Computers, Materials & Continua
DOI:10.32604/cmc.2022.020083
Article

Tech Science Press

Benchmarking Performance of Document Level Classification and Topic Modeling

Muhammad Shahid Bhatti[1,*], Azmat Ullah[1], Rohaya Latip[2], Abid Sohail[1], Anum Riaz[1] and Rohail Hassan[3]

[1]Department of Computer Science, COMSATS University Islamabad, Lahore, Pakistan
[2]Department of Communication Technology and Network, Faculty of Computer Science, Universiti Putra Malaysia, Selangor, 43400, Malaysia
[3]Othman Yeop Abdullah Graduate School of Business (OYAGSB), Universiti Utara Malaysia (UUM), Kuala Lumpur, 50300, Malaysia
[*]Corresponding Author: Muhammad Shahid Bhatti. Email: msbhatti@cuilahore.edu.pk
Received: 08 May 2021; Accepted: 18 June 2021

Abstract: Text classification of low resource language is always a trivial and challenging problem. This paper discusses the process of Urdu news classification and Urdu documents similarity. Urdu is one of the most famous spoken languages in Asia. The implementation of computational methodologies for text classification has increased over time. However, Urdu language has not much experimented with research, it does not have readily available datasets, which turn out to be the primary reason behind limited research and applying the latest methodologies to the Urdu. To overcome these obstacles, a medium-sized dataset having six categories is collected from authentic Pakistani news sources. Urdu is a rich but complex language. Text processing can be challenging for Urdu due to its complex features as compared to other languages. Term frequency-inverse document frequency (TFIDF) based term weighting scheme for extracting features, chi-2 for selecting essential features, and Linear discriminant analysis (LDA) for dimensionality reduction have been used. TFIDF matrix and cosine similarity measure have been used to identify similar documents in a collection and find the semantic meaning of words in a document FastText model has been applied. The training-test split evaluation methodology is used for this experimentation, which includes 70% for training data and 30% for testing data. State-of-the-art machine learning and deep dense neural network approaches for Urdu news classification have been used. Finally, we trained Multinomial Naïve Bayes, XGBoost, Bagging, and Deep dense neural network. Bagging and deep dense neural network outperformed the other algorithms. The experimental results show that deep dense achieves 92.0% mean f1 score, and Bagging 95.0% f1 score.

Keywords: Deep neural network; machine learning; natural language processing; TFIDF; sparse matrix; cosine similarity; classification; linear discriminant analysis; gradient boosting

1 Introduction

The Urdu language is one of the widely spoken languages. Moreover, Urdu news is also one of the vital news sources read all over the World [1]. In terms of the writing system, Urdu uses the Nastaliq script [2], a modified Persian script [3] that is a type of modified Arabic script. Unfortunately, Urdu has not been given significant recognition in the research area. It has not been exposed to new methodologies and technologies. There is not much research-related work in this language, particularly on Urdu news [4]. A major reason behind the lack of research and Urdu news classification is the non-availability of many Urdu news datasets. So far, there are not many publicly available datasets. To fill this gap, a dataset [5] consists of six distinct classes is collected. The information extracted from the news website contains the link of the news, the date, the title, and the new text to perform text classification [6]. The long textual content is more challenging, complicated, and computationally expensive. A document may comprise a sentence, a paragraph, or an extended text of hundreds of words. A large dataset typically has a massive vocabulary length, more noise, and redundant information than a small dataset.

Similarly, a couple of paragraphs inside the same dataset semantically belong to a couple of categories [7]. Due to more complex morphology, the lack of linguistic resources, and characteristics. The Urdu news classification [6] is a more complicated and challenging task. Further, the text classification implemented for other similar languages cannot be applied to Urdu. The reputation of the Urdu is rapidly growing. These rapid changes caught the interest of researchers in Urdu. Text classification models are based on machine learning, and machine learning models do not perform well on raw datasets. Therefore, the basic dataset must be preprocessed. Preprocessing steps include removing duplicate documents, removing any null values, tokenizing sentences, removing punctuation marks, and stop-words removal.

To improve the learning of a model, features must be extracted from a dataset because they directly learn from features. Feature extraction [8] is the most significant task in machine learning. But unfortunately, Urdu does not have proper delimitation in between words. Extraction of these features is computationally expensive, requires expertise, and domain knowledge. The main challenging task is selecting the essential elements from extracted features presented in high dimensional space [9]. This problem is resolved by using two techniques. First, we chose the essential features from the high dimension space, and finally, we apply the LDA [10] dimensionality reduction. These techniques not only enhance the performance of the machine learning algorithms but also reduces the dimensionality space.

Moreover, deep dense neural networks extract more complex and meaningful features from feature space compared to machine learning algorithms. The train-test split evaluation [11] methodology is applied. As the dataset is imbalanced, measuring the performance of models following metrics has been used precision, recall, f1-score, and Cohen kappa [12]. Similarity analysis [13] is an emerging research area. Digital data are increasing over time. Unstructured data need efficient methods to find any relevant or similar topic. Many systems have been implemented for different languages (Arabic, Hindi, and Bengali, etc.) to retrieve relevant documents based on the similarity score. But unfortunately, no research work has been done on Urdu. Our proposed system for the Urdu news similarity provides a good result that has been compiled after evaluation. In this study, the state-of-the-art algorithms multi Naïve Bayes [14], Bagging [15], XGBoost [15], and Deep dense neural network [16] have been trained on the Urdu news dataset.

