Electronic Letters on Science & Engineering 18(1) (2022)
Available online at dergipark.org.tr/else

Poet Classification Using ANN and DNN

Ekin EKİNCİ[1*], Hidayet TAKCI[2], Sultan ALAGÖZ[3]

[1]Sakarya University of Applied Sciences, Faculty of Technology, Department of Computer Engineering, Sakarya, Turkey, ekinekinci@subu.edu.tr, ORCID: 0000-0003-0658-592X
[2]Sivas Cumhuriyet University, Faculty of Engineering Department of Computer Engineering, Sivas, Turkey, htakci@cumhuriyet.edu.tr, ORCID: 0000-0002-4448-4284
[3]Sivas Cumhuriyet University, Faculty of Economics and Administrative Sciences, Department of Management Information Systems, Sivas, Turkey, alagozsultan536@gmail.com, ORCID: 0000-0002-5978-8731

Abstract: Since statistical analysis of poetry is a challenging task in Natural Language Processing (NLP), making inferences about the poets also becomes a very challenging task. In this study, a dataset of Turkish poems which is obtained for 5 different poets is used to compare classification performance of the Artificial Neural Network (ANN) and Deep Neural Network (DNN) architectures. While Multilayer Perceptron (MLP) is selected for ANN architecture, Convolutional Neural Network (CNN) is selected as DNN architecture. Two main feature representation approaches are used for the experiments- Term Frequency-Inverse Document Frequency (TF-IDF) is used for ANN and word embedding is used for DNN. As a result of the experiments it has been seen that MLP has the highest performance in terms of accuracy, precision, recall and F-score.

Keywords: Poet classification, Classification, Natural Language Processing (NLP), Artificial Neural Network (ANN), Deep Neural Network (DNN)

YSA ve DSA Kullanılarak Şair Sınıflandırma

Özet: Doğal Dil İşleme'de (DDİ) şiirin istatistiksel analizi zorlu bir görev olduğundan, şairler hakkında çıkarımlar yapmak da oldukça zorlu bir görev haline gelmektedir. Bu çalışmada, 5 farklı şair için elde edilen Türkçe şiirlerden oluşan veri kümesi, Yapay Sinir Ağı (YSA) ve Derin Sinir Ağı (DSA) mimarilerinin sınıflandırma performansını karşılaştırmak için kullanılmıştır. YSA mimarisi olarak Çok Katmanlı Algılayıcı (MLP) seçilirken, DNN mimarisi olarak Evrişimsel Sinir Ağı (CNN) seçilmiştir. Deneyler için iki ana özellik temsili yaklaşımı kullanılmıştır- YSA için Terim Frekansı-Ters Terim Frekansı (TF-IDF), DSA'lar için kelime gömme kullanılmıştır. Deneyler sonucunda MLP, doğruluk, kesinlik, hassasiyet ve F-skoru açısından en yüksek performansa sahip olduğu görülmüştür.

Anahtar Kelimeler: Şair Sınıflandırma, Sınıflandırma, Doğal Dil İşleme (DDİ), Yapay sinir Ağları (YSA), Derin Sinir Ağları (DSA)

Reference to this paper should be made as follows (bu makaleye aşağıdaki şekilde atıfta bulunulmalı):
Ekin Ekinci, Hidayet Takcı, Sultan Alagöz 'Poet Classification Using ANN and DNN' , Elec Lett Sci Eng, vol. 18(1), (2022), 10-20.

1. Introduction

Authorship Attribution (AA) is the task of identifying the authors of a text document from a set of candidate authors by benefiting from some writing style markers. The basic idea in AA is the manifestation of habits defined as the continuous repetition of acquired behaviors in the texts written by the authors. The acquired writing habits emerge as the authors use the same writing style in the same way while creating the texts. While these constantly repeating features constitute an important clue in attributing the author of the text [1].

The history of determining an author is based on Mendenhall's work in the 19th century [2]. Mendenhall followed an approach based on the number of characters in the words in his study on literary texts. The turning point in the use of statistical methods in authorship analysis was conducted by Mosteller and Wallace in 1963 [3]. In this study, functional words were used as

[*] *Corresponding author; ekinekinci@subu.edu.tr*

textual features, and Bayes Theorem, one of the classification methods, was used to process these extracted functional words. Most of the AA studies followed were statistical [4-7].

