Expert Systems With Applications 174 (2021) 114802

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Detecting abusive Instagram comments in Turkish using convolutional Neural network and machine learning methods

Habibe Karayiğit [a], Çiğdem İnan Acı [b,*], Ali Akdağlı [a]

[a] Department of Electrical and Electronics Engineering, Mersin University, 33343, Turkey
[b] Department of Computer Engineering, Mersin University, 33343, Turkey

ARTICLE INFO

Keywords:
Abusive comment
Hate speech
Classification
Social media
Instagram
Dataset

ABSTRACT

Instagram is a free photo-sharing platform where each user has a profile and can upload photos for followers to view, like, and comment. Abusive comments on images can be humiliating and harmful to those who share photos. Developing a comment filter in languages other than English is difficult and time-consuming. This paper proposes a dataset called Abusive Turkish Comments (ATC) to detect abusive Instagram comments in Turkish. It is composed of a large number of Instagram comments posted to tabloid and sports accounts (i.e., 10,528 abusive and 19,826 not-abusive). It is the first public dataset dedicated to detecting abusive Turkish messages, as far as we know. The sentiment annotation has been done in sentence-level by assigning polarity to each comment. The performance of the abusive message detection models was evaluated using several performance metrics: Convolutional Neural Network (CNN), five well-known classifiers (i.e., Naive Bayes, Support Vector Machine, Decision Tree, Random Forest, and Logistic Regression), and two reweighted classifiers (i.e., Adaptive Boosting (AdaBoost), eXtreme Gradient Boosting (XGBoost)) were compared in terms of F1-score, precision, and recall. The results showed that the best performance (i.e., Micro-averaged F1-score: 0.974, Macro-averaged F1-score: 0.973, Kappa-value: 0.946) was yielded by the CNN model on the oversampled ATC dataset. The abusive message detection model proposed in this study can contribute to the development of Turkish comment filters on Instagram. Different model combinations are considered to select the best model that gives better recognition accuracy.

1. Introduction

The use of social media has increased with the increase in mobile devices. According to 2019 statistics, there are 59.36 million Internet users that constitute 72% of the population in Turkey. Turkey is in the top five in the world in the use of social media. A large proportion of Internet users (52 million) also use an active part in social media (Statista, 2020).

Instagram, a free photo (and video) capturing and sharing service, has quickly emerged as a new medium in the spotlight in recent years. It provides users with an instantaneous way to capture and share their life moments with friends through a series of (filter manipulated) pictures and videos. Since its launch in October 2010, it has attracted more than 1 billion active users, and more than 40 billion photos shared so far (Hu, Manikonda, & Kambhampati, 2014; Instagram, 2020). There were 38,870,000 Instagram users in Turkey in March 2020, which accounted for 46.74% of its entire population. Turkey is ranked 6th in the world in the use of Instagram (Statista, 2020).

Comments on social media are not always positive for users; there is also potential to harm (Tang & Dalzell, 2019). Abusive messages in social media such as indiscriminate slang, abusive language, and profanity should not be seen as just a message. They are a severe and ruthless tool that opens up to cyber violence (Lee, Lee, Park, & Han, 2018). For years, social media companies such as Twitter, Facebook, and YouTube have been combating this issue. It has been estimated that hundreds of millions of euros are invested every year in countermeasures, including human resources (Zhang, Robinson, & Tepper, 2018a). Kowalski, Giumetti, Schroeder, and Lattanner (2014) reported that there are some connections between failures, absenteeism, and drop-out behaviours of students who have been cyber-bullied. Cyberbullying has potentially devastating psychological effects such as depression, low self-esteem, suicide ideation, and even suicide (Van Royen, Poels, Daelemans, & Vandebosch, 2015). One out of every five people is exposed to cyberbullying in Turkey, and it is the same as the

* Corresponding author.
E-mail addresses: d2014242@mersin.edu.tr (H. Karayiğit), caci@mersin.edu.tr (Ç. İnan Acı), akdagli@mersin.edu.tr (A. Akdağlı).

https://doi.org/10.1016/j.eswa.2021.114802
Received 20 August 2020; Received in revised form 6 January 2021; Accepted 27 February 2021
Available online 3 March 2021

world average. The most commonly used methods for cyberbullying are sending messages and posting comments (BBC, 2020). It is crucial to detect and prevent these risks by anticipating them as much as possible.

Abusive expressions in Turkish are used to express the most substantial feelings of people, such as anger, passion, fear, and are the words that have the highest capacity to cause emotional pain and provoke violent conflict. They are generally related to sexuality, sexual orientation, and defecation. The effect of the abused word is related to how taboo it is for the person being abused. Although its use is seen as an undesired form of communication, it is frequently used, especially in social media.

Recent years have seen an increasing number of research on abusive speech detection and other related areas such as "offensive", "profane", "hate speech", and "cyberbullying". Manual analysis of abusive speech on social networks is challenging and time-consuming. Sentiment analysis extracts the users' thoughts, feelings, and desires from the comments on social media and distinguishes their polarity (Cambria, 2016; Hemmatian & Sohrabi, 2019). Sentiment-Based Text Categorization (SBTC) builds a model using the pre-tagged dataset and attempts to assign unlabeled data to the correct category. Text categorization based on sentiment analysis helps us to classify cyberbullying as positive and negative by looking at the SBTC labels (Sebastiani, 2002).

There have been many studies on sentiment analysis focus on extracting information from the subjective corpus like social media comments by using Machine Learning (ML) techniques (Al-garadi, Varathan, & Ravana, 2016; Balakrishnan, Khan, & Arabnia, 2020; Briliani, Irawan, & Setianingsih, 2019; Burnap & Williams, 2015; Chatzakou et al., 2019; Hosseinmardi et al., 2015; Ibrohim & Budi, 2018; Lee et al., 2018; Naf'an, Bimantara, Larasati, Risondang, & Nugraha, 2019; Pratiwi, Budi, & Alfina, 2018). Although the ML-based techniques have been widely used and have shown quite successful performance, they strongly depend on manually-defined features, where the feature definition requires much effort from domain experts. For this reason, deep learning techniques have been drawing attention recently, as they may reduce the effort for the feature definition and achieve relatively high performance (Kim & Jeong, 2019).

The deep learning model consists of a more significant number of hidden layers and neurons to represent the data with different abstractions. It works efficiently and effectively with large datasets. Deep and Convolutional Neural Networks (CNN) with many hidden layers are examples of this (Alayba, Palade, England, & Iqbal, 2017). CNN has been recently used to classify sentiments in social media comments (Ayata, Saraclar, & Özgür, 2017; Kim & Jeong, 2019; Mozafari, Farahbakhsh, & Crespi, 2020; Park & Fung, 2017; Shushkevich & Cardiff, 2019; Zhang & Luo, 2019; Zhang, Robinson, & Tepper, 2018b).

In this study, a dataset called ATC (Abusive Turkish Comments), which consists of 10,528 abusive and 19,826 not-abusive Instagram comments in Turkish, is presented (Karayiğit, Acı, & Akdağlı, 2020). The dataset has been balanced using the oversampling method, and the two forms of the dataset (i.e., oversampled ATC and the original ATC) have evaluated by an ANN-based classifier (i.e., CNN), five well-known ML-based classifiers (i.e., Naive Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF) and Logistic Regression (LR)), and two reweighted classifiers (i.e., Adaptive Boosting (AdaBoost), eXtreme Gradient Boosting (XGBoost)) to detect abusive comments in Turkish. Bag of Words (BoW), bi-gram, and tri-gram feature selection methods were used in the feature selection phase. Word2Vec (i.e., Continuous Bag of Words (CBoW) and Skip-Gram) and Bidirectional Encoder Representations from Transformers (BERT) were utilized for embeddings. Since Turkish is an agglutinative language, sub-word embedding is as important as word embedding for sentiment analysis. Byte Pair Encoding (BPE) was performed to build a sub-word dictionary. CBoW and Skip-Gram were used for word-level embeddings, BPE was used for sub-word level embedding, and BERT was used for sentence-level embedding. Besides, the pre-processing phase was evaluated with and without data cleaning to analyze the effects of hashtags and punctuation marks on the sentiment. The algorithms mentioned above were applied separately to the ATC dataset in different combinations, and the performance metrics were compared in order to find the best prediction model.

This paper's contributions can be summarized as follows: (1) We introduce a new dataset that makes possible the detection of abusive comments in Turkish. To the best of our knowledge, no existing work has been done on abusive Turkish comments on social media platforms. (2) The vast majority of SBTC studies have been done for Twitter. Although cyberbullying studies have increased for Instagram recently, the number of studies using Twitter datasets is relatively higher. However, the photos and videos shared on Instagram create a suitable environment for abusive comments than Twitter. (3) The proposed dataset and CNN algorithm achieved a good categorization performance in terms of F1-score, precision, and recall.

The rest of this paper is organized as follows: In Section 2, previous works on abusive speech analysis and existing datasets related to hate and abusive speech are discussed. Section 3 presents the materials and methods used in the study. Section 4 presents the experimental study and discusses the results. Finally, conclusions are given in Section 5.

2. Related studies

There are many studies on SBTC, and a variety of approaches have been developed. English has the highest number of sentiment analysis studies, while research is more limited for other languages, including Turkish. This section discusses several papers in the field of SBTC using either English or other languages.

Cyber-violence fed by negativities (e.g., harmful, offensive, and aggressive interactions) is different from traditional violence because humiliating texts or visual materials served on social media can be accessible to everyone (Heirman & Walrave, 2008). A common form of cyber-violence is a simple act of commenting on someone's post using abusive language (Jones, Mitchell, & Finkelhor, 2013). Hate speech can be defined as textual sharing on social media to create hate against an individual or group (Charitidis, Doropoulos, Vologiannidis, Papastergiou, & Karakeva, 2020). In recent years, there have been many studies about hate speech analysis (Charitidis et al., 2020; Mossie & Wang, 2020; Waseem, Thorne, & Bingel, 2018), harassment detection (Huang & Raisi, 2018), cyberbullying (Balakrishnan et al., 2020), aggression detection (Chatzakou et al., 2019), misogyny detection (Shushkevich & Cardiff, 2019) and offensive language analysis (Burnap & Williams, 2015).

Abusive language detection is included in the scope of hate speech, and it has become an important research area for SBTC (Chatzakou et al., 2017; Chen, McKeever, & Delany, 2017b; Ibrohim & Budi, 2018; Johnson, 2018; Waseem et al., 2018). Abusive speech targets individuals' race, ethnicity, national origin, religion, gender, sexual orientation, disability, or illness (Mossie & Wang, 2020). Abusive speech contains an element of massive humiliation directly.

Recent researches have been focused on the implementation of SBTC algorithms to datasets that are obtained from social network sources (i.e., Facebook, Twitter, Instagram, YouTube) for the detection of hate speech (Charitidis et al., 2020). The most preferred datasets have consisted of tweets in English (Davidson, Warmsley, Macy, & Weber, 2017; Golbeck et al., 2017; Kwok & Wang, 2013; Waseem, Zeerak & Hovy, 2016). Although most hate speech-related studies are in English, there are studies in different languages (Charitidis et al., 2020; Mossie & Wang, 2020; Vigna, Cimino, Dell'orletta, Petrocchi, & Tesconi, 2017; Wiegand, Siegel, & Ruppenhofer, 2018).

The use of abusive language is considered under the umbrella of hate speech. This terminology also covers profanity (the use of inappropriate words). However, many researchers refer to abusive language as the offensive language (Al-Hassan & Al-Dossari, 2019). The studies conducted for detecting offensive/abusive messages can be summarized in chronological order as follows: Park and Fung (2017) first proposed a

Table 1
Summary of the previous studies on detecting offensive/abusive messages and their best results: precision (P), recall (R), F1-score (F).

