Sādhanā (2022) 47:194
https://doi.org/10.1007/s12046-022-01975-3

Author and genre identification of Turkish news texts using deep learning algorithms

PINAR TÜFEKCİ[1,*] and MELİKE BEKTAŞ[2]

[1]Department of Computer Engineering, Çorlu Faculty of Engineering, Tekirdag Namik Kemal University, Tekirdağ, Turkey
[2]Department of Information Technologies, Bursa Technical University, Bursa, Turkey
e-mail: ptufekci@nku.edu.tr

MS received 8 March 2022; revised 2 June 2022; accepted 16 August 2022

Abstract. Nowadays, the increasing amount of data has brought the need to classify the data. Text classification is the process of categorizing similar text data. This paper aims to make a modeling study for author and genre identification, which is one of the important challenges of text classification, for Turkish news texts by using machine and deep learning algorithms. For this purpose, firstly, a total of 13 large-scale datasets having multi classes are built as new datasets. In the modeling stage, Multinomial Naïve Bayes (MNB), Random Forest (RF), Convolutional Neural Network (CNN), and Long Short Term Memory (LSTM) algorithms were applied to the datasets. Results showed that for dataset AI-TNKU-7, the CNN algorithm demonstrated the highest accuracy for author identification at 95.81%. In relation to genre identification, the LSTM algorithm for the dataset GI-TNKU-6 demonstrated the highest accuracy at 96.73%.

Keywords. Author identification; genre identification; deep learning; text classification; Turkish news datasets; machine learning.

1. Introduction

Nowadays, the rapid increase in social media content, mail, blog, and news articles has revealed the need for automatic management and classification of this unstructured data. Text classification is the process of identification of which text belongs to a predetermined class by considering the characteristics of a text [1].

There are many studies related to text classification challenges such as, author identification [2–9], genre identification [2–4, 8, 10, 11], and identification of author gender [2, 4, 8, 12–20]. Successful models have been developed to solve text classification challenges using machine learning and deep learning algorithms in these studies.

The author identification problem is the process of identifying the author of a text. The genre identification problem is the process of determining which class the type of text belongs to. Author identification can be used in many different fields in order to help the author of a text, in cases where more than one person claims rights to a text and claims that he/she wrote it, in judicial practices, library applications, and in anti-terrorism investigations [7].

The genre classification challenge is to find out the class or category of a text such as a news text, and whether it relates to sports, health, economy, politics, magazine categories, etc. Genre identification can be useful for many text-based applications. For instance, if the type of document is known in advance, the retrieved information can be presented to the user more accurately [21]. Searching for a document on the internet can be found more accurately and easily according to its type. Online library automation that classifies the document can enable one to find a document quickly and classify new incoming documents correctly.

Amasyalı and Diri carried out the genre classification process by using the dataset consisting of 630 Turkish texts in total, 210 in each class. They used the Support Vector Machine (SVM), NB, RF, and C4.5 decision tree algorithms for the classification process. They reported that they achieved 93.6% accuracy with the SVM algorithm [2].

Diri and Kaban classified 250 Turkish news texts in total consisting of 5 classes: sports, economy, politics, health, and popular. They achieved 98.2% accuracy with the SVM algorithm [3]. Diri and Yasdi classified Turkish texts, for genre and gender. They reported that they achieved 97.5% accuracy with the k-Nearest Neighbors (kNN) algorithm for genre identification [4].

Tüfekci and Uzun tried to solve the author identification problem using 3 different datasets in their studies. The datasets used consisted of 430 Turkish texts in 10 classes, 910 Turkish texts in 69 classes, and 630 Turkish texts in 18

*For correspondence
Published online: 19 September 2022

classes. As a result of the study, they reported that they achieved 96.5% accuracy with the SVM algorithm [5].

Şahin *et al* classified 1255 texts of English poetry into 3 classes in terms of authors. As a result of the study, they obtained an accuracy of 70% with the Sequential Minimal Optimization (SMO) algorithm [6].

Stamatos carried out the author identification process in English and Arabic texts consisting of 10 classes and a total of 1000. In the dataset consisting of Arabic texts, 93.6% accuracy was achieved, and 79.4% accuracy was obtained in the dataset consisting of English texts [7].

Diri and Doğan carried out genre identification using a total of 480 Turkish texts consisting of news texts with a total success rate of 92.1% with the SVM algorithm [8].

Vijayakumar *et al* performed the author identification process using the Yelp dataset consisting of online restaurant and hotel reviews in English. They tested the success of MNB, Maximum Entropy (ME), and SVM algorithms by applying natural language processing techniques for the classification process. As a result of the study, they reported that they achieved the highest 90.5% accuracy with the SVM algorithm [9].

Tüfekci *et al* classified Turkish news texts in terms of genres by using two different datasets consisting of 5 classes and 750 and 1150 texts. In both datasets, they achieved 92.4% and 96.2% accuracy with the NB algorithm [10].

Wongso *et al* classified the texts in terms of genres, using a set of Indonesian data. In the study, they used a dataset consisting of 5 classes: economy, health, sports, politics, technology, and a total of 5000 news texts, 1000 in each class. As a result of the study, they performed the classification process of the NB algorithm with 98.4% success [11].