To solve a classification and similarity problem, state-of-the-art approaches such as feature extraction, feature selection, gradient boosting, and deep learning have been used. A research project an ensemble and a scalable boosting system XGBoost [17] which seeks to build a robust classifier based on all weak classifiers. It proposes a sparsity algorithm for small data to achieve better performance. A semantic similarity analysis [18] and [19] co-occurrences using FastText Embeddings for Urdu documents given a superior semantic similarity score. The methodologies of deep learning algorithms [20], such as convolution neural network (CNN) and recurrent neural network (RNN) for multiclass event classification, have been performed on labelled instances and achieved a promising result of 83%.

This paper is structured as follows: Section 2 explains the literature review in which we described some of the work previously done regarding text classification. Section 2 presents the characteristics of the Urdu language. Section 3 describes data collection, preprocessing, and feature optimization. It includes dataset description, data cleaning, and removal of stop words. Section 4 describes Machine learning (ML) Experimental Design, which includes discussing different ML models and adopted model descriptions. Section 5 is about topic modelling based on the similarity of two documents. Section 6 explains the results, and the last section concludes the summary of all of the work.

2 The Characteristics of Urdu

Urdu is derived from Turkish, Arabic, and Persian languages. It is written in a complex and context-sensitive style known as the Nastaliq style. The Urdu comprises of words that consist of a combination of several characters known as ligatures. To form a sentence, the terms are joined from right to left. The Urdu has gone through a progression of improvement during its development. Nevertheless, Urdu has its roots in Persian, Arabic, and similarities with most south Asian languages. For example, similarity in terms of lack of capitalization, lack of small and capital words, and free word order characteristic. Urdu is transcribed in the derivation of the Persian alphabet that is a derivation of the Arabic alphabet.

- It is read from right to left For example: "مجھے اپنے ملک پاکستان سے محبت ہے" (I love my country Pakistan).
- The shape assumed by a character in a word is context-sensitive.
- It consists of 38 basic characters which are known as "حروف تہجی" (HARUF-e-TAHAJI)(Alphabets) in the Urdu. Therefore, to learn and write the Urdu, everyone has to understand the "تہجی حروف" (HARUF-e-TAHAJI)(Alphabets) first.
- Urdu "اسم" (Ism)(Noun) has two grammatical genders "مذکر" (Muzakir)(Masculine) and "مونث" (Monus)(Feminine). Nouns may have particular gender suffixes or be unmarked for gender. Nouns are inflected to showcase and number (singular or plural).
- Verb "فعل" (Fael) corresponds to the occurrence or performing some action. That verb that does not take an object is called "لازم فعل" (Fael-Lazim) (Intransitive verb). When a verb needs a direct object, then it is called a "متعدی فعل" (Fael-Muatadi)(Transitive verb)
- A character of Urdu has different shapes, but when joined with other characters, it forms a word. For example, the characters "خ" (Khee), "ی" (Ye), "ا" (Alif), and "ل" (Laam) when joined is "خیال" (Khayal)(Idea).

- Vowels of the Urdu are written as "ا" (alif), "و" (wao), "ی" (choti ye), "ے" (bari ye).
- A "نقطہ" (Dot) plays a vital role in the Urdu alphabets. For example: "ج" (Jeem) with a "نقطہ" (Dot) below it, "خ" "Khe" with a "نقطہ" (Dot) above it, with not a "نقطہ" (Dot) it becomes "ح" (hey).
- The character "ب" (bay), has its basic shape in common with three other characters, "ت" (Tey), "ث" (Say), and "پ" (Pay). Therefore, it is one of the challenges for learners to differentiate between these characters. The 38 basic characters of Urdu are shown in Fig. 1.

Figure 1: The character set of the Urdu language

The implementation of computational methodologies for text classification has increased over time. This led to the data collection in various languages like Persian, Chinese, Hindi, and Arabic. However, despite being spoken widely across the World, the Urdu language has not much experimented with text classification. For this particular experiment, a dataset of 20 thousand instances has been used see Tab. 1, the Urdu news does not have easily available datasets which turn out to be the major reason for the collection of the medium-sized dataset. This paper provides an accurate and publicly available dataset that is manually extracted from authentic Pakistani Urdu news sources. The details of news articles are collected into six major categories: Health, technology, sports, politics, showbiz, technology, and business & economy. The dataset is manually collected from the following websites Ary News, Bol News, Express-News, BBC News, Daily Jang News, SAMAA News, GEO News, and DAWN News. The train-test split is used for this experimentation which includes 70% for training data and 30% for testing data [11]. Our contribution is to:

- Gather a large dataset of Urdu news documents and make it available for future research.
- Preprocess the dataset.
- Examine the effect on the performance of classification using feature extraction and feature selection.
- To train machine learning and deep dense neural network classifier on the Urdu news documents.
- Examine the efficiency of Naïve Bayes, Bagging, XGBoost, and deep dense neural network on the dataset.
- Find the similarity among Urdu news.

Table 1: Details of dataset's instances and features

Dataset	Size
Total classes	6
Politics	6,274
Health	4,715
Business & Economy	3,607
Sports	2,309
Showbiz	2,017
Technology	1,183
Total words	5,62,0465
Total features	19,087
Total instances (After preprocessing)	20,105

3 Machine Learning Model Selection

3.1 Data Preprocessing

The dataset collected for Urdu news classification is not useful unless it is pre-processed. The data must be represented in such a way that the performance of the machine learning algorithms and deep dense neural networks can be improved. For this experiment, the most useful pre-processing techniques like tokenization, stopwords removal, and punctuation marks removal have been used.

I. Tokenization

The tokenization of the Urdu news involves splitting the news articles into small tokens. Each word in the news article is treated as a token.

