Verify: Breakthrough accuracy in the Urdu fake news detection using Text classification

Santosh
Institute of Business Administration, Karachi
Sahtiya68@gmail.com

Dr. Zarmeen Nasim
Institute of Business Administration, Karachi
znasim@iba.edu.pk

Toto
Institute of Business Administration, Karachi
toto.14879@khi.iba.edu.pk

Parkash
Institute of Business Administration, Karachi
parkash.14910@khi.iba.edu.pk

Abstract

Researchers around the world have been struggling to minimize the rising spread of fake news through several Natural Language Processing techniques and a great amount of work has been done for resource-rich languages like English, French, German, Spanish, Chinese, etc. Alternatively, minimal research has been carried out on the Urdu language, which is spoken by millions of people around the globe. This study works on solving the problem of detecting the authenticity of Urdu news through Text analytics and Natural language processing methods. Upon studying the previously conducted research on text analysis and classification in Urdu and other resource-rich languages, it was found that machine translation does not work very effectively for authenticity due to compromises in structure, grammatical accuracy, and vocabulary. Hence, during this study, a Text analytics model has been developed on the only publicly available Urdu news articles dataset, originally composed in Urdu and comprising 900 articles, 500 real and 400 fake. During the preprocessing, stop words, English words, characters and numbers, and punctuations were removed which affected negatively the accuracy of the model. Apart from that, the data was lemmatized and tokenized and their effects on judging the authenticity of the news articles were examined to be a positive development. The supervised learning models include Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), Support Vector Machines (SVM), Logistic Regression (LR), Random Forests (RF), Decision Tree (DT), AdaBoost (AB), and XGBoost classifiers along with the combination of different sets of the word and character n-grams were applied to the data and their results were compared. As a result, it was found that the XGBoost classifier accompanied with the word unigram and 1-4 character n-grams generated 91% accuracy, the highest reported on this dataset so far.

Keywords: Fake news detection, Natural Language Processing, Media, Social Media, Urdu

1. Introduction

Fake news is defined as a factually incorrect news article, which aims at misguiding and misleading the readers. Researchers classify fake news into five categories (1) fabrication, (2) news satire, (3) manipulation (4) advertising (5) propaganda, and (6) news parody [4]. The events of the dissemination of fake news date back to the 15th century, centuries before the launch of mobile phones, social media, and the internet and have been reasons for many mishappenings among people of different countries, castes, communities, religions, and beliefs throughout the world.

With the development of digital and social media, information disseminates in seconds. People living in one corner of the world receive the information of another corner with just a click. This transformation in the methods of communication and speed of traveling of information has both pros and cons. Along with the spread of real information, it has been a medium for disseminating fake news quickly. Fake news has become a worldwide issue. People use it for several motives, like personal grudges, gains, and to change public opinion. It is perfectly evident from the scenario of the Covid-19 pandemic [2]. The circulation of fake stories about the victims, medical and homeopathic treatments, and the spread of the virus fueled panic among the people. WHO struggled to stop the dispersion of the fake Covid-19 stories related to the spread, symptoms, treatment, and prevention of the virus, but people in many countries followed the inbox forwarded methods and infected themselves with other diseases which led to panic in the public and disbelief in the competent authorities [2] [6].

For the prevention of dissemination of fake news, programmers and researchers apply National

Language Processing Techniques to identify the authenticity of news by identifying the characteristics, word choice, and writing patterns [1]. Extensive work has been done for high-resource languages like English because of the availability of the content, and literature [1][3]. Urdu, despite being spoken by more than 100 million speakers [1] around the world, is still considered a low-resource language because of the little literature and content available for Natural Language Processing Tasks [3]. This research aims at implementing Natural Language processing algorithms on Urdu scripted news and providing an open-source code as a resource for the confirmation of the authenticity of information for the Urdu-speaking public.

The unavailability of an authentic web source for the confirmation of the genuineness of the information in Urdu has caused several problems like Dr. Shahid Masood's imprisonment, and the Indian website's claim of a civil war in several cities of Pakistan. It has led to the arousal of conflicts within and outside and has deterred the country's image in national and international media [4]. Through this project, the researchers tend to solve the problems raised by the dissemination of false news and study the existing studies on fake news detection and NLP techniques applications in Urdu and other languages. This study examines the text classification techniques for confirming the trustworthiness of Urdu news articles, tries different sets of NLP techniques such as n-grams, lemmatization, etc, and compares their results, and in the result, it suggests the classification technique which serves the best in finding the genuineness of the content of an article of Urdu language.