Author	Year	Platform	Language	Feature representation	Best Classifier	P	R	F
Park and Fung (2017)	2017	Twitter	English	Character and Word2vec	Hybrid CNN	0.71	0.75	0.73
Chen et al. (2017a)	2017	YouTube, Myspace, SlashDot	English	Word embedding	FastText	–	0.76	–
Pratiwi et al. (2018)	2018	Instagram	Indonesian	N-gram	FastText	–	–	0.68
Wiegand, Ruppenhofer et al. (2018)	2018	Twitter, Wikipedia, UseNet	English	Lexical, linguistics, and word embedding	SVM	0.82	0.80	0.81
Ibrohim and Budi (2018)	2018	Twitter	Indonesian	N-gram	Naive Bayes	–	–	0.86
Alakrot et al. (2018)	2018	YouTube	Arabic	N-gram	SVM	0.88	0.80	0.82
Chakraborty and Seddiqui (2019)	2019	Facebook	Bengali	Unicode, Bengali words, and emotions	SVM	78% Accuracy		
Briliani et al. (2019)	2019	Instagram	Indonesian	TF-IDF	KNN	0.98	0.98	0.98
Omar et al. (2020)	2020	Facebook, Twitter, Instagram, YouTube	Arabic	Word embedding	RNN	0.98	0.98	0.98

two-step approach to detecting abusive language, and then they classified it into specific types and compared it with a one-step approach of doing one multiclass classification on sexist and racist language. Chen, McKeever, and Delany (2017a) carried out experiments on abusive text detection using fundamental neural network techniques. They investigated an off-the-shelf neural network-based classifier (FastText). Pratiwi et al. (2018) used FastText as the classifier to conduct hate speech detection on Instagram comments for the Indonesian language. Wiegand, Ruppenhofer†, Schmidt, and Greenberg (2018) were constructed the lexicon of abusive words automatically. They used both corpora and lexical resources to detect abusive words on Twitter, Wikipedia, and UseNet platforms. Ibrohim and Budi (2018) conducted a study to detect the abusive language in Indonesian social media using ML-based methods. Alakrot, Murray, and Nikolov (2018) presented a predictive model for the detection of anti-social behavior in online communication in Arabic, such as comments which contain obscene or offensive words and phrases. They collected and labeled a large dataset of YouTube comments and trained an SVM classifier with combinations of word-level features, N-gram features, and a variety of pre-processing techniques. Chakraborty and Seddiqui (2019) built an ML-based automatic system for abusive language detection in the Bengali language. They have implemented NB, SVM, and CNN to classify the data as a threat and abusive or not. Briliani et al. (2019) created a system that detects hate speech or not on the Instagram comment section with the K-Nearest Neighbor (KNN) classification method in the Indonesian language. Omar, Mahmoud, and Abd-El-Hafeez (2020) collected datasets from Facebook, Twitter, Instagram, and YouTube and manually labeled by three Arabic annotators into two balanced classes: Hate and not hate. Twelve ML-based algorithms and two deep learning architectures were used. Recurrent Neural Network (RNN) outperformed other classifiers with an accuracy of 98.7%. Table 1 summarizes recent studies that deal with the detection of abusive behaviour on social media platforms.

Considering the social media platform where data is obtained, some studies have been carried out to detect cyberbullying messages on Instagram (Bimantara, Larasati, Risondang, Naf'an, & Nugraha, 2019; Hosseinmardi et al., 2015; Ozel, Sarac, Akdemir, & Aksu, 2017; Priyoko & Yaqin, 2019). Since the focus of our study is offensive/abusive comments, these studies were not included in Table 1.

3. Materials and methods

In this section, we present the details of the ATC dataset and the algorithms utilized for abusive comment detection.

3.1. Materials

3.1.1. Turkish language structure

Turkish belongs to the Altaic sub-group of the Ural-Altaic group of languages (Kılınc et al., 2016). The Turkish language, which uses the Latin alphabet, consists of twenty-one consonants and eight vowels (Parlar, Özel, & Song, 2019). Turkish is an agglutinative language such as Finnish and Hungarian, where a single verb in it can be translated into longer sentences (Çakıcı, Steedman, & Bozşahin, 2018). Therefore, names and verbs can cause incorrect sentence forms. Turkish words can also increase rapidly, and this condition can significantly increase the number of word forms (Eryiğit, Nivre, & Oflazer, 2008). Since Turkish is an agglutinative language, it is difficult to divide sentences into simpler parts (subject, verb, object) (El-Kahlout & Akin, 2013).

Table 2
The two versions of the ATC dataset.

Dataset	Number of Abusive Comments	Number of Not-abusive Comments
Original ATC	10,258	19,826
Oversampled ATC	19,826	19,826

3.1.2. Abusive Turkish comments (ATC) dataset

The dataset acquisition step is fundamental, and it is the most time-consuming part of the text categorization process (Omar et al., 2020). Instagram accounts of a Turkish magazine page (i.e., @2.sayfaofficial), football teams (i.e., @fenerbahce, @galatasaray), and football players (i.e., @ardaturan, @1volkandemirel) which contains controversial issues and opinions were chosen to build the ATC dataset. It is more likely to find abusive and hateful comments in such accounts. According to the Turkish slang dictionary (TDK, 2020), all samples in the dataset were manually labeled as abusive or not-abusive. The ATC dataset contains 10,528 abusive and 19,826 not-abusive comments which were posted between 2017 and 2019 years. Python programming language and an Application Programming Interface (API) were coded to collect the text data from Instagram. The ATC dataset is publicly available (Karayiğit et al., 2020) for researchers' non-commercial use.

As can be seen from the number of abusive and not-abusive comments, the ATC dataset is imbalanced. The distribution of documents in classes is highly variable; one class may contain a few terms, while the other class may have a large number of terms in imbalanced datasets (Agnihotri, Verma, & Tripathi, 2017). When an imbalance class exists in a dataset, classification algorithms such as DT and SVM mainly consider the majority class. The classifiers usually treat labels in the minority class as mislabeled (Le, Hoang Son, Vo, Lee, & Baik, 2018). Therefore, balancing techniques such as undersampling and oversampling are used to solve the problem. The undersampling technique balances the data by reducing the number of the majority class. Contrary to undersampling, a sufficient number of random samples is added to minority data by oversampling (Gazzah & Amara, 2008). Since the ATC dataset is not large, it is more reasonable to use the oversampling method rather than undersampling (Table 2).

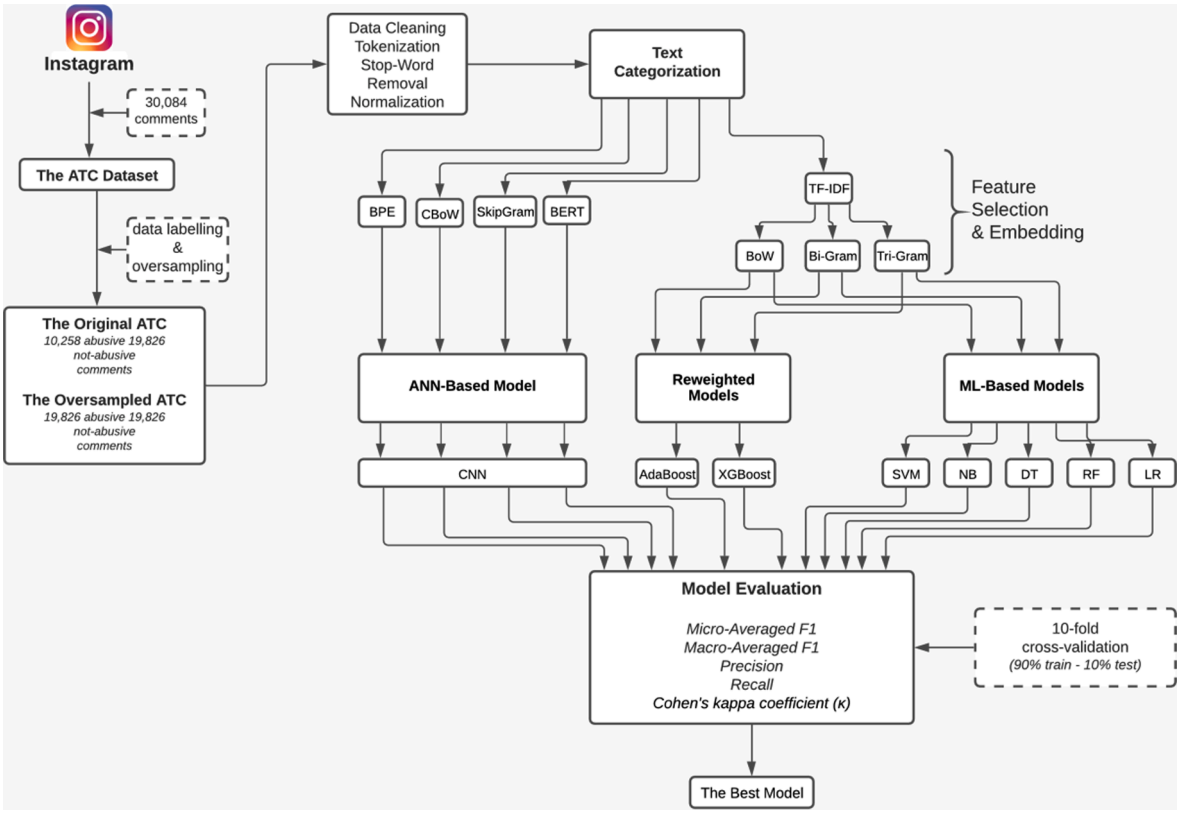

Fig. 1. The overall design of the abusive content detection system.

Table 3
The algorithms used in the experiments.

Dataset	Feature Extraction & Embedding	Classifiers	Training Set (90%)	Test Set (10%)
Original ATC	BoW + TF-IDF bi-gram + TF-IDF tri-gram + TF-IDF BPE CBoW Skip Gram BERT	CNN, NB, SVM, DT, RF, LR, Adaboost XGBoost	27,076 comments	3,008 comments
Oversampled ATC			35,687 comments	3,965 comments

3.2. Methods

The purpose of this study is to create an SBTC system that detects abusive texts. The best abusive comment detection model was chosen between different feature extraction, embedding, and text categorization methods. This section gives general information about the algorithms used in the study. Since all of the algorithms are well-known techniques, they are briefly explained, and theoretical details are given in references. The overall model design is presented in Fig. 1.

The abusive comment detection system has several data processing steps: labeling, pre-processing, feature selection, and categorization. First, Instagram data was obtained, and the ATC dataset was built. After that, the dataset was manually labeled into two classes (i.e., abusive, not-abusive), and the oversampling was carried out to create one more dataset. BPE, Word2Vec, and BERT embedding models were used as representations of comments. BoW, bi-gram, and tri-gram feature selection methods were utilized with Term Frequency-Inverse Document Frequency (TF-IDF) for categorization using ML-based classifiers.

The ATC dataset was tested using 10-fold cross-validation. In 10-fold cross-validation, the original sample is randomly partitioned into ten equal size subsamples. Of the ten subsamples, a single subsample is retained as the validation data for testing the model, and the remaining nine subsamples are used as training data. The cross-validation process is then repeated ten times (the folds), with each of the ten subsamples used exactly once as the validation data. The ten results from the folds can then be averaged (or otherwise combined) to produce a single estimation. This method's advantage is that all observations are used for both training and validation, and each observation is used for validation exactly once.

Table 3 shows an overview of the algorithms used in the categorization phase. Experiments have been conducted with different combinations of these algorithms to find out the best categorization model.

3.2.1. Pre-Processing

The pre-processing step eliminates unnecessary data and converts it to a structure that allows statistical steps to be performed (Omar & Al-Tashi, 2018). Well-known pre-processing steps used in this study are given as follows:

- ***Data cleaning***: URLs, hashtags, numeric characters, and punctuation marks are generally considered non-crucial for natural language processing applications. However, simply removing hashtags and punctuation, etc., might not be the best way to clean the textual data. The hashtags generally contain rich semantic information that could be important for abusive comment detection, and punctuation marks can be alternative emojis to express the users' options. Therefore, the pre-processing phase has been conducted with/without the data cleaning step to monitor its effects on the results.
- **Tokenization**: Comments in the ATC dataset were tokenized based on whitespaces and punctuation marks.
- **Stop-word removal**: Stop words (i.e., formatting tags, digits, prepositions, pronouns, conjunction, and auxiliary verbs) were removed from the ATC dataset.

- **Normalization**: All texts were converted to lowercase.

3.2.2. Feature selection and weighting

Feature selection is a method that removes the irrelevant features from datasets before using classifiers if they are not related to the wanted result (Sebastiani, 2002). BoW and n-gram are the most commonly used and traditional feature selection methods.

- The BoW approach simply considers all text as an unordered set of word features, with each separate word becoming a single feature (H. Chen et al., 2017b).
- N-gram models are among the most used feature selection techniques in automatic abusive speech detection and related tasks (i.e., hateful, aggression, toxic, racism, sexism). The N-gram models predict the existence of n-1 words before the word is analyzed (Mossie & Wang, 2020). In this study, we have implemented the word-based n-gram models where the length of n varies from 2 to 3.
- TF-IDF is a popular method that is used to calculate the weight of terms (Wang, Wang, & Zhang, 2010). The IDF weighting scheme indicates the importance of a term in a text and selects the best terms with the highest weight (Al-Radaideh & Al-Abrat, 2019). In this study, various combinations of feature selection methods with TF-IDF weighting were evaluated to figure out how lexical features would affect accuracy.

3.2.3. Word2Vec, BPE and BERT

Word embeddings represent words in a vector space, where the position is based on how they are used in combination with other words. Each word w is represented as a word vector, and distances between these vectors encode relationships between words. Trained models can be either context-free or contextual. In the context-free (or context-independent) models, each word has a single unique vector that does not vary with the other words found around it in a document (Babaeianjelodar, Lorenz, Gordon, Matthews, & Freitag, 2020).