II. Stop words removal

The stopwords do not carry any information and it affects the performance of machine learning model(s). The stopwords have been removed by using the TFIDF algorithm.

III. Punctuation marks removal

Similarly, the punctuation does not carry any information and it affects the performance of machine learning model(s). The punctuation marks have been removed by using the NLTK algorithm.

The pre-processing techniques are specifically defined for the Urdu language only and it will not work for any other language(s). The input attributes and output attributes are correlated with each other, but the given output attribute is in categorical form, it was required to encode the output attribute into numeric form.

3.2 Evaluation Methodology

After preprocessing, the Urdu dataset follows the characteristics of high quality and large in amount. In this paper, the train-test split approach is applied. Thus, 70% has been used as training data to build the machine learning and deep dense neural network models, and 30% has been used as test data to evaluate the performance of models. The distribution of training data and testing data is shown in Tab. 2.

Table 2: Distribution of the dataset

Dataset	Train data size	Test data size
Total classes	6	6
Politics	4,360	1,925
Health	3,312	1,401
Business & Economy	2,549	1,058
Showbiz	1,393	624
Technology	807	369
Sports	1,652	655
Total instances	14,073	6,032

3.3 Feature Extraction

Urdu is a rich but complex language that makes it difficult for automatic text processing. For Urdu news classification, the sentences are split into tokens. The range of n-gram taken is 1 for minimum n-gram and 1 for maximum n-gram. By tokenization of the sentences into words makes it easy to interpret the meaning of the sentence. It also helps to improve the performance of machine learning and deep dense neural network models. The experiment is performed by using the most common stop words [21] "ہے", "ہیں", "سے", "تھے", "کے" ("is", "are", "from", "were", "of"). TFIDF transforms the documents of the dataset into sparse-matrix [22]. The sparse matrix contains numeric values which represent how important a term is to a document. The equation Eqs. (1) and (2) is a statistical measure; it calculates that how important a word to a document is.

$$IDF(t) = log\frac{1+n}{1+df(d,t)} + 1 \tag{1}$$

$$tfidf = TF(t,d) + IDF(t). \tag{2}$$

3.4 Dimensionality Reduction

Feature extraction using the term TFIDF [8] is computationally expensive because the extracted features are represented in sparse-matrix's format. In this particular problem, the representation of sparse-matrix [22] for 21,105 instances have 19,087 dimensions, the sparse-matrix [22] holds more 0 s than meaningful values. To overcome this issue, the chi-2 feature selection is applied, which reduces the matrix's size by selecting the highly dependent features on the responses. We selected 10,000 most important features out of 19,087, and finally, we applied the LDA [10]. LDA projects the data points in a new linear 2-dimensional feature space. It reduces the dimensionality from the original number of features to (Number of classes-1). The result of the TFIDF feature vector for each article, projected on two dimensions, is shown in Fig. 2.

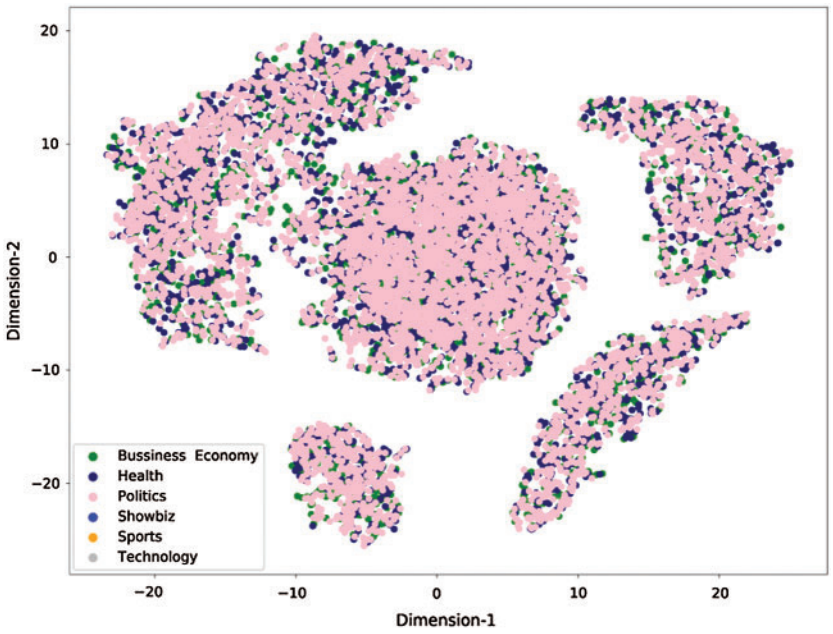

Figure 2: TFIDF feature vector projected on two dimensions

3.5 Evaluation Measure

The data collected is imbalanced. The four main metrics used to evaluate the performance of trained models. Precision, recall, f1_score, and cohen kappa. All measures distinguish correct classification of target variable within the different classes. The recall is a function of correctly classified true positives and misclassified false positives. Precision is the function of correctly classified examples and misclassified as false positives. F1-score is a way of combining both precision and recall. Cohen kappa [12] is a statistical test to find interrater reliability. To evaluate the performance of multiclass classification, a confusion matrix has been used, and the result is shown in Fig. 3.

