Decision Analytics Journal 4 (2022) 100111

Contents lists available at ScienceDirect

Decision Analytics Journal

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

A deep learning classification model for Persian Hafez poetry based on the poet's era

Jannatul Ferdous Ruma, Sharmin Akter, Jesrin Jahan Laboni, Rashedur M. Rahman *

Department of Electrical & Computer Engineering, North South University, Dhaka 1229, Bangladesh

ARTICLE INFO

Keywords:
Natural Language Processing
Persian text classification
Neural network
Long Short-Term Memory (LSTM)
Bidirectional Long Short-Term Memory (Bi-LSTM)
Gated Recurrent Unit (GRU)
Paragraph Vector (Doc2Vec)
Distributed Memory

ABSTRACT

More than any other literary genre, poetry presents a significant challenge for Natural Language Processing (NLP) algorithms. Small poetries in the Persian language are called ghazal. Ghazal classification by document embedding technique and sequential learning on poetic era is an under-explored area of research till now. Deep learning and document embedding technique is explored in the current study. We have worked with Persian Ghazal, which Hafez writes. We have found and employed useful NLP approaches to facilitate and automate the classification of Hafez's poetry. We developed and implemented a set of rigorous and repeatable techniques that may be extended to different types of poetries. It is a part of Persian text classification and NLP. We have implemented neural network models that automatically classify Hafez's Persian poetry chronologically with around 85% accuracy. This proposed model is significantly better than previously reported work in the Persian Language on poetry data. In the Persian language, meter classification and machine learning-based poetry classification were done before. We have introduced a classification method based on the poet's era using sequential architectures. We found the highest accuracy when we used the Distributed Memory model for document embedding and Long Short-Term Memory (LSTM) model for training the Persian Hafez ghazals. We have achieved approximately 87% precision, 85% F1-score and 85% recall score by using our model. To perform this classification, we have used refined Hafez ghazals' labels and found better accuracy than Bidirectional Long Short-Term Memory (Bi-LSTM) and Gated Recurrent Units (GRU) models.

1. Introduction

Poetry is a literary work in which diction, rhythm, and visuals are used to emphasize the expression of feelings and ideas. Poems are typically written to portray feelings such as love, generosity, emotion, and respect. A poem can be as broad or as detailed as the poet desires. The words of a poem may not be used literally, but they may have a deeper meaning in context. This is what attracts computational linguistics to poetry. Poetry is a literary genre based on a certain style that produces a rhyme, whereas other literature forms contain phrases and paragraphs but lack a metrical framework.

A ghazal is poetry or, to put it another way, a formal structure within which a number of ideas can be presented in the true essence of the moment [1]. The ghazal includes five to fifteen rhyming couplets with a repetition at the start of the following line in its traditional form [2]. The practice of using many adjectives to describe the same subject is common in Persian poetry. Using two or even three adjectives in succession is customary in Iran. All of these terms may have the same meaning in English, yet each adjective in Persian denotes something somewhat different. Light verbs and the recurrence of agreement indicators in Persian are crucial literary strategies, whereas global language like English lacks.

Poetry, like no other literary form, poses a substantial challenge when it comes to employing Natural Language Processing (NLP) algorithms. Classification of poetry texts is a form of text classification [3] or document classification [4]. The Persian language is more challenging to classify than the English language. The use of a large range of declensional suffixes is one of the problems of the Persian language [5]. Word spacing is another typical issue in Persian literature. In Persian, a word's component is detached by an intra-word space called pseudo-space, in addition to white space as inter-word space. Many obstacles in Persian text processing arise from different writing styles, such as informal and colloquial terms, declensional suffixes, various ways of writing for a word, and word space [6]. Another distinction is that the Persian language lacks a rigid syntax, allowing a single sentence to be restated in a variety of ways while still conveying the same meaning. We use sequential learning methods. Persian ghazals are not classified based on era. The goal of our classification of ghazals is to determine the relative chronology of every poem in relation to the author's lifetime.

* Corresponding author.
E-mail address: rashedur.rahman@northsouth.edu (R.M. Rahman).

https://doi.org/10.1016/j.dajour.2022.100111
Received 13 March 2022; Received in revised form 20 July 2022; Accepted 29 July 2022
Available online 3 August 2022

We propose a deep learning-based classification system with the following research objectives.

(1) Applying document embedding technique and Deep Learning (DL) models for classification of Persian ghazals based on chronologically labeled data.
(2) Assessing the effectiveness of the proposed methodology in comparison to multiple traditional Machine Learning (ML) models.
(3) Evaluating the effectiveness of the suggested approach in the context of similar research using a number of evaluation metrics.

Distinct work was published on Persian poet identification [7] and Persian poetry generation [8] based on neural network methods. However, to the best of our knowledge, no other work has been found on Persian ghazal classification based on sequential models. We strongly believe that the findings from this study on Persian ghazal will help in further research initiatives in this field.

2. Literature review

Poetry classification has been done for various languages such as English [9], Hindi, Bengali [10], Persian, Marathi, Punjabi [11], and also other languages. Different techniques have been adopted by the authors to categorize the poetries in terms of several criteria. The most common classification of poetry is subject based [9–11] and emotion-based classification of poetry [12,13]. A closely related concept to our work based on different eras was done by Praveenkumar et al. [14] on Indian English poetry, where they implemented a method to classify Indian English poetry using a Random Forest classifier. Poems in Indian English Poetry are divided into pre-independence and post-independence eras. The poetry style and subjects' shift from one era to the next depends on the poem author's era. As a result, this study puts multiple feature selection methods and ensembled characteristics to automatically determine the poems' era. Semantics, topics, and stylistic characteristics are used to classify poetry. The authors employed Latent Semantic Analysis (LSA) to identify semantic features, Latent Dirichlet Allocation (LDA) topic modeling to find topic features, and phonemics, grammatical components, and poetic structure as style features in the experiment. They used 760 poems in their experiment, where the pre-independence class contains 344 poems and the post-independence class contains 416 poems. The poems are written by 28 different authors. The authors used the combination of LDA and LSA for features and applied these features to classify the poems based on two eras. By using the combined feature set and Random Forest classifier, they achieved the highest 91.20 percent accuracy in poem classification based on era. Ahmed et al. [13] worked on English poems written by Indian poets based on DL and suggested an emotional state classification system for poetry text in the study. The authors applied the attention-based C-BiLSTM model in their study, which helped to classify the poems in terms of emotion. They achieved 88% accuracy in poem classification and classified the poems into 13 classes. Initially, they preprocessed the data by stop words removal, tokenization and also applied Keras embedding layer for feature representation. Finally, they proposed an attention-based C-BiLSTM model, which provides the highest accuracy, precision, recall, and F1-measure value of 88% in all the performance metrics. Their proposed model outperformed the baseline ML models. Mahmood and Qasim [15] recently worked on document classification on English language and applied various forms of Paragraph vector (Doc2Vec) in their studies, such as DBOW, DMC, DMM and a combination of DBOW and DMM. Though they applied an optimized form of ML classifiers along with that, another work by Khan and Chang [16] is based on DNN and LSTM.

Gharbat et al. [17] introduced an Arabic poetry classification method based on four different eras of the poets. Also, Orabi et al. [18] reported a CNN-based deep learning model that classifies Arabic poetry based on the era for the first time. The initial step in building this model was to create a dataset; hence they used an updated Arabic Poetry Dataset. They employed FastText word embedding, which was based on the whole corpus of poetry. Two classifiers, a supervised deep-learning classifier and a FastText-based classifier were trained. Several tests were carried out on their classification approach. Initially, they used a DL model without frequent words to create a polarity classifier of poetry to modern and non-modern eras, achieving the maximum accuracy and F1-score of 91.3 percent and 91.4 percent, respectively. The authors classified poetry into three eras in their experiment, where the accuracy and F1-score of the classifier were both 0.875. Finally, in that work, they have classified the poetry into five distinct eras and earned the best accuracy and F1-score of 80.1 percent and 79.6 percent, respectively.

Emotion classification was introduced in Arabic poetry using ML approach and later, other languages are classified in this aspect by Alshari et al. [12]. Another work on modern Arabic poetry was accomplished on ML, where authors in [19] proposed an approach to classify the poems into four different types such as love, Islamic, social, and political. Three ML methods were used for the categorization of current Arabic poetry in their approach. These methods showed good performance in the categorization of English text. With respect to performance, Support Vector Machines (SVM) was the first, Naive Bayes (NB) was the second, and Linear Support Vector Classification (LSVC) was the third. Their datasets were divided into four divisions such as 23 Islamic poems, 25 Love poems, 22 Political poems, and 22 Social poems. In each class of this Arabic dataset, around 500 to 600 verses were present in every class. In the first step of their study, the authors removed the non-Arabic terms and words, punctuation, numbers, and stop words in the data preprocessing. They used the Boolean vector model to extract the features for ML classifiers, and among three of the classifiers, they used LSVC, which performed much better than the other two classifiers. They achieved the highest 72 percent precision, 47 percent recall and 51 percent F1-measure score when LSVC was applied. But when they used SVC, results were inferior compared to LSVC and NB. The precision score of SVC was 17.75 percent, whereas Naïve Bayes precision score was 64 percent. Abandah et al. [20] proposed a recurrent neural network method to classify Arabic poetry. They have proposed a model which provided around 97.27% accuracy. They suggested a method for classifying the input Arabic text into 16 poetry meters and prose meters. They used a ML technique to create neural networks to categorize Arabic poetry using a big dataset of 1,657k verses of poems and prose. To solve these challenges, the authors used deep and narrow recurrent neural networks with Bidirectional Long Short-Term Memory (Bi-LSTM) cells.

Hamidi et al. [21] worked on meter classification of 138 poetries of the Persian the language and classified them based on meter style with SVM classifier. Another work was done on Persian language based on topic modeling and classification criteria were the genre of the ghazals by Asgari et al. [22]. This genre-based classification, compared with the basic SVM model and topic modeling outperformed in the study. Rahgozer [23] conducted a research on Hafez's ghazals which automatically extracted Hafez's poetry's semantic properties by adopting Houman's classification method. They used the ML methods to classify the ghazals. Houman did the chronological labels by hand around eighty years ago, which did not cover all the Hafez ghazals. To classify all the Hafez ghazals in chronological order, Rahgozer [23] showed the high-level methodology architecture. The main purpose of the automatic chronological classification of Hafez's ghazals was to help us understand them and guide the corrections when necessary. Hafez corpus is complied with Houman's order of ghazals, and the timing annotation is the actual location of the ghazal in the corpus, with discrete labels. This method was the most efficient means to record Houman's classification, and it sets the timing attribute of the poems during the preparation of our Hafez corpus. The author searched for the most appropriate text classification method for the research and applied SVM, which is considered a state-of-the-art classification algorithm. The author of this work used feature engineering based on the layers of

Bag-of-Words (BOW), TF-IDF and LDA. He used the LDA-based cosine similarity features to all poems in the training set to determine their top-performing SVM classifier. Later, he applied different techniques in isolation and then compared them to identify the best LDA-based similarity features for SVM. Later in this work, the author used LSA and LDA-based features to improve performance. He developed a bilingual Hafez poetry corpus of size 249, which are annotated with Hafez classes that are used for training. The author initialized LSI, LDA, Log-Entropy and Doc2Vec models using both Persian and Persian–English corpus as training. They used gensim library and the HAZM Python library for Persian pre-processing tasks such as tokenization, normalization, lemmatization and filtering. They used SVM for classification, trained with LDA-based similarity features. The author reported 79.2 percent highest accuracy when incorporate LDA cosine similarity for Persian training data. Other than that, 78.4 percent accuracy was also detected by using SVM and bilingual data in the study. Davari et al. [24] worked on Persian document classification recently using deep learning method. The authors applied 'Hamshahri' dataset, which contains 166,000 documents with various themes. In the first step of their experiment, they preprocessed the data using normalization, elimination of different language characters and removed the stopwords. In the feature extraction step, they adopted two renowned word embedding models e.g. FastText and Word2Vec. They classified their data using deep neural network models and reported the highest accuracy, 85% when they used 512 neurons with ReLu function on the initial layer. In this study, they achieved the precision, recall and F1-score 96% using Word2Vec word embedding technique. Table 1 represents a summary of works on Persian and Non-Persian Languages.

