positive 830 6833 b = negative </pre>

As shown in Table 4.8, in the classification made by WEKA using Naive Bayes Multinomial classifier, 14311 of 15658 instances are classified correctly and 91.3974% accuracy is achieved. A satisfactory result is obtained for the classification. In terms of Confusion Matrix, 7478 out of a total of 7995 positive-tagged data labeled correctly, while 517 samples are classified incorrectly. In total,

6833 of 7663 negative labeled data are classified correctly, while 830 of them are labeled incorrectly as in the other class. In this classifier tested with a 10-fold cross validation model, the system spent 0.01 seconds to learn each classification model.

4.3.2. The Results of the Support Vector Machine Classifier

This study uses the LibSVM library to run the Support Vector Machine classifier in WEKA. The results obtained with SVM classifier after preprocessing and feature selection are shown in Table 4.9. Again for this classifier, TF*IDF is used as the weighting method and Chi-Square method is applied for feature selection after stemming the data.

Table 4.9. Results of the Support Vector Machine Classifier

Classifier	Support Vector Machines (WEKA – LibSVM)
Instances	15658
Attributes	5000
Test mode	10-fold cross-validation
Stemmer	Snow Ball (Turkish)
Feature Selection	Chi-Square
Time Taken to Build Model	25.03 seconds
Precision	0.855
Recall	0.810
F-Measure	0.805
Correctly Classified Instances	<u>12690 (81.0448 %)</u>
Confusion Matrix	<pre> a b <-- classified as 5131 2864 a = positive 104 7559 b = negative </pre>

When the results in Table 4.9 are examined, 12690 samples out of 15658 instances are correctly classified in the classification made by using SVM classifier. This result has 81.0448% classification accuracy for the dataset. It has a lower success rate with respect to Naïve Bayes Multinomial classifier, while 25.03 seconds are spent for learning the classifier model for each fold. This is a poor

result in terms of time costs. When confusion matrix is evaluated, it is observed that 5131 out of a total of 7995 positive labeled data are correctly classified, while 2864 positive samples are classified incorrectly. The classification success of the data labeled as positive is low. 7559 out of a total of 7663 negative labeled data are classified correctly, 104 negative samples are labeled as positive and classified incorrectly. The SVM classifier is particularly more successful in classifying negative labeled comments, whereas for the positive class, the opposite is true.

4.3.3. The Results of the Decision Tree-J48

In this section, the results of a Decision Tree classifier found in WEKA, referred to as J48, are explained. 5000 words are selected by using the Chi-square method after stemming and TF*IDF weighting are used as in the previous experiments. The experimental results are as shown in Table 4.10.

Table 4.10. Results of the J48 Classifier

Classifier	Decision Tree-J48 (WEKA J48 pruned tree)
Instances	15658
Attributes	5000
Test mode	10-fold cross-validation
Stemmer	Snow Ball (Turkish)
Feature Selection	Chi-Square
Time taken to build model	674.09 seconds
Precision	0.836
Recall	0.829
F-Measure	0.829
Correctly Classified Instances	<u>12988 (82.948 %)</u>
Confusion Matrix	<pre> a b <-- classified as 6134 1861 a = positive 809 6854 b = negative </pre>

As shown in Table 4.10, there is 82.948% classification success rate for this algorithm. This result shows that 12988 of the total of 15658 instances are correctly classified. The elapsed time for modeling is 674.09 seconds, that involves too much computation cost for the classification. It shows poor performance in terms of time for the classification. In terms of Confusion Matrix, 6134 out of a total of 7995 positive labeled samples are correctly classified, while 1861 positive comments are classified incorrectly. On the other hand, out of a total of 7663 negative labeled data, 6854 of them are classified correctly and 809 negative comments are classified incorrectly. J48 has slightly better classification accuracy with respect to SVM.