In this study, we have proposed some models for the identification of author and genre in Turkish news texts by using machine and deep learning algorithms. Furthermore, we have created 13 new large-scaled and multi-class datasets. These datasets are composed of Turkish news texts of columnists, which are extracted from a newspaper (https://www.hurriyet.com.tr/). In the models, Multinomial Naïve Bayes (MNB), Random Forest (RF), Convolutional Neural Network (CNN), and Long Short Term Memory (LSTM) algorithms were applied to these datasets. To obtain the best model, firstly, natural language processing steps were applied to these datasets, and then classifier algorithms were applied to the datasets.

The rest of this paper is organized as follows: Section 2 describes machine and deep learning algorithms. Section 3 presents the evaluation metrics used in this study. In section 4, the flow of this study is outlined and all data processing stages are explained. The models, which are built for the newly collected datasets are presented in section 5. Evaluation of the model and results are discussed in section 6. Finally, in section 7 the conclusions are given.

2. Methods

In this section, Naïve Bayes, Random Forest, Convolutional Neural Network, and Long Short Term Memory algorithms are explained as classifier methods for author and genre identification of Turkish news texts.

2.1 *Naïve Bayes*

Naïve Bayes is a probabilistic classification algorithm based on Bayes theorem. This theorem is based on statistically calculating the relationships of conditional probabilities. Naïve Bayes classification helps to find the probability of a label given some observed features, which write as P(L | features) Bayes theorem is calculated using the formula in Equation (1).

$$P(L|features) = \frac{P(features|L)P(L)}{P(features)} \tag{1}$$

There are different types of Naïve Bayes classifiers: Bernoulli Naïve Bayes, Gaussian Naïve Bayes and Multinomial Naïve Bayes [22]. In this study, using Multinomial Naïve Bayes (MNB) classifier, a model is created for author and genre identification of texts. MNB is an algorithm that is generally used to solve text classification problems where features such as word count and word frequency are important and relevant [22].

2.2 *Random forest*

Random Forest algorithm is a supervised machine learning algorithm belonging to ensemble learning methods used for both classification and regression. This algorithm consists of many individual decision trees that work as an ensemble [23].

Random Forest algorithm first selects random instances from the dataset using sampling with replacement. Then, a decision tree is grown for each sample using a random subset of the features. In the next step, voting is done for each predicted outcome. Finally, the prediction result with the most votes is determined as the final prediction result.

2.3 *Convolutional neural network*

Convolutional neural network algorithm is an artificial neural network model that includes operations such as a convolutional layer, pooling layer, and fully connected layer as shown in figure 1. Compared to other neural networks, CNN's most distinctive feature is its use of convolutional layers. In the CNN algorithm, the convolution process is used to determine the features, the pooling process is used to reduce the number of weights, and the full connected layer is used to classify. Fully connected layers

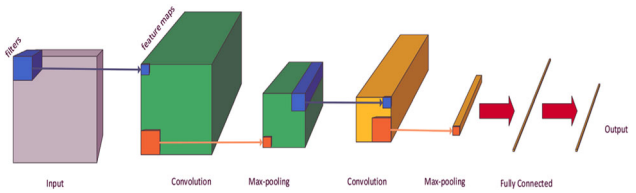

Figure 1. General structure of the convolutional neural network.

decide the final class with the highest probability using a softmax function [24].

2.4 *Long short term memory*

Long Short Term Memory is a type of recurrent neural network algorithm designed by Sepp Hochreiter and Jürgen Schmidhuber [25]. This algorithm consists of three gates: input gate, forget gate, output gate, and cell state. Long Short Term Memory cell is shown in figure 2.

The Cell State is the memory of the network and structure that enables communication that transmits meaningful information across cells to make predictions [26]. The forget gate decides which information will be transferred to other LSTM cells and forgotten. By applying the sigmoid function (σ), the information to be forgotten is decided [27]. On the other hand, the input gate performs the updating of the cell state. The output gate is the gate that determines the input of the next cell.

3. Evaluation metrics

The success of each model has been evaluated using accuracy, precision, recall, and F1 score metric. In the evaluation metrics formula, true positive expression is TP, true negative is TN, false positive is FP, and false negative is FN.

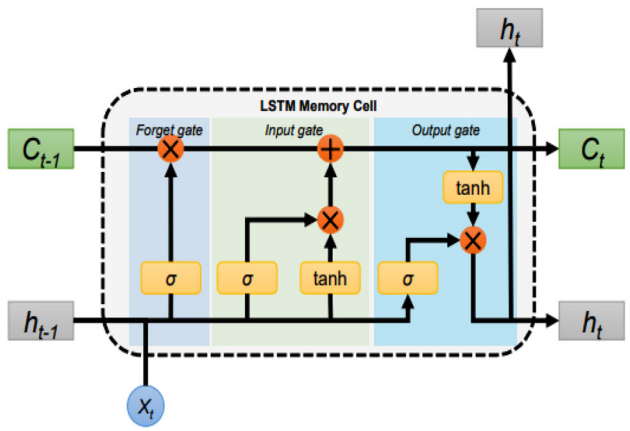

Figure 2. The basic structure of LSTM [25].

Accuracy metric calculates the ratio of correct predictions over the total number of samples evaluated [28]. The accuracy value is calculated using Equation (2).