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

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

$$F1 - score = 2 * \frac{(precision * recall)}{(precision + recall)} \tag{5}$$

$$CohenKappa = \frac{\sum^{a} - \sum^{e} f}{N - \sum^{e} f} \tag{6}$$

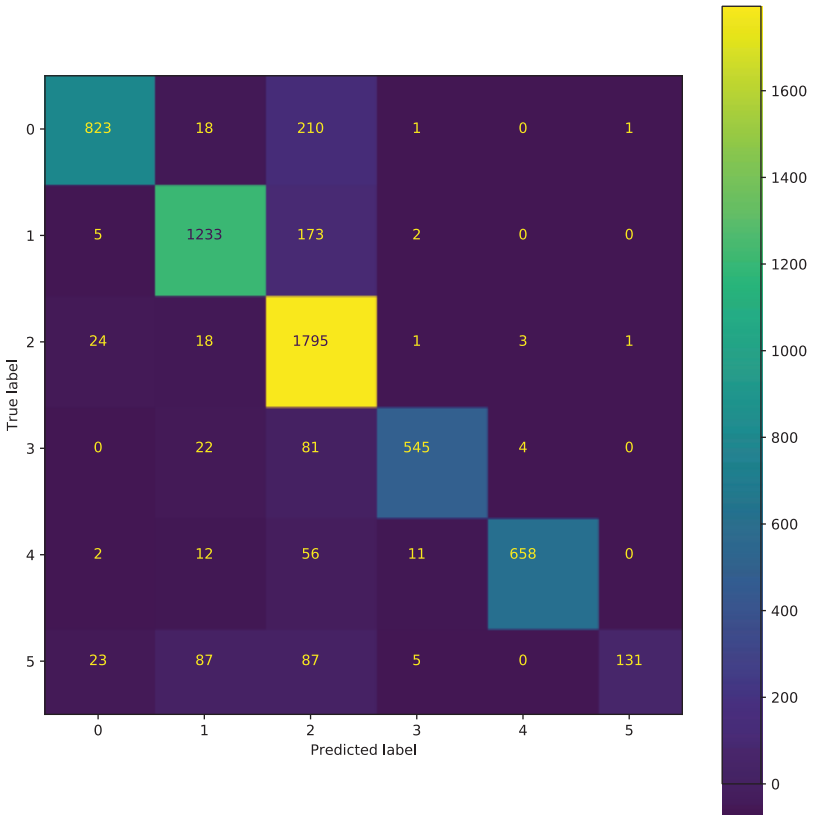

Figure 3: Confusion matrix showing TP, TN, FP, and FN of six classes

4 Machine Learning Experimental Design

Text classification with machine learning algorithms and deep dense neural networks are much more efficient and accurate than manual methods. State-of-the-art machine learning and deep dense neural network algorithms have been used to perform Urdu news classification. Multi Naïve Bayes, Bagging, XGBoost, and deep dense neural network. The architecture of the machine learning models is shown in Fig. 4.

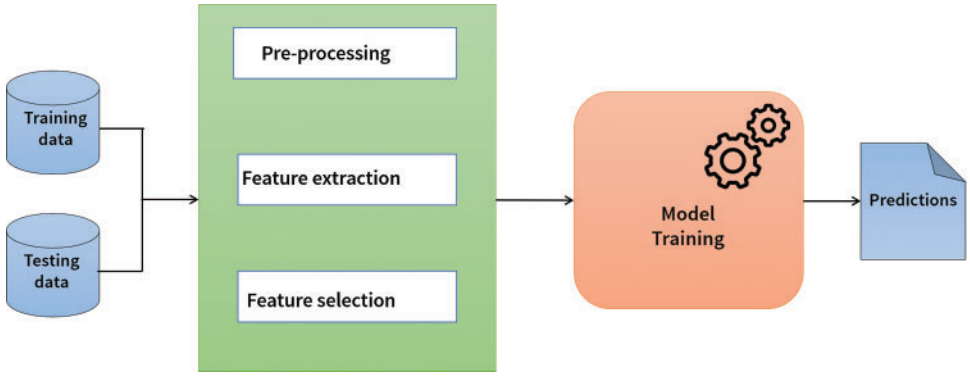

Figure 4: Architecture of the system

4.1 Multinomial Naïve Bayes

Multinomial Naïve Bayes is based on the Bayes theorem; it assumes that the drawn features exist in multinomial distribution. We have used it as a baseline classifier for Urdu news classification. However, the parameters of multinomial Naive Bayes are not tuned.

4.2 Bagging

Bagging is an ensemble estimator. It fits N-base estimators on a random subset of the original data. And it aggregates their scores(predictions) to form a final score(prediction). In this experiment, we have used Sklearn's MLP Classifier (multilayer-perceptron) as a base estimator. Tab. 3 shows the parameters of the base-estimator, and Tab. 4 shows the parameters of the bagging classifier.

Table 3: Parameters of base- estimator (Multilayer-Perceptron) classifier

Parameters	Values
Size of hidden layers	100
Activation function	Relu
Optimizer	Adam
Learning rate	0.001
Batch size	Auto

Table 4: Parameters of bagging classifier

Parameters	Values
Base-estimator	MLP classifier
Number of base-estimators	50

4.3 XGBoost

EXtreme Gradient Boosting (XGBoost) is a decision tree-based ensemble algorithm that uses the gradient-boosting framework. It outperforms all other traditional machine learning algorithms in tabular data parameters are shown in Tab. 5. Moreover, XGBoost has been credited with winning Kaggle's competitions [14]. It applies the principle of weak learners. It accurately predicts an output variable by combining the estimates of weaker models.