3. Methodology

3.1. Selection of Persian poet for research

In the medieval time of Iran, Nezami, Rumi and Hafez were the most famous poets for writing allegorical imaginary poetries or ghazals [32]. Among all of them, Hafez became famous all around the world for his survival poems written in his lifetime. Hafez (1315–1390) is a well-known ancient Persian poet who is known for his collection of about 495 ghazals (Divan-e Hafiz) and his full name is Khwāja Shams-ud-Dīn Muḥammad Ḥāfeẓ-e Shīrāzī. Hafez is widely renowned for mastering the ghazal, a short sentimental poetry that mirrors the English sonnet [33,34]. He is also acknowledged as 'Hafiz of Shiraz'. The exceptional appeal of Hafez's poetry in all Persian-speaking regions originates from his unaffected use of homey images and common idioms, as well as his plain and frequently colloquial albeit melodic language, free of false virtuosity [35,36]. Because of the restrictive social properties and freedom of expression impediments of his era, Hafez's poetry is enigmatic and complex on the one hand and elegant on the other because of his enumeration of high-calibre world-views, mystical and philosophical attributes, artistically knitted within stunning designs.

Hafez was born in the Iranian city of Shiraz. During his lifetime, the political condition of Iran was quite unstable. Hafez joined in the court of Shiraz in his early twenties as a young poet by Shah Sheikh Abu Eshaq Inju, who had ruled over Shiraz till 1353 [37]. According to Sha'bānī [38], after that period, Amir Mohammed captured the Shiraz authority in 1353 and headed over there till 1357. Limbert [36] discussed in his book that during Amir Mohammed's ruling period, Hafez was removed from the college where he was teaching Quran. Shah Shoja was the son of Amir Mohammad Mozaffar (Amir Mobarezeddin), who ruled over the Shiraz from 1357 to 1384. Amir was a ruthless leader, and his son Shah Shoja blinded and imprisoned him soon after the ruling period. Shah Shoja reassigned Hafez to the college and also constructed a good relationship with him. Hafez had written most of the poems in the period about subtle spirituality. After the ruling period of Shah Shoja, Hafez was in the sixties in his lifetime and kept focusing on writing poems about God's relationship.

Persian is still spoken as a first language in Iran, Afghanistan, Tajikistan, and other countries in Central Asia. The Persian lyrical poet Hafiz is still the most well-known poet. In the nineteenth century, writers like Goethe in Germany imitated him and were revered by writers like Tennyson in England and revered by writers like Emerson in the United States. He continues to draw admiration and inspire fans everywhere. Hafiz produced various panegyrics for Shiraz's kings and leaders, poetry about nature, poems with ethical and moral themes, and poems demonstrating his great societal awareness. In his investigation of love as the core value of human existence, he most likely took inspiration from the other medieval poets, such as Sanai, Attar, Rumi, and Nezami, who stood out as particularly influential. Hafez was hailed as the standard-bearer for Persian lyrical perfection because of his brilliant use of rhetorical methods, poetic devices, and imagery. Ebrahami et al. [39] compared Hafez's poetry with England's National poet William Shakespeare for their aesthetic views on ghazals (sonnet). The ghazal's visual similarities and shared topics show their spirituality and beauty found in that study. We used Hafez's poetry in our study because students of Islamic philosophy, Sufism, and Middle Eastern studies would be beneficial to a larger readership interested in comparative poetics, Eastern literature and spirituality, medieval romance and philosophy of love.

3.2. Dataset

We have used the Persian poet Hafez's ghazals in our study. The Hafez ghazals are available on the ganjoor website [40]. Hafez scholar Houman has labeled 249 out of 495 Persian ghazals manually in chronological order [41]. Houman labeled the poems according to Hafez's lifetime and classified them into six individual classes. Fig. 1(a) is the representation of the Houman label. The translations of the ghazals are made by Shahriar Shahriari and collected from [42]. The Hafez scholar Raad's labels have been collected from the Rahgozer's work [23]. The dataset contains 496 Persian poems, and 249 of them are chronologically classified by Houman. Contemporary Hafez scholar Raad has tried to label the poems according to two politicians' names e.g. Amir Mohammed and his son Shah Shoja. Dr. Raad has also reduced the number of classes from six to four. He has also removed the irrelevant data from the whole dataset. After cleaning the dataset, 233 ghazals have been classified manually as 'before Amir Mobarezeddin', 'Amir Mobarezeddin', 'Shah Shoja' and 'after Shah Shoja'.

Before Amir Mobarezeddin's ruling period, Hafez wrote ghazals related to youth; during Amir Mobarezeddin's period, most of the ghazals were written in the context to protest because the ruler removed him from the college where Hafez taught Islamic topics. Shah Shoja took over the ruling position after his father and reassigned Hafez to his job. In that period of Hafez's life, he realized subtle spirituality and wrote ghazals related to these topics. Later in his life, after Shah Shoja's ruling period, Hafez realized cosmic consciousness, and he started writing sad poems corresponding to God-realization. Dr. Raad classified the Hafez poems according to these topics and the corresponding ruling time of the politicians of Shiraz.

Very few of the poems are translated into English by Shahriar, which is collected from the hafizonlove.com website. We have found 76 translations among 249 ghazals of Hafez. Fig. 1 illustrates the data distribution of Hafez and Raad labels.

The following Fig. 2 is the sample dataset of Hafez ghazal [42] (Ghazal No. 1 and 183).

We have used only Persian and English data, but we could not apply more than two languages in our work for the unavailability of the translations of the Hafez ghazals.

The wordcloud of Fig. 3 shows the highest frequency of the words of ghazals. We can see that the word 'دل' and 'غزل' are the highest occurred words in the data. Data visualization is important because if any unnecessary word is present in the dataset with a higher frequency, it will help remove those words.

Table 1
Tabular representation of related works of poetry classification with Non-Persian and Persian languages.

Study	Study purpose	Dataset	Methods/Techniques	Results
Lou et al. [9]	English (Subject-based Classification)	7214 English poetry collected from poetryfoundation.org	SVM with TF-IDF and LDA	84.8% accuracy
Ahmed et al. [13]	English (Emotion Classification)	9142 poetic posts consisting of 13 emotion classes	C-BiLSTM, LSTM, BiLSTM	88% accuracy
Praveenkumar et al. [14]	Indian English Poetry (Classification based on two eras)	760 poems written by 28 authors	RF classifier with reduced feature set	91.20% accuracy
Kumar et al. [25]	English and Indian poetry	13,747 poetries with 3 classes collected from Poemhunter.com	Bi-LSTM, SGD, Conv1D, MNB models	87% accuracy with Conv1D
Rakshit et al. [10]	Bengali (Classification based on four categories)	1341 poem with 4 classes collected from tagoreweb.in	SVM with shallow parser	56.8% accuracy
Orabi et al. [18]	Arabic (Classified based on five distinct era)	Scrapped Arabic poems from adab.com	CNN based	80.1% accuracy, 79.6% F1-score
Promrit and Waijanya [26]	Thai Poetry (based on category)	500 poems for training and 55 for testing	CNN model with Word2Vec embedding compared with SVM and Naïve Bayes	83% accuracy
Pal and Patel [27]	Hindi Poetry (3 category	154 poetries including Shringar, Karuna and Veera class	SVM, NB, RF, CART KNN	56.66% performance with SVM
Jamal et al. [28]	Malay Poetry	1500 pantum with 10 themes	SVM with RBF and linear kernel using cross validation	58.44% accuracy SVM with RBF kernel
Kaur [11]	Punjabi Poetry (Content based classification)	2034 poetries	SVM and Textual feature	76.02% accuracy
Kaur and Saini [29]	Punjabi Poetry	250 poetries with 4 categories	10 machine learning algorithms using Weka (Hyperpipes, K- nearest neighbor, Naive Bayes, SVM	50.63%, 52.92%, 52.75% and 58.79% accuracy with Hyperpipes, K- nearest neighbor, Naive Bayes and SVM respectively
Saini and Kaur [30]	Punjabi Poetry (Emotion detection based on 'Navrasa')	948 poetries with 9 classes	SVM classifier	70.02% accuracy
Kaur et al. [31]	Punjabi poetry (Thematic Classifier)	2000 poetries with 8 theme based categories	14 ML algorithms divided by baseline learners, ensemble learners, deep learning. Super hybrid textual features	76.14% accuracy SVM (baseline learner) 64.10% accuracy (ensemble learner) 80.32% accuracy with Bi-LSTM
Hamidi et al. [21]	Persian Poetry (Meter Classification)	138 poetry with 12 persian meter styles	10-fold cross validation with SVM Classifier Meta-Parameter Optimization	91% accuracy with SVM
Asgari et al. [22]	Persian poetry (Genre Classification)	7,326 ghazals collected from ganjoor.net	10-fold cross validation with topic modeling and SVM	55% accuracy on ghazals
Rahgozer [23]	Persian (Poetry classification)	249 ghazals with six classes	SVM with BOW, TF-IDF and LDA similarity	79.11% accuracy
Current Study	Persian Ghazal Classification (Based on Poet's era)	233 ghazals with bilingual four classes	LSTM, GRU and Bi-LSTM with document embedding technique	85% accuracy, 85% F1-score, 87% precision with LSTM and Distributed Memory Mean

3.3. Research design

The following flowchart in Fig. 4 shows the steps we have followed in classifying the ghazals of Hafez. The key steps of the methodology include data pre-processing, feature extraction, and model selection. We have used two types of class labels for our experiment and also used only Persian data and bilingual data. In the pre-processing step, we filtered the stopwords, normalized, stemmed and tokenized the text data. We have also filtered the rare words from the whole dataset. After cleaning the data, we have adopted different types of feature extraction methods for our study.

Initially, we extracted features from the cleaned text using BOW and LDA with cosine similarity. We have applied both feature extraction methods and used these features in ML classifiers. We have experimented with two machine learning classifiers which are Random Forest and Logistic Regression. We have also implemented Distributed Bag-of-Words (DBOW), Distributed Memory (DM) and concatenation of DBOW and DM for document embedding purposes in the deep learning architecture. When we have applied the DL method, we have used LSTM, GRU and BiLSTM models to classify the Hafiz ghazals. We have analyzed both ML and DL models to classify the ghazals. Fig. 4 illustrates the overall methodology of our study.

3.4. Data preprocessing

We applied data pre-processing and cleaning techniques to create understandable raw data for the classification. This step is mandatory to deal with raw text data.

In the first step of pre-processing, we removed the bad characters, stop words from the Persian ghazals. We also filtered the low-frequency words from the whole dataset because if any word is present only once in the whole dataset, it can be present in either train or test data and it will hamper the overall accuracy.