Word2Vec, a word embedding system developed by Mikolov, Sutskever, Chen, Corrado, and Dean (2013), is a shallow neural network with one hidden layer which takes text from the input layer and produces vector representations of the words as output. There are two Word2Vec models: Skip-Gram and the CBoW. In the CBoW model, the distributed representations of context (or surrounding words) are combined to predict the word in the middle. While in the Skip-Gram model, the distributed representation of the input word is used to predict the context. They are trained self-supervised on sequences of tokens with a loss function that is based on the categorical cross-entropy of an approximation to the softmax function (Chamberlain, Rossi, Shiebler, Sedhain, & Bronstein, 2020).

In word embedding models, words are mapped or "embedded" into low-dimensional real-valued vectors. Such mapping is based, implicitly or explicitly, on word co-occurrence statistics (Levy & Goldberg, 2014). Naturally, frequent words provide a better representation of their distributional properties; thus, the quality of word embeddings is in direct relation to the frequency of words (Drozd, Gladkova, & Matsuoka, 2015). However, even in large corpora, most words occur very few times. Most of the in-vocabulary words (for a given task/corpora) have to be discarded or embedded into low-quality vectors. Therefore, word-level models suffer from data sparsity. Another issue with word-level models is that they do not make use of morphological information. Different forms of the same word are treated as completely unrelated entities (Li, Drozd, Liu, & Du, 2018).

Sub-word level information is crucial for capturing morphology and improving compositional representations for out-of-vocabulary entries (Li et al., 2018). BPE (Sennrich, Haddow, & Birch, 2016) is a variable-length encoding that views the text as a sequence of symbols and iteratively merges the most frequent symbol pair into a new symbol (Heinzerling & Strube, 2018).

BPE performs a statistical analysis of the training dataset to discover common symbols within a word, such as consecutive characters of arbitrary length. Starting from symbols of length 1, BPE iteratively merges the most frequent consecutive symbols to produce new longer symbols. Note that for efficiency, pairs crossing word boundaries are not considered. In the end, we can use such symbols as sub-words to segment words (Zhang, Lipton, Li, & Smola, 2020).

BPE brings the perfect balance between character- and word-level hybrid representations, which makes it capable of managing large corpora. This behavior also encodes any rare words in the vocabulary with appropriate sub-word tokens without introducing any "unknown" tokens.

BERT (Devlin, Chang, Lee, & Toutanova, 2018) is a new language representation model, which uses a Bidirectional Transformer Network to pre-train a language model on a large corpus, and fine-tunes the pre-trained model on other tasks. The task-specific BERT design can represent either a single sentence or a pair of sentences as a consecutive array of tokens. Its input representation is constructed by summing its corresponding token, segment, and position embeddings for a given token. BERT is designed as a deeply bidirectional model. The network effectively captures information from both the right and left context of a token from the first layer itself and all the way through to the last layer. For a classification task, the first word of the sequence is identified with a unique token, and a fully-connected layer is connected at the position of the last encoder layer. Finally, a softmax layer completes the sentence or sentence-pair classification (Gao, Feng, Song, & Wu, 2019).

3.2.4. Text categorization algorithms

Text categorization is the process of assigning a given text to one or more categories. This process is considered a supervised classification technique since pre-classified documents are provided as a training set. The goal of text categorization is to assign a category to a new document (Fatima & Srinivasu, 2017). Brief information about the text categorization algorithms used in the study are given below:

Deep learning put forward by Hinton et al. in 2006 was a class of unsupervised learning (Hinton & Salakhutdinov, 2006). Its concept comes from the studies of Artificial Neural Networks (ANN). A multilayer perceptron with multiple implicit strata is a deep learning structure. By combining lower level features to form more abstract, higher-level representing property classifications or features, deep learning covers distributed feature representation of data (Wang & Raj, 2017).

Deep learning technology is applied in common NLP (natural language processing) tasks, such as semantic parsing (Yih, He, & Meek, 2014), information retrieval (Shen, He, Gao, Deng, & Mesnil, 2014), sentimental analysis (Severyn & Moschitti, 2015), text categorization (Liu, Qiu, Chen, Wu, & Huang, 2015), summarization (Marujo et al., 2016), and text generation (He & Deng, 2017). CNN has been shown as a powerful deep learning tool in image analytics, especially for feature extraction with transfer learning. CNN has been adapted to text categorization and showed to be useful for categorization tasks in which we expect to find strong local clues regarding class membership, such as a few key sentences or phrases. Convolution layers involve one-dimensional (1D) convolution with a small size kernel to extract features and max-pooling to condense or summarize the features extracted from the convolution. Finally, the fully connected layer takes the features through activations, fits the training data, and makes predictions (Wei, Qin, Ye, & Zhao, 2018).

Although the success of the CNNs in categorizing text is known, five well-known ML-based algorithms (i.e., NB, SVM, DT, RF, and LR) were employed to compare the categorization performance. Each of those classifiers is explained briefly in the following paragraphs:

The NB classifier relies on conditional probability (Abooraig et al., 2018), and it has high bias and low variance. NB classifier uses Bayes theorem, which assumes features that are conditionally independent. The advantages of the NB algorithm are simplicity, robustness, and interpretability (Hmeidi et al., 2015). NB is frequently used in SBTC due to its simplicity and computational capabilities (Bay & Çelebi, 2016).

The SVM classifier is used to categorize linear and non-linear data (Abooraig et al., 2018), and it is a perfect classifier to extract abusive expressions. SVM creates an n-dimensional hyper-plane that divides the dataset into two classes (Han, Kamber, & Pei, 2011). The advantages of the SVM classifier are robustness against noise, low risk of overfitting, and solving complex classification problems due to its kernel functions (Fatima & Pasha, 2017). It is used frequently in large textual datasets. Key disadvantages are the requirement to formulate the problem as a 2-class classification, longer training time, and being a black box model making it hard to evaluate (Renjith, Sreekumar, & Jathavedan, 2020).

The DT algorithm uses a top-down approach with if-then conditional expressions (Bay & Çelebi, 2016). Leaf nodes in its structure indicate the class with the highest number of training examples to find this node (Demirsoz & Ozcan, 2017). The advantages of the DT are simplicity, interpretability, and the ability to handle features (Hmeidi et al., 2015).

The RF algorithm creates different DTs, and final choices are taken by evaluating the individual trees (Mahmood et al., 2020). RF can show a high classification performance even in large data (Kilinç et al., 2017). If there are enough trees in the RF algorithm, the likelihood of an over adaptation problem is reduced in categorization.

LR is an algorithm that is commonly used for hateful based text categorization problems. The advantage of LR is that it interprets the text categories learned from the training set more easily than other classifiers. The disadvantage of LR is that it needs more data than different classifiers for more classification results (Segura-Bedmar, Colón-Ruíz, Tejedor-Alonso, & Moro-Moro, 2018).

Boosting is a *meta*-learning technique designed to improve the classification performance of imbalanced datasets. In recent years, significant efforts have been devoted to improving the performance of traditional classifiers on imbalanced datasets, largely following two trajectories: the data level and the algorithmic level. By directly manipulating the class distributions (data level), positive samples can be better represented in the dataset (i.e., oversampling). Different costs can be assigned to minority and majority classes at the algorithmic level, effectively updating the objective function. Also, different weights can be given to minority and majority classes so that minority class instances take higher importance when determining the class label of a query point (Seiffert, Khoshgoftaar, Van Hulse, & Napolitano, 2008; Yuan & Ma, 2012).

AdaBoost (Freund & Schapire, 1996) creates a set of base classifiers sequentially by adaptively adjusting the weights over instances in the training set. In AdaBoost, instances that are misclassified by the current classifier are given higher weights and vice versa. By doing so, classifiers are forced to concentrate on instances that are not properly identified by previous classifiers, and diverse base classifiers can be created. In theory, the upper bound of the training error of AdaBoost decreases monotonically as long as the base classifier is a little more accurate than a random guess. Each base classifier is also weighted when combined to create the final decision function (Yuan & Ma, 2012).

XGBoost is derived from the gradient boosting decision tree (Friedman, 2001) and proposed by Chen and Guestrin (2016). Similar to gradient boosting, XGBoost is combining a weak base classifier into a stronger classifier. At each iteration of the training process, the residual of a base classifier is used in the next classifier for optimizing the objective function. XGBoost provides a parallel tree boosting that solves many data science problems in a fast and accurate way (Shi et al., 2019).

3.2.5. Performance metrics

To evaluate the performance of proposed categorization models, we have used four evaluation metrics: F1-score (micro and macro averaged), Cohen's kappa coefficient (κ), precision, and recall. Definitions of performance metrics used in our study are given below:

The confusion matrix provides the basis for evaluating the performance

of any classifier with the help of its four components, True Positive (TP), False Negative (FN), False Positive (FP), and True Negative (TN). TP represents the number of abusive comments labeled as abusive while TN is the number of not-abusive comments labeled as not-abusive. Also, FP represents the number of abusive comments labeled as not-abusive, and FN is the number of not-abusive comments labeled as abusive.

Classification accuracy of a classifier on a given dataset refers to the percentage of test set tuples correctly classified by the classifier. It reflects how well the classifier recognizes tuples of various classes (Mohan Patro & Ranjan Patra, 2015). Precision has a significant role in classifier performance in any statistical analysis. The number of item labeled classes is divided by the sum of numbers labeled elements of a class, including both correct and incorrect classifications. The recall is a value of true classification that comes after dividing the total classification of a class. Low recall depicts how many known numbers are missing from the class. F1-score is calculated through precision and recall, and it shows the accuracy of the sentences (Dwivedi, Aggarwal, Keshari, & Kumar, 2019).

F1-score (Eq. (1)) is expressed as the harmonic mean of precision (Eq. (2)) and recall (Eq. (3)):

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

$$\text{F1} - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

Macro-averaged F1 (Eq. (4)) determines the measure for each label and then calculates the average over the number of labels in the dataset.

$$\text{Macro\_averaged F1} = \frac{1}{|Classes|} \sum_{i \in Classes} \text{F1} - \text{score}(i) \tag{4}$$

where *Classes* refer to the labeled class blocks.

Micro-averaged F1 (Eq. (5)) first determines the summation of TP, FP, FN, and TN for each label, respectively, and then finds the measures over these results. This formula is described in Eq. (5):

$$\text{Micro\_averaged F1} = \frac{1}{D} \sum_{i \in Classes} |\text{i}|\text{F1\_score}(i) \tag{5}$$

where D is the count of samples in the test set and $|i|$ is the cardinality of class i (Terragni, Fersini, & Messina, 2020).

The macro-averaged F1 metric is more suitable for fair evaluation than the micro-averaged F1 in imbalanced datasets, as it shows the different abilities of the classifier in categories with few documents more effectively (Dogan & Uysal, 2019). If a micro-averaged F1-score is used, it will be better to get the classification results with a macro-averaged F1-score. Because the macro-averaged F1-score ensures the arithmetic average of true positives' F1 and false negatives' F1, it also gives the true classification accuracy value even in imbalanced datasets (Saraç & Özel, 2016).

The κ value uses classifier performances as a benchmark to test the robustness and usefulness of datasets. It shows the degree of agreement between the predicted and actual samples for all categories (0, 1). The κ value approaches + 1 indicates a perfect match between the two observation results while it approaches −1 indicates incompatibility (Kilic, 2015).

4. Results and discussions

The experiments were performed separately on the two versions of the ATC dataset, and the results were obtained using 10-fold cross-validation. The results were then averaged across the folds, using suitable performance measures, to determine the accuracy of the positive and negative classes' detection.

All of the classifiers were applied and tested using a Python-based Scikit-Learn Library (Scikit, 2020). Appropriate parameter values of

Table 4
CNN model's hyperparameters and descriptions.

Parameter	Value	Description
Convolutional layer	Three 1D convolution layers	Learning features from the input data.
Kernel size	2,3,4	Specifying the length of the 1D convolution window.
Padding	same	Padding evenly to the left/right or up/down of the input
Activation	ReLu and Sigmoid	Adjusting output values
Filters	100,50,50	Extracting the output feature maps
Pooling	GlobalMaxPooling1D	Downsampling the input representation by taking the maximum value over the time dimension.
Dense Layer	The first dense layer has 512 neurons, and the second one has 1 neuron.	Returning the output
Optimizer	ADAM	Optimizer that implements the ADAM algorithm.
Loss	binary_crossentropy	Calculating the gradients
Epochs	30 for the BPE model, 15 for others	The number of times the dataset is passed forward and backward through the CNN
Batch Size	128	The number of samples that need to be worked on before updating internal model parameters

the classifiers were identified by grid-search. The grid-search is a precise searching method through a default specified subset of an ML classifier (Aya, Ormanci Acar, & Tufekci, 2016).

All of the experiments have been carried out on Google's free Colaboratory service (Google, 2020) to speed up the model's training. Google Colaboratory is a free Jupyter notebook environment that does not require setup and runs entirely in the cloud. The training time is the time required to train the categorization model using the specified algorithm combination with 90% of the dataset. The test time is the time required to test the categorization model using the same algorithm combination with 10% of the dataset.