4.3.4. The Results of Random Forest Classifier

The results obtained for Random Forest classifier which is one of the popular machine learning and classification algorithms are shown in Table 4.11. When the results in Table 4.11 are analyzed, it is observed that 12103 data instances are classified as correct among 15658 instances in this classification made by using Random Forest classifier. This classification has 77.296% accuracy for the dataset. It has a low accuracy rate and it is tested with a 10-fold cross validation as in the previous experiments 640.81 seconds are spent for each fold. This is a very bad result in terms of time costs. In terms of Confusion Matrix, out of a total of 7995 positive labeled data instances, 5382 of them are classified correctly, while 2613 data samples are classified incorrectly. The success of the classification for the data labeled as positive is very low.

Table 4.11. Results of Random Forest Classifier

Classifier	Random Forest
Instances	15658
Attributes	5000
Test mode	10-fold cross-validation
Stemmer	Snow Ball (Turkish)
Feature Selection	Chi-Square
Time taken to build model	640.81 seconds
Precision	0.787
Recall	0.773
F-Measure	0.771
Correctly Classified Instances	12103 (77.296 %)
Confusion Matrix	<pre> a b <-- classified as 5382 2613 a = positive 942 6721 b = negative </pre>

In total, 6721 of the 7663 negative labeled data are classified correctly, while 942 negative comments are labeled as the other class and classified incorrectly. For Turkish texts, Random Forest may not be preferred in terms of time cost and success rate in classification.

4.3.5. Comparison of Classification Results and Discussion

In this section, the classifiers used in this study are compared and evaluated in terms of F-Measure values, time takes to learn the model, and accuracy rates. In this study, the most appropriate method for the determination of cyberbullying texts in Turkish content is proposed.

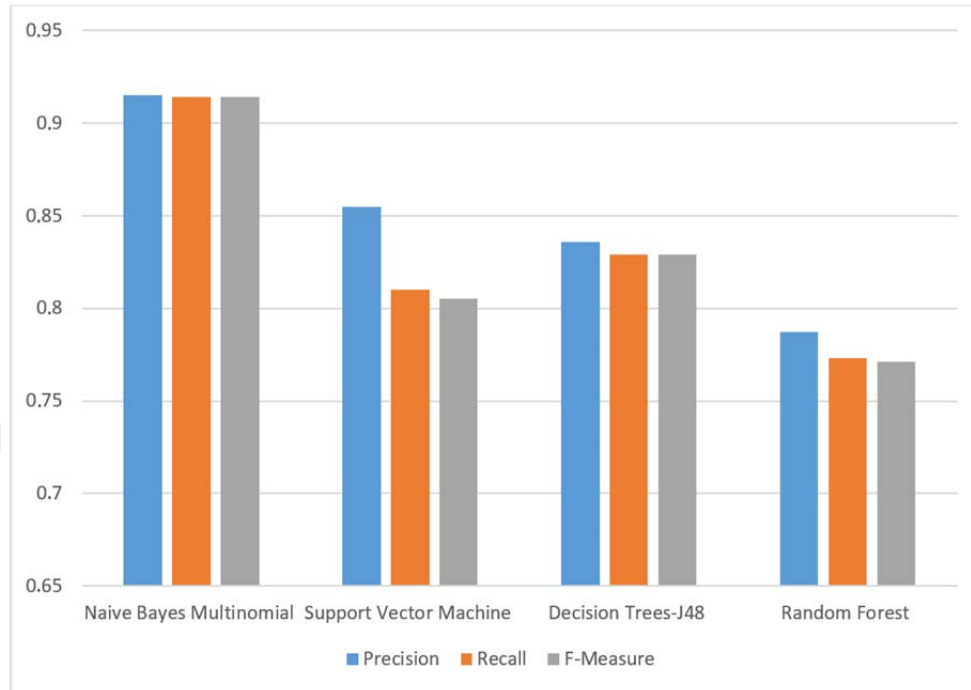


Figure 4.1. Comparison of Precision, Recall and F-Measure Values

Figure 4.1 compares the F-Measure, Precision and Recall values for each classifier. For this study, Naive Bayes Multinomial is found as the most successful method in all three measure. The SVM algorithm is the second best in the Precision value but its Recall and F-Measure values are lower than that of Decision Trees. Random Forest has the worst performance. In the next step, the accuracy rates of the classifiers are compared.