Table 5: Parameter description of XGBoost

Parameters	Values
Learning rate	0.001
Estimators	2500
Objective	Multi-Softmax
Number of classes	6
Iterations	700

4.4 Deep Dense Neural Network

In this paper, Kera's deep dense neural network has been used. The hidden layers and hidden neurons are selected randomly. To avoid overfitting, the dropout layer is used. The parameters and their values are shown in Tab. 6, and the architecture is shown in Fig. 5 and Tab. 7. A sample run of the input vector is shown in Eq. (7).

$$X = \begin{bmatrix} 0.57615236 \\ 0.40993715 \\ 0.57615236 \\ 040993715 \end{bmatrix} \tag{7}$$

Table 6: Deep neural network architecture

Layer	Neurons/Values	Activation function
Input layer	10,000	–
Drop out layer	0.3	–
Hidden layer 1	200	Relu
Drop out layer	0.5	–
Hidden layer 2	360	Relu
Drop out layer	0.8	–
Hidden layer 3	520	Relu
Drop out layer	0.7	–
Hidden layer 4	280	Relu
Drop out layer	0.75	–
Output layer	6	Softmax

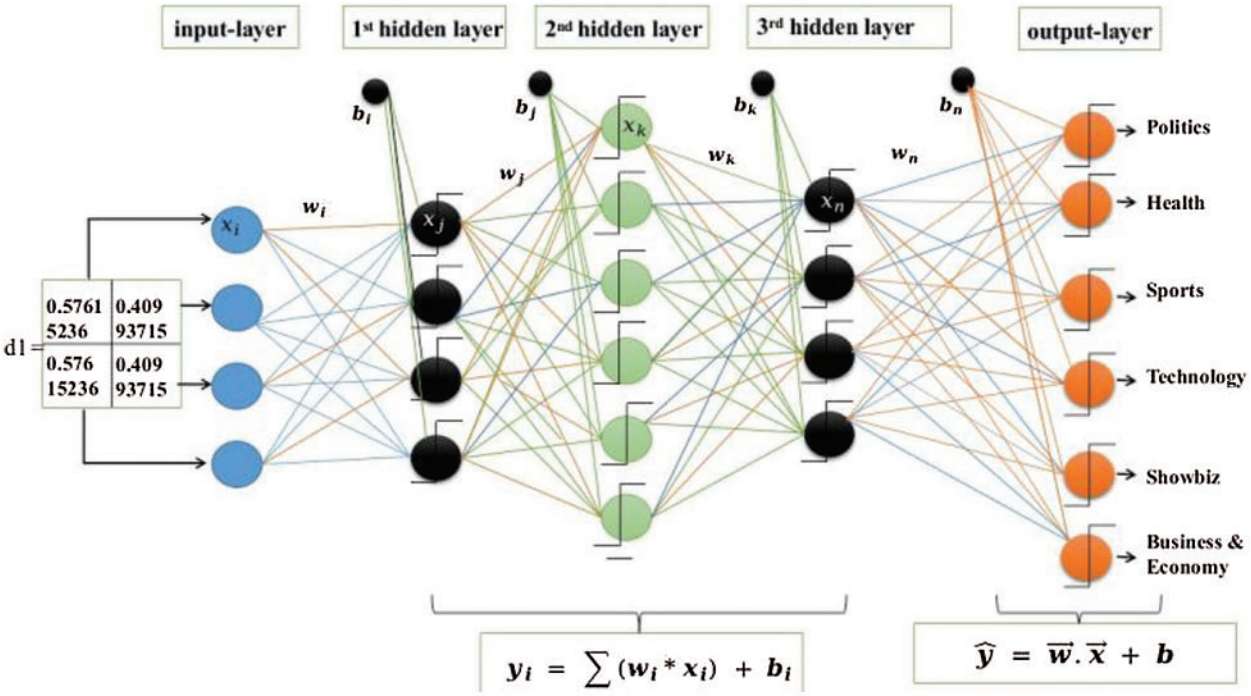

Figure 5: Deep dense neural network parameters used

Table 7: Deep dense neural network parameters used

Parameters	Values
Learning rate	0.01
Metric	Accuracy
Optimizer	Adam
Loss	Sparse categorical entropy
Batch size	200
Epochs	500

Mathematical representation of deep dense neural network feed-forward propagation.

4.4.1 Input Layer

The input layer contains the vector of n-features extracted by using the TFIDF approach.

$$X = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \tag{8}$$

4.4.2 Relu Activation Function

The rectified linear activation function (reLU) is a simple activation function that returns zero if the passed value x is zero or less than zero, but for any positive value, it returns the value.

$$y = max(0, x) \tag{9}$$

4.4.3 Hidden Layer

The hidden layer neuron's values are calculated using the summation of the input nodes X multiplied by their assigned weights w. In addition, the value of bias neuron b is also added to the summation. Finally, the output from the hidden layer is calculated using the reLU activation g.

$$h_i = g(w_i.X_i + b_i) \tag{10}$$

4.4.4 Output Layer

The weighted sum of inputs into the output layer is calculated using the summation of the neuron x fired from the hidden layer multiplied by their assigned weights w. The value of bias neuron b is also added to the summation. The output y-hat is calculated using the Softmax activation g, which converts the output scores into probability distributions.

$$\hat{y} = g(w_i.h_i + b_i) \tag{11}$$

4.4.5 Loss Function (Sparse Categorical Cross-Entropy)

The sparse categorical cross-entropy works on integers, these integers are the class indices y-hat, this loss computes logarithm only for output index $J(w)$.

$$J(w) = -log(\hat{y}_y). \tag{12}$$

Back propagation for fully connected dense layers. The error term for output unit.