The pre-processing step was carried out with/without cleaning punctuation marks, hashtags, stop words (Karayiğit et al., 2020), emojis, mentions, and web links in order to observe the effects of the data cleaning phase on the results. An emoji (i.e., Unicode: 1F4A9) has not removed in both "data cleaning" and "no data cleaning" experiments, as it is frequently used in abusive expressions.

The following subsections present the technical details and experimental results of each abusive comment detection model.

4.1. Experimental results of the ANN-Based model

CNNs are a specific type of ANNs best known for processing data with grid-like structure (Abroyan, 2017). CNN models consist of two main layers: convolutional and fully connected. The convolutional layer includes convolution and pooling operations. Convolution operation represents the extraction of the features of the text depending on the selected kernel. Pooling operation refers to reducing in the dimensions of the obtained feature map according to pooling size and selected methods such as average and maximum pooling (Ornek, Ceylan, & Ervural, 2019).

Activation functions were used to set the output values. The rectified linear unit (ReLu) activation function converts the input values (a) to$f(a) = \max(0, a)$. On the other hand, the sigmoid activation function converts the input value (a) to $f(a) = 1/(1 + e - a)$, which is a value between 0 and 1.

We have employed a CNN model as an ANN-based classifier for abusive message detection. We have applied the CNN model on the original and oversampled ATC dataset by combining different embedding methods (i.e., CBoW, SkipGram BPE, and BERT) and pre-processing steps (i.e., with/without data cleaning).

Table 4 shows the parameters used in the CNN classifier. Table 5 gives the average of 10-fold cross-validation results (i.e., micro/macro-averaged F1-scores and κ values) and training/test durations of the CNN models. Fig. 2 presents the Precision and Recall results of each fold on the CNN model combinations.

As shown in Table 4, the CNN model imported from Keras (Keras, 2020) includes three convolutional, one global max pooling, and two dense layers. The convolutional layers consist of 100, 50, 50 filters (activation function = ReLu), respectively. The first dense layer consists of 512 neurons (activation function = ReLu) and the second layer, called the output layer, includes one neuron (activation function = sigmoid). We have used Adaptive Moment Estimation (ADAM) as an optimizer. It is an algorithm for the first-order gradient-based optimization of stochastic objective functions and also based on lower-order moments adaptive estimates (Huang, Cao, & Wang, 2019). We have adopted Binary Cross Entropy as a loss function and set the batch size as 128, the weight decay as 0.0001, the momentum as 0.9, and the number of training epochs as 15.

We have employed a Turkish corpus (Wikipedia, 2020) to build the CBoW and Skip-Gram embeddings. Embedding vector dimension was selected as 400, and sliding window maximum length was selected as 5. A pre-trained embedding model (Heinzerling & Strube, 2018) was adopted for Turkish sub-word embedding (BPEmb_TR, 2020). The pre-trained distributed model's vocabulary size was selected as 25,000,

Table 5
Average F1-scores and κ values of the CNN model using 10-fold cross-validation.

Dataset	Algorithm Combination of the Experiment	Micro-averaged F1	Macro-averaged F1	Cohen's kappa coefficient (κ)	Avg. Training Time (sec.)	Avg. Test Time (sec.)
Original ATC	NDC + CBoW + CNN	0.924	0.914	0.829	1552.744	0.769
	NDC + Skip-Gram + CNN	0.925	0.914	0.829	1573.601	0.788
	NDC + BPE + CNN	0.913	0.906	0.803	2379.531	2.026
	NDC + BERT + CNN	0.947	0.942	0.883	280.012	0.039
	DC + CBoW + CNN	0.918	0.930	0.834	1482.808	0.748
	DC + Skip-Gram + CNN	0.923	0.913	0.827	1462.641	0.579
	DC + BPE + CNN	0.931	0.924	0.847	2214.171	2.038
	DC + BERT + CNN	0.942	0.939	0.878	228.979	0.008
Oversampled ATC	NDC + CBoW + CNN	0.971	0.971	0.941	2001.791	0.961
	NDC + Skip-Gram + CNN	0.970	0.969	0.938	2051.781	1.009
	NDC + BPE + CNN	0.945	0.945	0.893	2771.635	2.586
	NDC + BERT + CNN	0.973	0.974	0.940	411.745	0.009
	DC + CBoW + CNN	**0.974**	**0.973**	**0.946**	**1996.982**	**1.016**
	DC + Skip-Gram + CNN	0.967	0.968	0.941	2191.014	64.905
	DC + BPE + CNN	0.956	0.956	0.913	2805.852	2.468
	DC + BERT + CNN	0.970	0.970	0.939	406.056	0.009

* DC = Data Cleaning; NDC = No Data Cleaning; sec = second; avg = average

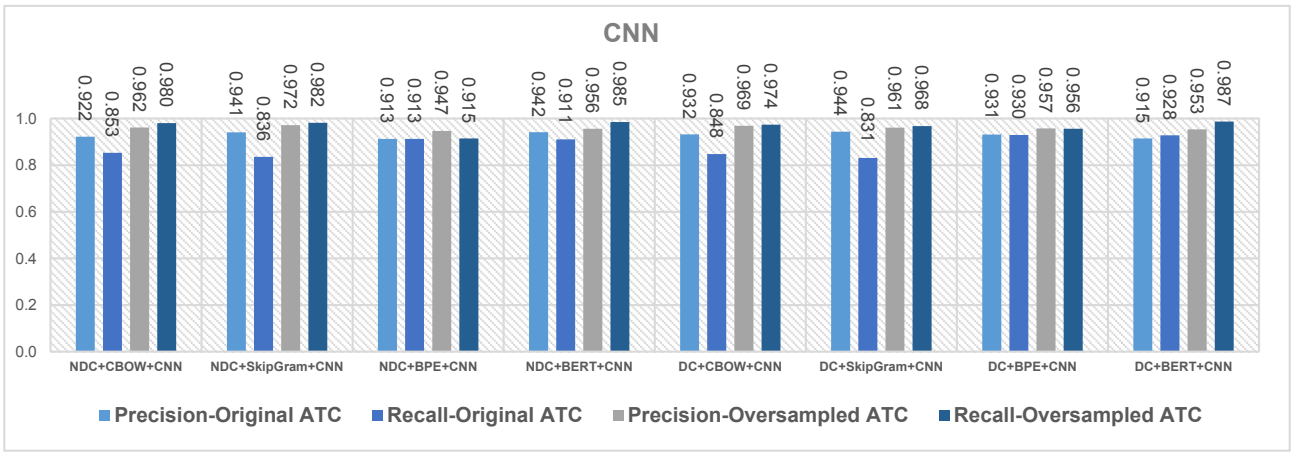

Fig. 2. The average Precision and Recall results of the CNN model combinations using 10-fold cross-validation on the original ATC and the oversampled ATC (Karayiğit et al., 2020).

Table 6
The hyperparameters used in the ML-based classifiers.

Classifier	Parameters
NB	*alpha* = 0.1
SVM	*C* = 10.01, LinearSVC() function was used
DT	random state value = 25
RF	Number of decision trees = 50
LR	default values in Scikit-Learn were used

and the embedding dimension was selected as 50. A BERT-based Turkish Sentiment Model (Yıldırım, 2020a) was utilized to apply BERT embedding to the CNN model. The model was also used in other Turkish NLP studies (Demirtas & Pechenizkiy, 2013; Yıldırım, 2020b). The embedding dimension was selected as 200, and the maximum sequence length value was selected as 128.

4.2. Experimental results of the ML-Based models

Well-known NLP approaches generally adopt ML-based algorithms

Table 7
Average F1-scores and κ values of the SVM model using 10-fold cross-validation.

Dataset	Algorithm Combination of the Experiment	Micro-averaged F1	Macro-averaged F1	Cohen's kappa coefficient (κ)	Avg. Training Time (sec.)	Avg. Test Time (sec.)
Original ATC	NDC + BoW + TF-IDF + SVM	0.932	0.923	0.847	1.577	0.041
	NDC + bi-gram + TF-IDF + SVM	0.933	0.924	0.849	2.926	0.062
	NDC + tri-gram + TF-IDF + SVM	0.927	0.919	0.838	5.635	0.099
	DC + BoW + TF-IDF + SVM	0.932	0.923	0.846	1.710	0.048
	DC + bi-gram + TF-IDF + SVM	0.933	0.925	0.850	4.057	0.082
	DC + tri-gram + TF-IDF + SVM	0.930	0.922	0.845	6.689	0.108
OversampledATC	NDC + BoW + TF-IDF + SVM	0.972	0.972	0.945	2.286	0.057
	NDC + bi-gram + TF-IDF + SVM	0.970	0.970	0.940	4.958	0.107
	NDC + tri-gram + TF-IDF + SVM	0.967	0.964	0.928	7.754	0.149
	DC + BoW + TF-IDF + SVM	**0.973**	**0.973**	**0.945**	**2.310**	**0.061**
	DC + bi-gram + TF-IDF + SVM	0.971	0.971	0.942	5.428	0.110
	DC + tri-gram + TF-IDF + SVM	0.967	0.967	0.935	8.488	0.152

* DC = Data Cleaning; NDC = No Data Cleaning; sec = second; avg = average.

Table 8
Average F1-scores and κ values of the NB model using 10-fold cross-validation.

Dataset	Algorithm Combination of the Experiment	Micro-averaged F1	Macro-averaged F1	Cohen's kappa coefficient (κ)	Avg. Training Time (sec.)	Avg. Test Time (sec.)
Original ATC	NDC + BoW + TF-IDF + NB	0.924	0.915	0.831	0.997	0.050
	NDC + bi-gram + TF-IDF + NB	0.921	0.911	0.823	2.566	0.089
	NDC + tri-gram + TF-IDF + NB	0.919	0.908	0.817	4.058	0.123
	DC + BoW + TF-IDF + NB	0.920	0.910	0.821	0.914	0.044
	DC + bi-gram + TF-IDF + NB	0.920	0.910	0.819	2.301	0.075
	DC + tri-gram + TF-IDF + NB	0.918	0.907	0.814	3.713	0.101
OversampledATC	NDC + BoW + TF-IDF + NB	0.924	0.924	0.848	1.273	0.068
	NDC + bi-gram + TF-IDF + NB	**0.933**	**0.933**	**0.866**	**3.111**	**0.152**
	NDC + tri-gram + TF-IDF + NB	0.934	0.934	0.868	5.327	0.179
	DC + BoW + TF-IDF + NB	0.921	0.920	0.842	1.395	0.065
	DC + bi-gram + TF-IDF + NB	0.927	0.927	0.855	3.393	0.118
	DC + tri-gram + TF-IDF + NB	0.929	0.929	0.859	5.230	0.162

* DC = Data Cleaning; NDC = No Data Cleaning; sec = second; avg = average.

Table 9
Average F1-scores and κ values of the RF model using 10-fold cross-validation.

Dataset	Algorithm Combination of the Experiment	Micro-averaged F1	Macro-averaged F1	Cohen's kappa coefficient (κ)	Avg. Training Time (sec.)	Avg. Test Time (sec.)
Original ATC	NDC + BoW + TF-IDF + RF	0.921	0.909	0.820	29.911	0.503
	NDC + bi-gram + TF-IDF + RF	0.914	0.901	0.803	83.731	0.689
	NDC + tri-gram + TF-IDF + RF	0.908	0.894	0.788	143.128	0.800
	DC + BoW + TF-IDF + RF	0.930	0.919	0.837	34.908	0.678
	DC + bi-gram + TF-IDF + RF	0.922	0.914	0.828	101.555	0.881
	DC + tri-gram + TF-IDF + RF	0.921	0.909	0.816	165.700	0.995
OversampledATC	NDC + BoW + TF-IDF + RF	0.966	0.966	0.933	42.183	0.541
	NDC + bi-gram + TF-IDF + RF	0.967	0.964	0.933	115.373	0.760
	NDC + tri-gram + TF-IDF + RF	0.965	0.966	0.931	197.081	0.916
	DC + BoW + TF-IDF + RF	0.970	0.970	0.940	43.924	0.687
	DC + bi-gram + TF-IDF + RF	**0.970**	**0.971**	**0.939**	**121.880**	**0.909**
	DC + tri-gram + TF-IDF + RF	0.969	0.970	0.941	200.923	1.045

* DC = Data Cleaning; NDC = No Data Cleaning; sec = second; avg = average.

Table 10
Average F1-scores and κ values of the LR model using 10-fold cross-validation.