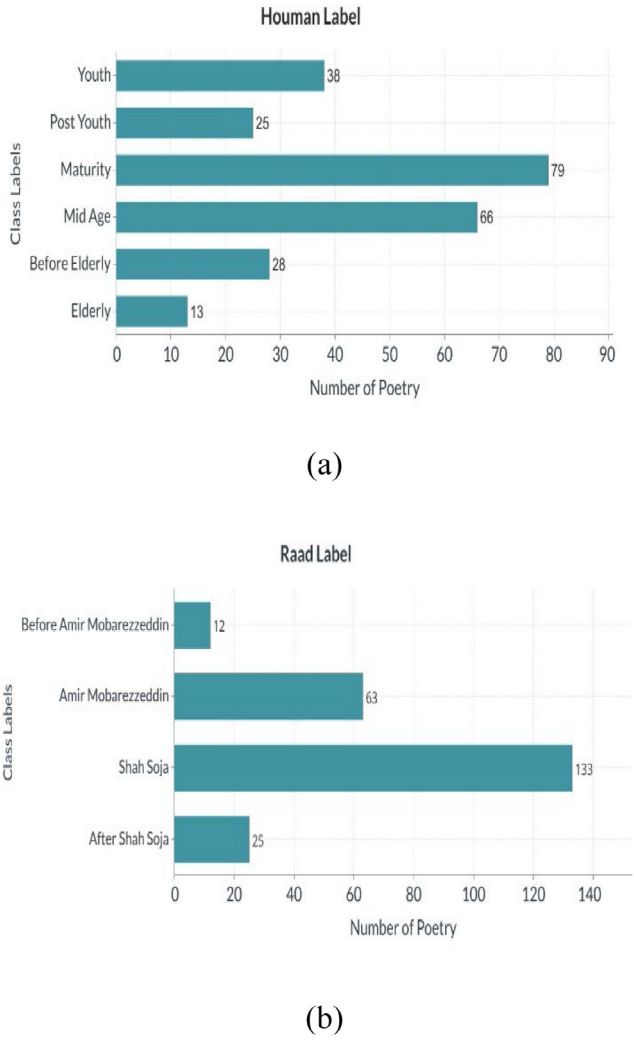


Fig. 1. (a) Class labels distribution by Houman and (b) Class labels distribution by Raad of Hafez Poems.

We also normalized the data in the next step of pre-processing, which helps to reduce the variations of text data. This normalization includes the removal of Unicode punctuations, capitalization of characters and white spaces. Persian text includes space delimiter, separated by half-space, or be attached to the next tokens. Any changes of these spaces can make the sentence incorrect. The incorrect form of the spaces removed and thus normalized the incorrect data. When we worked with English data, we removed capitalization from the data. Persian data does not contain any lowercase or upper case variation in the character. It also helps to get rid of a large amount of false positives.

Data stemming is also applied in our data which mainly removed prefixes and postfix from the text data. Another NLP technique we used is data tokenization which splits the corpus into individual words. This step is one of the major steps in NLP.

We used Hazm [43] python library for pre-processing the Persian data and NLTK [44] for English translations.

3.5. Feature extraction

3.5.1. Bag-of-words (BOW)

BOW is the baseline statistical model for textual feature extraction [45]. In the BOW model, features have been extracted using the frequency of the words in the data. This will create a matrix of individual words and the number of data. In our case, the number of words we have found after filtering is 3412 distributed over 249 rows.

Table 2
DBOW and DM parameter.

Parameter	Explanation
dm	0 means select DBOW from Doc2Vec model, and one means distributed memory
vector_size	The number of dimensions of features in the document
negative	Negative sampling of the document applied when value is > 0.
min_count	min_count is the minimum number of occurrences of any word in the document
alpha	The starting learning rate when initialization of epoch
min_alpha	Learning rate drops down over the epochs and min_alpha set because learning rate cannot drop than the value
epochs	The number of iterations of learning the data

3.5.2. Latent Dirichlet analysis (LDA)

LDA is a generative statistical model that enables unobserved groups to understand why certain sections of the data are related. LDA portrays texts as a mix of subjects depending on the probability of a word representing a variety of topics [46].

3.5.3. Distributed bag-of-words (DBOW)

DBOW model is a method that predicts words randomly picked from the document in the output, avoiding the context words in the input, which is the corresponding id of the document [47]. DBOW corresponds to Skip-Gram in Word2vec. In order to build its vector, DBOW randomly guesses a probability distribution of words in a document using the document's identifier. It does not take into account the sequence of the words. The document vector and word weights are randomly initialized and modified throughout training using stochastic gradient descent. Fig. 5 shows the representation of the DBOW model. We have used a part of Hafez ghazal for the demonstration.

We have used the following parameter values when we have applied DBOW for feature extraction *dm = 0, vector_size = 500, negative = 5, min_count = 2, alpha = 0.065, min_alpha = 0.05*. We have also trained this model using 50 epochs.

3.5.4. Distributed memory (DM)

DM correlates to CBOW of Word2Vec implemented by Mikolov et al. [48]. The document id can be compared to a word. It functions like a recollection, recalling what is absent from the current context—or the document's topic [47]. As a result, every paragraph has its own unique vector, represented by a column in matrix D, whereas every word in its own unique vector is represented by a column in matrix W. It anticipates a term based on the context of the content. DM takes a random set of words from a paragraph and a document identification number as input and attempts to predict a center word from the corresponding document.

Fig. 6 is the representation of DM model. We have used the following parameter values in gensim library when we have applied DM for feature extraction: *dm = 1, vector_size = 500, negative = 5, min_count = 2, alpha = 0.065, min_alpha = 0.05*. We have also trained this model until 50 epochs (see Fig. 6).

To illustrate the examples of DBOW and DM, we have used a short sentence from a Hafez ghazal. We have used "به لب یار دلنواز کنید" from Ghazal 244 for simplicity.

In Table 2, we have described the parameter necessity of Doc2Vec, which is implemented in Gensim[1] Python library, and the effects of changing the parameter values briefly.

[1] https://radimrehurek.com/gensim/models/doc2vec.html

Persian Poem	English Translation	Houman Label	Raad Label
دوش وقت سحر از غصه نجاتم دادند واندر آن ظلمت شب آب حیاتم دادند بیخود از شعشعه پرتو ذاتم کردند باده از جام تجلی صفاتم دادند چه مبارک سحری بود و چه فرخنده شبی آن شب قدر که این تازه براتم دادند بعد از این روی من و آینه وصف جمال که در آن جا خبر از جلوه ذاتم دادند من اگر کامروا گشتم و خوشدل چه عجب مستحق بودم و این‌ها به زکاتم دادند هاتف آن روز به من مژده این دولت داد که بدان جور و جفا صبر و ثباتم دادند این همه شهد و شکر کز سخنم می‌ریزد اجر صبریست کز آن شاخ نباتم دادند همت حافظ و انفاس سحرخیزان بود که ز بند غم ایام نجاتم دادند	"At the break of dawn from sorrows I was saved In the dark night of the Soul, drank the elixir I craved. Ecstatic, my soul was radiant, bright, Sanctified cup of my life, drunk I behaved. O, what exalted sunrise, what glorious night That holy night, to the New Life was enslaved. From now on, in the mirror, O what a sight The mirror, glory of my soul, proclaimed and raved. Wonder not if I am bathed in heart's delight I deserved and was given, though may have seemed depraved. Angelic voice brought news of my God-given right My patience is the fruit of hardships that I braved. Sweet nectar drips from my lips, as my words take their flight Beloved, my sweetheart, upon my soul patiently had engraved. 'T was Hafiz, divinely inspired that I attained such height It was God's mercy that time's sorrows for me waived."	Before Elderly	Shah Shoja
الا یا ایها الساقی ادر کأسا و ناولها که عشق آسان نمود اول ولی افتاد مشکل‌ها به بوی نافه‌ای کاخر صبا زان طره بگشاید ز تاب جعد مشکینش چه خون افتاد در دل‌ها مرا در منزل جانان چه امن عیش چون هر دم جرس فریاد می‌دارد که بربندید محمل‌ها به می سجاده رنگین کن گرت پیر مغان گوید که سالک بی‌خبر نبود ز راه و رسم منزل‌ها شب تاریک و بیم موج و گردابی چنین هایل کجا دانند حال ما سبکباران ساحل‌ها همه کارم ز خود کامی به بدنامی کشید آخر نهان کی ماند آن رازی کز او سازند محفل‌ها حضوری گر همی‌خواهی از او غایب مشو حافظ مبی ما تلق من تهوی دع الدنیا و اهملها	"O beautiful wine-bearer, bring forth the cup and put it to my lips Path of love seemed easy at first, what came was many hardships. With its perfume, the morning breeze unlocks those beautiful locks The curl of those dark ringlets, many hearts to shreds strips. In the house of my Beloved, how can I enjoy the feast Since the church bells call the call that for pilgrimage equips. With wine color your robe, one of the old Magi's best tips Trust in this traveler's tips, who knows of many paths and trips. The dark midnight, fearful waves, and the tempestuous whirlpool How can he know of our state, while ports house his unladen ships. I followed my own path of love, and now I am in bad repute How can a secret remain veiled, if from every tongue it drips? If His presence you seek, Hafiz, then why yourself eclipse? Stick to the One you know, let go of imaginary trips."	Elderly	Shah Shoja

Fig. 2. Example of Hafez ghazal Dataset.

Fig. 3. Wordcloud visualization of Hafez ghazals.

3.5.5. Concatenation of DBOW and DM

This method of feature extraction is the summation of DBOW and DM features, which we have already discussed in 3.5.1 Bag-of-Words (BOW) and 3.5.2 Latent Dirichlet Analysis (LDA) In this step, we have concatenated the output vectors of DBOW and DM and use this concatenation as an embedding feature. According to Le and Mikolov [47], conjunction with DBOW and DM is typically more consistent throughout a wide range of tasks.

Rhanoui et al. [49] have investigated Doc2vec embedding for document categorization since Doc2vec has consistently outperformed Word2vec on various datasets. They have used these two strategies to obtain excellent precision. It is a more recent approach than Word2vec, and it suits well for document processing. Fig. 7 is the representation of the concatenation technique of DBOW and DM, which we have discussed in earlier sections.

When we have concatenated the DBOW and DM the parameter values, we have kept the same as before. After concatenation, the vector_size value has become 1000.

3.6. Machine learning models

As we have a very small imbalanced dataset, we have used stratified k-fold cross validation when we have used machine learning classifiers. The class ratio in the original dataset is preserved across the *k* folds using this stratified *k*-fold cross validation procedure in the machine learning techniques.

3.6.1. Random forest

A random forest classifier is a ML classifier which is constructed on numerous decision trees, which decreases the danger of overfitting the model. We have initialized the parameters *max_depth = 80, bootstrap = true, max_features = 10, n_estimators = 1000, criterion = 'entropy', random_state = 1, min_smaples_split = 8.*

3.6.2. Logistic regression

We have also applied logistic regression to classify the text data. We have tuned the parameters to apply this classifier. We have used the following parameter values in our experiment: *C = 10, multi_class = 'multinomial', solver = 'lbfgs', max-iter = 1000*

3.7. Deep learning (DL) models

When we experiment with neural network models, we have incorporated the sample_weight for the classes to handle the imbalanced dataset. This method adds more weight to the classes which contain a low amount of data. In the following section, we have discussed the applied DL models.