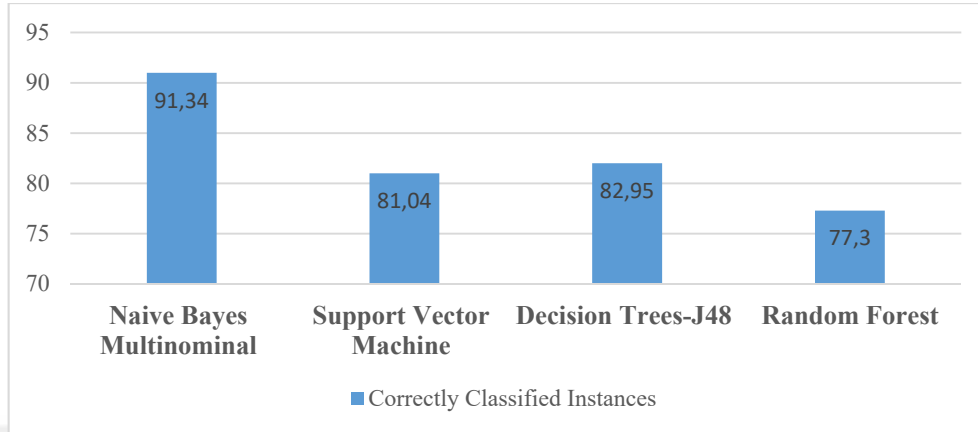


Figure 4.2. Accuracy Comparison of Classifier Results

As can be seen in Figure 4.2., Naive Bayes Multinomial classifier is the most successful classifier for the detection of cyberbullying in Turkish text with an accuracy rate of 91.34%. The Random Forest classifier is the worst classifier compared to other classifiers with an accuracy of 77.3%.

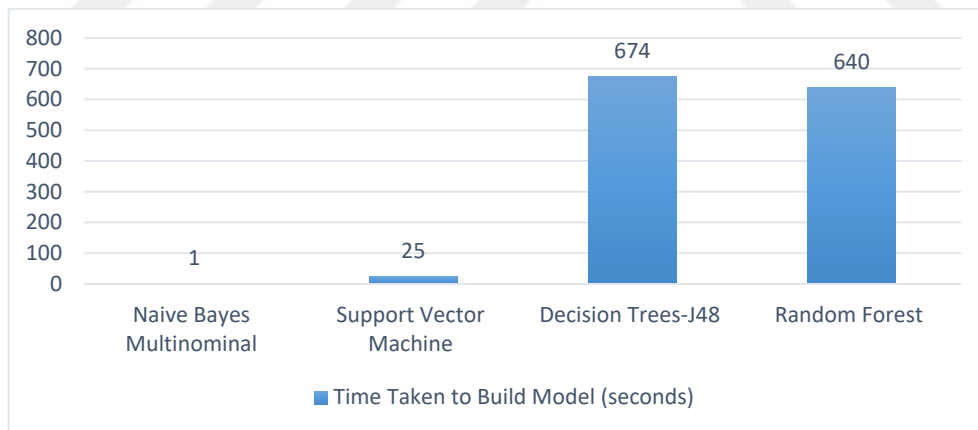


Figure 4.3. Time Taken to Build Model

Figure 4.3 compares the time taken for learning the classification model. Naive Bayes Multinomial shows the fastest performance based on the times used for learning in the WEKA environment. Although the SVM classifier requires

more time to build the classification model with respect to NBM, it has better performance than J48 and Random Forest. However, J48 and Random Forest have extremely the costly performance based on the times given in Figure 4.3, and they have a poor result.

As a result, NBM shows the most successful performance when F-Measure values and time elapsed for modeling are compared. NBM, which is one of the classifiers used in WEKA, has been deemed suitable for the detection of cyberbullying from Turkish text content.