2. Literature Review

There is little literature solely focused on fake news detection in the Urdu language. Hence, the researchers picked the commonly practiced NLP techniques for text processing and specifically for fake news detection in other renowned languages. In [1], for applying the NLP tasks, the data was cleaned by discarding the auxiliary character sequences and tokens, performing tokenization on words and characters, and the ramification of the sentences. For features, word n-grams from 1 to 6 and character n-grams from 1 to 6 were applied and the function words n-grams boosted their performances. A standard stop word list for Urdu was used as function words. For the binary classification of the news articles in [1], several methods were implemented which include Multinomial I Bayes, BernoINaive Bayes, Support Vector Machines, Logistic Regression, Random Forests, Decision Tree, and AdaBoost. The AdaBoost lent the maximum F1 score with particular combinations of character-word 2-grams and unigrams.

The team working on the study [1] continued their work in the study [3]. In this study, the dataset was enhanced by adding 400 news articles that were originally in the English Language and were machine translated into the Urdu Language by Google translate. Combined, the dataset contained 700 real news and 600 fake news articles, out of which 900 articles were used in the study [1]. The best classifiers from the study [1], SVM and AdaBoost were applied and it was found that the experimentation on the augmented dataset performed lesser than the original Urdu news articles because of the imperfect quality of the machine translation, which was also confirmed manually.

In 2020, the Center for Computing Research (CIC), Instituto Politécnico Nacional (IPN), Mexico conducted a fake news detection challenge for the Urdu Language, in which 39 teams participated. The teams had to perform the task with maximum accuracy on the dataset used in [1]. The team for [5] applied the generalized autoregressors technique for the binary classification task. They trained the XLNet model that uses the AR pre-training method and employs the use of language modeling objectives based on permutation. Their system reported an overall accuracy of 0.84 and an overall F1 macro score of 0.83.

The team from [1] and [3] also participated in this competition with the name BERT 4 EVER and stood first with the highest accuracy which led to the study [4]. In this task, the dataset was increased and more news articles were collected in the same way as done in the already available dataset from the study [1], adding 400 news articles, and bringing the test dataset to a total of 400 news articles and the train dataset to 900 news articles.

Furthermore, quoting the techniques used by the teams it must be mentioned that only one of the teams removed stop words during the preprocessing while other teams did not remove the stop words from the data, the teams used different techniques of text representation. Three of them used weighted tf-idf, three of them represented the articles using word embeddings while two teams applied different approaches of Word2Vec and FastText embeddings respectively. BERT4EVER used the contextual representation using BERT, which is a recent and advanced manner of text representation. To classify the corpus some teams used the classical non-neural algorithms while others' submissions consisted of various neural network architectures.

Among the submitted models, the team BERT 4 EVER outperformed the character bi-grams with logistic regression baseline achieving an F1-macro score of 0.90. This fact confirms that contextual representation and large neural network techniques perform better than the classical feature-based models [4]. It has also been shown in many recent studies in all branches of natural language processing.

Due to the unavailability of the originally labeled datasets in Urdu, the study [7] was conducted on English-translated news articles. The study was conducted on a translated QProp English language dataset containing 5,322 fake and 6,252 real news articles. On manual confirmation, it was found that around 95.4% of the translation was accurate. They introduce Propaganda Spotting in Online Urdu Language –ProSOUL) - a framework to identify content and sources of propaganda spread in the Urdu language. The team developed a Linguistic Inquiry and Word Count dictionary for the extraction of psycho-linguistic features of the Urdu Language. For the text representation, n-gram, NELA, word2Vec, and BERT features and the combination of word n-gram, character n-gram, and NELA features led to the best performance with 0.91 accuracies. In the comparison of the BERT features, Word2Vec performed better than BERT technique for the word embeddings [7].

In the studies [8], and [9] data mining techniques have been used for the detection of fake news by classifying the posts, and online reviews in publicly available corpora. The research team working on [9] achieved 99.7% accuracy by using a logistic classifier implemented in the browsers of the users. In [10], a rumor verification model has been proposed that achieves improved performance for veracity classification by leveraging task relatedness with auxiliary tasks, specifically rumor detection and stance classification, through a multi-task learning approach.