3.7.1. Long short-term memory (LSTM)

LSTM is a recurrent neural network model which contains a hidden layer introduced by Hochreiter and Schmidhuber [50]. The LSTM

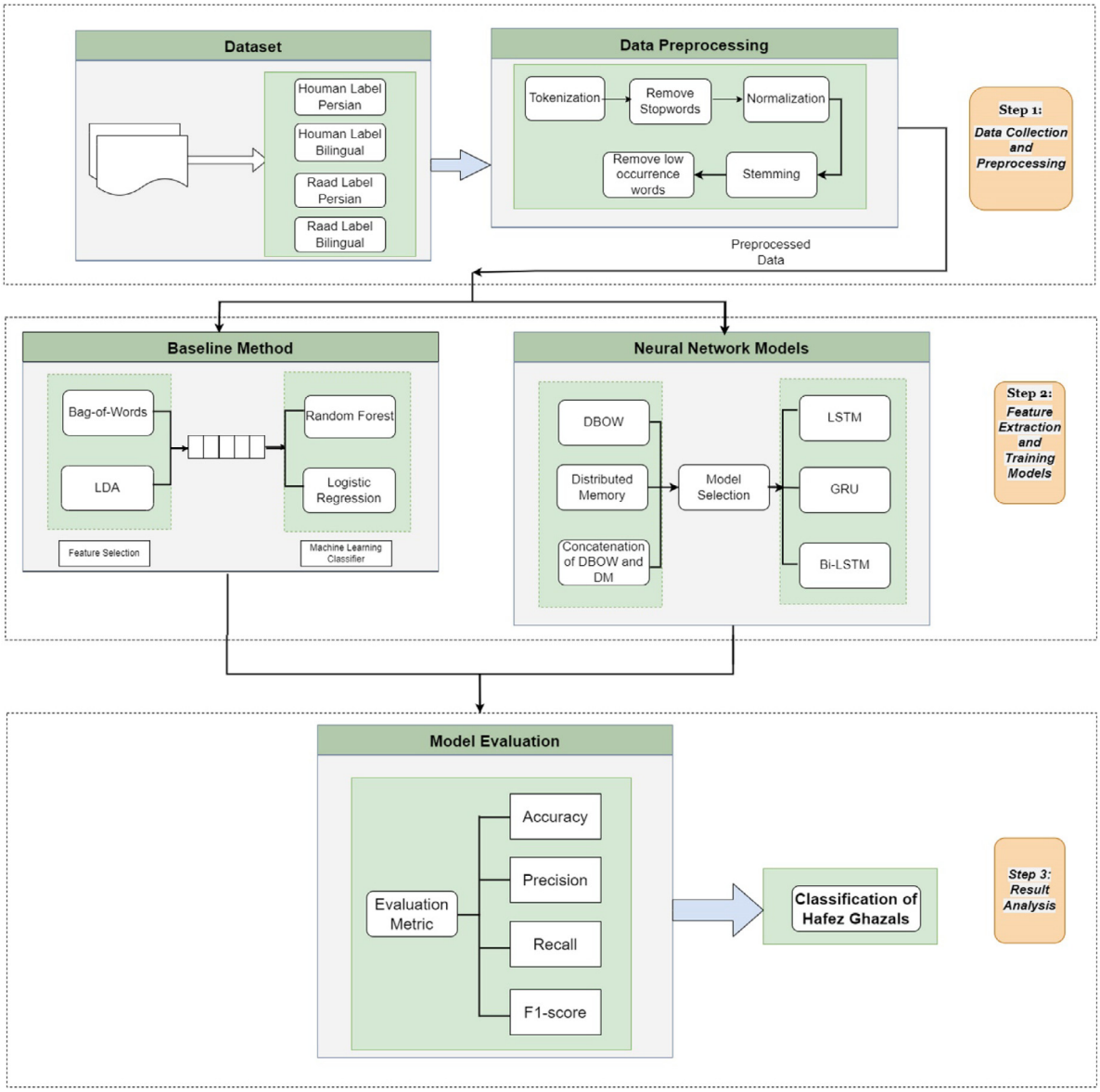


Fig. 4. The analysis procedure of the classification of poetry using various feature extraction methods, models.

design is built around a memory unit and includes a forget gate that prioritizes the state of memory cells. Fig. 8 shows an LSTM cell.

To avoid overfitting, we employed two dropout layers. We have initially used a sequential layer and then added the embedding layer over this model. We have also added two dense layers with '*relu*' activation functions. The final layer is also a dense layer with *softmax* activation function (see Fig. 9).

3.7.2. Gated recurrent unit (GRU)

Gated neural network is also used in text classification problems research by Chung et al. [51] and poet identification problems by Salami et al. [7]. This neural network model used GRU unit in the model. We have initialized 128 as the GRU unit value, one dropout layer with a value of 0.6, two dense layers with activation function '*relu*'.

3.7.3. Bidirectional long short-term memory (Bi-LSTM)

Bidirectional LSTM is proposed by Schuster & Paliwal [52], which runs the input in two directions. Bi-LSTM differentiates from unidirectional is that it can preserve information both from the past and future at any point. In a recent study of poetry classification, Kaur et al. [31] found Bi-LSTM model gives the highest performance in their work. We have implemented Bi-LSTM model in our study with two Bidirectional layers and one dropout layer with a value of 0.8.

We used 80 percent of the whole data for training and the remaining 20 percent data for validation of the model. We applied *'adam'* optimizer and *categorical crossentropy* as loss functions for every deep learning model we have used. We have also used the *learning rate = 0.0001*.

3.8. Evaluation metrics

We have used accuracy, F1-score, precision and recall to evaluate the Hafez ghazals and analyze the performance. Precision, Recall, F1-score, and Accuracy are the most often used measures for poetry classification in the NLP field. These metrics provide us the ability to assess a classifier's performance from several angles.

TruePositive: Any ghazal that has successfully categorized is referred to as TruePositive.

TrueNegative: If a ghazal is accurately classified, excepting one class, the other class will get a result of 0. For instance, the remaining class

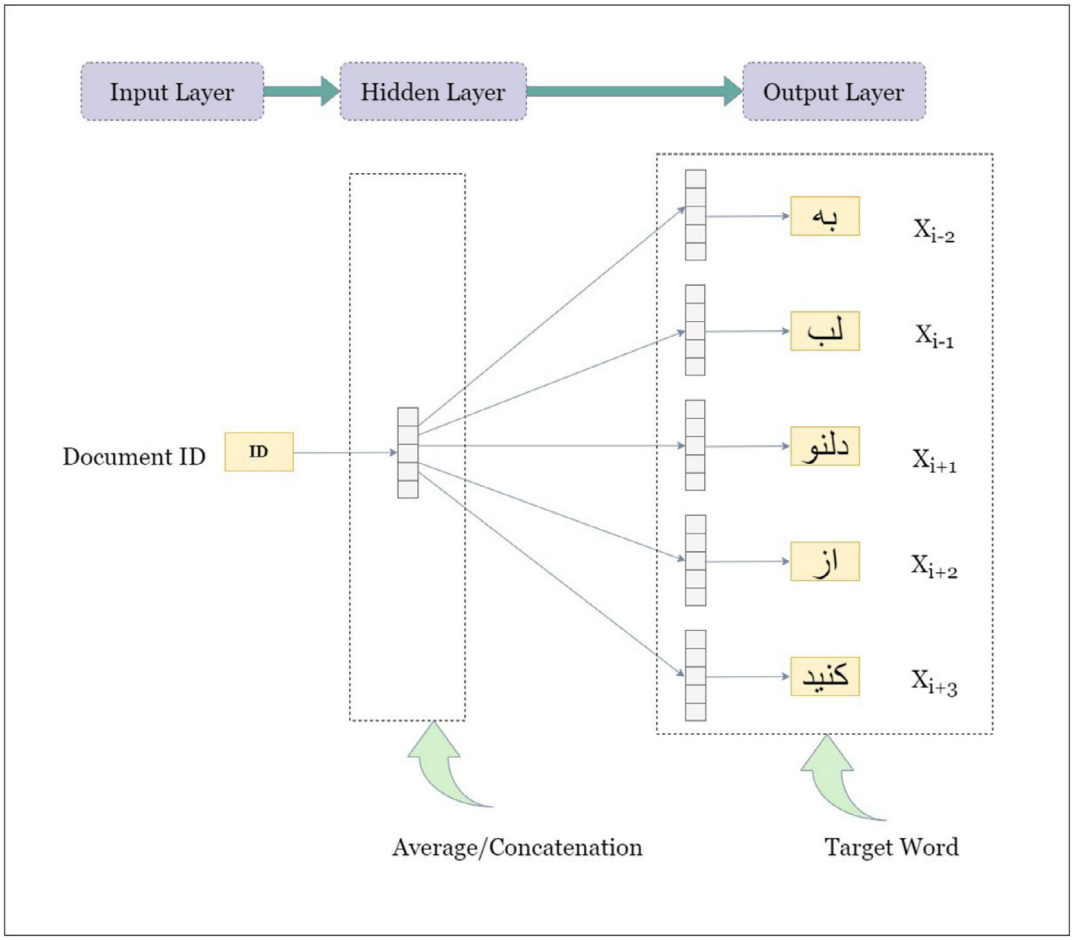


Fig. 5. Distributed Bag-of-Words model using Persian data.

values will be 0 if a ghazal is actually in the "Shah Shoja" class and was accurately categorized in all other classes as well.

FalsePositive: If any ghazal is wrongly assigned to a class, the value will calculate as 1.

FalseNegative: The value will be defined as 0 for the real class of that particular data when any ghazal is mistakenly categorized.

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative} \quad (1)$$

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \quad (2)$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \quad (3)$$

$$F1 - score = \frac{2 * recall * Precision}{Recall + Precision} \quad (4)$$

Accuracy: The fraction of accurately categorized positive and negative examples is known as accuracy.

Precision: The number of positive class predictions of any class that really falls within the positive class is measured by precision.

Recall: Recall measures how many correct class predictions were produced using all of the successful cases in the dataset.

F1-score: F1-score is widely used to compare more than one algorithm performance. According to Jeni et al. [53], this metric can provide a more specific result on an imbalanced dataset by adding weights using precision and recall. Compared to accuracy, F1-score offers more information about how well a classifier performs while being sensitive to the data distributions.

4. Results and analysis

We have applied two types of class labels and both Persian and bilingual data for our experiment. The following section illustrates the experimental results found from our models.

4.1. Performance analysis of machine learning classifiers on Houman labeled data

In the first step of our experiment, we have used BOW for feature extraction initially and analyzed the ML classifiers' performance first. We applied k values as 5 and 10 to find out the best performance results of k-fold cross validation. We have reported comparatively lower values in stratified 5-fold cross validation experiments than 10-fold methods. We used both Persian and bilingual data in this case. Rahgozer used SVM in their study and found 37.34% accuracy when they used only BOW in the Persian data. We have applied two other machine learning classifiers, e.g., Random Forest and Logistic Regression, to find a better performance in the case of ML. Later, we have applied LDA with cosine similarity for feature extraction and we have found the highest F1-score, 65.9%, in the logistic regression classifier, whereas, in Rahgozer, F1-score was about 57.1%. But we cannot outperform with ML classifier in terms of accuracy. Table 3 reports this.