4.4.6 The Error Term for the Output Unit

If the observed example $o(E)$ of output unit k is not equal to the target example $t(E)$, then the error for unit k is calculated as

$$\delta_{Ok} = o_k(E)(1 - o_k(E))(t_k(E) - o_k(E)) \tag{13}$$

4.4.7 The Error Term for Hidden Unit 'k'

If the observed example $h(E)$ of hidden unit k is not equal to the target example $t(E)$, then the error for unit k is calculated as

$$\delta_{Hk} = h_k(E)(1 - h_k(E))(t_k(E) - h_k(E)) \tag{14}$$

4.4.8 Updating Weights Between Hidden Units and Output Units

The error term at the output units δ_O and the error term at hidden units h is multiplied with the learning rate η.

$$\Delta_{ij} = \eta \delta_{Oj} h_i(E). \tag{15}$$

4.4.9 Updating Weights Between Input Units and Hidden Units

Input to the system x and the error of hidden unit δ_H is multiplied with the learning rate η.

$$\Delta_{ij} = \eta \delta_{Hj} x_i. \tag{16}$$

Following is an example of an Urdu sentence inside a news article:

"مصنوعی ذہانت مسئلہ حل کرنے والا ہے" (Artificial intelligence is solving problems)

TFIDF vector representation)

5 Topic Modeling Based on Similarity

Cosine similarity [23] is a widely used metric in the retrieval of similar information. It models a document in terms of a vector. We can find the similarity between the two documents by determining their cosine values. This metric can be applied to sentences and as well as to paragraphs. Urdu documents are represented in terms of sparse-matrix generated by TFIDF [8].

Representation of Urdu document 1.

"اردو زبان نے تحقیق کی توجہ حاصل کی"} "(The Urdu language received the attention of research)

Vector representation of Urdu document 1.

$$V_i = \begin{bmatrix} 0.26844636 & 0.26844636 & 0.37729199 & 0.37729199 & 0.0 \\ 0.26844636 & 0.0 & 0.0 & 0.0 & 0.37729199 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.26844636 \\ 0.37729199 & 0.0 & 0.0 & 0.0 & 0.37729199 \end{bmatrix}$$

Representation of Urdu document 2.

"ہمارے تحقیقی مقالے سے اردو زبان کے مسئلے کو حل کرنے میں مدد ملے گی" (Our research paper will help to solve the problem of the Urdu language)

Vector representation of documents 2.

$$V_i = \begin{bmatrix} 0.19714759 & 0.19714759 & 0.0 & 0.0 & 0.27708406 \\ 0.19714759 & 0.27708406 & 0.27708406 & 0.27708406 & 0.0 \\ 0.27708406 & 0.27708406 & 0.27708406 & 0.27708406 & 0.19714759 \\ 0.0 & 0.27708406 & 0.27708406 & 0.27708406 & 0.0 \end{bmatrix}$$

The similarity between two documents can be determined using Eq. (17).

$$\cos(A, B) = \frac{A.B}{\|A\|.\|B\|}. \tag{17}$$

The similarity of document 1 and document 2 is:

cosine (A, B) = 0.211

But the issue of cosine similarity is that it cannot calculate the semantic meaning of words. So, Word2Vec and FastText [24] models are trained. Word2Vec and FastText are word embedding techniques that use the neural network approach. Word2Vec has one particular issue called "Out of Vocabulary" if the word does not exist in a vector, it raises an error. This issue was resolved by using the FastText model. We have trained FastText on more than five million words. The trained model calculated the semantic meaning of the word, and Tab. 8 shows the results of the input word "وزیر" (Minister).

Table 8: FastText's most-similar words

Similarity	Words
0.5325	وزیر اعلی
0.5237	وزیر اعظم
0.4851	وزیر خارجہ
0.4831	میئر
0.4769	مینجمنٹ

6 Results and Discussion

This section discusses the results of machine learning algorithms and deep dense neural network trained on an imbalanced Urdu news dataset. For the Urdu news classification, we prepared and pre-processed the Urdu news dataset. The dataset is splitted into train and test sets containing 70% and 30% news instances from each class. Since the dataset contains raw data, we extracted and selected the most important features, first we used all the features of the training and testing dataset. Then, we selected the top k features and fed those features to different classifiers. The top selected features are transformed into a sparse-matrix [22] using TF-IDF. Even after selecting the most important features from the dataset the sparse-matrix is still computationally expensive, in order to reduce the dimensionality space we have used Linear Discriminant Analysis [10], to further reduce the dimensions and improve the performance of classifiers.

To carry out the experiment, the state-of-the-art machine learning algorithms Multinomial Naïve Bayes, Bagging, XGBoost, and Keras deep dense neural network are applied.

The experiments carried out two iterations for each machine learning algorithm. One by using the default parameters of algorithms and second by choosing the best parameters.

Since the dataset is imbalanced, we have evaluated the performance of models by using the following metrics precision, recall, f1_score, and Cohen kappa. After tuning the parameters of models, we trained our four classifiers by including the most common stops-words "ہے", "ہیں", "سے", "تھے", "کے", ("is", "are", "from", "were", "of") of the Urdu Language. The results of the models are shown in Tab. 9.

Table 9: Results of classifiers

Measures	NB	Bagging	XGBoost	DDNN
Precision	92.0%	95.0%	88.0%	92.0%
Recall	83.0%	94.0%	84.0%	92.0%
F1-score	86.0%	95.0%	85.0%	92.0%
Cohen Kappa	84.0%	94.0%	84.0%	91.0%
Training time(hrs)	0.0027	1.47	0.16	0.27

In investigating the imbalanced Urdu news dataset, it was discovered that the Bagging (with base-estimator Multilayer Perceptron) and deep dense neural network out performed Multinomial Naïve Bayes and XGboost. According to the current results deep dense neural network achieved 92.0% mean precision, 92.0% mean recall, 92.0% mean f1_score, and 91.0% mean Cohen kappa and Bagging achieved 95.0% precision, 94.0% recall, 95.0% f1_score, and 94.0% Cohen kappa. The performance analysis of classifiers is shown in Fig. 6. The Receiver Operator Characteristic (ROC) curve of Bagging is shown in Fig. 7. The ROC curve of the Deep dense neural network is shown in Fig. 8.