Dataset	Algorithm Combination of the Experiment	Micro-averaged F1	Macro-averaged F1	Cohen's kappa coefficient (κ)	Avg. Training Time (sec.)	Avg. Test Time (sec.)
Original ATC	NDC + BoW + TF-IDF + LR	0.907	0.892	0.785	2.575	0.048
	NDC + bi-gram + TF-IDF + LR	0.904	0.888	0.777	6.380	0.085
	NDC + tri-gram + TF-IDF + LR	0.901	0.886	0.772	11.026	0.116
	DC + BoW + TF-IDF + LR	0.906	0.890	0.782	2.363	0.044
	DC + bi-gram + TF-IDF + LR	0.904	0.888	0.778	6.308	0.130
	DC + tri-gram + TF-IDF + LR	0.903	0.886	0.774	10.474	0.098
OversampledATC	NDC + BoW + TF-IDF + LR	0.917	0.917	0.833	2.982	0.064
	NDC + bi-gram + TF-IDF + LR	0.911	0.911	0.822	7.607	0.119
	NDC + tri-gram + TF-IDF + LR	0.904	0.904	0.807	13.215	0.168
	DC + BoW + TF-IDF + LR	**0.918**	**0.918**	**0.837**	**2.787**	**0.060**
	DC + bi-gram + TF-IDF + LR	0.916	0.916	0.833	6.983	0.109
	DC + tri-gram + TF-IDF + LR	0.916	0.916	0.831	11.233	0.147

*DC = Data Cleaning; NDC = No Data Cleaning; sec = second; avg = average.

Table 11
Average F1-scores and κ values of the DT model using 10-fold cross-validation.

Dataset	Algorithm Combination of the Experiment	Micro-averaged F1	Macro-averaged F1	Cohen's kappa coefficient (κ)	Avg. Training Time (sec.)	Avg. Test Time (sec.)
Original ATC	NDC + BoW + TF-IDF + DT	0.904	0.893	0.713	32.926	0.088
	NDC + bi-gram + TF-IDF + DT	0.899	0.887	0.775	108.646	0.104
	NDC + tri-gram + TF-IDF + DT	0.888	0.877	0.754	201.349	0.157
	DC + BoW + TF-IDF + DT	0.914	0.904	0.810	32.383	0.058
	DC + bi-gram + TF-IDF + DT	0.907	0.897	0.794	102.488	0.089
	DC + tri-gram + TF-IDF + DT	0.906	0.896	0.791	181.865	0.112
OversampledATC	NDC + BoW + TF-IDF + DT	0.949	0.949	0.898	29.937	0.079
	NDC + bi-gram + TF-IDF + DT	0.943	0.943	0.887	94.171	0.140
	NDC + tri-gram + TF-IDF + DT	0.935	0.934	0.869	174.687	0.192
	DC + BoW + TF-IDF + DT	**0.957**	**0.957**	**0.913**	**34.623**	**0.073**
	DC + bi-gram + TF-IDF + DT	0.951	0.951	0.901	108.994	0.124
	DC + tri-gram + TF-IDF + DT	0.947	0.947	0.894	189.594	0.161

* DC = Data Cleaning; NDC = No Data Cleaning; sec = second; avg = average.

or, more generally based on statistical ML. ML-based algorithms usually consisted of intelligent modules that can learn from historical data (Khan, Daud, Nasir, & Amjad, 2016). ML-based categorization models have been designed and compared with the ANN-based and the reweighted models in our study.

Table 6 shows the hyperparameters used in the ML-based classifiers. Tables 7 through 11 give the average F1-scores, κ values, and training/test durations of the ML-based models using 10-fold cross-validation. Figs. 3 through 7 depict the Precision and Recall results of each fold on the ML-based classifiers.

The *Alpha* value used as additive (Laplace/Lidstone) smoothing parameter (0 for no smoothing) was selected as 0.1 for the NB classifier. The C parameter was used for regularization. The strength of the regularization is inversely proportional to *C*. The LinearSVC() is a useful technique in text categorization and improves classification performance when used with TF-IDF (Zin, Mustapha, Murad, & Sharef, 2018).

The random state value creates and tries the best plan for situations where the DT becomes different. It was selected as a 25 in the DT classifier. Fifty tree structures were used in the RF classifier. This value is the number of DTs generated by the RF classifier. It should be noted that there is not a rule in defining the parameters of ML-based algorithms. In general, the optimal values are empirically chosen based on the physical complexity of the problem.

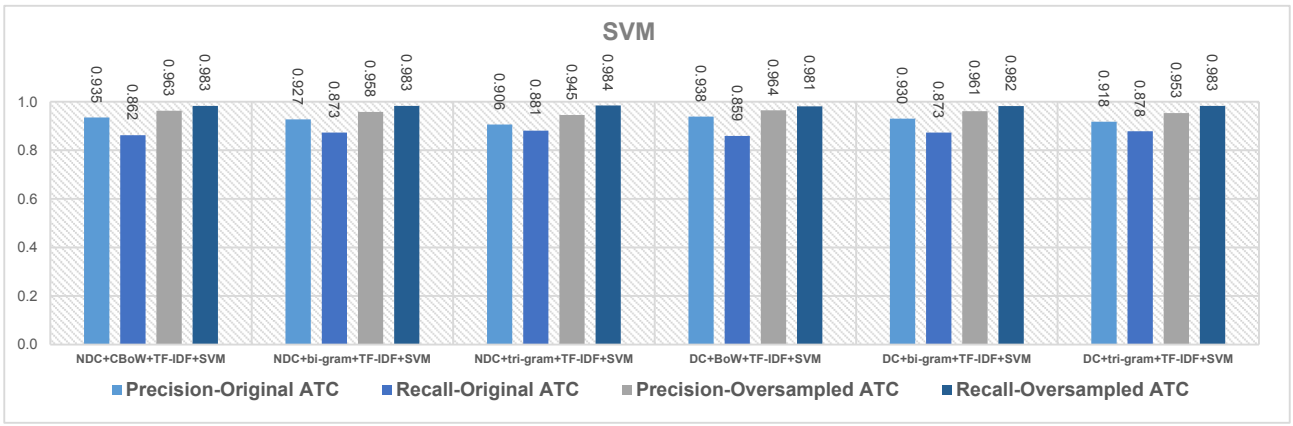

Fig. 3. The average Precision and Recall results of the SVM model combinations using 10-fold cross-validation on the original ATC and the oversampled ATC.

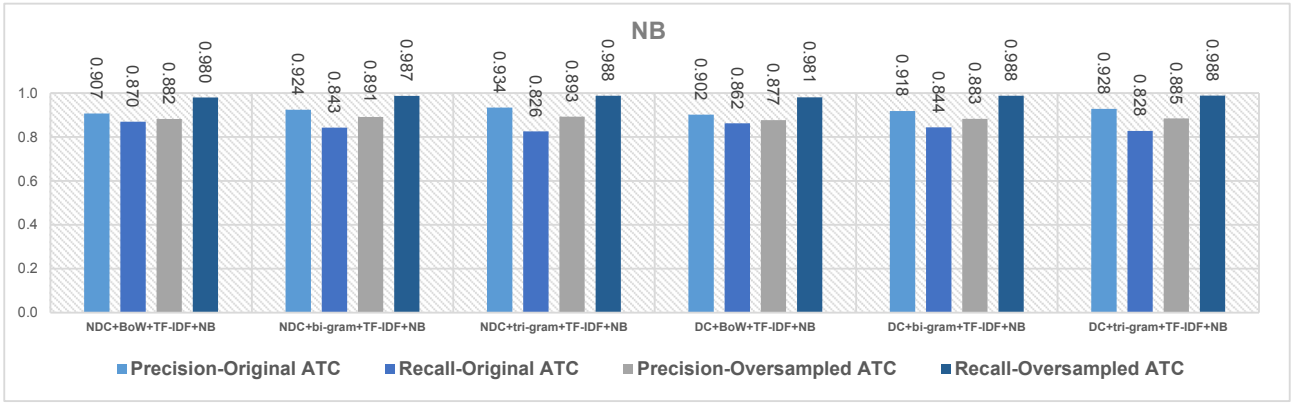

Fig. 4. The average Precision and Recall results of the NB model combinations using 10-fold cross-validation on the original ATC and the oversampled ATC.

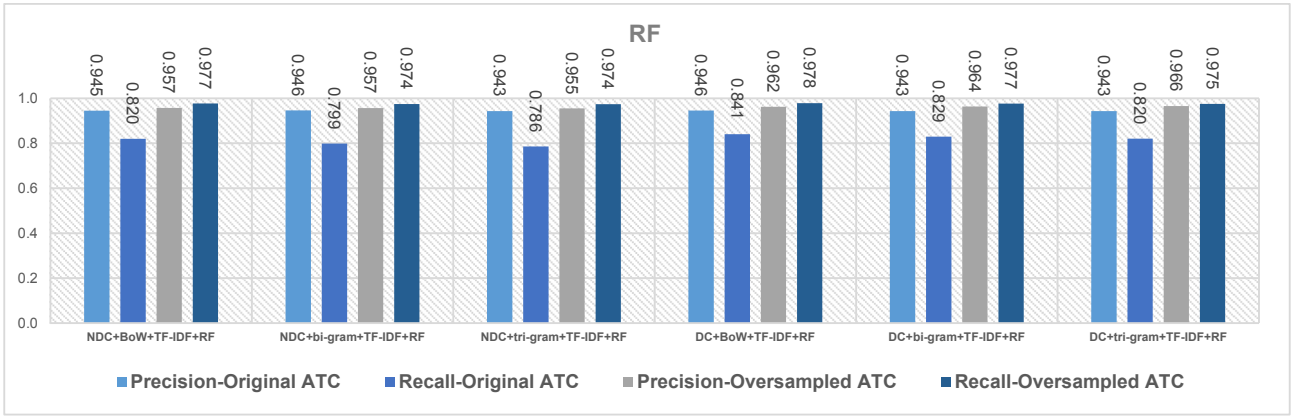

Fig. 5. The average Precision and Recall results of the RF model combinations using 10-fold cross-validation on the original ATC and the oversampled ATC.

4.3. Experimental results of the reweighted models

Boosting has been shown to improve classifiers' performance in many situations, including when data is imbalanced. When performing boosting by reweighting, the numerical weights for each example are passed directly to the base learner. The base learner uses weighting information when forming its hypothesis (Seiffert et al., 2008). AdaBoost and XGBoost were implemented as reweighted classifiers in our study. DT was used as a base learner for the boosting process. Since AdaBoost and XGBoost algorithms already have a boosting process (i.e., reweighting), it was not combined with another boosting process (i.e., resampling). So, AdaBoost and XGBoost algorithms have only been applied to the original ATC dataset.

Table 12 shows the hyperparameters used in the reweighted classifiers. Table 13 explains the performance results of the AdaBoost and XGBoost algorithms. Figs. 8 and 9 illustrate the Precision and Recall results of each fold on the reweighted classifiers.

4.4. Performance comparison

The two datasets (the original ATC and the oversampled ATC) were applied to the CNN model, traditional ML-based models (NB, SVM, DT,

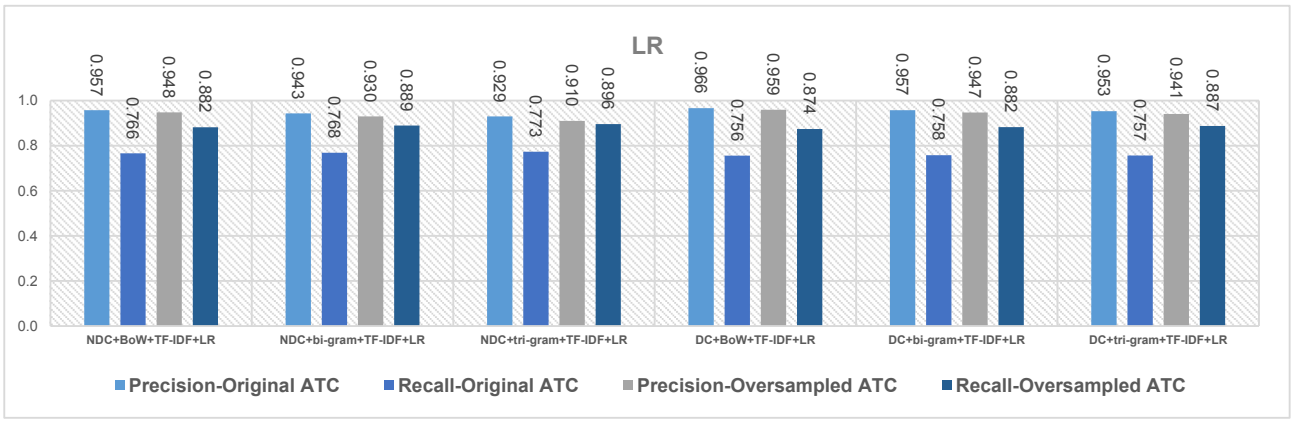

Fig. 6. The average Precision and Recall results of the LR model combinations using 10-fold cross-validation on the original ATC and the oversampled ATC.