4.2. Performance analysis of deep learning models on Houman labeled data

Later in our work, we applied deep learning models to improve performance. We experimented with three different deep learning models, which are LSTM, GRU, and Bi-LSTM. From Table 4, we can observe

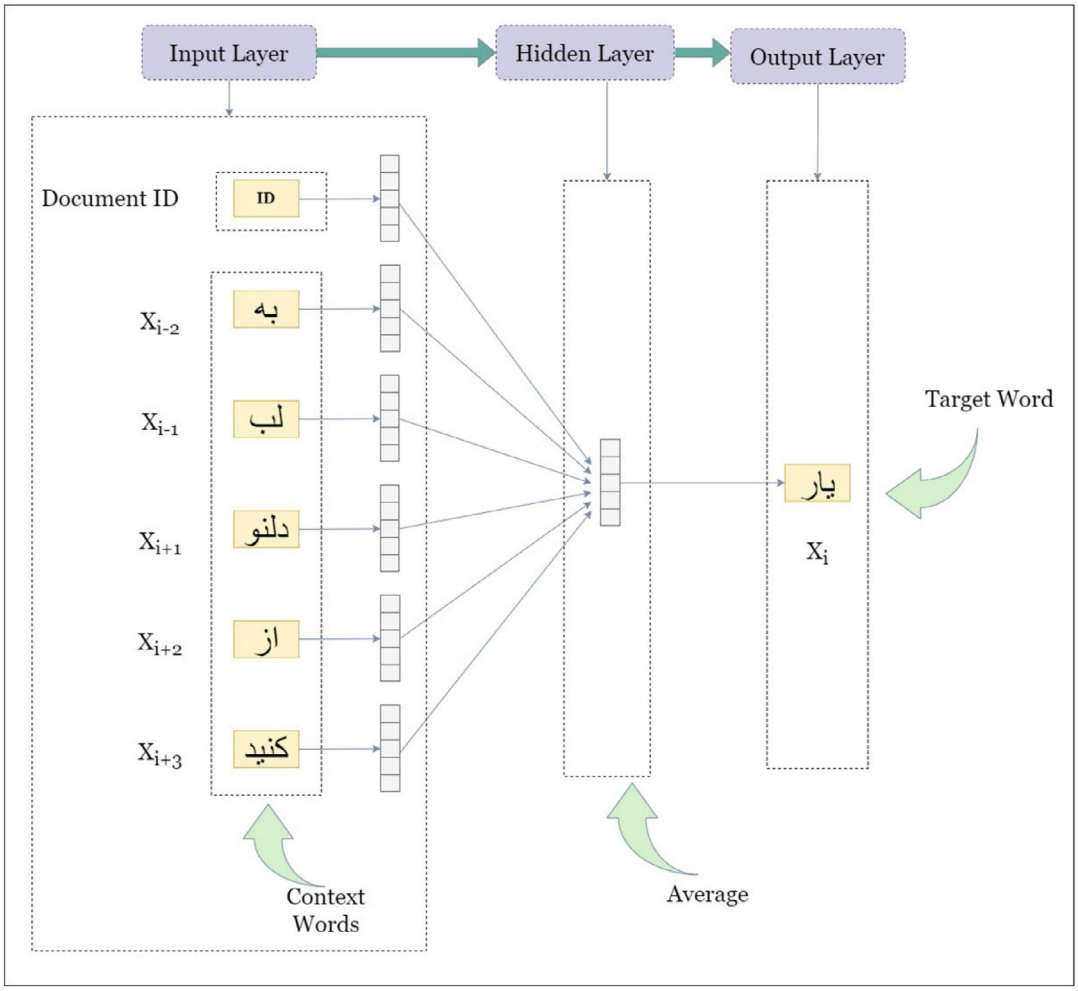


Fig. 6. Distributed Memory Model using Persian data.

Table 3
ML Classifiers performance using BOW and LDA similarities on Houman labeled data.

k-value	ML classifiers	Features	Accuracy (%)	F1-Score (%)
5	SVM	Persian BOW (6 class)	34.77	23.9
		Bilingual BOW (6 class)	28.52	22.5
		Persian LDA Similarity (6 class)	77.94	62.78
	Random Forest	Persian BOW (6 class)	31.44	17.2
		Bilingual BOW (6 class)	31.73	20.6
		Persian LDA Similarity (6 class)	70.55	56.7
	Logistic Regression	Persian BOW (6 class)	32.69	25.9
		Bilingual BOW (6 class)	30.23	22.76
		Persian LDA Similarity (6 class)	72.46	64.39
10	SVM (Rahgozer [23] using WEKA tool)	Persian BOW (6 class)	37.34	24.1
		Bilingual BOW (6 class)	39.75	23.8
		Persian LDA Similarity (6 class)	79.11	57.1
	Random Forest	Persian BOW (6 class)	32.2	17.9
		Bilingual BOW (6 class)	35.0	19.8
		Persian LDA Similarity (6 class)	71.6	53.0
	Logistic Regression	Persian BOW (6 class)	33.0	26.2
		Bilingual BOW (6 class)	30.5	24.55
		Persian LDA Similarity (6 class)	73.1	65.9

that when we use DBOW for feature extraction, Bi-LSTM provides the highest 71.5% in terms of accuracy and LSTM gives 69% test accuracy for only Persian data. But if we consider precision value, then LSTM is showing 71% value for Persian DBOW. When we applied DM in the Persian data, the model accuracy was noticeably improved. LSTM gives a 76.6% F1-score, which is the highest among all the experiments we conducted using six classes. GRU model performs comparatively less than the LSTM and Bi-LSTM model, which gives 71.6% accuracy and 70% precision and F1-score value. But LSTM shows 77% recall value and 77% precision value. We also used the concatenation of DBOW and DM in our experiments. This feature extraction method works quite well in bilingual data compared to only Persian data. It gives around 75.8% accuracy in the BiLSTM model and a 75.7% recall score which is the highest among bilingual data. But in this feature extraction

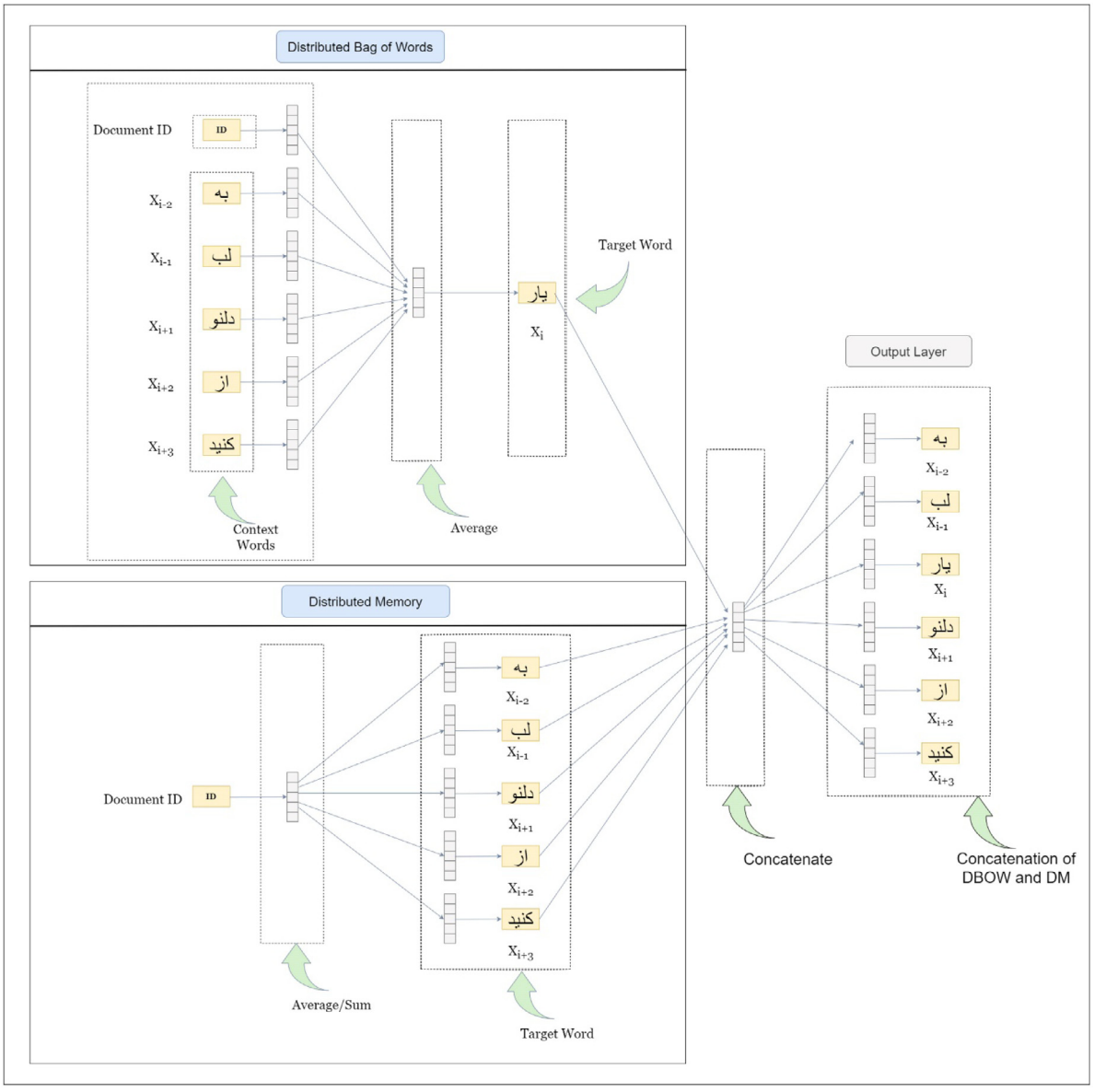


Fig. 7. Concatenation of DBOW and DM model using Persian data.

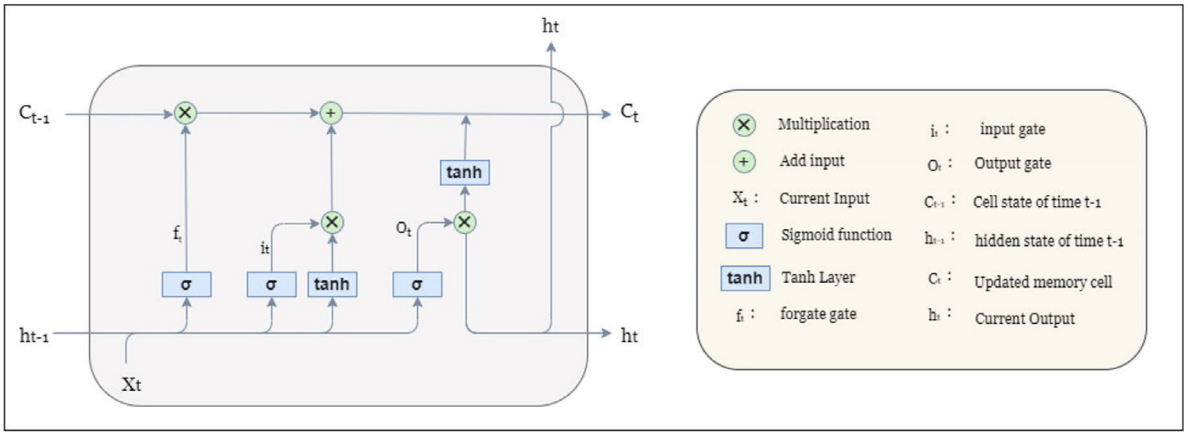


Fig. 8. Baseline Long Short Term Memory Architecture (Hochreiter and Schmidhuber [50]).

method, LSTM could not work that well. The accuracy value has been dropped here, which is around 7% lower than the previous method. From Table 4, accuracy is highest when we used distributed memory in the Persian data for feature extraction and the LSTM model for training

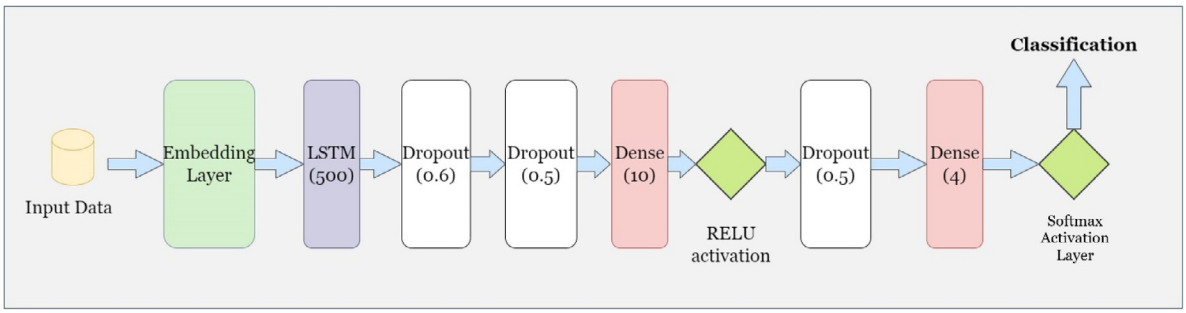


Fig. 9. Applied Long Short Term Memory Model for Classification ghazals.