Due to the successful applications of machine learning and deep learning, researchers in the AA area have begun to focus on the field of machine learning and deep learning as well. In addition, machine learning and deep learning methods have yielded higher accuracy than statistical methods [8]. This is because statistical methods are black box models and depend on data [9]. Also, machine learning and deep learning methods are noise tolerant and can handle nonlinear relations among features [10].

AA can be designed as a binary or multi-class single label classification problem. Therefore, machine learning and deep learning methods deal with AA problems through assigning class labels to text data. In the literature, as machine learning approaches Support Vector Machine (SVM) [11, 12], Decision Tree (DT) [13], Naïve Bayes (NB) [14, 15], k-nearest neighbor (KNN) [16], ANN [17, 18], ensemble classifiers [19, 20] are realized and as deep learning approaches Recurrent Neural Networks (RNN) [21], LSTM [22], CNN [23, 24] and so on are realized for AA problem.

Although the history of the AA dates back much further, it is still one of the research areas on which there is a lot of work. Especially, thanks to the internet, today, millions of different types of natural language text documents are provided to users electronically. These types include code [25, 26], e-mail [27, 28], micro messages (such as twitter feeds, blogs, reddit comments) [29-33], literature texts [34, 35] and so on. When the literature is examined, it is seen that while a lot of studies have been done for these types of documents, the number of studies on poems is very limited.

Hoorn et al. [36] benefited from letter sequences to realize poet classification. For this purpose, the authors represented poems in Dutch with tri-grams and different sizes of windows. While tri-gram features were trained with KNN, NB and ANN, window based features were trained with only neural networks. The most successful method for tri-gram features was the neural network. For window size based features, the most successful result was obtained when the size was 8. Shahmiri et al. [37] developed a two-class poet identification and aimed to find out whether the existing poems were written by Shahnameh of Ferdowsi. As a result of the reduction of the poetic features, the authors determined that ANN is much more successful than NB as a result of the classification. Mohammad [38] classified Arabic poetries into twenty different poet classes. As a classification algorithm NB was used and 66% micro average were obtained. Can et al. [39] categorized Ottoman poems by poets. To achieve the categorization task, most frequent words, token lengths, two-word collocations, and type lengths were selected as features and as classifiers NB, SVM were used. Rakshit et al. [40] utilized a combination of lexical and stylometric features namely orthographic, syntactic and phonemic features for poet identification. Pandian et al. [13] extracted lexical, syntactic and semantic features from poems to determine poets with C4.5 algorithm. Ahmed et al. [41, 42] made poet attribution based Arabic poems. They extracted characters, sentence length of poems, word length, rhyme, meter and first word of the sentence as features and fed these features separately and combined into NB, SVM and Sequential Minimal Optimization (SMO). Waijanya and Promrit [43] classified Thai poems based on poets with CNN. The authors made classifications over different numbers of poets and observed that while achieving 100% accuracy for two authors, the accuracy decreased to 55% with the increase in the number of poets. Şahin et al. [44] to classify poems based on poets created a dataset of three different poets in English. They applied five different classification algorithms namely SMO, C4.5, Random Forest (RF), and KNN by performing filtering on the TF-IDF features with Chi Square (CHI) they obtained from poems. The SMO had the best success with 70% F-score. Tariq et al.

[45] proposed to identify poets of Urdu Ghazal by applying feature reduction using TF-IDF matrix. As a result of the experiments, it observed that feature reduction provided an increase in performance for classical machine learning algorithms. Salami and Momtazi [46] compared RNN in terms of poet identification.

In this study, MLP from classical ANN architectures and CNN networks from DNN architectures are used for poet classification. Although DNNs often outperform classical ANN architectures, the MLP algorithm in this study outperforms CNN based on the accuracy, precision, recall and F-score, interestingly. The success of classical machine learning algorithms or deep learning algorithms is related to the data used. While deep learning algorithms give higher results in larger sized data, classical machine learning algorithms can give higher results in lower dimensional data. Therefore, this study has shown us once again that not every algorithm gives the best results everywhere, but gives the best results under suitable conditions.

The rest of the paper is organized as follows. In section 2, pre-processing steps, weighting scheme and classification algorithms are mentioned. In Section 3, experiments and experimental results are given in detail. Finally, discussions and conclusions for the future work are summarized in Section 4.