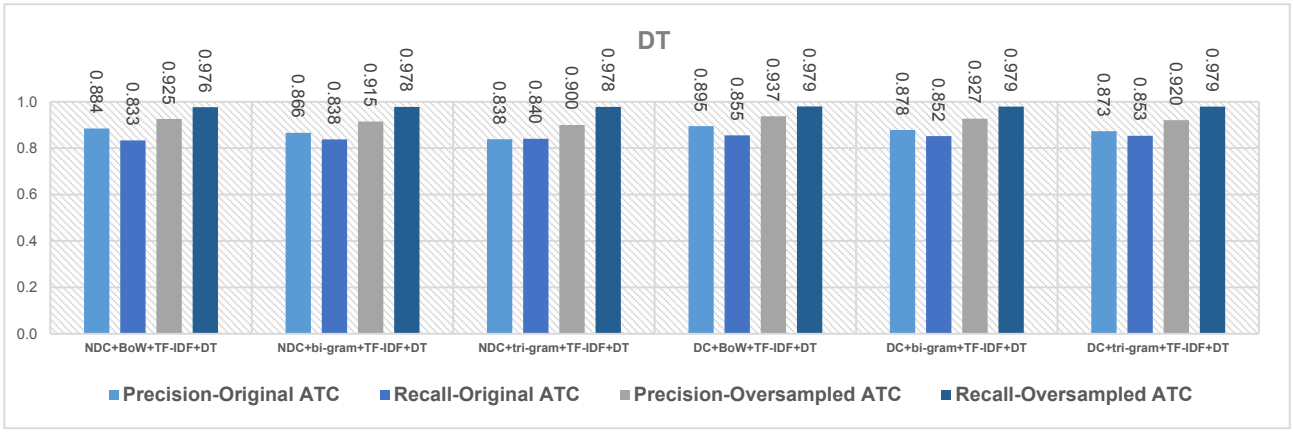

Fig. 7. The average Precision and Recall results of the DT model combinations using 10-fold cross-validation on the original ATC and the oversampled ATC.

Table 12
The parameters used in the reweighted classifiers.

Classifier	Parameter	Value	Description
AdaBoost	Algorithm	SAMME.R	SAMME real boosting algorithm Hastie, Rosset, Zhu, and Zou (2009).
	Base_estimator	DT Classifier	The base estimator which the boosted ensemble is built.
	n_estimators	3000	The maximum number of estimators at which boosting is terminated.
	Learning_rate	1	Learning rate shrinks the contribution of each classifier.
XGBoost	n_estimators	3000	Number of rounds (estimators) for boosting
	Max_dept	10	Maximum depth of a tree
	Learning_rate	0.1	Shrinkage factor for the corrections by new trees when added to the model
	Subsample	1	Subsample ratio of the training instance to prevent overfitting
	Min_child_weight	1	Minimum sum of instance weight needed in a child.
	Colsample_bytree	1	Subsample ratio of random columns (features) when constructing each tree.

RF, and LR), and the reweighted models (i.e., AdaBoost and XGBoost). The pre-processing step was carried out with/without data cleaning, and the obtained results were compared. The experimental results of classifiers in terms of F1-score, precision, and recall are presented in the previous sections.

The last two columns of performance result tables were dedicated to showing the average training and test durations of each categorization model for 10-fold cross-validation. Training and test durations for each fold are given in the detailed experimental results document uploaded to a Kaggle page (Karayiğit et al., 2020) to keep the paper's length at a reasonable level.

Performance comparison of the categorization models are given a point-to-point manner below:

- It is evident from the figures and tables that the oversampling method has yielded better results in all performance metrics for all feature selection, embedding, and classification algorithms. The oversampling method has a positive effect on the results.
- For all the algorithms used in our evaluation, satisfactory results have been obtained. However, the best classification performance (Micro-averaged F1-score: 0.974, Macro-averaged F1-score: 0.973, κ: 0.946) is obtained from the CNN model on the oversampled ATC. The other combinations of the CNN model and the SVM model on the oversampled ATC yielded very similar results. The CNN model is the best classifier that separates the abusive and not-abusive classes, and it minimizes misclassification errors. Higher precision and recall values of the CNN classifier also prove that it can separate positive (abusive) and negative (not-abusive) comments with a correct rate in the ATC datasets.
- The SVM classifier has the second-best performance results after CNN. The effect of the oversampling method has become less pronounced than the CNN model in terms of precision. The SVM model

Table 13
Average F1-scores and κ values of the AdaBoost and XGBoost models using 10-fold cross-validation.

Dataset	Algorithm combination of the experiment	Micro-averaged F1	Macro-averaged F1	Cohen's kappa coefficient (κ)	Avg. training time (sec.)	Avg. Test time (sec.)
Original ATC	**NDC + BoW + TF-IDF + AB**	**0.919**	**0.909**	**0.818**	**202.654**	**2.690**
	NDC + bi-gram + TF-IDF + AB	0.914	0.904	0.809	533.504	4.123
	NDC + tri-gram + TF-IDF + AB	0.907	0.896	0.798	930.810	5.816
	DC + BoW + TF-IDF + AB	**0.919**	**0.909**	**0.819**	**176.067**	**2.567**
	DC + bi-gram + TF-IDF + AB	0.916	0.906	0.812	471.727	3.728
	DC + tri-gram + TF-IDF + AB	0.912	0.902	0.804	794.009	4.818
	NDC + BoW + TF-IDF + XB	**0.923**	**0.911**	**0.823**	**282.081**	**0.429**
	NDC + bi-gram + TF-IDF + XB	0.918	0.906	0.813	909.373	0.478
	NDC + tri-gram + TF-IDF + XB	0.917	0.903	0.807	1605.153	0.600
	DC + BoW + TF-IDF + XB	0.921	0.910	0.821	265.888	0.357
	DC + bi-gram + TF-IDF + XB	0.920	0.907	0.816	807.654	0.433
	DC + tri-gram + TF-IDF + XB	0.922	0.906	0.811	1340.781	0.558

*DC = Data Cleaning; NDC = No Data Cleaning; sec = second; avg = average; AB = AdaBoost; XB = XGBoost.

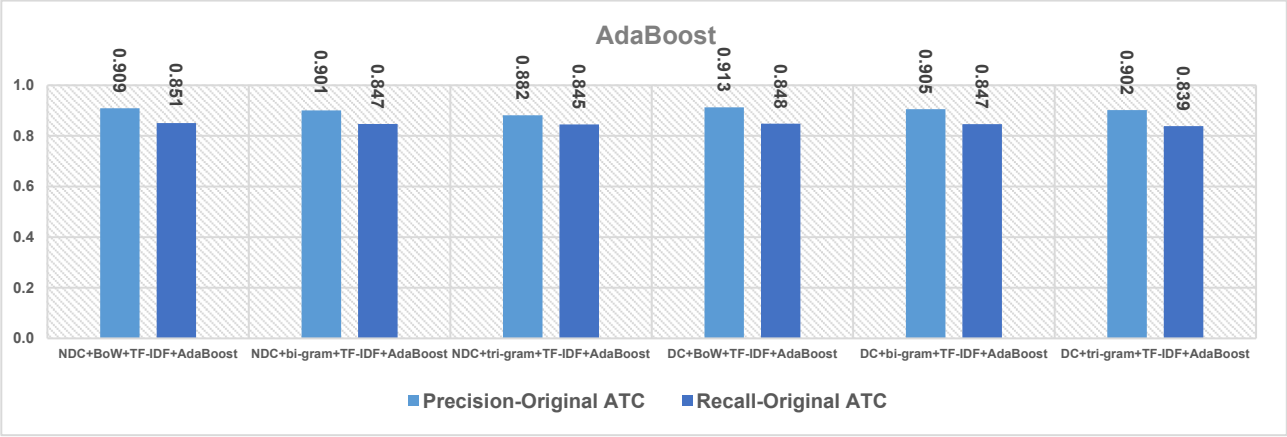

Fig. 8. The average Precision and Recall results of the AdaBoost model combinations using 10-fold cross-validation on the original ATC.

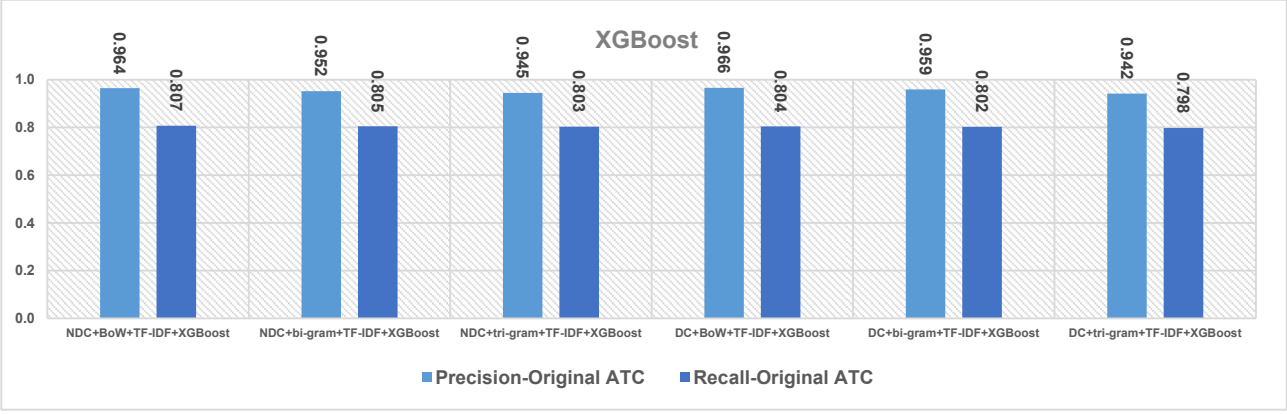

Fig. 9. The average Precision and Recall results of the XGBoost model combinations using 10-fold cross-validation on the original ATC.

gives very fast and accurate results when considering the time spent in training the models.

- The RF classifier has the third-best performance results after SVM. It produced nearly the same results as the SVM model in some combinations. The effect of oversampling has become more pronounced in the RF model than the others.
- While the DT and NB models showed similar performance, it can be said that the DT model is better than the NB model in most of the performance metrics. The LR model and the reweighted models (i.e., AdaBoost and XGBoost) have shown worse performance than the other algorithms.
- It is derived from the tables, "data cleaning" and "no data cleaning" steps do not have a definite advantage over each other. However, BoW has mostly produced the best performance results in terms of embedding algorithms.
- κ value can be used to measure the accuracy, sensitivity, robustness of datasets (Talukder & Ahammed, 2020). According to κ values achieved, it can be observed that the discriminative precision of the datasets is quite robust.

5. Conclusions

The first aim of this study was to create a dataset to detect abusive message content. As far as we know, no study surveyed the presence of a dataset that includes Turkish abusive expressions. The ATC dataset is the first public dataset that can be used in SBTC studies. Although hate speech analysis in social networks has been done in many languages, our study is the first Turkish dataset as far as we know. We have created the ATC dataset and developed one more version with the oversampling data balancing method.

The second aim was developing a robust and accurate SBTC model to detect abusive expressions in the ATC datasets. ANN-based (i.e., CNN), five different ML-based (i.e., NB, SVM, DT, RF, and LR), and reweighted-based (i.e., AdaBoost and XGBoost) classifiers were used to classify abusive Turkish comments on Instagram. When we consider the classification results, it can be said that the proposed SBTC models generally performed well.

To summarize, this study addresses Turkish abusive comment analysis in social media using well-known classification and feature selection methods. Future work can be performed in developing more SBTC models using different sentiments (i.e., hate, racism, sexism, and aggression) on the ATC dataset.

CRediT authorship contribution statement

Habibe Karayiğit: Conceptualization, Investigation, Methodology, Software, Data curation. **Çiğdem İnan Acı:** Conceptualization, Investigation, Validation, Writing - original draft, Writing - review & editing, Visualization. **Ali Akdağlı:** Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abooraig, R., Al-Zu'bi, S., Kanan, T., Hawashin, B., Al Ayoub, M., & Hmeidi, I. (2018). Automatic categorization of Arabic articles based on their political orientation. *Digital Investigation, 25*, 24–41. https://doi.org/10.1016/j.diin.2018.04.003

Abroyan, N. (2017). Convolutional and recurrent neural networks for real-time data classification. In 7th International Conference on Innovative Computing Technology, INTECH 2017 (pp. 42–45). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/INTECH.2017.8102422.

Agnihotri, D., Verma, K., & Tripathi, P. (2017). Variable Global Feature Selection Scheme for automatic classification of text documents. *Expert Systems with Applications, 81*, 268–281. https://doi.org/10.1016/j.eswa.2017.03.057

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

Al-Hassan, A., & Al-Dossari, H. (2019). Detection of Hate Speech in Social Networks: A Survey on Multilingual Corpus. In *In 6th International Conference on Computer Science and Information Technology* (pp. 83–100).