Table 4
Deep learning models on Houman labels.

Language	Approach	Accuracy	F1-score	Precision	Recall
Persian	LSTM + DBOW	0.69	0.67	0.71	0.69
	GRU + DBOW	0.68	0.68	0.70	0.6
	Bi-LSTM + DBOW	0.715	0.692	0.701	0.715
	LSTM + DM	0.778	0.766	0.75	0.77
	GRU + DM	0.716	0.70	0.70	0.71
	Bi-LSTM + DM	0.737	0.727	0.732	0.736
	LSTM + DBOW +DM	0.736	0.711	0.737	0.716
	GRU + DBOW + DM	0.736	0.746	**0.764**	0.736
	Bi-LSTM +DBOW +DM	0.684	0.65	0.67	0.684
Bilingual	LSTM + DBOW	0.736	0.731	0.735	0.731
	GRU + DBOW	0.671	0.672	0.699	0.671
	Bi-LSTM + DBOW	0.747	0.731	0.756	0.747
	LSTM + DM	0.715	0.692	0.688	0.715
	GRU + DM	0.652	0.656	0.697	0.656
	Bi-LSTM + DM	0.715	0.711	0.729	0.716
	LSTM + DBOW +DM	0.705	0.669	0.662	0.694
	GRU + DBOW + DM	0.684	0.665	0.669	0.684
	Bi-LSTM +DBOW + DM	**0.758**	0.732	0.741	0.757

Table 5
Machine learning classifiers performance using BOW and LDA similarities on Raad labels.

k-value	ML Classifiers	Features	Accuracy (%)	F1-Score (%)
5	SVM	Persian BOW	49.11	49.0
		Bilingual BOW	52.49	51.7
		Persian LDA Similarity	80.43	75.91
	Random Forest	Persian BOW	57.09	48.6
		Bilingual BOW	57.18	49.8
		Persian LDA Similarity	78.67	69.4
	Logistic Regression	Persian BOW	49.38	42.78
		Bilingual BOW	49.1	42.49
		Persian LDA Similarity	79.43	67.7
10	SVM	Persian BOW	52.87	49.6
		Bilingual BOW	53.12	52.36
		Persian LDA Similarity	81.02	82.13
	Random Forest	Persian BOW	57.11	49.2
		Bilingual BOW	57.87	50.65
		Persian LDA Similarity	79.24	70.1
	Logistic Regression	Persian BOW	51.14	46.7
		Bilingual BOW	51.80	46.98
		Persian LDA Similarity	80.02	70.73

our dataset in the DL models. If we have considered the precision score, Bi-LSTM gives the highest precision score on bilingual data (see Fig. 10).

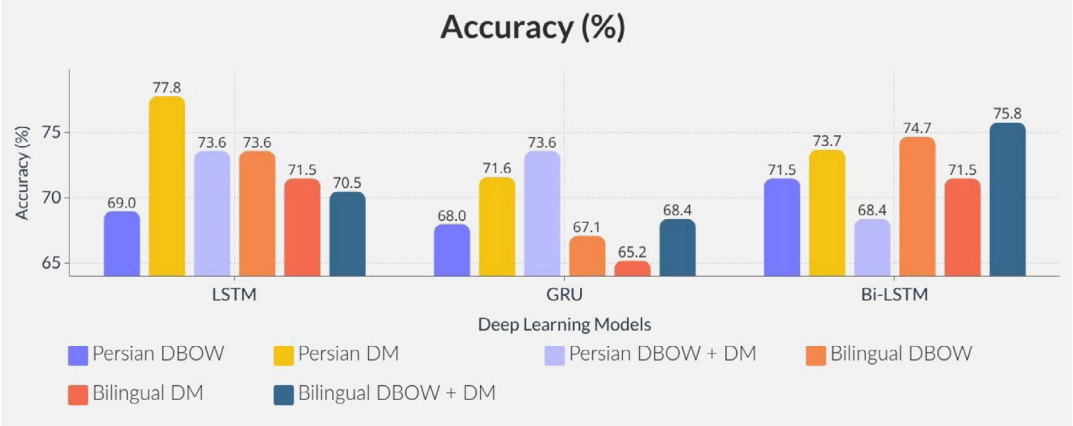
4.3. Performance analysis of machine learning classifiers on Raad labeled data

We applied ML classifiers to the refined labeled data with different k values in cross-validation methods. From Table 5, we can observe that the highest accuracy is achieved by using SVM algorithm with LDA similarity feature, which is 81.02%, whereas the F1 value is 82.13% in this case.

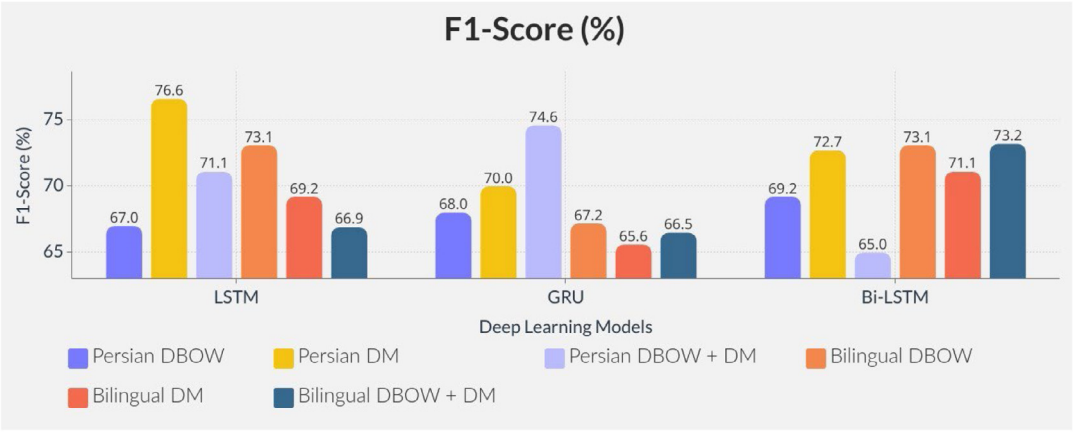
Here, we can see that, for the Persian BOW and Bilingual BOW feature extraction method, the Random Forest classifier gives the highest accuracy value compared to other ML classifiers. But for Bilingual BOW and Persian BOW, SVM results are 53.87% and 52.87% accuracy, respectively. Logistic Regression classifier with LDA similarity provides 80.02% accuracy, which is higher compared to Random Forest classifier we applied.

4.4. Performance analysis of deep learning models on Raad labeled data

We have applied refined data in our models to get better accuracy. When we have applied DBOW in the word embedding of deep learning



(a)



(b)

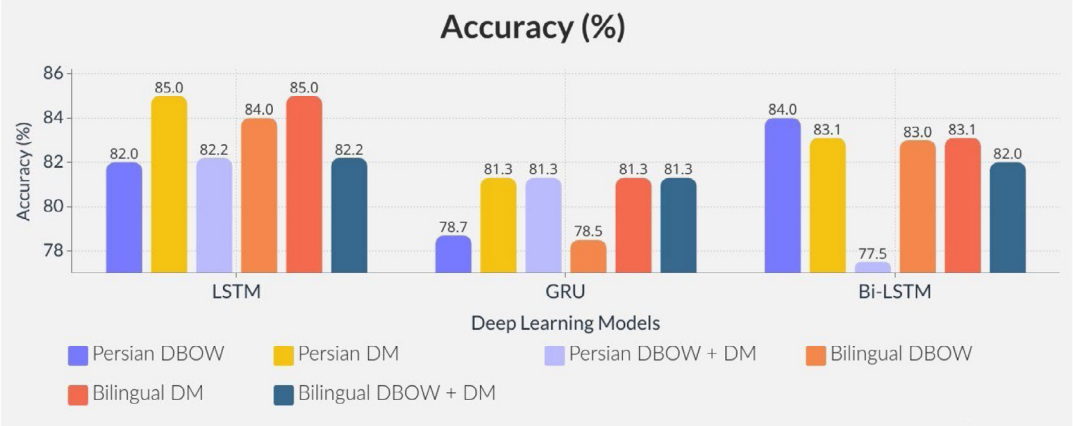
Fig. 10. Performance analysis of Deep learning models using Houman Labels (a) accuracy, and (b) F1-score.

Table 6
Deep learning models using refined labels.

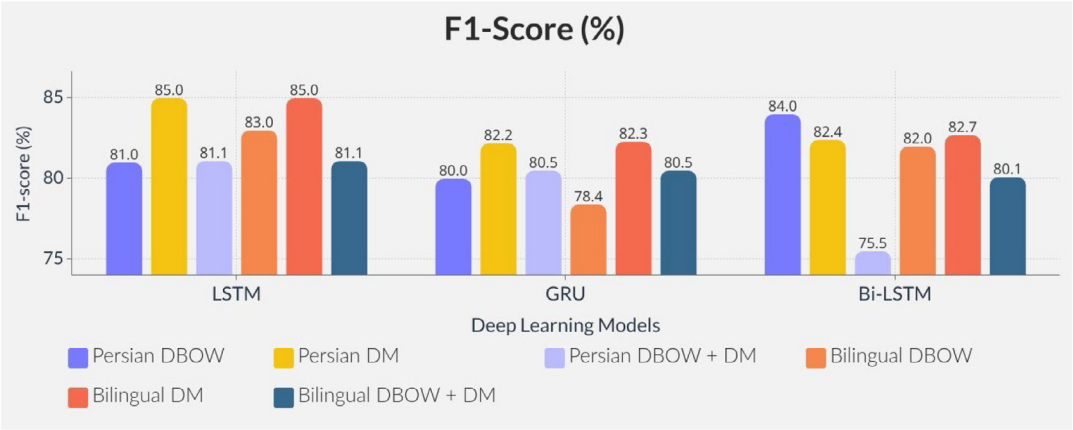
Language	Approach	Accuracy	F1-score	Precision	Recall
Persian	LSTM + DBOW	0.82	0.81	0.82	0.82
	GRU + DBOW	0.787	0.80	0.818	0.787
	Bi-LSTM + DBOW	0.84	0.84	0.84	0.85
	LSTM + DM	**0.850**	**0.85**	**0.87**	0.85
	GRU + DM	0.813	0.822	0.852	0.813
	Bi-LSTM + DM	0.831	0.824	0.85	0.83
	LSTM + DBOW +DM	0.822	0.811	0.823	0.822
	GRU + DBOW + DM	0.813	0.805	0.815	0.813
	Bi-LSTM +DBOW +DM	0.775	0.755	0.782	**0.886**
Bilingual	LSTM + DBOW	0.84	0.83	0.83	0.84
	GRU + DBOW	0.785	0.784	0.818	0.787
	Bi-LSTM + DBOW	0.83	0.82	0.84	0.83
	LSTM + DM	**0.850**	**0.85**	**0.87**	**0.85**
	GRU + DM	0.813	0.823	0.85	0.813
	Bi-LSTM + DM	0.831	0.827	0.847	0.831
	LSTM + DBOW +DM	0.822	0.811	0.823	0.822
	GRU + DBOW + DM	0.813	0.805	0.815	0.813
	Bi-LSTM +DBOW + DM	0.82	0.801	0.819	0.82

models, Bi-LSTM shows around 84 percent accuracy, and LSTM shows 82 percent accuracy. But GRU is providing the lowest accuracy in this case. From Table 6, when we have used bilingual data LSTM gives higher accuracy, which is around 84%, precision value 83%. Bi-LSTM, in this case, shows slightly lower accuracy, which is 83%.