2. Proposed Methodology

2.1. Feature Engineering

2.1.1. Pre-processing

In AA problems, pre-processing is the most important step due to transforming the text into a meaningful form from which information can be extracted. This transforming step is realized by eliminating stop-words, stemming, tokenizing, lemmatizing, normalizing, converting to lowercase and removing out-of-vocabulary (OOV) words. These steps are general and adapted to the task to be proposed by using one or more of them.

In the pre-processing step of the poems, firstly, conversion of text in lower case is carried out because of lack of case sensitivity. The OOV words which are accepted as noise are removed from the texts. Then stop-words, which do not give clues about the writing style of the poets, are eliminated. Then poems are tokenized. All these steps are realized by using the Python NLP library called Natural Language Toolkit (NLTK)[1]. After stemming is performed on the poems by using Snowball stemmer[2]. At the end of the preprocessing steps, the original form of the poem line, which is "ne bülbül kaldı ne gül kül oluverdi dünya", becomes "kal bülbül kal gül kül oluver dünya".

2.1.2. Feature Construction

The TF-IDF is a measure of weight which tells the importance of a word to a poem in the corpus [47]. In the realm of machine learning for AA tasks, TF-IDF is the most preferred real-valued representation because of its simplicity and efficiency. The heuristic intuition behind the TF-IDF is if a word occurs in too many poems that word is of little importance for the poetry collection,

[1] https://www.nltk.org/
[2] https://snowballstem.org/

that is, its weight is low. On the contrary, the related word is of high importance, that is, its weight is also high. A high weight means that the word has a high distinctiveness in classification.

The equation of the TF-IDF is given with Eq. 1 as below:

$$\mathrm{TF-IDF}(t,d) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t) \tag{1}$$

In the equation above $\mathrm{TF}(t,d)$ represents the number of times term t occurs in the document d and calculated based on Eq. 2. $\mathrm{IDF}(t)$ represents the inverse document frequency of term t and calculated by using Eq. 3.

$$\mathrm{TF}(t,d) = \frac{\#\mathrm{term\_t\_occurs\_in\_document\_d}}{\mathrm{total\_\#terms\_in\_document\_d}} \tag{2}$$

$$\mathrm{IDF}(d) = \frac{\#\mathrm{documents}}{\#\mathrm{documents\_with\_term\_t\_in\_it}} \tag{3}$$

Mathematical representation of texts is informative for studies of NLP such as classification. For deep learning based NLP studies texts are represented by means of mathematical representation. One of these is Global Vectors (GloVe). GloVe is an unsupervised word-vector learning algorithm. The GloVe model is trained on the word-word co-occurrence matrix and generates the word-vector.

2.2. Classification Models

2.2.1. MLP

ANNs, which are inspired by the human brain and can produce new information, associate and generalize information through the learning way of the human brain; are computer systems that are connected to each other via weighted links and that process information in parallel with process elements (neurons), each of which has its own distributed memory [1]. Artificial neural networks, which are used in areas such as classification, pattern recognition, and prediction, are very powerful and popular methods thanks to their ability to learn non-linear and implicit relationships between inputs and outputs through mathematical functions. In the solution of non-linear problems in artificial neural networks, the MLP algorithm is used. This network consists of an input layer, one or more hidden layers, and an output layer. In each layer, there are one or several neurons which are used to connect the layer with the following. The generalized delta learning rule is used in this artificial neural network model, which learns from the inputs given to it and the outputs corresponding to these inputs. The generalized delta learning rule provides learning in two stages: forward feed and backward propagation. In forward feed, the network produces outputs according to the inputs given to it, while in backward propagation, the connection weights are updated. The purpose of updating the connection weights is to minimize the error value. The mathematical formula of MLP is given below.

$$O_k = f_2\left(\sum_{i=1}^{N} w_{k,i} f_1\left(\sum_{j=1}^{M} w_{i,j} x_j + w_{i,o}\right) + w_{k,o}\right) \tag{4}$$

In the Eq. 4, O_k is the net input to the k^{th} neuron in the output layer as a result of the activation function f_2. M and N represent the neuron size in the input and hidden layers, respectively. $w_{k,i}$ is the weight between kth and i^{th} neuron in the output and hidden layers, respectively. Activation function f_1 is used for the net input to the i^{th} neuron in the hidden layer. $w_{(i,j)}$ is the weight between i^{th} and j^{th} neuron in the hidden and input layers, respectively. $w_{(i,o)}$ and $w_{(k,o)}$ are the bias values for the neurons in the hidden and output layer, respectively. MLP architecture with one hidden layer is given with Figure 1.