4.3.6. The Proposed Classifier and Results

In this section, we discuss the results of our proposed classifier which is implemented to test the efficiency of the dataset collected in this study. The results of our proposed classifier, whose details are mentioned in the previous sections, are shown in the table below.

Table 4.12. Results of the Proposed Classifier

Classifier	Proposed Classifier
Instances	15658
Attributes (Bad words)	5000
Stemmer	Zemberek
Feature Selection	-
Time taken to build model	9 seconds
Correctly Classified Instances	<u>12840 (82 %)</u>

When we look at the results of the proposed classifier, it is seen that Zemberek library is used to get word stems unlike other classifiers. Another difference is in the Test Mode. The bad words list is generated from the whole dataset, and they are used to classify the all instances in the dataset instead of 10-fold cross validation test mode. After performing these operations, 9 seconds are

spent for classification modeling. As a result, 12840 instances of 15658 instances are correctly classified and 82% accuracy is achieved. When the accuracy of this classification is compared with the accuracy of the other classifiers made in this study, the result is as shown in Figure 4.4.

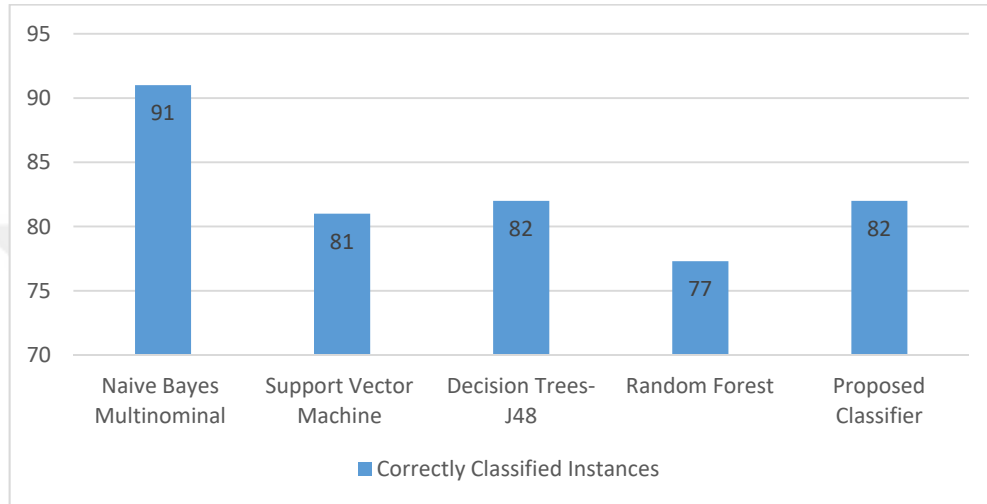


Figure 4.4. Comparison of Accuracies of Classifiers

As a result, NBM shows the most successful performance when classifier accuracies, F-Measure values, and time elapsed for modeling are compared. NBM, which is one of the classifiers used in WEKA, has been deemed suitable for the detection of cyberbullying texts containing Turkish content. As shown in Figure 4.4 the proposed classifier has the second best classification accuracy.



Figure 4.5. Time Taken to Build Model

When the times required for learning the classifiers are compared, our proposed classifier has rank second in terms of time performance after NBM.

5. CONCLUSION

In this thesis, the success of automatically detection of cyberbullying from text contents written in Turkish is investigated. In this regard, cyberbullying texts on Turkish-language sites are collected. For this purpose, 4 different social networks are used and as a result a dataset containing 15658 instances is prepared. This dataset has the distinction of being the largest dataset prepared for detecting cyberbullying from Turkish text contents. After collecting the dataset, the effects of preprocessing and classification methods are studied on the performance of cyberbullying detection. It is determined that TF*IDF weighting has a positive effect on the accuracy of classification. In addition, SnowBall Stemmer, which is a root finder algorithm running in WEKA environment, increases the classification performance in this study. The removal of stopwords from the dataset adversely affects the results. This has shown us that stopwords are widely used, especially in texts that do not contain cyberbullying, thus making it easier to identify texts that do not contain cyberbullying. Therefore, stopwords are not deleted in this study.