When we have used distributed memory for feature extraction in the LSTM model, it gives the highest accuracy among all of the methods we have experimented. The precision score is the highest at 87% in this method. GRU model performs much better than Bi-LSTM model when we have concatenated DBOW and DM.



(a)



(b)

Fig. 11. Performance analysis of Deep learning models using Raad Labels (a) accuracy and (b) F1-score.

Table 6 is the representation of performance on Raad labeled data. We can see from Fig. 11(a) that accuracy is highest when we have worked with only Persian DM. But if we consider recall value, Persian DBOW + DM shows the highest values in Bi-LSTM models. Precision values are highest in both Persian and bilingual DM feature extraction methods compared to others.

From Table 7, we can see the total training time we need to train the datasets using all of the features. We have reported 3353 ms training time when we have achieved the highest 85% accuracy on the test data. The BiLSTM model takes a higher training time compared to the other two models in our study. When we have worked with four class data and concatenation of DBOW and DM, the training time is higher in all the cases.

From Fig. 12(a), we can see that the training accuracy is around 99 percent after 50 epochs, and it is quite stable. But the test accuracy fluctuates after 30 epochs from 70 to 80 percent. From Fig. 12(b), model loss degrades till 10 epochs, but after that, test loss increases mostly with fluctuating values. Training loss is quite stable and degrades over the epochs.

Table 8 presents the performance of some related works with this work for poetry classification. We have tried ML classifiers to improve the performance, however, ML models did not perform well in our study, as we reported in Table 3 and Table 5. The very recent work based on deep learning on English and Arabic data motivates us to work on this experiment with different word embedding techniques to report the best model and improve the overall performance of the Persian Ghazals.

5. Discussion

To address the first research objective, we implemented LSTM model with different document embedding techniques in our experiment. We implement DMM, DBOW and combination of DBOW and DM for the feature extraction in this study.

We applied both six class data and four class data annotated by Houman and Raad, respectively. When we used the DBOW feature extraction method in Houman labeled data (only Persian), accuracy reached 69%, F1-score 67%, precision 71% and recall 69%. But when we experimented with bilingual data with the same document embedding technique, accuracy increased to 73.6%, which is around 4.6% greater than the monolingual data classification. The F1-score found 73.1%, whereas precision is 73.5% and the recall value is 73.1%. All of the metrics values were improved when we switched to bilingual data with DBOW features. But from Table 4, we can see that the highest accuracy and F1-score were achieved using only Persian data, not with bilingual data. We achieved the highest accuracy value using Bi-LSTM and a combination of DBOW and DM feature in bilingual data, which is 75.8%, whereas, from LSTM and DM feature extraction, we get the highest 77.8% accuracy, 76.6% F1-score, 75% precision and 77% recall. From Table 6, we achieved 85% accuracy using LSTM model in both

Table 7
Training time using deep learning models (50 epochs).

	LSTM		GRU		BiLSTM	
	Feature	Training time (ms)	Feature	Training time (ms)	Feature	Training time (ms)
Houman Label (Six Class)	Persian DBOW	2350	Persian DBOW	457	Persian DBOW	6650
	Bilingual DBOW	1650	Bilingual DBOW	406	Bilingual DBOW	2785
	Persian DM	3376	Persian DM	766	Persian DM	6450
	Bilingual DM	2250	Bilingual DM	643 ns	Bilingual DM	3187
	Persian DBOW+DM	6850	Persian DBOW+DM	829	Persian DBOW+DM	9543
	Bilingual DBOW+DM	3804	Bilingual DBOW+DM	653	Bilingual DBOW+DM	4595
Raad Label (Four Class)	Persian DBOW	2206	Persian DBOW	424	Persian DBOW	4662
	Bilingual DBOW	2850	Bilingual DBOW	550	Bilingual DBOW	4703
	Persian DM	**3353**	Persian DM	455	Persian DM	4720
	Bilingual DM	3950	Bilingual DM	703	Bilingual DM	4753
	Persian DBOW+DM	5914	Persian DBOW+DM	800	Persian DBOW+DM	7589
	Bilingual DBOW+DM	6150	Bilingual DBOW+DM	820	Bilingual DBOW+DM	5080

Table 8
Comparative analysis of poetry work in different languages and approaches.

Study	Language	Approach	Result
Ahmed et al. [13]	English (Emotion Classification)	C-BiLSTM, LSTM, BiLSTM	88% accuracy using C-BiLSTM
Lou et al. [9]	English (Subject-based Classification)	SVM with TF-IDF and LDA	84.8% accuracy
Rahgozer [23]	Persian (Poetry classification)	SVM with BOW, TF-IDF and LDA similarity	79.11% accuracy
Praveenkumar et al. [14]	Indian English Poetry (Classification based on two eras)	RF classifier with reduced feature set	91.20% accuracy
Rakshit et al. [10]	Bengali (Classification based on four categories)	SVM with shallow parser	56.8% accuracy
Orabi et al. [18]	Arabic (Classified based on five distinct era)	CNN based	80.1% accuracy, 79.6% F1-score
Promrit and Waijanya [26]	Thai Poetry (based on category)	CNN model with Word2Vec	83% accuracy
Pal and Patel [27]	Hindi Poetry (3 category)	SVM, NB, RF, CART KNN	56.66% performance with SVM
Kaur [11]	Punjabi Poetry (Content based classification)	SVM and Textual feature	76.02% accuracy
Kaur et al. [31]	Punjabi poetry (Thematic Classifier)	Bi-LSTM, LSTM, CNN	80.32% accuracy with Bi-LSTM
Current Study	**Persian (with 4 class bilingual chronological data)**	**LSTM with Distributed memory mean**	**85% accuracy, 85% F1-score, 87% precision**

monolingual and bilingual data of refined labels. F1-score is also 85%, precision 87% and recall 85%.

To resolve the second research objective, we conduct classical ML algorithms as well as other DL models. We applied BOW feature extraction for classical ML classifiers and used the same document embedding technique for GRU and Bi-LSTM models. In our study, compared to ML classifiers, DL outperforms in most of the experiments we have done in this research. LSTM with DM outperforms compared to other learning models in our study.

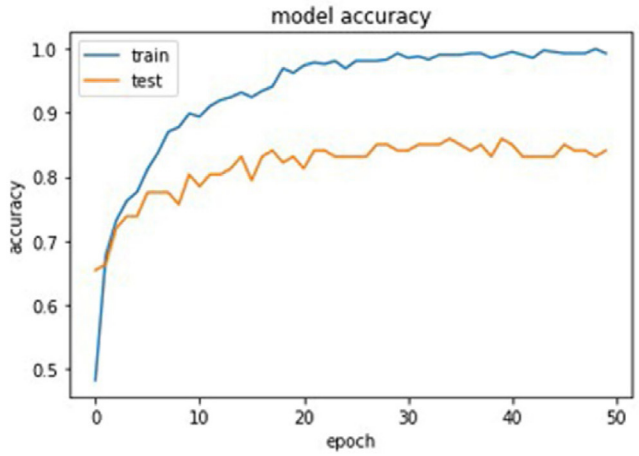
The last research objective compares performance with related research. From Table 8, we can see that closely related work on non-Persian language by Orabi et al. [18] reported 80.1% accuracy and 79.6% F1-score using CNN-based algorithms. We outperformed this where the accuracy value of around 5% and F1-score value of around 5.4% were improved. Whereas Indian English Poetry classification based on two eras by Praveenkumar et al. [14] used RF classifier with a reduced feature set and achieved 91.20% accuracy. Our proposed approach did not perform better than that binary classification system. Because in their work, two class labels such as pre-independence and post-independence were used. In those two eras, writing type was mostly changed so it is quite easier to differentiate and classify. But our data contains one author's lifespan data, so the word or language changes in the timespan do not vary much. We could not outperform Praveenkumar research work's performance. The most related literature work by Rahgozer [23] is listed in Table 8, and from the table, we can see that our strategy outperforms the work reported by Rahgozer [23]. With the feature extraction technique and LSTM model we improved the accuracy around 5.9% and F1-score around 25.9%.

6. Conclusion

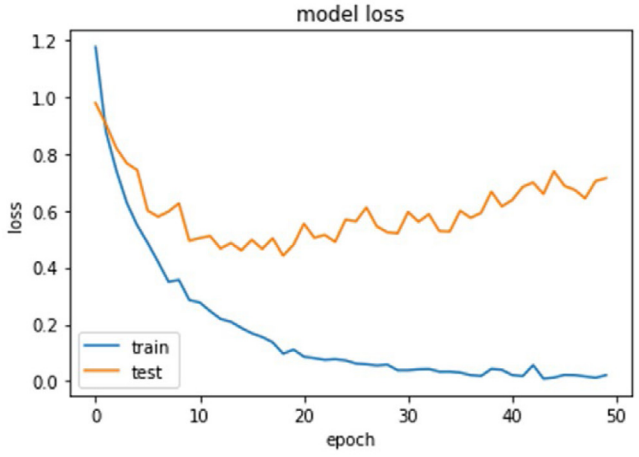
The poetry classification is a component of a broader architecture of poetry interpretation. Automatic text classification is a part of the discipline of Electronic Humanities, which connects humanities and computer science to make literary research easier. The purpose of our ghazal classification is to ascertain the relative chronology of each poem with respect to the author's lifespan. The main goal of this work is to propose and implement various DL models on Persian and bilingual data and analyze the results with traditional ML algorithms.

LSTM model with DM feature extraction provides the highest accuracy and precision value of around 85%. The applied model achieved better performance than the other studied baseline models. We also achieved around 7% higher accuracy when applying refined labels in our deep learning models, which gives 85% accuracy and 87% precision score on the bilingual data. We improved the accuracy value by around 6% from the previous study on Persian data.

Due to the insufficiency of labeled data based on chronological order, we could not perform the experiment with other poetry datasets. This dataset contains extremely imbalanced data, which may hinder more accurate results. If more data could be augmented in this poetry



(a)



(b)

Fig. 12. LSTM model performance on refined class labels (a) accuracy and (b) loss.

dataset, the robustness of the models could also be experimented with. The incorporation of data from other languages can also make this research more interesting. The inclusion of transformer-based models can also be an area of future research.