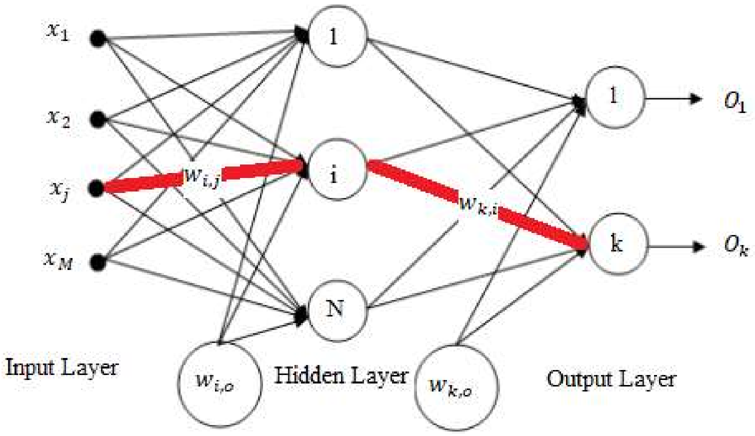

Figure 1. MLP architecture with one hidden layer

2.2.2. CNN

CNN has superior capability in classification of image, audio and text data compared to RNN and LSTM. CNN is a mathematical model which is designed to transform input into a more useful representation [48]. CNN is composed of three layers namely convolution, pooling, and fully connected layers. While the convolution and pooling layers feature extraction is realized, a fully connected layer uses the extracted features to obtain output.

Convolutional layer is the key among layers due to applying mathematical and linear operations. In this layer feature extraction is realized. Pooling layer performs down-sampling to reduce the dimension of the feature map. Thus reducing the number of parameters to be learned. The fully connected layer transforms the feature maps of the convolution layer and pooling layer to a one-dimensional feature vector. The architecture of the CNN is given with Figure 2.

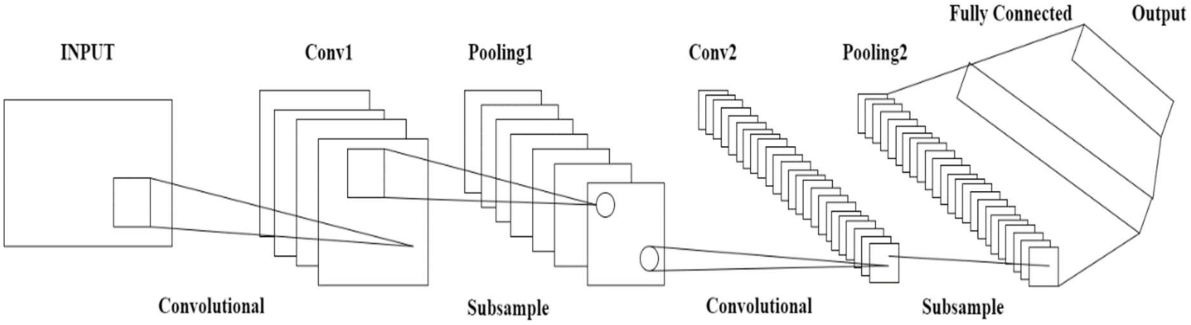

Figure 2. The inner structure of CNN unit

3. Experiments

3.1. Dataset and Evaluation Metrics

To carry out the experiments, we collected a total of 314 poems from 5 different poets from different websites. The whole datasets are in Turkish and publicly available. The dataset was divided into two parts as training and test sets with 80% and 20% rates, respectively. The training set contains 251 poems and the test set contains 63 poems. The summary of the dataset is given with Table 1.