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{2}$$

Precision metric is used to measure the positive patterns that are correctly predicted from all of the predicted patterns in a positive class [28]. The precision value is calculated using Equation (3).

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall, also known as the sensitivity metric, is used to measure the fraction of positive patterns that are correctly classified [28]. The recall value is calculated using Equation (4).

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

F1 Score metric is equal to the harmonic mean of the precision and recall metrics [28]. The F1 Score value is calculated using Equation (5).

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

4. Methodology

Seven, new and large-scale datasets named AI-TNKU-1, 2, 3, 4, 5, 6, and 7 for author identification, and six datasets named GI-TNKU-1, 2, 3, 4, 5, and 6 for genre identification were created to use in the solution of the problem of identification of author and genre in Turkish news texts. The datasets were imported into Python, and then preprocessing and modeling steps were applied to the datasets. The flowchart of the methodology is shown in figure 3.

4.1 *Data extraction*

The first stage of this study began with data extraction as indicated in figure 4. The raw dataset used in this study was extracted from a newspaper (http://www.hurriyet.com.tr/yazarlar/) from 08.11.1997 until 24.04.2019 using a web crawler [29, 30] written in Python. In this process, some authors on (https://www.hurriyet.com.tr/yazarlar/tum-yazarlar/) page were selected and a JSON rule was created for each author. When the JSON format rule of the author is run in the program, a file consisting of the name and surname of the author was created and the page of the news was saved in .txt format as a HTML document as well as

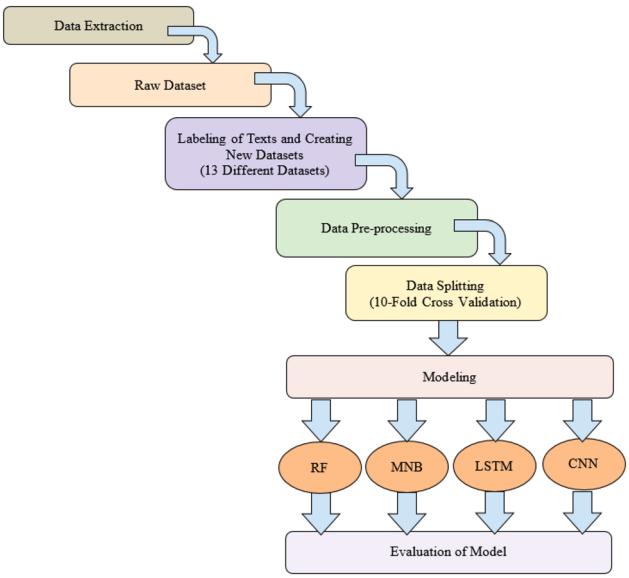

Figure 3. The flowchart of methodology.

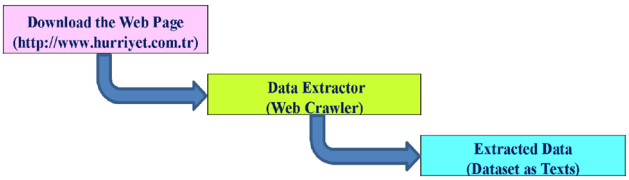

Figure 4. Basic structure of data extraction.

news texts. The title, publication date, and published news text were kept in each .txt file.

4.2 *Datasets*

In the second stage, the texts in the raw data obtained as a result of data extraction were labeled according to their genre and author, and 13 different, large-scale, and new datasets were created including multi-classes in Turkish. The details of these datasets are shown in tables 1 and 2. In addition, the datasets created for author and genre identification were shared on the Kaggle platform (https://www.kaggle.com/melikebektas/datasets).

4.3 *Data pre-processing*

The data pre-processing stage, which is a critical stage in any data science project, comes as the next stage, which prepares text data for modeling. In this stage, first, the datasets are imported into python. If the project is related to raw text data, the stages of pre-processing are followed as shown in figure 5 to clean unnecessary information, which are numbers, special characters, and unwanted spaces from raw text.

Tokenization is the first stage in text pre-processing. It is the process of splitting the given text into smaller pieces such as sentences or words called tokens. If the token is a word, it is called a word tokenizer, or is a sentence, it is called a sentence tokenizer. The word tokenizer, which splits sentences in a paragraph into words, is used in this study.

Second, all capital letters in the dataset have been converted to lowercase. Stopwords are considered noise in the text, and these words should be removed from the dataset. For this purpose, the list of tokens is filtered out from the list of stopwords for Turkish. Afterward, numbers and punctuation marks are cleaned from the list of tokens.

The last stage of the data pre-processing is word stemming, which is a procedure of normalization, which reduces words to their word stem of the derivational affixes. In this paper, the Zemberek [31] library is used to reduce the words to their stem in Turkish.

4.4 *Data splitting*

K-fold cross-validation is a statistical method for training machine and deep learning models that splits datasets into parts according to the k parameter [32]. In our study,

Table 1. Details of The Author Identification Datasets.