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

[1] C.H. de Fouchecour, Hafez: The Golden Age of Persian Literature, UNESCO Cour. a Wind. Open World. XLII, 1989, pp. 13–16, http://dx.doi.org/10.17953/amer.24.3.1761506755224221.

[2] E. Yarshater, Hafez, Encycl. Iran. XI, 2012, pp. 461–465, https://www.iranicaonline.org/articles/hafez-i.

[3] M.S. Khorsheed, A.O. Al-Thubaity, Comparative evaluation of text classification techniques using a large diverse Arabic dataset, Lang. Resour. Eval. 47 (2013) 513–538, http://dx.doi.org/10.1007/S10579-013-9221-8/FIGURES/5.

[4] P. Bafna, J.R. Saini, Hindi poetry classification using eager supervised machine learning algorithms, in: 2020 Int. Conf. Emerg. Smart Comput. Informatics, ESCI 2020, 2020, pp. 175–178, http://dx.doi.org/10.1109/ESCI48226.2020.9167632.

[5] A. Bagheri, M. Saraee, F. de Jong, Sentiment classification in Persian: Introducing a mutual information-based method for feature selection, in: 2013 21st Iran. Conf. Electr. Eng., 2013, pp. 1–6, http://dx.doi.org/10.1109/IranianCEE.2013.6599671.

[6] M. Farhoodi, A. Yari, Applying machine learning algorithms for automatic Persian text classification, in: 2010 6th Int. Conf. Adv. Inf. Manag. Serv., 2010, pp. 318–323.

[7] D. Salami, S. Momtazi, Recurrent convolutional neural networks for poet identification, Digit. Scholarsh. Humanit. 36 (2021) 472–481, http://dx.doi.org/10.1093/LLC/FQZ096.

[8] G. Ümit Yolcu, Binârî: A Poetry Generation System for Ghazals, 2020.

[9] A. Lou, D. Inkpen, C. Tanasescu, Multilabel subject-based classification of poetry, in: Twenty-Eighth Int. Flairs Conf., 2015.

[10] G. Rakshit, A. Ghosh, P. Bhattacharyya, G. Haffari, Automated analysis of bangla poetry for classification and poet identification, in: Proc. 12th Int. Conf. Nat. Lang. Process., NLP Association of India, Trivandrum, India, 2015, pp. 247–253, https://aclanthology.org/W15-5937.

[11] J. Kaur, J. Saini, Designing Punjabi poetry classifiers using machine learning and different textual features, Int. Arab J. Inf. Technol. 17 (2020) http://dx.doi.org/10.34028/iajit/17/1/5.

[12] O. Alsharif, D. Alshamaa, N. Ghneim, Emotion classification in arabic poetry using machine learning, Int. J. Comput. Appl. 65 (2013).

[13] S. Ahmad, M.Z. Asghar, F.M. Alotaibi, S. Khan, Classification of poetry text into the emotional states using deep learning technique, IEEE Access 8 (2020) 73865–73878, http://dx.doi.org/10.1109/ACCESS.2020.2987842.

[14] K. Praveenkumar, V. Naresh Mandhala, D. Bhattacharyya, D. Banerjee, Classification of Indian English poetry into pre-independence and post-independence eras using combination of semantics, topics and style features, NVEO - Nat. VOLATILES Essent. OILS J. | NVEO. 8 (2021) 162–170, http://www.nveo.org/index.php/journal/article/view/153. (Accessed 15 November 2021).

[15] S.A. Mahmood, Q.Q. Qasim, Big data sentimental analysis using document to vector and optimized support vector machine, UHD J. Sci. Technol. 4 (2020) 18–28, http://dx.doi.org/10.21928/UHDJST.V4N1Y2020.PP18-28.

[16] S.A. Khan, H.T. Chang, Comparative analysis on facebook post interaction using DNN, ELM and LSTM, PLoS One 14 (2019) e0224452, http://dx.doi.org/10.1371/JOURNAL.PONE.0224452.

[17] M. Gharbat, H. Saadeh, R.Q. Al Fayez, Discovering the applicability of classification algorithms with arabic poetry, in: 2019 IEEE Jordan Int. Jt. Conf. Electr. Eng. Inf. Technol. JEEIT 2019 - Proc., Institute of Electrical and Electronics Engineers Inc., 2019, pp. 453–458, http://dx.doi.org/10.1109/JEEIT.2019.8717387.

[18] M. Orabi, H. El Rifai, A. Elnagar, Classical arabic poetry: Classification based on era, in: 2020 IEEE/ACS 17th Int. Conf. Comput. Syst. Appl. 2020-November, 2020, pp. 1–6, http://dx.doi.org/10.1109/AICCSA50499.2020.9316520.

[19] M.A. Ahmed, R.A. Hasan, A.H. Ali, M.A. Mohammed, The classification of the modern arabic poetry using machine learning, TELKOMNIKA (Telecommunication Comput. Electron. Control. 17 (2019) 2667–2674, http://dx.doi.org/10.12928/TELKOMNIKA.V17I5.12646.

[20] G.A. Abandah, M.Z. Khedher, M.R. Abdel-Majeed, H.M. Mansour, S.F. Hulliel, L.M. Bisharat, Classifying and diacritizing Arabic poems using deep recurrent neural networks, J. King Saud Univ.-Comput. Inf. Sci. (2020) http://dx.doi.org/10.1016/J.JKSUCI.2020.12.002.

[21] S. Hamidi, F. Razzazi, M.P. Ghaemmaghami, Automatic meter classification in Persian poetries using support vector machines, in: 2009 IEEE Int. Symp. Signal Process. Inf. Technol., 2009, pp. 563–567, http://dx.doi.org/10.1109/ISSPIT.2009.5407514.

[22] E. Asgari, M. Ghassemi, M.A. Finlayson, Confirming the themes and interpretive unity of Ghazal poetry using topic models, in: Proc. NIPS Work. Top. Model. Comput. Appl. Eval., 2013, p. Submission 18, http://mimno.infosci.cornell.edu/nips2013ws/nips2013tm_submission_18.pdf.

[23] A. Rahgozar, Automatic Poetry Classification and Chronological Semantic Analysis, University of Ottawa, 2020, http://dx.doi.org/10.20381/RUOR-24749.

[24] N. Davari, M. Mahdian, A. Akhavanpour, N. Daneshpour, Persian document classification using deep learning methods, in: 2020 28th Iran. Conf. Electr. Eng. ICEE 2020, 2020, http://dx.doi.org/10.1109/ICEE50131.2020.9260650.

[25] K.P. Kumar, S.K.L. Kumar, et al., Analysis of Indian and American poetry using topic modeling and Deep learning, Mater. Today Proc. (2022).

[26] N. Promrit, S. Waijanya, Convolutional neural networks for thai poem classification, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 10261 LNCS, 2017, pp. 449–456, http://dx.doi.org/10.1007/978-3-319-59072-1_53.

[27] K. Pal, B.V. Patel, Data classification with k-fold cross validation and holdout accuracy estimation methods with 5 different machine learning techniques, in: Proc. 4th Int. Conf. Comput. Methodol. Commun. ICCMC 2020, 2020, pp. 83–87, http://dx.doi.org/10.1109/ICCMC48092.2020.ICCMC-00016.

[28] N. Jamal, M. Mohd, S.A. Noah, Poetry classification using support vector machines, J. Comput. Sci. 8 (2012) 1441.

[29] J. Kaur, J.R. Saini, Punjabi poetry classification: The test of 10 machine learning algorithms, in: Proc. 9th Int. Conf. Mach. Learn. Comput., Association for Computing Machinery, New York, NY, USA, 2017, pp. 1–5, http://dx.doi.org/10.1145/3055635.3056589.

[30] J.R. Saini, J. Kaur, Kāvi: An annotated corpus of Punjabi poetry with emotion detection based on 'Navrasa', Procedia Comput. Sci. 167 (2020) 1220–1229, http://dx.doi.org/10.1016/j.procs.2020.03.436.
[31] J. Kaur, J. Saini, Deep learning and super-hybrid textual feature based multi-category thematic classifier for punjabi poetry, in: Int. Conf. Comput. Eng. & Technol., 2022, pp. 42–52.
[32] J.S. Meisami, Allegorical gardens in the Persian poetic tradition: Nezami, Rumi, Hafez, Int. J. Middle East Stud. 17 (1985) 229–260, http://dx.doi.org/10.1017/S0020743800029019.
[33] A. Mohammadi, A comparative study of Ma'shuq or beloved in the ghazals of Hafez and sonnets of Shakespeare, Int. J. Innov. Sci. Res. Rev. 03 (2021) 1648–1655, http://www.journalijisr.com.
[34] Hafiz, W. Jones, Hafiz, the Prince of Persian Lyric Poets, Frederick A. Stokes' Company Publishers, New York, 2013.
[35] J.S. Meisami, A Life in Poetry: Hafiz's First Ghazal', Necklace Pleiades. 24 Essays Persian Lit. Cult. Relig, 2010, pp. 163–181.
[36] J. Limbert, Shiraz in the Age of Hafez: The Glory of a Medieval Persian City, University of Washington Press, 2004.
[37] J.S. Meisami, Medieval Persian Court Poetry, Princeton University Press, 2014, http://dx.doi.org/10.1515/9781400858781.
[38] R. Shabānī, The Book of Iran: a Selection of the History of Iran, 2005, p. 370.
[39] M. Ebrahimi, M. Ebrahimi, The aesthetic comparison of Hafez Shirazi's and William, Iran. EFL J. 4 (2014) 185.
[40] Ganjoor Hafez, https://ganjoor.net/hafez/. (Accessed 27 October 2021).
[41] M. Houman, Hafez, Tahouri Library, Iran, 1938.
[42] S. Shahriari, Ghazaliyat of Hafiz, 1999, https://www.hafizonlove.com/divan/. (Accessed 25 October 2021).
[43] A. Nourian, hazm, PyPI, 2018, https://pypi.org/project/hazm/. (Accessed 10 March 2022).
[44] S. Bird, E. Klein, E. Loper, Natural language processing with Python: analyzing text with the natural language toolkit, O'Reilly Media, Inc., 2009.
[45] Y. Zhang, R. Jin, Z.H. Zhou, Understanding bag-of-words model: a statistical framework, Int. J. Mach. Learn. Cybern. 11 (1) (2010) 43–52, http://dx.doi.org/10.1007/S13042-010-0001-0.
[46] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[47] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents, in: 31st Int. Conf. Mach. Learn. ICML 2014, vol. 4, 2014, pp. 2931–2939, https://arxiv.org/abs/1405.4053v2. (Accessed 27 October 2021).
[48] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: 1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc., 2013, https://arxiv.org/abs/1301.3781v3. (Accessed 16 November 2021).
[49] M. Rhanoui, M. Mikram, S. Yousfi, S. Barzali, A CNN-BiLSTM model for document-level sentiment analysis, Mach. Learn. Knowl. Extr. 1 (2019) 832–847, http://dx.doi.org/10.3390/MAKE1030048.
[50] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780, http://dx.doi.org/10.1162/neco.1997.9.8.1735.
[51] J. Chung, C. Gulcehre, K. Cho, Y. Begio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, http://arxiv.org/abs/1412.3555 (accessed September 9, 2021).
[52] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (1997) 2673–2681, http://dx.doi.org/10.1109/78.650093.
[53] L. Jeni, J. Cohn, F. De la Torre, Facing Imbalanced Data - Recommendations for the Use of Performance Metrics, 2013, http://dx.doi.org/10.1109/ACII.2013.47.