Table 1. Summary of dataset

Poet	# of Poems	Average # words per poem
Poet1	63	86
Poet2	56	115
Poet3	60	106
Poet4	60	137
Poet5	75	320

To evaluate the ANN and DNN architectures precision, recall, accuracy and F-measure scores are calculated. The definitions of these metrics are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \tag{7}$$

$$\text{F} - \text{Measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

where true positive (TP) is the number of poets correctly classified as X, false positive (FP) is the number of X poets classified as incorrectly, false negative (FN) is the number of poets incorrectly classified as X and true negative (TN) is the number of correctly classified poets (except X).

3.2. Experimental Settings and Results

To carry out the experiments, we collected a total of 314 poems from 5 different poets from different websites. In this study we aim to compare neural network architectures to learn which model is the best for classification of poets. All of our experiments run on Intel(R) Core(TM) i5-6500@3.20GHz, with 8 GB RAM. The operating system is Windows10. While for realizing MLP we use sklearn Library of Python[3], for realizing RNN and LSTM architectures, we use a high-level Python library which is called as Keras[4].

In order to run the models, the model parameters namely number of input neurons, hidden layers, hidden neurons, epoch and batch sizes must first be determined. The number of hidden neurons is determined based on feature size in the dataset. In this study, our features are extracted with TF-IDF and maximum feature size is selected as 1500. Therefore, the number of input neuron sizes is equal to 1500 for all architectures. In MLP, there is one hidden layer with 100 neurons. The regularization parameter alpha is set as 0.0001 and maximum iteration count is set as 300. For the optimization of the weight adam is used as a solver. In CNN, for the convolutional network layer, we use word embeddings with the dimension of 100 that are initialized by GloVe as in LSTM. The first dense layer has 1024 neurons and relu as activation function. The second dense layer has 524 neurons and relu as activation function. The last dense layer has 63 neurons and softmax as activation function. For the optimization of the weight adam is used as a solver with learning rate is equal to 0.000055. Number of epoch and batch size set to 1000 and 1280, respectively. Comparison results of the models are given with Table 2.

Table 2. Comparison results

Architecture	Precision	Recall	F-Score	Accuracy
MLP	0.83	0.81	0.81	0.81
CNN	0.63	0.61	0.59	0.61

When the comparison results are examined it has been seen that prediction accuracy of MLP is higher than that of DNN architecture CNN for this problem. Hence, we can conclude that MLP is the best suited classifier for poet classification.

For MLP, the confusion matrix is given with Table 3 to evaluate architecture in detail for each author.

Table 3. Confusion matrix for MLP

	Poet1	Poet2	Poet3	Poet4	Poet5
Poet1	11	1	2	1	0
Poet2	5	6	6	0	1
Poet3	2	0	7	0	0
Poet4	0	2	0	2	1
Poet5	1	0	1	1	13

According to Table 3, the results obtained for the Poet5 are the most successful. Of the 16 poems written by the Poet5, 13 are predicted correctly. It is followed by the Poet1. 11 of the 15 poems written by the Poet1 are predicted correctly. While these two poets give the best values in terms of both precision and recall, Poet2 gives the most unsuccessful result. Only 6 of the 18 poems written by the Poet 2 are predicted correctly. This situation tells us that the Poet 1 and Poet5 have a distinctive style for them, while the Poet does not have a distinctive style of his own. Therefore, the data used is as important as the algorithms in the success of poet classification. A data set composed of poets with distinctive style may give higher classification accuracy, while those with unambiguous style may give lower accuracy.

[3] https://scikit-learn.org/stable/

[4] https://pypi.org/project/Keras/

4. Conclusions

Since poet classification is a difficult task, there are not many studies on it in the literature. In this paper, to address the poet classification, we compare ANN and DNN architectures by using Turkish poem dataset. As ANN architecture we use MLP and as DNN architecture we use CNN. Results show that the MLP is the best compared with CNN. The reason for this difference in the success of the classification models depends on the selected data set, the preprocessing steps applied to the dataset, the extracted features and algorithm parameters. When the classification success per poet is evaluated, it is seen that the poets with distinctive stylistic features are classified well. In future, we would like to improve classification accuracy by enhancing feature construction and neural network architectures.

References

[1]. E. Ekinci, "Using authorship analysis techniques in forensic analysis of electronic mails," M.S. thesis, Dept. Computer Eng., Gebze Techinal Univ., Kocaeli, Turkey, 2013.