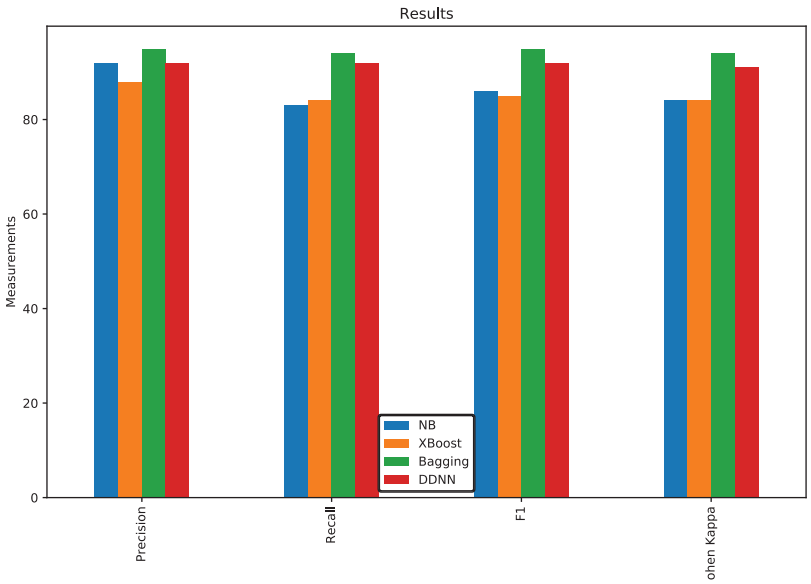

Figure 6: Performance analysis of algorithms

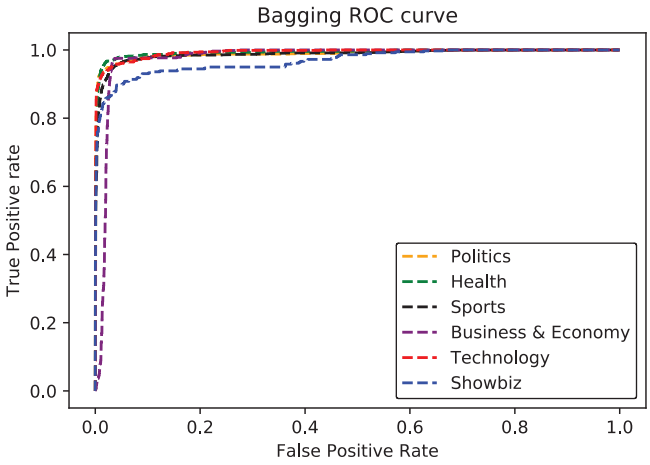

Figure 7: ROC of bagging

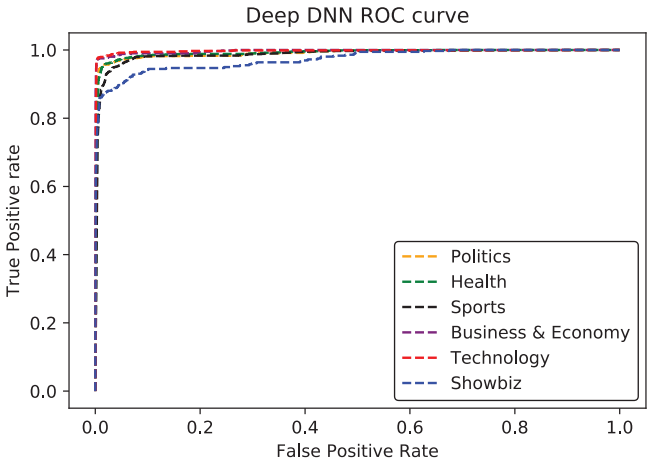

Figure 8: ROC of deep dense neural network

7 Conclusion

This paper can be considered a significant landmark to perform Urdu news classification, finding cosine similarity among Urdu news. It employs FastText to find the semantic meaning of words. Furthermore, it defines extracting features from documents and selecting features for sparsity reduction to make the machine learning algorithms more efficient. It demonstrates the application of state-of-the-art machine learning algorithms, Multinomial Naïve Bayes, Bagging, XGBoost, Deep dense neural network. Experimental results show Bagging with Multilayer-Perceptron as a base- estimator and deep dense neural network algorithms perform better on the Urdu news dataset than Multinomial Naive Bayes XGBoost. Deep dense neural network achieved 92.0% mean precision, 92.0% mean recall, 92.0% mean f1_score, and 91.0% mean Cohen kappa and Bagging achieved 95.0% precision, 94.0% recall, 95.0% f1_score, and 94.0% Cohen kappa.

Funding Statement: This research has been funded by Universiti Putra Malaysia and supported by the Ministry of Higher Education Malaysia.

Conflicts of Interest: The authors declare that they have no interest in reporting regarding the present study.

References

[1] M. Lal, K. Kumar, A. A. Wagan, M. A. Khuhro, U. Saeed *et al.*, "A systematic study of urdu language processing its tools and techniques," *International Journal of Engineering Research Technology*, vol. 9, no. 12, pp. 39–43, 2020.