After the preprocessing step, the feature space of our dataset is reduced by applying the feature selection methods. For this purpose, two well-known feature selection methods that are Chi-square and Information Gain are applied. Chi-square feature selection method is found to be slightly more successful than Information Gain. After feature selection, features having scores below 0.001 are removed from the feature set. After feature selection, classification methods are tested and four different classification algorithms, which are popular in text processing, are used. In addition, a filter based classifier is proposed for this thesis in order to test the efficiency of the dataset and compare it with other classifiers.

Naive Bayes Multinomial is found to be the most successful classifier in terms of classification time and classification accuracy. In addition, the proposed classifier is found to be the second best in terms of accuracy and runtime, which shows that the collected dataset is actually large enough.

As future work, it is aimed to make further research and include more instances to the dataset collected. In order to improve the Turkish content dataset, we plan to increase its size to above of 15658 instances. To help studies done in this subject, this dataset prepared within the scope of this thesis is aimed to be shared on the internet as an open source. With this dataset, which can be used as an open source on the internet, new improvements can be made for the detection of cyberbullying in Turkish texts.



REFERENCES

- Aggarwal, C. C., & Zhai, C., 2012. A survey of text classification algorithms. In Mining Text Data (Vol. 9781461432234, pp. 163-222). Springer US. https://doi.org/10.1007/978-1-4614-3223-4_6
- Akbaba, S., Şahin, M., 2018. Okulda Ve Sanal Ortamda Zorbalık ve Empati. NOBEL Akademik Yayıncılık.
- Akin, A.A., Akin, M.D., 2007. Zemberek, An Open Source NLP Framework for Turkish Languages. Structure, 10, 1-5.
- Aricak, O.T., 2009. "Psychiatric Symptomatology as A Predictor of Cyberbullying Among University Students", Eurasian Journal of Educational Research, 34, s. 167-184.
- Aricak, O.T., 2011. "Siber Zorbalık: Gençlerimizi Bekleyen Yeni Tehlike" Kariyer Penceresi 2 (6): 10-12.
- Bahat, C. S., 2008. Cyberbullying: Overview and Strategies for School Counsellors, Guidance Officers, and All School Personnel.
- Bargh, J.A. And Mckenna, K.Y.A., 2004. The Internet and Social Life. Annual Review of Psychology, 55, 573-590.
- Bauman, S., 2010. Cyberbullying in a rural intermediate school: An exploratory study, Journal of Early Adolescence, 30: 803-833.
- Belsey, B. 2017. Cyberbullying: An emerging threat to the 'always on' generation. Retrieved Jan 14, 2018, from <http://www.cyberbullying.ca>
- Beran, T., and LI, Q., 2005. Cyber-harassment: A study of a new method for an old behavior, Journal of Educational Computing Research, 32: 265-277.
- Bozyigit, A., Utku, S., Nasibov, E. (2018). Sanal Zorbalık İçeren Sosyal Medya Mesajlarının Tespiti. (UBMK)
- Breiman, L., 2001. Random Forests. Machine Learning, 45(1), 5-32. doi: 10.1023/A:1010933404324