[2]. T. C. Mendenhall, "Characteristic curves of composition," AAAS, vol. 9, no. 214, pp. 237–246, Mar. 1887, doi: 10.1126/science.ns-9.214S.237.

[3]. F. Mosteller and D. L. Wallace, "Inference in authorship problem," J. Am. Stat. Assoc., vol. 58, no. 302, pp. 275–309, Jun. 1963, doi: 10.2307/2283270.

[4]. D. Holmes, "Authorship attribution," Comput. Hum., vol. 28, no. 2, pp. 87–106, Apr. 1994, doi: 10.1007/BF01830689.

[5]. D. Holmes, "The evolution of stylometry in humanities scholarship," Lit. Ling. Comput., vol. 13, no. 3, pp. 111–117, Sept. 1998, doi: 10.1093/llc/13.3.111.

[6]. A. McEnery and M. Oakes, "Authorship studies/textual statistics," in Handbook of Natural Language Processing, R. Dale, H. Moisl and H. Somers, Eds., Dallas, USA: Marcel Dekker Inc., 2000, pp. 234–248.

[7]. A. Rico-Sulayes, "Statistical authorship attribution of Mexican drug traficking online forum posts," Int. J. Speech, Lang. Law, vol. 18, no. 1, pp. 53–74, Sept. 2011, doi: 10.1558/ijsll.v18i1.53.

[8]. P. Hajek and R. Henriques, "Mining corporate annual reports for intelligent detection of financial statement fraud--A comparative study of machine learning methods," Knowl.-Based Syst., vol. 128, pp. 139–152, Jul. 2017, doi: 10.1016/j.knosys.2017.05.001.

[9]. R. Jindal, R. Malhotra and A. Jain, "Techniques for text classification: literature review and current trends," Webology, vol. 12, no. 2, 2015.

[10]. R. Zheng, J. Li, H. Chen and Z. Huang, "A framework for authorship identification of online messages: writing-style features and classification techniques," JASIST, vol. 57, no. 3, pp. 378–393, Dec. 2005, doi: 10.1002/asi.20316.

[11]. S. Ouamour and H. Sayoud, "Authorship attribution of ancient texts written by ten Arabic travelers using a smo-svm classifier," in Proc. 2012 Int. Conf. on Communications and Information Technology (ICCIT), Jun. 2012, pp. 44–47, doi: 10.1109/ICCITechnol.2012.6285841.

[12]. P. Varela, M. Albonico, E. Justino and J. Assis, "Authorship attribution in Latin languages using stylometry," IEEE Lat. Am. Trans., vol. 18, no. 4, pp. 729–735, Apr. 2020, doi: 10.1109/TLA.2020.9082216.

[13]. A. Pandian, V. V. Ramalingam and R. P. V. Preet, "Authorship identification for Tamil classical poem (Mukkoodar Pallu) using C4.5 algorithm," Indian J. Sci., vol. 9, no. 47, pp. 1–5, Dec. 2016, doi: 10.17485/ijst/2016/v9i47/107944.

[14]. A. S. Altheneyan and M. C. Menai, "Naïve Bayes classifiers for authorship attribution of Arabic texts," J. King Saud Univ., Comp. & Info. Sci., vol. 26, no. 4, pp. 473–484, Dec.

2014, doi: 10.1016/j.jksuci.2014.06.006.

[15]. D. M. Anisuzzaman and A. Salam, "Authorship attribution for Bengali language using the fusion of n-gram and naive bayes algorithms," IJITCS, vol. 10, pp. 11–21, Oct. 2018, doi: 10.5815/ijitcs.2018.10.02.

[16]. H. Zamani, H. N. Esfahani, P. Babaie, S. Abnar, M. Dehghani and A. Shakery, "Authorship identification using dynamic selection of features from probabilistic feature set," in Information Access Evaluation. Multilinguality, Multimodality, and Interaction, vol. 8685, E. Kanoulaset al., Eds., Switzerland: Springer, Cham, 2014, pp. 128–140.

[17]. R. L. Priya and G. Manimannan, "Authorship attribution of Tamil articles using artificial neural network," IJSIMR, vol. 3, no. 6, pp. 22–28, Jun. 2015.