Dataset	Number of Classes	Number of Texts per Class	Number of Total Texts
AI-TNKU-1	68	100	6800
AI-TNKU-2	50	200	10000
AI-TNKU-3	38	300	11400
AI-TNKU-4	33	400	13200
AI-TNKU-5	27	500	13500
AI-TNKU-6	16	1000	16000
AI-TNKU-7	9	2000	18000

Table 2. Details of The Genre Identification Datasets.

Dataset	Number of Classes	Class (Genre) Name	Number of Texts per Class	Number of Total Texts
GI-TNKU-1	7	Economy/General/ General & Politics/ Magazine/Politics/ General Social Life/ Daily Social Life	3188	22316
GI-TNKU-2	6	Economy/General/ General & Politics/ Magazine/Politics/ Daily Social Life	3343	20058
GI-TNKU-3	5	Economy/ General & Politics/ Magazine/Politics/ Daily Social Life	3848	19240
GI-TNKU-4	4	Economy/ General & Politics/ Politics/ Daily Social Life	4064	16256
GI-TNKU-5	3	General & Politics/ Politics/ Daily Social Life	4760	14280
GI-TNKU-6	2	General & Politics/ Daily Social Life	5831	11662

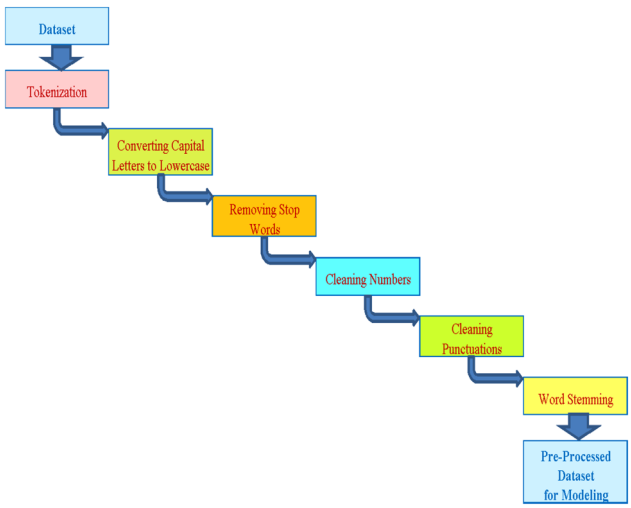

Figure 5. The flowchart of data pre-processing stage.

10-fold cross-validation was applied to the author and genre identification datasets. The datasets were randomly split into 10 equal parts and then at 10 different stages, one of these pieces was used as the test dataset and the other 9 pieces were used as the training dataset. Thus, each subset was used at least once for testing purposes.

5. Modeling

The pre-processed data were applied to four classifiers for modeling. The first two models were traditional Machine Learning (ML) models, in which MNB and RF classifiers were used. CNN and LSTM classifiers were used in the Deep Learning (DL) models.

5.1 *Machine learning models*

In the traditional ML models, the content of each string in the dataset is needed to be converted into a vector of numbers. For this purpose, first, a pipeline, which in the output of each step is given as input to the next one, is created with a TF-IDF Vectorizer, and the classifiers MNB and RF. The texts of the processed dataset have been converted to a matrix of token counts using Vectorizer, and then transformed a matrix into a TF-IDF representation using the TF-IDF Vectorizer. The results of the MNB and RF models of each dataset are given in table 3.

5.2 *Deep learning models*

CNN and LSTM were implemented using the Keras open-source neural network library. There are many ways to build the CNN and LSTM architectures. Thus, it is important to find an efficient and optimal deep learning

Table 3. Results of the MNB and RF models for the datasets.

Dataset	Model	Accuracy	Precision	Recall	F1 Score
AI-TNKU-1	MNB	68.47	76.32	68.47	72.18
	RF	76.80	76.74	76.80	76.76
AI-TNKU-2	MNB	75.64	81.52	75.64	78.47
	RF	80.60	80.62	80.60	80.61
AI-TNKU-3	MNB	80.09	84.98	80.09	82.46
	RF	83.93	84.39	83.93	84.15
AI-TNKU-4	MNB	81.15	85.35	81.15	83.19
	RF	85.17	85.47	85.17	85.31
AI-TNKU-5	MNB	85.47	88.11	85.47	86.76
	RF	88.88	89.16	88.88	89.01
AI-TNKU-6	MNB	88.41	89.44	88.41	88.92
	RF	90.41	90.39	90.41	90.40
AI-TNKU-7	MNB	92.87	93.59	92.87	93.22
	RF	95.72	95.85	95.72	95.78
GI-TNKU-1	MNB	75.96	77.92	75.96	76.92
	RF	83.04	83.54	83.04	83.28
GI-TNKU-2	MNB	79.02	80.42	79.01	79.70
	RF	84.52	84.96	84.52	84.73
GI-TNKU-3	MNB	84.79	85.69	84.79	85.23
	RF	89.18	89.45	89.18	89.31
GI-TNKU-4	MNB	84.97	85.99	84.96	85.47
	RF	88.34	88.87	88.33	88.59
GI-TNKU-5	MNB	86.64	86.90	86.64	86.76
	RF	92.47	92.60	92.47	92.53
GI-TNKU-6	MNB	90.72	90.60	90.12	90.35
	RF	95.93	95.95	95.93	95.94

model. Unfortunately, in deep learning models, many tests with many different hyperparameters are required to find the most optimal model. Hyperparameters are the parameters determined by the person who designed the model that the applied model cannot change by learning or predicting from the data [33]. In this study, in the embedding layer; max_len, max_words, embedding_dim hyperparameters, while creating models; neuron number, kernel size, activation function, layer number and dropout coefficient hyperparameters while running the model. The appropriate values of epoch, batch_size, optimizer, and loss function hyperparameters that increase the success of the model were obtained as a result of experimental studies.