- Cassidy, W., Brown, K., and Jackson, M., 2012. 'Under the radar': Educators and cyberbullying in schools, *School Psychology International*, 33: 520-532.
- Chang, C., Lin, C., 2011. Libsvm: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology (TIST)*, v.2 n.3, p.1-27
- Dadvar, M., and Jong F. D., 2012. Improved cyberbullying detection through personal profiles, *International Conference on Cyberbullying*, Paris, France.
- Dadvar, M., Jong, F. D., Ordelman, R., And Trieschnigg, D., 2012. Improved cyberbullying detection using gender information, *Twelfth Dutch-Belgian Information Retrieval Workshop*, 23-25.
- Dinakar, K., Reichart, R., and Lieberman, H., 2011. Modelling the Detection of Textual Cyberbullying, *Social Mobile Web Workshop at International Conference on Weblog and Social Media*, Barcelona, Spain.
- Hinduja, S., & Patchin, J.W., 2009. *Bullying beyond the schoolyard: Preventing and responding to cyberbullying*. California: Carwin Pres
- Hinduja, S., and Patchin, J. W., 2008. Cyberbullying: An exploratory analysis of factors related to offending and victimization, *Deviant Behavior*, 29: 129-156.
- Katzer, C., Fetchenhauer, D., and Belschak, F., 2009. Cyberbullying: Who Are the Victims? A Comparison of Victimization in Internet Chatrooms and Victimization in School, *Journal of Media Psychology*, 21(1): 25-36, DOI 10.1027/1864-1105.21.1.25.
- Kavzoğlu, T., 2014. "Heyelan Duyarlılık Analizinde Ki-Kare Testine Dayalı Faktör Seçimi", *V. Uzaktan Algılama ve Coğrafi Bilgi Sistemleri Sempozyumu*, 14-17, İstanbul.

- Koç, M., Horzum, M. B., Ayas, T., Aydin, F., Özbay, A., Uğur, E. & Çolak, S., 2016. Coping With Cyberbullying Scale: Study of reliability and validity. *Sakarya University Journal of Education*, 6(3), 116–128
- Kontostathis, A., Edwards, L., and Leatherman, A., 2010. Text mining and cybercrime, *Text Mining: Applications and Theory*, M. W. Berry and J. Kogan Eds., John Wiley and Sons, New York, NY.
- Korde, V., Mahender, C. N., 2012. “Text classification and classifiers: a survey”, *International Journal of Artificial Intelligence & Applications*, 3(2): 85-99
- Kowalski, R., & Limber, S., 2007. Electronic bullying among middle school students. *Journal of Adolescent Health*, 41(6), 22-30.
- Li, Q., 2006. Cyberbullying in schools: A research of gender differences. *School Psychology International*, 27(2): 157-170.
- Mason, K., 2008. Cyberbullying: A preliminary assessment for the school personnel. *Psychology in the Schools*, 45, 323–348.
- Mccallum, A., Nigam, K., 1998. A comparison of event models for Naive Bayes text classification. *AAAI-98 Workshop on Learning for Text Categorization*, pp. 41–48.
- Munezero, M., Mozgovoy, M., Kakkonen, T., Klyuev, V., And Sutinen, E., 2013. Antisocial behavior corpus for harmful language detection, *Federated Conference on Computer Science and Information Systems*, Krakow, Poland.
- Nahar, V., Li, X., Pang, C., & Zhang, Y., 2013. Cyberbullying detection based on text-stream classification. In *The 11th Australasian Data Mining Conference (AusDM 2013)*.
- Nahar, V., Unankard, S., Li, X., and Pang, C., 2012. Sentiment analysis for effective detection of cyberbullying, *14th Asia-Pacific international conference on Web Technologies and Applications APWeb 2012*, 767-774.

- Olweus, D., 1993. *Bullying at school: What we know and what we can do.* Cambridge, MA: Blackwell Publishers, Inc.
- Özel, S. A., Saraç, E., Akdemir, S., Aksu, H., 2017. "Detection of cyberbullying on social media messages in Turkish," in 2017 International Conference on Computer Science and Engineering (UBMK), pp. 366–370.
- Özel, S. A., Saraç, E., 2017. Effects of Feature Extraction and Classification Methods on Cyberbully Detection. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 21 (1), 190-200. Retrieved from <http://dergipark.org.tr/sdufenbed/issue/30898/334429>.
- Öztemel, E., 2003. *Yapay Sinir Ağları*, Papatya Yayınları, İstanbul.
- Parris, L., Varjas, K., Meyers, J., & Cutts, H., 2012. High school students' perceptions of coping with cyberbullying. *Youth & Society*, 44(2), 284-306.
- Quinlan, J. R., 1993. *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA.
- Saad, M. K., 2010. Arabic Morphological Tools for Text Mining. 6 th International Conference on Electrical and Computer Systems (EECS'10)
- Salton, G., 1968. *Automatic Information Organization and Retrieval*, McGraw Hill Book Co., New York, chapter 4.
- Saraç, E. and Özel, S. A., 2013. Web Page Classification Using Firefly Optimization. 2013 IEEE International Symposium on Innovations in Intelligent Systems and Applications in Albena, Bulgaria, IEEE, 1-5.
- Saraç, E., 2016. *Selecting Optimum Feature Subsets With Nature Inspired Algorithms for Cyberbully Detection*. Çukurova University.
- Sayar, K., 2009. *Ruh Hali*. İstanbul: Timaş Yayınları.
- Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., Wang, Z., 2007. A novel feature selection algorithm for text categorization. *Expert Systems with Applications*, 33(1), 1–5. doi:10.1016/J.ESWA.2006.04.001 [63]