[18]. X. Yang, G. Xu, Q. Li, Y. Guo and M. Zhang, "Authorship attribution of source code by using back propagation neural network based on particle swarm optimization," PLoS ONE, vol. 12, no. 11, pp. 1–18, Nov. 2017, doi: 10.1371/journal.pone.0187204.

[19]. M. Al-Sarem, F. Saeed, A. Alsaeedi, W. Boulila and T. Al-Hadhrami, "Ensemble methods for instance-based Arabic language authorship attribution," IEEE Access, vol. 8, pp. 17331–17345, Jan. 2020, doi: 10.1109/ACCESS.2020.2964952.

[20]. C. Suman, A. Raj, S. Saha and P. Bhattacharyya, "Source code authorship attribution using stacked classifier," presented at the 12th meeting of the Forum for Information Retrieval Evaluation (FIRE 2020), Hyderabad, India, Dec. 16-20, 2020.

[21]. C. Zhao, W. Song, X. Liu, L. Liu and X. Zhao, "Research on authorship attribution of article fragments via RNNs," in Proc. 2018 IEEE 9th Int. Conf. on Software Engineering and Service Science (ICSESS), Nov. 2018, pp. 156–159, doi: 10.1109/ICSESS.2018.8663814.

[22]. B. Alsulami, E. Dauber, R. Harang, S. Mancoridis and R. Greenstadt, "Source code authorship attribution using long short-term memory based networks," in Computer Security – ESORICS 2017, vol. 10492, S. Foley, D. Gollmann and E. Snekkenes, Eds., Switzerland: Springer, Cham, 2017

[23]. P. Shrestha, S. Sierra, F. A. Gonzalez, M. Montes, P. Rosso and T. Solorio, "Research on authorship attribution of article fragments via RNNs," in Proc. 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 2017, pp. 669–674.

[24]. A. Khatun, A. Rahman, S. Islam and M. E. Jannat, "Authorship attribution in Bangla literature using character-level CNN," in Proc. 2019 22nd Int. Conf. on Computer and Information Technology (ICCIT), 2019, pp. 1–5, doi: 10.1109/ICCIT48885.2019.9038560.

[25]. V. Kalgutkar, R. Kaur, H. Gonzalez, N. Stakhanova and A. Matyukhina, "Code authorship attribution: methods and challenges," ACM Comput. Surv., vol. 52, no.1, pp. 1–36, Feb. 2019, doi: 10.1145/3292577.

[26]. R. Mateless, O. Tsur and R. Moskovitch, "Pkg2Vec: Hierarchical package embedding for code authorship attribution," Future Gener. Comput. Syst., vol. 116, pp. 49–60, Mar. 2021, doi: 10.1016/j.future.2020.10.020.

[27]. M. R. Schmid, F. Iqbal and B. C. M. Fung, "E-mail authorship attribution using customized associative classification," Digit. Investig., vol. 14, no. 1, pp. S116–S126, Aug. 2015, doi: 10.1016/j.diin.2015.05.012.

[28]. Y. Fang, Y. Yang and C. Huang, "EmailDetective: an email authorship identification and verification model," Comput. J., vol. 63, no. 11, pp. 1775–1787, Jul. 2020, doi: 10.1093/comjnl/bxaa059.

[29]. M. H. Altakrori, F. Iqbal, B. C. M. Fung, S. H. H. Ding and A. Tubaishat, "Arabic authorship attribution: an extensive study on twitter posts," ACM Trans. Asian Low-Resour. Lang. Inf. Process., vol. 18, no. 1, pp. 1–51, Nov. 2018, doi: 10.1145/3236391.

[30]. C. Suman, A. Raj, S. Saha and P. Bhattacharyya, "Authorship attribution of microtext using capsule networks," IEEE Trans. Comput. Soc. Syst., Apr. 2021. [Online]. Available:

https://ieeexplore.ieee.org/document/9393500.

[31]. T. Cavalcante, A. Rocha and A. Carvalho, “Large-scale micro-blog authorship attribution: beyond simple feature engineering,” in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, vol. 8827, E. Bayro-Corrochano and E. Hancock, Eds., Switzerland: Springer, Cham, 2014, pp. 399–407.

[32]. P. Canbay, E. A. Sezer and H. Sever, “Binary background model with geometric mean for author-independent authorship verification,” J. Inf. Sci., pp. 1–17, May. 2021, doi: 10.1177/01655515211007710.