Al-Radaideh, Q. A., & Al-Abrat, M. A. (2019). An Arabic text categorization approach using term weighting and multiple reducts. *Soft Computing, 23*(14), 5849–5863. https://doi.org/10.1007/s00500-018-3249-z

Alakrot, A., Murray, L., & Nikolov, N. S. (2018). Towards accurate detection of offensive language in online communication in Arabic. *Procedia Computer Science, 142*, 315–320. https://doi.org/10.1016/J.PROCS.2018.10.491

Alayba, A. M., Palade, V., England, M., & Iqbal, R. (2017). Arabic language sentiment analysis on health services. In 2017 1st International Workshop on Arabic Script Analysis and Recognition (ASAR) (pp. 114–118). IEEE. https://doi.org/10.1109/ASAR.2017.8067771.

Aya, S. A., Ormanci Acar, T., & Tufekci, N. (2016). Modeling of membrane fouling in a submerged membrane reactor using support vector regression. *Desalination and Water Treatment, 57*(51), 24132–24145. https://doi.org/10.1080/19443994.2016.1140080

Ayata, D., Saraclar, M., & Özgür, A. (2017). Political opinion/sentiment prediction via long short term memory recurrent neural networks on Twitter. In *In 25th IEEE Conference on Signal Processing and Communications Applications* (pp. 1–4).

Babaeianjelodar, M., Lorenz, S., Gordon, J., Matthews, J., & Freitag, E. (2020). Quantifying Gender Bias in Different Corpora. In Companion Proceedings of the Web Conference 2020 (pp. 752–759). New York, NY, USA: ACM. https://doi.org/10.1145/3366424.3383559.

Balakrishnan, V., Khan, S., & Arabnia, H. R. (2020). Improving cyberbullying detection using Twitter users' psychological features and machine learning. *Computers and Security, 90*, Article 101710. https://doi.org/10.1016/j.cose.2019.101710

Bay, Y., & Çelebi, E. (2016). Feature selection for enhanced author identification of turkish text. In *Lecture Notes in Electrical Engineering* (Vol. 363, pp. 371–379). Springer Verlag. https://doi.org/10.1007/978-3-319-22635-4_34.

BBC. (2020). News. Retrieved from https://www.bbc.com/turkce/haberler-turkiye-51614553#:~:text=Türkiye'de 2017 yılında Samsung, siber zorbalık yaptığını ifade etti.

Bimantara, A. A., Larasati, A., Risondang, E. M., Naf'an, M. Z., & Nugraha, N. A. S. (2019). Sentiment analysis of cyberbullying on instagram user comments. *Journal of Data Science and Its Applications, 2*(1), 88–98. https://doi.org/10.21108/jdsa.2019.2.20.

BPEmb_TR,. (2020). BPEmb_TR. Retrieved January 1, 2021, from https://nlp.h-its.org/bpemb/tr/.

Briliani, A., Irawan, B., & Setianingsih, C. (2019). Hate Speech Detection in Indonesian Language on Instagram Comment Section Using K-Nearest Neighbor Classification Method. In 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS) (pp. 98–104). IEEE. https://doi.org/10.1109/IoTaIS47347.2019.8980398.

Burnap, P., & Williams, M. L. (2015). Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy and Internet, 7*(2), 223–242. https://doi.org/10.1002/poi3.85

Çakıcı, R., Steedman, M., & Bozşahin, C. (2018). Wide-Coverage Parsing, Semantics, and Morphology (pp. 153–174). https://doi.org/10.1007/978-3-319-90165-7_8.

Cambria, E. (2016). Affective computing and sentiment analysis. *IEEE Intelligent Systems, 31*(2), 102–107. https://doi.org/10.1109/MIS.2016.31

Chakraborty, P., & Seddiqui, M. H. (2019). Threat and Abusive Language Detection on Social Media in Bengali Language. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (pp. 1–6). IEEE. https://doi.org/10.1109/ICASERT.2019.8934609.

Chamberlain, B. P., Rossi, E., Shiebler, D., Sedhain, S., & Bronstein, M. M. (2020). Tuning Word2vec for Large Scale Recommendation Systems. In RecSys 2020 - 14th ACM Conference on Recommender Systems (pp. 732–737). New York, NY, USA: Association for Computing Machinery, Inc. https://doi.org/10.1145/3383313.3418486.

Charitidis, P., Doropoulos, S., Vologiannidis, S., Papastergiou, I., & Karakeva, S. (2020). Towards countering hate speech against journalists on social media. *Online Social Networks and Media, 17*, Article 100071. https://doi.org/10.1016/j.osnem.2020.100071

Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., & Vakali, A. (2017). Hate is not Binary: Studying Abusive Behavior of #GamerGate on Twitter. In *HT 2017 - Proceedings of the 28th ACM Conference on Hypertext and Social Media* (pp. 65–74).

Chatzakou, D., Leontiadis, I., Blackburn, J., De Cristofaro, E., Stringhini, G., Vakali, A., & Kourtellis, N. (2019). Detecting cyberbullying and cyberaggression in social media. *ACM Transactions on the Web, 13*(3). https://doi.org/10.1145/3343484

Chen, H., McKeever, S., & Delany, S. J. (2017). Abusive Text Detection Using Neural Networks. In *In CEUR Workshop Proceedings* (pp. 258–260).

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

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (Vol. 13-17-Augu, pp. 785–794). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785.

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. *Retrieved from.* http://arxiv.org/abs/1703.04009.

Demirsoz, O., & Ozcan, R. (2017). Classification of news-related tweets. *Journal of Information Science, 43*(4), 509–524. https://doi.org/10.1177/0165551516653082

Demirtas, E., & Pechenizkiy, M. (2013). *Cross-lingual polarity detection with machine translation. In Proceedings of the 2nd International Workshop on Issues of Sentiment Discovery and Opinion Mining* (pp. 1–8). New York, New York, USA: Association for Computing Machinery. WISDOM 2013 - Held in Conjunction with SIGKDD 2013.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language Understanding. NAACL HLT 2019–2019 Conference of the North American Chapter of the Association for Computational Linguistics. Retrieved from *Human Language Technologies - Proceedings of the Conference, 1*, 4171–4186.

Dogan, T., & Uysal, A. K. (2019). Improved inverse gravity moment term weighting for text classification. *Expert Systems with Applications, 130*, 45–59. https://doi.org/10.1016/j.eswa.2019.04.015

Drozd, A., Gladkova, A., & Matsuoka, S. (2015). Discovering Aspectual Classes of Russian Verbs in Untagged Large Corpora. In IEEE International Conference on Data Science and Data Intensive Systems (pp. 61–68). Sydney, NSW, Australia.

Dwivedi, R. K., Aggarwal, M., Keshari, S. K., & Kumar, A. (2019). Sentiment analysis and feature extraction using rule-based model (RBM). In Lecture Notes in Networks and Systems (Vol. 56, pp. 57–63). Springer. https://doi.org/10.1007/978-981-13-2354-6_7.

El-Kahlout, I. D., & Akin, A. A. (2013). Turkish constituent chunking with morphological and contextual features. In Lecture Notes in Computer Science (including subseries

Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 7816 LNCS, pp. 270–281). https://doi.org/10.1007/978-3-642-37247-6_22.
Eryiğit, G., Nivre, J., & Oflazer, K. (2008). Dependency Parsing of Turkish. *Computational Linguistics, 34*(3), 357–389. https://doi.org/10.1162/coli.2008.07-017-R1-06-83
Fatima, M., & Pasha, M. (2017). Survey of machine learning algorithms for disease diagnostic. *Journal of Intelligent Learning Systems and Applications, 09*(01), 1–16. https://doi.org/10.4236/jilsa.2017.91001
Fatima, S., & Srinivasu, D. B. (2017). Text Document categorization using support vector machine. *International Research Journal of Engineering and Technology, 4*(2), 141–147.
Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm |. In *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning. In Proceedings of the Thirteenth International Conference on International Conference on Machine Learning* (pp. 148–156).
Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine on JSTOR. Retrieved from *The Annals of Statistics, 29*(5), 1189–1232 https://www.jstor.org/stable/2699986?seq=1#metadata_info_tab_contents.
Gao, Z., Feng, A., Song, X., & Wu, X. (2019). Target-dependent sentiment classification with BERT. *IEEE Access, 7*, 154290–154299. https://doi.org/10.1109/ACCESS.2019.2946594
Gazzah, S., Amara, N. E., & Ben.. (2008). New oversampling approaches based on polynomial fitting for imbalanced data sets. In *In DAS 2008 - Proceedings of the 8th IAPR International Workshop on Document Analysis Systems* (pp. 677–684). https://doi.org/10.1109/DAS.2008.74
Golbeck, J., Ashktorab, Z., Banjo, R. O., Berlinger, A., Bhagwan, S., Buntain, C., … Wu, D. M. (2017). In *A large human-labeled corpus for online harassment research* (pp. 229–233). Inc: Association for Computing Machinery. https://doi.org/10.1145/3091478.3091509.
Google,. (2020). Colaboratory. Retrieved August 13, 2020, from https://colab.research.google.com/.
Han, J., Kamber, M., & Pei, J. (2011). *Data Mining. Concepts and Techniques* (3rd). Edition (The Morgan Kaufmann Series in Data Management Systems).
Hastie, T., Rosset, S., Zhu, J., & Zou, H. (2009). Multi-class AdaBoost. *Statistics and Its Interface, 2*(3), 349–360. https://doi.org/10.4310/sii.2009.v2.n3.a8
He, X., & Deng, L. (2017). Deep learning for image-to-text generation: A technical overview. *IEEE Signal Processing Magazine, 34*(6), 109–116. https://doi.org/10.1109/MSP.2017.2741510
Heinzerling, B., & Strube, M. (2018). BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In Eleventh International Conference on Language Resources and Evaluation (LREC 2018) (pp. 1–5). Miyazaki, Japan: European Language Resources Association (ELRA). Retrieved from https://github.com/facebookresearch/.
Heirman, W., & Walrave, M. (2008). Assessing Concerns and Issues about the Mediation of Technology in Cyberbullying.
Hemmatian, F., & Sohrabi, M. K. (2019). A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review, 52*(3), 1495–1545. https://doi.org/10.1007/s10462-017-9599-6
Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507. https://doi.org/10.1126/SCIENCE.1127647
Hmeidi, I., Al-Ayyoub, M., Abdulla, N. A., Almodawar, A. A., Abooraig, R., & Mahyoub, N. A. (2015). Automatic Arabic text categorization: A comprehensive comparative study. *Journal of Information Science, 41*(1), 114–124. https://doi.org/10.1177/0165551514558172
Hosseinmardi, H., Mattson, S. A., Rafiq, R. I., Han, R., Lv, Q., & Mishra, S. (2015). Detection of cyberbullying incidents on the instagram social network. *Retrieved from*. http://arxiv.org/abs/1503.03909.
Hu, Y., Manikonda, L., & Kambhampati, S. (2014). What we instagram: A first analysis of instagram photo content and user types. In 8th International Conference on Weblogs and Social Media (pp. 595–598). Ann Arbor,: The AAAI Press. Retrieved from https://asu.pure.elsevier.com/en/publications/what-we-instagram-a-first-analysis-of-instagram-photo-content-and.
Huang, B., & Raisi, E. (2018). Weak Supervision and Machine Learning for Online Harassment Detection (pp. 5–28). https://doi.org/10.1007/978-3-319-78583-7_2.
Huang, Z., Cao, Y., & Wang, T. (2019). Transfer Learning with Efficient Convolutional Neural Networks for Fruit Recognition. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC) (pp. 358–362). IEEE. https://doi.org/10.1109/ITNEC.2019.8729435.
Ibrohim, M. O., & Budi, I. (2018). A Dataset and Preliminaries Study for Abusive Language Detection in Indonesian Social Media. In Procedia Computer Science (Vol. 135, pp. 222–229). Elsevier B.V. https://doi.org/10.1016/j.procs.2018.08.169.
Instagram. (2020). Statistics. Retrieved July 12, 2020, from https://www.socialmediatoday.com/news/top-10-instagram-stats-for-2020-infographic/569641/.
Johnson, B. G. (2018). Tolerating and managing extreme speech on social media. *Internet Research, 28*(5), 1275–1291. https://doi.org/10.1108/IntR-03-2017-0100
Jones, L. M., Mitchell, K. J., & Finkelhor, D. (2013, January). Online harassment in context: Trends from three youth internet safety surveys (2000, 2005, 2010). Psychology of Violence. https://doi.org/10.1037/a0030309.
Karayiğit, H., Acı, Ç.İ., & Akdağlı, A. (2020). Abusive Turkish Comments Dataset. *Retrieved from*. https://www.kaggle.com/habibekarayiit/datasets.
Keras. (2020). API. Retrieved from https://keras.io/api/layers/convolution_layers/convolution1d/.
Khan, W., Daud, A., Nasir, J. A., & Amjad, T. (2016). A survey on the state-of-the-art machine learning models in the context of NLP. *Kuwait Journal of Science, 43*(4), 95–113.
Kilic, S. (2015). Kappa test. *Journal of Mood Disorders, 5*(3), 142. https://doi.org/10.5455/jmood.20150920115439
Kilinç, D., Özçift, A., Bozyigit, F., Yildlrlm, P., Yücalar, F., & Borandag, E. (2017). TTC-3600: A new benchmark dataset for Turkish text categorization. *Journal of Information Science, 43*(2), 174–185. https://doi.org/10.1177/0165551515620551
Kim, H., & Jeong, Y.-S. (2019). Sentiment classification using convolutional neural networks. *Applied Sciences, 9*(11), 2347. https://doi.org/10.3390/app9112347
Kılınc, D., Borandağ, E., Yücalar, F., Tunalı, V., Şimşek, M., & Özçift, A. (2016). Classification of scientific articles using text mining with KNN Algorithm and R Language. *Marmara Journal of Science, 28*(3), 89–94. https://doi.org/10.7240/mufbed.69674
Kowalski, R. M., Giumetti, G. W., Schroeder, A. N., & Lattanner, M. R. (2014). Bullying in the digital age: A critical review and meta-analysis of cyberbullying research among youth. *Psychological Bulletin, 140*(4), 1073–1137. https://doi.org/10.1037/a0035618
Kwok, I., & Wang, Y. (2013). Locate the Hate: Detecting Tweets against Blacks. In *In Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence* (pp. 1621–1622).
Le, T., Hoang Son, L., Vo, M., Lee, M., & Baik, S. (2018). A cluster-based boosting algorithm for bankruptcy prediction in a highly imbalanced dataset. *Symmetry, 10*(7), 250. https://doi.org/10.3390/sym10070250
Lee, H. S., Lee, H. R., Park, J. U., & Han, Y. S. (2018). An abusive text detection system based on enhanced abusive and non-abusive word lists. *Decision Support Systems, 113*, 22–31. https://doi.org/10.1016/j.dss.2018.06.009
Levy, O., & Goldberg, Y. (2014). Linguistic regularities in sparse and explicit word representations. In CoNLL 2014 - 18th Conference on Computational Natural Language Learning, Proceedings (pp. 171–180). Association for Computational Linguistics (ACL). https://doi.org/10.3115/v1/w14-1618.
Li, B., Drozd, A., Liu, T., & Du, X. (2018). *Subword-level Composition Functions for Learning Word Embeddings* (pp. 38–48). Association for Computational Linguistics (ACL).
Liu, P., Qiu, X., Chen, X., Wu, S., & Huang, X. (2015). Multi-Timescale Long Short-Term Memory Neural Network for Modelling Sentences and Documents. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 2326–2335). Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1280.
Mahmood, Z., Safder, I., Nawab, R. M. A., Bukhari, F., Nawaz, R., Alfakeeh, A. S., … Hassan, S. U. (2020). Deep sentiments in Roman Urdu text using recurrent convolutional neural network model. *Information Processing and Management, 57*(4), Article 102233. https://doi.org/10.1016/j.ipm.2020.102233
Marujo, L., Ling, W., Ribeiro, R., Gershman, A., Carbonell, J., Martins de Matos, D., & Neto, J. P. (2016). Exploring events and distributed representations of text in multi-document summarization. *Knowledge-Based Systems, 94*, 33–42. https://doi.org/10.1016/J.KNOSYS.2015.11.005
Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Retrieved from*. http://arxiv.org/abs/1310.4546.
Mohan Patro, V. M., & Ranjan Patra, M. (2015). A Novel Approach to Compute Confusion Matrix for Classification of n-Class Attributes with Feature Selection. Transactions on Machine Learning and Artificial Intelligence, 3(2), 52–52. https://doi.org/10.14738/tmlai.32.1108.
Mossie, Z., & Wang, J. H. (2020). Vulnerable community identification using hate speech detection on social media. *Information Processing and Management, 57*(3), Article 102087. https://doi.org/10.1016/j.ipm.2019.102087
Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). A BERT-Based Transfer Learning Approach for Hate Speech Detection in Online Social Media. In H. Cherifi, S. Gaito, J. F. Mendes, E. Moro, & L. M. Rocha (Eds.), *Complex Networks and Their Applications VIII* (pp. 928–940). Cham: Springer International Publishing.
Naf'an, M. Z., Bimantara, A. A., Larasati, A., Risondang, E. M., & Nugraha, N. A. S. (2019). Sentiment analysis of cyberbullying on instagram user comments. *Journal of Data Science and Its Applications, 2*(1), 88–98. https://doi.org/10.21108/jdsa.2019.2.20.
Omar, A., Mahmoud, T. M., & Abd-El-Hafeez, T. (2020). *Comparative Performance of Machine Learning and Deep Learning Algorithms for Arabic Hate Speech Detection in OSNs* (pp. 247–257). Cham: Springer.
Omar, N., & Al-Tashi, Q. (2018). Arabic nested noun compound extraction based on linguistic features and statistical measures. *GEMA Online Journal of Language Studies, 18*(2), 93–107. https://doi.org/10.17576/gema-2018-1802-07.
Ornek, A. H., Ceylan, M., & Ervural, S. (2019). Health status detection of neonates using infrared thermography and deep convolutional neural networks. *Infrared Physics & Technology, 103*, Article 103044. https://doi.org/10.1016/J.INFRARED.2019.103044
Ozel, S. A., Sarac, 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). IEEE. https://doi.org/10.1109/UBMK.2017.8093411.
Park, J. H., & Fung, P. (2017). One-step and two-step classification for abusive language detection on twitter, 41–45. *Retrieved from*. http://arxiv.org/abs/1706.01206.
Parlar, T., Özel, S. A., & Song, F. (2019). Analysis of data pre-processing methods for sentiment analysis of reviews. *Computer Science, 20*(1), 123–141. https://doi.org/10.7494/csci.2019.20.1.3097
Pratiwi, N. I., Budi, I., & Alfina, I. (2018). Hate Speech Detection on Indonesian Instagram Comments using FastText Approach. In 2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS) (pp. 447–450). IEEE. https://doi.org/10.1109/ICACSIS.2018.8618182.
Priyoko, B., & Yaqin, A. (2019). Implementation of Naive Bayes Algorithm for Spam Comments Classification on Instagram. In 2019 International Conference on Information and Communications Technology (ICOIACT) (pp. 508–513). IEEE. https://doi.org/10.1109/ICOIACT46704.2019.8938575.

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

Saraç, E., & Özel, S. A. (2016). Effects of feature extraction and classification methods on cyberbully detection. *Süleyman Demirel University Journal of Natural and Applied Sciences, 21*(1), 190. https://doi.org/10.19113/sdufbed.20964.

Scikit,. (2020). Scikit- Learn Library. Retrieved from https://scikit-learn.org/stable/.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys. Association for Computing Machinery (ACM).*. https://doi.org/10.1145/505282.505283

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

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2008). Resampling or reweighting: A comparison of boosting implementations. In Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI (Vol. 1, pp. 445–451). https://doi.org/10.1109/ICTAI.2008.59.

Sennrich, R., Haddow, B., & Birch, A. (2016). Neural machine translation of rare words with subword units. In 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers (Vol. 3, pp. 1715–1725). Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/p16-1162.

Severyn, A., & Moschitti, A. (2015). Twitter Sentiment Analysis with Deep Convolutional Neural Networks. In Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15 (pp. 959–962). New York, New York, USA: ACM Press. https://doi.org/10.1145/2766462.2767830.

Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014). Learning semantic representations using convolutional neural networks for web search. In Proceedings of the 23rd International Conference on World Wide Web - WWW '14 Companion (pp. 373–374). New York, New York, USA: ACM Press. https://doi.org/10.1145/2567948.2577348.

Shi, H., Wang, H., Huang, Y., Zhao, L., Qin, C., & Liu, C. (2019). A hierarchical method based on weighted extreme gradient boosting in ECG heartbeat classification. *Computer Methods and Programs in Biomedicine, 171*, 1–10. https://doi.org/10.1016/j.cmpb.2019.02.005

Shushkevich, E., & Cardiff, J. (2019). Automatic misogyny detection in social media: A survey. *Computacion y Sistemas, 23*(4), 1159–1164. https://doi.org/10.13053/CyS-23-4-3299.

Statista,. (2020). Turkey: Number of Instagram users 2020 | Statista. Retrieved July 12, 2020, from https://www.statista.com/statistics/1024714/instagram-users-turkey/.

Talukder, A., & Ahammed, B. (2020). Machine learning algorithms for predicting malnutrition among under-five children in Bangladesh. *Nutrition, 110861*. https://doi.org/10.1016/j.nut.2020.110861

Tang, Y., & Dalzell, N. (2019). Classifying hate speech using a two-layer model. *Statistics and Public Policy, 6*(1), 80–86. https://doi.org/10.1080/2330443x.2019.1660285

TDK. (2020). Turkish Language Society. Retrieved June 14, 2020, from https://sozluk.gov.tr/.

Terragni, S., Fersini, E., & Messina, E. (2020). Constrained relational topic models. *Information Sciences, 512*, 581–594. https://doi.org/10.1016/j.ins.2019.09.039

Van Royen, K., Poels, K., Daelemans, W., & Vandebosch, H. (2015). Automatic monitoring of cyberbullying on social networking sites: From technological feasibility to desirability. *Telematics and Informatics, 32*(1), 89–97. https://doi.org/10.1016/j.tele.2014.04.002

Vigna, F. Del, Cimino, A., Dell'orletta, F., Petrocchi, M., & Tesconi, M. (2017). Hate me, hate me not: Hate speech detection on Facebook. In ITA-SEC 17. Retrieved from https://curl.haxx.se.

Wang, H., & Raj, B. (2017). On the Origin of Deep Learning. *Retrieved from*. http://arxiv.org/abs/1702.07800.

Wang, N., Wang, P., & Zhang, B. (2010). An improved TF-IDF weights function based on information theory. In CCTAE 2010 - 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering (Vol. 3, pp. 439–441). https://doi.org/10.1109/CCTAE.2010.5544382.

Waseem, Z., & Hovy, D. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter | Request PDF. In *In Proceedings of the NAACL Student Research Workshop* (pp. 88–93).

Waseem, Z., Thorne, J., & Bingel, J. (2018). Bridging the Gaps: Multi Task Learning for Domain Transfer of Hate Speech Detection (pp. 29–55). Springer, Cham. https://doi.org/10.1007/978-3-319-78583-7_3.

Wei, F., Qin, H., Ye, S., & Zhao, H. (2018). Empirical Study of Deep Learning for Text Classification in Legal Document Review. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 3317–3320). IEEE. https://doi.org/10.1109/BigData.2018.8622157.

Wiegand, M., Ruppenhofer, J., Schmidt, A., & Greenberg, C. (2018). Inducing a Lexicon of Abusive Words – A Feature-Based Approach. In *In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 1046–1056).

Wiegand, M., Siegel, M., & Ruppenhofer, J. (2018). Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. *In GermEval 2018 Shared Task on the Identification of Offensive Language*.

Wikipedia. (2020). Wikimedia Dump. Retrieved from https://dumps.wikimedia.org/trwiki/.

Yih, W., He, X., & Meek, C. (2014). Semantic Parsing for Single-Relation Question Answering. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers) (pp. 643–648). Baltimore, Maryland, USA.

Yıldırım, S. (2020a). BERT-base Turkish Sentiment Model. Retrieved January 1, 2021, from https://huggingface.co/savasy/bert-base-turkish-sentiment-cased/blob/main/README.md.

Yıldırım, S. (2020b). Comparing Deep Neural Networks to Traditional Models for Sentiment Analysis in Turkish Language. In *Algorithms for Intelligent Systems* (pp. 311–319). Singapore: Springer. https://doi.org/10.1007/978-981-15-1216-2_12.

Yuan, B., & Ma, X. (2012). Sampling + reweighting: Boosting the performance of AdaBoost on imbalanced datasets. In *In Proceedings of the International Joint Conference on Neural Networks*. https://doi.org/10.1109/IJCNN.2012.6252738

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2020). Subword Embedding. In Dive into Deep Learning (pp. 664–666).

Zhang, Z., & Luo, L. (2019). Hate speech detection: A solved problem? The challenging case of long tail on Twitter. *Semantic Web, 10*(5), 925–945. https://doi.org/10.3233/SW-180338

Zhang, Z., Robinson, D., & Tepper, J. (2018a). Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 10843 LNCS, pp. 745–760). Springer Verlag. https://doi.org/10.1007/978-3-319-93417-4_48.

Zhang, Z., Robinson, D., & Tepper, J. (2018). *Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network* (pp. 745–760). Cham: Springer.

Zin, H. M., Mustapha, N., Murad, M. A. A., & Sharef, N. M. (2018). Term weighting scheme effect in sentiment analysis of online movie reviews. *Advanced Science Letters, 24*(2), 933–937. https://doi.org/10.1166/asl.2018.10661