- Shariff, S. & Gouin, R., 2005. Cyber - dilemmas: Gendered hierarchies free expression and cyber-safety in schools, presented at safety and security in a networked world: balancing cyber-rights and responsibilities. Oxford Internet Institute Conference, Oxford, U.K.
- Slonje, R. & Smith, P. K., 2008. Cyberbullying: Another main type of bullying? *Scandinavian Journal of Psychology*, 49, 147-154.
- Willard, N.E., 2006. *Cyberbullying and Cyberthreats: Responding to the challenge of online social cruelty, threats and distress*. Eugene, Oregon: Center for Safe and Responsible Internet Use.
- Yang, Y., Pedersen, J. O., 1997. A Comparative Study on Feature Selection in Text Categorization, 412-420.
- Yates, F., 1934. Contingency tables involving small numbers and the χ^2 test, *Supplement to the Journal of the Royal Statistical Society*, 1: 217-235.
- Ybarra, M.L., and Mitchell, K.J., 2007. Prevalence and frequency of Internet harassment instigation: implications for adolescent health, *J Adolesc Health* 41 (2): 189- 95.doi:10.1016/j.jadohealth.2007.03.005. PMID 17659224.
- Yin, D., Xue, Z., Hong, L., Davison, B. D., Kontostathis, A., And Edwards, L., 2009. Detection of harassment on Web 2.0, *Content Analysis in the WEB 2.0 (CAW2.0)*, Madrid, Spain.
- Yu, E. S., Liddy, E. D., 1999. Feature selection in text categorization using the Baldwin effect. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339) (Vol. 4, pp. 2924–2927)*. IEEE. doi:10.1109/IJCNN.1999.833550



CURRICULUM VITAE

Erhan Öztürk was born in Adana, in 1989. He has completed his elementary education at Adana. He went to high school at Adana Kurttepe Anatolian High School. He has completed his university education at department of Computer Engineering of Çukurova University in 2013. Since 2019, he has been working as UI/UX Designer and Computer Engineer at Directorate of Computing of Yıldız Technical University in İstanbul.