[33]. G. R. Casimiro and L. A. Digiampietri, “Authorship attribution using data from Reddit forum,” in Proc. SBSI'20: XVI Brazilian Symposium on Information Systems, 2020, pp. 1–8, doi: 10.1145/3411564.3411616.

[34]. M. Kestemont, “Stylometric authorship attribution for the middle Dutch mystical tradition from Groenendaal,” Dutch Crossing, vol. 42, no. 1, pp. 1–51, Nov. 2018, doi: 10.1145/3236391.

[35]. T. Boran, M. Martinaj and M. S. Hossain, “Authorship identification on limited samplings,” Comput. Secur., vol. 97, pp. 101943, Oct. 2020, doi: 10.1016/j.cose.2020.101943.

[36]. J. F. Hoorn, S. L. Frank, W. Kowalczyk and F. van der Ham, “Neural network identification of poets using letter sequences,” Lit. Linguistics Comput., vol. 14, no. 3, pp. 311–338, Sep. 1999, doi: 10.1093/llc/14.3.311.

[37]. A. S. Shahmiri, R. Dezhkam and S. Shirey, “Poet identification for Shahnameh of Ferdowsi using artificial neural networks,” The CSI Journal on Computer Science and Engineering, vol. 4, no. 3(a), pp. 17–26, 2006.

[38]. I. A. Mohammad, “Naïve Bayes for Classical Arabic Poetry Classification,” Al-Nahrain Journal of Science, vol. 12, no. 4, pp. 217–225, Dec. 2009.

[39]. E. F. Can, F. Can, P. Duygulu and M. Kalpakli, “Automatic categorization of Ottoman literary texts by poet and time period,” in Computer and Information Sciences II, E. Gelenbe, R. Lent and G. Sakellari, Eds., UK: Springer, London, 2012, pp. 52–57.

[40]. G. Rakshit, A. Ghosh, P. Bhattacharyya and G. Haffari, “Automated analysis of Bangla poetry for classification and poet identification,” in Proc12th Intl. Conference on Natural Language Processing, Trivandrum, India, 2015, pp. 247–253.

[41]. A. Ahmed, R. Mohamed and B. Mostafa, “Authorship attribution in Arabic poetry using NB, SVM, SMO,” in Proc. 2016 11th International Conf. on Intelligent Systems: Theories and Applications (SITA), 2016, pp. 1–5, doi: 10.1109/SITA.2016.7772287.

[42]. A. Ahmed, R. Mohamed and B. Mostafa, “Machine learning for authorship attribution in Arabic poetry,” Int. J. Future Comput. Commun., vol. 6, no. 2, pp. 42–46, June 2017, doi: 10.18178/ijfcc.2017.6.2.486.

[43]. S. Waijanya and N. Promrit, “The poet identification using convolutional neural networks,” in Intelligent Systems and Computing, vol. 566, P. Meesad, S. Sodsee and H. Unger, Eds., Switzerland: Springer, Cham, 2017, pp. 179–187.

[44]. D. Ö. Şahin, O. E. Kural, E. Kılıç and A. Karabina, “A text classification application: poet detection from poetry,” in Proc. International Conf. on Engineering Technologies (ICENTE’17), Konya, Turkey, 2017, pp. 228–230.

[45]. N. Tariq, I. Ezaj, M. K. Malik, Z. Nawaz and F. Bukhari, “Identification of Urdu ghazal poets using SVM,” Mehran University Research Journal of Engineering & Technology, vol. 38, no. 4 pp. 935–944, Oct. 2019, doi: 10.22581/muet1982.1904.07.

[46]. D. Salami and S. Momtazi, “Recurrent convolutional neural networks for poet identification,” Digit. Scholarsh. Humanit., vol. 36, no. 2, pp. 472–481, April 2020, doi: 10.1093/llc/fqz096.

[47]. H. Christian, M. P. Agus and D. Suhartono, “Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF),” ComTech: Computer,

Mathematics and Engineering Applications, vol. 7, no. 4, pp. 285–294, Dec. 2016.

[48]. W. Singleton and M. El-Sharkawy, “Increasing cnn representational power using absolute cosine value regularization,” in 2020 IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), Springfield, MA, USA, 2020, pp. 391-394.