In deep learning models, both the CNN model and the LSTM model, the model has started to be created with Sequential and Embedding layers, respectively. The embedding layer is used in deep learning models to digitize the dataset and transform it into a vector. Embedding is a method of associating a vector to a word. When we create an embedding layer, the values of the weights are determined randomly as in other layers. During the training, it is updated according to the gradient values with backpropagation and subsequently adapted to the structure of the incoming model [24].

Table 4. Parameters of deep learning models.

Dataset	max_len	max_words	embedding_dim
AI-TNKU-1	2000	100000	300
AI-TNKU-2	2000	100000	300
AI-TNKU-3	2000	100000	300
AI-TNKU-4	2000	100000	300
AI-TNKU-5	2000	150000	400
AI-TNKU-6	1000	100000	400
AI-TNKU-7	1000	100000	400
GI-TNKU-1	1000	100000	300
GI-TNKU-2	1000	100000	400
GI-TNKU-3	1000	100000	400
GI-TNKU-4	1000	100000	400
GI-TNKU-5	1000	100000	400
GI-TNKU-6	1000	100000	400

In the embedding layers of the models, the values specified in table 4 were found for the maximum length dimension (max_len) parameter, the maximum word number (max_words) parameter, and the embedding size (embedding_dim) parameter as a result of the experimental modeling studies made specifically for each dataset.

The best models for the datasets were found as a result of long experimental studies, which were applied to the datasets after passing through natural language processing stages, using deep learning algorithms to CNN and LSTM classifiers. Here, the results of the best CNN and LSTM models of each dataset found as a result of experimental studies are given in table 5.

6. Evaluation of models and results

According to table 3, which indicates the results of the ML models, and table 5, which indicates the results of the DL models for the datasets, in figure 6, all the models for author identification are illustrated and compared with each other's successes. In figure 7, all the models for genre identification are shown by comparing the results of the models.

When the success of the models for author and genre identification in figures 6 and 7 are compared for the datasets, the best models of the datasets are obtained as indicated in table 6.

According to table 6, it is seen that the best models found for author identification created by using the RF algorithm for the datasets named AI-TNKU-1, 2, 3, and 4, were obtained with the accuracies of 76.80%, 80.60%, 83.93%, and 85.17%, respectively. For the AI-TNKU-5 dataset, the model in which the LSTM algorithm was used was the highest success rate with an accuracy of 88.97%. The best models for the datasets named AI-TNKU-6 and 7 were

Table 5. Results of the CNN and LSTM models for the datasets.

Dataset	Model	Accuracy	Precision	Recall	F1 Score
AI-TNKU-1	CNN	47.48	52.50	45.38	48.68
	LSTM	55.16	61.39	53.51	57.17
AI-TNKU-2	CNN	71.06	75.74	68.95	72.18
	LSTM	56.41	62.18	52.29	56.80
AI-TNKU-3	CNN	77.85	80.58	76.62	78.55
	LSTM	67.46	72.46	64.85	68.44
AI-TNKU-4	CNN	78.79	80.48	76.43	78.40
	LSTM	73.49	77.42	72.00	74.61
AI-TNKU-5	CNN	88.18	90.17	87.26	88.69
	LSTM	88.97	90.62	88.37	89.48
AI-TNKU-6	CNN	91.73	92.71	91.29	91.99
	LSTM	91.63	92.59	91.22	91.89
AI-TNKU-7	CNN	95.81	96.22	95.68	95.94
	LSTM	95.81	96.24	95.66	95.94
GI-TNKU-1	CNN	83.97	85.87	83.00	84.41
	LSTM	84.15	86.81	82.53	84.61
GI-TNKU-2	CNN	86.54	88.42	85.65	87.01
	LSTM	86.62	88.54	85.53	87.00
GI-TNKU-3	CNN	90.98	91.96	90.53	91.23
	LSTM	89.59	90.90	89.06	89.97
GI-TNKU-4	CNN	92.00	93.03	91.31	92.16
	LSTM	91.57	92.67	90.97	91.81
GI-TNKU-5	CNN	94.01	94.11	93.62	93.86
	LSTM	93.43	93.96	93.24	93.59
GI-TNKU-6	CNN	96.69	96.77	96.69	96.72
	LSTM	96.73	96.79	96.73	96.75

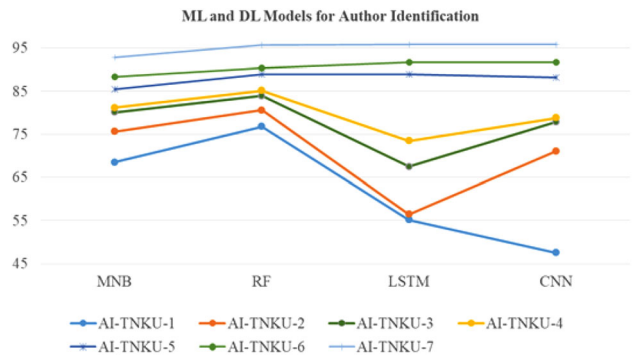

Figure 6. The accuracies of the models for each dataset for Author Identification.

obtained with 91.73% and 95.81% accuracies, respectively, using the CNN algorithm.