APPENDIX



Appendix 1: List of Stopwords

1	A	72	Dolayısıyla	143	Kimi	214	Şekilde
2	Acaba	73	Dört	144	Kimin	215	Sekiz
3	Altı	74	E	145	Kimisi	216	Seksen
4	Altmış	75	Edecek	146	Kimse	217	Sen
5	Ama	76	Eden	147	Kırk	218	Senden
6	Ancak	77	Ederek	148	Madem	219	Seni
7	Arada	78	Edilecek	149	Mi	220	Senin
8	Artık	79	Ediliyor	150	Mı	221	Şey
9	Asla	80	Edilmesi	151	Milyar	222	Şeyden
10	Aslında	81	Ediyor	152	Milyon	223	Şeye
11	Aslında	82	Eğer	153	Mu	224	Şeyi
12	Ayrıca	83	Elbette	154	Mü	225	Şeyler
13	Az	84	Elli	155	Nasıl	226	Şimdi
14	Bana	85	En	156	Ne	227	Siz
15	Bazen	86	Etmesi	157	Neden	228	Siz
16	Bazı	87	Etti	158	Nedenle	229	Sizden
17	Bazıları	88	Ettiği	159	Nerde	230	Sizden
18	Belki	89	Ettiğini	160	Nerede	231	Size
19	Ben	90	Fakat	161	Nereye	232	Sizi
20	Benden	91	Falan	162	Neyse	233	Sizi
21	Beni	92	Filan	163	Niçin	234	Sizin
22	Benim	93	Gene	164	Nin	235	Sizin
23	Beri	94	Gereği	165	Nın	236	Sonra
24	Beş	95	Gerek	166	Niye	237	Şöyle
25	Bile	96	Dolayısıyla	167	Nun	238	Şu
26	Bilhassa	97	Dört	168	Nün	239	Şuna
27	Bin	98	E	169	O	240	Şunları
28	Bir	99	Edecek	170	Öbür	241	Şunu
29	Biraz	100	Eden	171	Olan	242	Ta
30	Birçoğu	101	Ederek	172	Olarak	243	Tabii
31	Birçok	102	Edilecek	173	Oldu	244	Tam
32	Biri	103	Ediliyor	174	Olduğu	245	Tamam
33	Birisi	104	Edilmesi	175	Olduğunu	246	Tamamen
34	Birkaç	105	Ediyor	176	Olduklarını	247	Tarafından
35	Birşey	106	Eğer	177	Olmadı	248	Trilyon
36	Biz	107	Elbette	178	Olmadığı	249	Tüm
37	Bizden	108	Elli	179	Olmak	250	Tümü

38	Bize	109	En	180	Olması	251	U
39	Bizi	110	Etmesi	181	Olmayan	252	Ü
40	Bizim	111	Etti	182	Olmaz	253	Üç
41	Böyle	112	Ettiği	183	Olsa	254	Un
42	Böylece	113	Ettiğini	184	Olsun	255	ÜN
43	Bu	114	Fakat	185	Olup	256	Üzere
44	Buna	115	Falan	186	Olur	257	Var
45	Bunda	116	Filan	187	Olur	258	Vardı
46	Bundan	117	Gene	188	Olursa	259	Ve
47	Bunlar	118	Gereği	189	Oluyor	260	Veya
48	Bunları	119	Gerek	190	On	26	Ya
49	Bunların	120	Dolayısıyla	191	Ön	262	Yani
50	Bunu	121	Dört	192	Ona	263	Yapacak
51	Bunun	122	E	193	Önce	264	Yapılan
52	Burada	123	Edecek	194	Ondan	265	Yapılması
53	Bütün	124	Eden	195	Onlar	266	Yapıyor
54	Çoğu	125	Ederek	196	Onlara	267	Yapmak
55	Çoğunu	126	Edilecek	197	Onlardan	268	Yaptı
56	Çok	127	Ediliyor	198	Onları	269	Yaptığı
57	Çünkü	128	Edilmesi	199	Onların	270	Yaptığını
58	Da	129	Ediyor	200	Onu	271	Yaptıkları
59	Daha	130	Eğer	201	Onun	272	Ye
60	Dahi	131	Elbette	202	Orada	273	Yedi
61	Dan	132	Elli	203	Öte	274	Yerine
62	De	133	En	204	Ötörü	275	Yetmiş
63	Defa	134	Etmesi	205	Otuz	276	Yi
64	Değil	135	Etti	206	Öyle	277	Yı
65	Diğer	136	Ettiği	207	Oysa	278	Yine
66	Diğeri	137	Ettiğini	208	Pek	279	Yirmi
67	Diğerleri	138	Fakat	209	Rağmen	280	Yoksa
68	Diye	139	Falan	210	Sana	281	Yu
69	Doksan	140	Filan	211	Sanki	282	Yüz
70	Dokuz	141	Gene	212	Sanki	283	Zaten
71	Dolayı	142	Gereği	213	Şayet	284	Zira