[2] M. Alam and S. U. Hussain, "Sequence to sequence networks for roman-urdu to urdu transliteration," in *Proc. of Int. Multi-Topic Conf. (INMIC)*, pp. 1–7, IEEE. Lahore, Pakistan, 2017.

[3] S. Izadi, J. Sadari, F. Solimanpour and C. Y. Suen, "A review on Persian script and recognition techniques," in *Summit on Arabic and Chinese Handwriting Recognition*, pp. 22–35, Springer, Berlin, Heidelberg, 2006.

[4] M. V. Patil and A. M. N. Yogim, "Importance of data collection and validation for systematic software development process," *International Journal of Computer Science & Information Technology*, vol. 3, no. 2, pp. 260–278, 2011.

[5] K. Cengiz, R. Sharma, K. Kottursamy, K. K. Singh, T. Topac *et al.*, "Recent emerging technologies for intelligent learning and analytics in big data," *Multimedia Technologies in the Internet of Things Environment*, Springer, Singapore, pp. 69–81, 2021.

[6] S. M. Hassan and S. A. A. Zaidi, "Urdu news headline text classification by using different machine learning algorithms," in *Proc. of Int. Conf. on Data Science*, pp. 4–8, Karachi Pakistan, 2019.

[7] A. Tripathy, A. Anand and S. K. Rath, "Document-level sentiment classification using hybrid machine learning approach," *Knowledge and Information Systems*, vol. 53, no. 3, pp. 805–831, 2017.

[8] Z. Zhou, J. Qin, X. Xiang, Y. Tan, Q. Liu *et al.*, "News text topic clustering optimized method based on tf-idf algorithm on spark," *Computers, Materials & Continua*, vol. 62, no.1, pp. 217–231, 2020.

[9] H. Wu, Y. Liu and J. Wang, "Review of text classification methods on deep learning," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1309–1321, 2020.

[10] A. Sarveniazi, "An actual survey of dimensionality reduction," *American Journal of Computational Mathematics*, vol. 2014, no. 4, pp. 55–72, 2014.

[11] R. Medar, V. S. Rajpurohit and B. Rashmi, "Impact of training and testing data splits on accuracy of time series forecasting in machine learning," in *Proc. of Int. Conf. on Computing, Communication, Control and Automation*, Pune, India, pp. 1–6, IEEE, 2017.

[12] M. Grandini, E. Bagli and G. Visani, "Metrics for multiclass classification: An overview," arXiv preprint arXiv: 2008.05756, Cornell university press, Ithaca, New York, USA, 2020.

[13] L. Huang, D. Milne, E. Frank and I. H. Witten, "Learning a concept-based document similarity measure," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 8, pp. 1593–1608, 2012.

[14] A. McCallum and K. Nigam, "A comparison of event models for naive Bayes text classification," in *Proc. of AAAI-98 Workshop on Learning for Text Categorization*, Madison, Wisconsin US, vol. 752, no. 1, pp. 41–48, 1998.

[15] G. Kumari, "A study of bagging and boosting approaches to develop meta-classifier," *An International Journal of Engineering Science and Technology*, vol. 2, no. 5, pp. 850–855, 2012.

[16] A. Kumara, S. Saumyab and J. P. Singha, "NITP-Ai-nLP@ urdu fake fIRE2020: Multi-layer dense neural network for fake news detection in urdu news articles," in *Proc. of FIRE 2020: Forum for Information Retrieval Evaluation*, Hyderabad, India, pp. 16–20, 2020.

[17] T. Chen and T. He, "Xgboost: Extreme gradient boosting," *R package version 0.4-2 1.4, Report published by regularizing gradient boosting framework*, Xgboost open source community, package version 1.4.1.1, pp. 1–3, 2021.

[18] R. H. Basit, M. Aslam, A. M. Martinez-Enriquez and A. Z. Syed, "Semantic similarity analysis of urdu documents," in *Proc. of Mexican Conf. on Pattern Recognition*, pp. 234–243, Berlin, Heidelberg, Springer, 2017.

[19] U. Khalid, A. Hussain, M. U. Arshad, W. Shahzad and M. O. Baig, "Co-occurrences using fast text embeddings for word similarity tasks in urdu," arXiv preprint arXiv: 2102.10957, Cornell university press, Ithaca, New York, USA, 2021.

[20] M. N. Asim, M. U. Ghani, M. A. Ibrahim, S. Ahmed, W. Mehmood *et al.*, "Benchmark per-formance of machine and deep learning based methodologies for urdu text document classification," arXiv e-prints, arXiv–2003, Cornell university press, Ithaca, New York, USA, 2020.

[21] J. Kaur and P. K. Buttar, "Stopwords removal and its algorithms based on different methods," *International Journal of Advanced Research in Computer Science*, vol. 10, no. 5, pp. 81–87, 2018.

[22] A. E. Waters, A. C. Sankaranarayanan and R. G. Baraniuk, "SpaRCS: Recovering low-rank and sparse matrices from compressive measurements," *Neural Information Processing Systems*, pp. 1089–1097, 2011.

[23] I. Guellil, A. Adeel, F. Azouaou, F. Benali, A. Hachani *et al.*, "A semi-supervised approach for sentiment analysis of arab (ic + izi) messages: Application to the Algerian dialect," *SN Computer Science*, vol. 2, no. 118, pp. 1–18, 2021.

[24] K. Irshad, M. T. Afzal, S. S. Rizvi, A. Shahid, R. Riaz *et al.*, "Swcs: Section-wise content similarity approach to exploit scientific big data," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 877–894, 2021.