For author identification, the relations between the model achievements and the number of classes and text of the dataset were examined, and it is shown in figure 8. As seen in this figure, the performance of the models using the RF algorithm for the datasets named AI-TNKU-1, 2, 3, and 4, which have 68, 50, 38, and 33 classes, respectively, was found to be more successful than the other models.

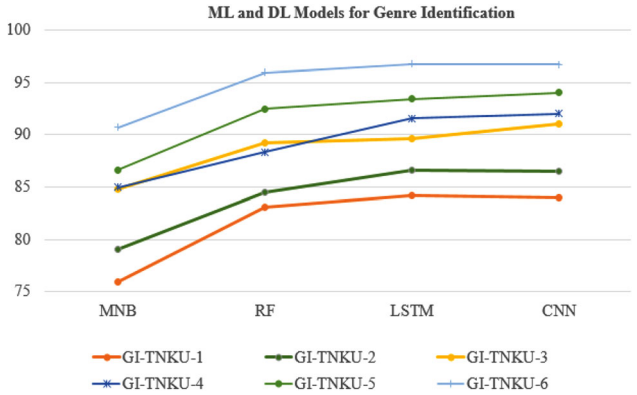

Figure 7. The accuracies of the models for each dataset for Genre Identification.

Table 6. The best models for the datasets.

Dataset	Model	Accuracy	Precision	Recall	F1 Score
AI-TNKU-1	RF	76.80	76.74	76.80	76.76
AI-TNKU-2	RF	80.60	80.62	80.60	80.61
AI-TNKU-3	RF	83.93	84.39	83.93	84.15
AI-TNKU-4	RF	85.17	85.47	85.17	85.31
AI-TNKU-5	LSTM	88.97	90.62	88.37	89.48
AI-TNKU-6	CNN	91.73	92.71	91.29	91.99
AI-TNKU-7	CNN	95.81	96.22	95.68	95.94
GI-TNKU-1	LSTM	84.15	86.81	82.53	84.61
GI-TNKU-2	LSTM	86.62	88.54	85.53	87.00
GI-TNKU-3	CNN	90.98	91.96	90.53	91.23
GI-TNKU-4	CNN	92.00	93.03	91.31	92.16
GI-TNKU-5	CNN	94.01	94.11	93.62	93.86
GI-TNKU-6	LSTM	96.73	96.79	96.73	96.75

Considering the number of texts in these datasets, it was observed that the numbers of texts per class were 100, 200, 300, and 400 texts, and the numbers of total texts were 6800, 10000, 11400, and 13200 texts. These results showed us that the RF algorithm as an ML algorithm was a more eligible algorithm in cases where the number of texts was low and the number of classes was high.

According to table 6, it is observed that the best models found for genre identification created by using the LSTM algorithm for the datasets named GI-TNKU-1, 2, and 6 were obtained with the accuracies of 84.15%, 86.62%, and 96.73%, respectively. The best models for the datasets named GI-TNKU-3, 4, and 5 were obtained with 90.98%, 92.00%, and 94.01% accuracies, respectively, using the CNN algorithm. These results showed that the DL models using CNN and LSTM algorithms were found more successful than ML models.

In figure 9, the relations between the model achievements and the number of classes and text of the dataset were examined for genre identification. As shown in this figure, the performance of the models using the LSTM algorithm for the datasets named GI-TNKU-1, 2, and 6,

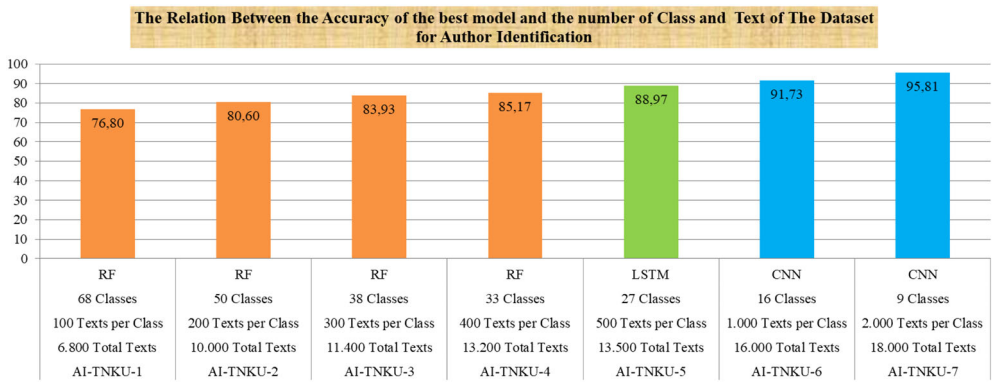

Figure 8. Best Model Achievements for Author Identification by Class and Text Numbers.

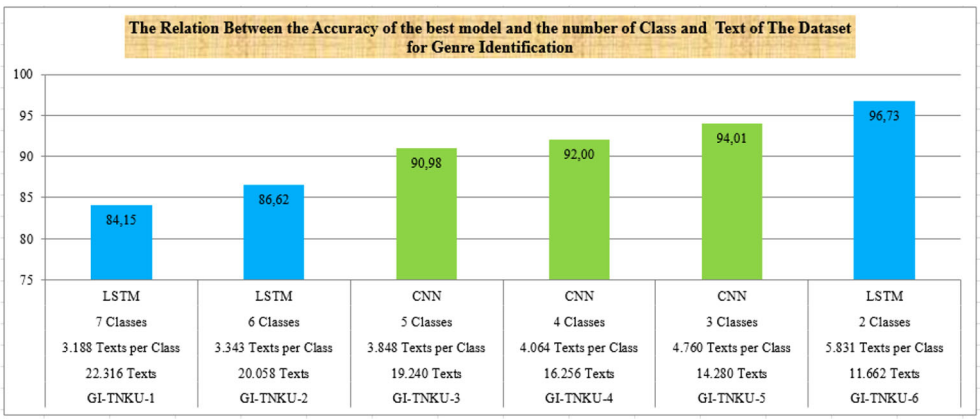

Figure 9. Best Model Achievements for Genre Identification by Class and Text Numbers.

which have 7, 6, and 2 classes, respectively, was found to be more successful than the other models. Considering the number of texts in these datasets, it was observed that the numbers of texts per class were 3188, 3343, and 5831 texts, and the numbers of total texts were 22316, 20058, and 11662 texts. It was observed that from the GI-TNKU-1 dataset to the GI-TNKU-6 dataset, the number of classes decreased, the number of texts in each class increased, and the model success also increased.

As a result, it was observed that the performance of DL models increased with the decrease in the number of classes and the increase in the number of texts per class. It was concluded that DL models would be more useful for the datasets, which include fewer classes and high texts per class, and ML models would be more appropriate for the datasets, which contain multiple classes and fewer texts per class.

7. Conclusions

In this study, some models were built for the identification of author and genre in Turkish news texts by using machine and deep learning algorithms. First, a total of 13 new datasets, which are 7 datasets named AI-TNKU-1, 2, 3, 4, 5, 6, and 7 for author identification and 6 datasets named GI-TNKU-1, 2, 3, 4, 5, and 6 for genre identification, with large-scale and multiple classes containing columnist articles of a newspaper, were created. These datasets have been made ready for the modeling stage by passing through natural language processing steps specific to the Turkish language and applying learning algorithms for classification processes. The considered algorithms are among the most successful and widely used learning algorithms, such as MNB, RF, CNN, and LSTM.

In the modeling steps, these algorithms have been applied to the datasets as a classifier, and the hyperparameter values with the highest performance from these classifiers have been tried to be found after long experimental studies. As a result of the modeling, the best models, that is, the models with the best accuracy, precision, recall, and F1 score values were obtained for each dataset and classifier.

In modeling for author identification, for the datasets named AI-TNKU-1, 2, 3, and 4, the best models, in which the RF algorithm was used, were obtained with the success rates of 76.80%, 80.60%, 83.93%, and 85.17%,

respectively. For the AI-TNKU-5 dataset, with the model using the LSTM algorithm, the highest accuracy success was 88.97%, for AI-TNKU-6 and 7 datasets, 91.73%, and 95.81%, respectively, with the model using the CNN algorithm.

For genre identification, the best models using the LSTM algorithm for the datasets named GI-TNKU-1, 2, and 6 were obtained with accuracies of 84.15%, 86.62%, and 96.73%, respectively. The best models for the datasets named GI-TNKU-3, 4, and 5 were obtained with 90.98%, 92.00%, and 94.01% accuracies, respectively, using the CNN algorithm. These results showed that the DL models for the task of genre identification using CNN and LSTM algorithms were found more successful than ML models for the datasets.

Considering the results of the best models obtained for author and genre identification, it was observed that there was a correlation between model achievement and the number of classes and texts per class. The relations for the datasets in the models were examined and the effects of these on the modeling performance were tried to be found as a general rule. As a result, it was observed that the performance of DL models increased with the decrease in the number of classes and the increase in the number of texts per class. It was concluded that DL models would be more useful for the datasets, which include fewer classes and high texts per class, and ML models would be more appropriate for the datasets, which contain multiple classes and fewer texts per class.

In future work, the new DL models will be attempted to build by using Capsule Networks for the text classification problems in Turkish.

Author's contribution The authors are solely responsible for the experimental works conducted in this paper, drafting of the paper and presentation of all the sections.

Funding The authors did not receive financial support from any organization for the submitted work.

Availability of data and material The authors hereby declare that the datasets created by themselves were shared on the Kaggle platform (https://www.kaggle.com/melikebektas/datasets).

Code availability Since, future works are based on the custom codes developed in this work, the code may not be available from the authors.

Declaration

Conflict of interest The authors have no relevant financial or nonfinancial interests to disclose.

References

[1] Bassam A, Masri A, Graham K and Shahrul A M N 2019 Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms. *Information Processing & Management* 56: 212–227

[2] Mehmet F A and Banu D 2006 Automatic Turkish Text Categorization in Terms of Author, Genre and Gender. In: *International Conference on Application of Natural Language to Information Systems*, pp. 221–226

[3] Zafer K and Banu D 2008 Genre and author detection in Turkish texts using artificial immune recognition systems. In: *16th Signal Processing, Communication and Applications Conference*, pp. 1–4

[4] Murat Y and Banu D 2012 Author recognition by Abstract Feature Extraction. In: *20th Signal Processing and Communications Applications Conference,* pp. 1-4

[5] Pınar T and Erdinç U 2013 Author detection by using different term weighting schemes. In: *21st Signal Processing and Communications Applications Conference.* pp. 1-4

[6] Durmuş O S, Oguz E K, Erdal K, and Armagan K 2018 A Text Classification Application: Poet Detection from Poetry. *arXiv e-prints*, p. arXiv:1810.11414

[7] Efstathios S 2008 Author identification: Using text sampling to handle the class imbalance problem. *Information Processing & Management* 44: 90–799

[8] Sibel D and Banu D 2010 Türkçe dokümanlar için N-gram tabanlı yeni bir sınıflandırma (Ng-ind): yazar, tür ve cinsiyet. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi* 3: 11–19

[9] Biveeken V and Muhammad M M F 2019 A New Method to Identify Short-Text Authors Using Combinations of Machine Learning and Natural Language Processing Techniques. *Procedia Computer Science* 159: 428–436

[10] Pınar T, Erdinç U, and Burak S 2012 Text classification of web based news articles by using Turkish grammatical features. In: *20th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4

[11] Rini W, Ferdinand A L, Brandon C T, Olivia R and Rudy, 2017 News Article Text Classification in Indonesian Language. *Procedia Computer Science* 116: 137–143

[12] Aleksandr S, Tatiana L, Dmitry G, Roman R and Ivan M 2016 Machine Learning Models of Text Categorization by Author Gender Using Topic-independent Features. *Procedia Computer Science* 101: 135–142

[13] Aleksandr S, Ivan M, Dmitry G, Anton S, Roman R and Tatiana L 2018 Automatic gender identification of author of Russian text by machine learning and neural net algorithms in case of gender deception. *Procedia Computer Science* 123: 417–423

[14] Aleksandr S, Ivan M, Dmitry G, Anton S, Roman R and Tatiana L 2018 Deep Learning neural nets versus traditional machine learning in gender identification of authors of RusProfiling texts. *Procedia Computer Science* 123: 424–431

[15] Na C, Rajarathnam C and Koduvayur P S 2011 Author gender identification from text. *Digital Investigation* 8: 78–88

[16] Shereen H, Mona F and ElSayed H 2019 Gender identification of egyptian dialect in twitter. *Egyptian Informatics Journal* 20: 109–116

[17] Kholoud A, Mahmoud A, Riyad A and Ghassan K 2017 Author gender identification from Arabic text. *Journal of Information Security Applications* 35: 85–95

[18] N R, Goenawan, William C, Derwin S, and Fredy P 2019 Gender Demography Classification on Instagram based on User's Comments Section. *Procedia Computer Science*, 157: 64–71

[19] Ritesh and Chakravarthy B 2018 Word Representations For Gender Classification Using Deep Learning. *Procedia Computer Science*, 132: 614–622

[20] Emad E A, Jamil R A and Muath A 2020 Age and Gender prediction in Open Domain Text. *Procedia Computer Science* 170: 563–570

[21] Yong-Bae L and Sung H M 2004 Automatic identification of text genres and their roles in subject-based categorization. In: *37th Annual Hawaii International Conference on System Sciences,* pp. 10–pp

[22] Jake V 2016 *Python data science handbook: Essential tools for working with data*. O'Reilly Media, Inc

[23] Leo B 2001 Random forests. *Machine Learning* 45: 5–32

[24] Ian G, Yoshua B and Aaron C 2016 *Deep learning*. MIT press

[25] Hongxang F, Mingliang J, Ligang X, Hua Z, Junxiang C and Jiahu J 2020 Comparison of long short term memory networks and the hydrological model in runoff simulation. *Water* 12: 175

[26] Kai S T, Richard S, and Christopher D M 2015 Improved semantic representations from tree-structured long short-term memory networks. *arXiv Prepr. arXiv1503.00075*

[27] Martin S, Ralf S and Hermann N 2012 LSTM neural networks for language modeling. In: *Thirteenth annual conference of the international speech communication association*

[28] Mohammad H and Md Nasir S 2015 A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process*, vol. 5

[29] Erdinç U 2020 A novel web scraping approach using the additional information obtained from web pages. *IEEE Access* 8: 61726–61740

[30] Erdinç U 2020 A regular expression generator based on CSS selectors for efficient extraction from HTML pages. *Turkish Journal of Electrical Engineering & Comput. Sciences* 28: 3389–3401

[31] Ahmet A A and Mehmet D A 2007 Zemberek, an open source NLP framework for Turkic languages. *Structure* 10: 1–5

[32] Richard R P and Dennis C 1984 Cross-validation of regression models. *Journal of American Statistical Association* 79: 575–583

[33] Erkan T and Fadime D 2019 Hiper Parametre Optimizasyonu Hyper Parameter Optimization. In: *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, pp. 1–5