Similarly, the studies [10],[11],[12] have focused on the feature extraction from the text and then used those features in the classification models that include Logistic Regression, K-Nearest Neighbor, Random Forest, and Support Vector Machines. Study [10] reports on comparative style analysis of hyperpartisan (extremely one-sided) news and fake news. This study shows how a style analysis can distinguish hyperpartisan news from the mainstream (F1 = 0.78), and satire from both (F1 = 0.81). In [11], the researchers present a comprehensive review of detecting fake news on social media, including fake news characterizations on psychology and social theories, existing algorithms from a data mining perspective, evaluation metrics, and representative datasets.

Several studies have been conducted to find the relationship between the title and the content instead of classifying them into real or fake [13][14]. In the study [13], 73% accuracy was achieved, which was 26% higher than the previously conducted research by Excitement Open Platform[13]. For the study [14], the proposed approach is based on a maximum entropy classifier, which uses surface-level, sentiment, and domain-specific features represented in the Tweet Stance Detection task in SemEval 2016.

For this study, all these previously conducted fake news detection studies in Urdu and other languages were studied and the techniques were adopted for experimentation on the Urdu dataset used in this research.

3. Research Methodology

As displayed in figure 1, the workflow of the project was divided into several steps which include Dataset Collection, Data preprocessing, modeling, and evaluation. There were multiple substeps under these steps which further elaborated the diversity of the task and opened more doors to possible experimentations for further research. The classifiers and techniques were evaluated using accuracy, F1 real, and F1 Fake measures, and out of them, accuracy (Test score) was the deciding factor for the efficiency of a specific algorithm. Each step shown in the workflow is described as follows:

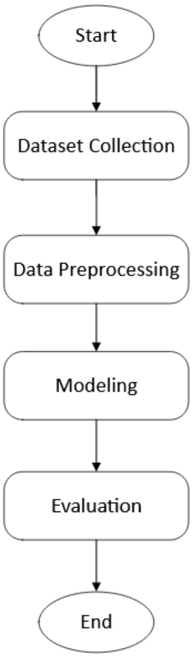

Fig. 1. Stepwise research methodology

3.1 Dataset Collection

The dataset for this study has been adopted completely from GitHub[1], originally posted by the team working on the study[1]. As mentioned above, it is the only labeled dataset available in the Urdu language, which helps to run the techniques and get the results with maximum efficiency, as translating the articles from another language deteriorates the performance. For the collection of the real news, the researchers of the study [1] crawled the data from trustworthy websites like BBC Urdu, Geo News, etc, or by verifying the news on multiple platforms. They crawled the articles using the Newspaper library of Python[2] which provides ease in dealing with the HTML, noisy texts, images, and advertisements on different web pages.

On the other hand, for gathering the fake news, professionals, and journalists were hired to write the fake news articles manually during the study [1]. It saved the team a great hustle of searching and verifying the fake news on online platforms. The journalists were directed to imitate the real news writing pattern so that there are no clear indications for the articles to be judged their authenticity easily[1].

The dataset contains a total of 900 news articles, 500 labeled real and 400, fake. The data has been divided into the train and test datasets by 638 and 262 articles respectively, where the train part is comprised of 350 real and 288 fake articles, and the test set is comprised of 150 real and 112 fake articles. The news in these articles belongs to five categories, (i) Business, (ii) Health, (iii) Showbiz,(iv) Sports, and (v) Technology. The category-wise distribution of the articles is displayed in Table 1.

Category	Train		Test	
	Real	Fake	Real	Fake
Business	70	36	30	14
Health	70	70	30	30
Showbiz	70	70	30	30
Sports	70	42	30	8
Technology	70	70	30	30

Table 1. Dataset distribution by category of news articles

3.2 Data Preprocessing

Before starting the classification experiments, several preprocessing and data quality-enhancing experiments were conducted to help increase the accuracy of the model. The following four tasks were implied.

1) Removing the English and Urdu Punctuation
2) Removing English and Urdu Numbers
3) Removing English words and characters
4) Removing stop words

These tasks were implemented individually as well as combined and the effect was studied.

3.3 Modeling

This section of the study discusses the experiments conducted on the dataset for solving the problem of identifying fake news by applying the various sets of classification algorithms along with different sets of character and word n-grams. Upon applying lemmatization and tokenization, the effects on the accuracy of all seven (7) classification models were studied. Each supervised learning model was tested against each set of character and word n-grams, and the results were compiled. Finally, the techniques which helped to reach the goal of enhancing the evaluation measures were used in the final classification model.

[1] https://github.com/MaazAmjad/Datasets-for-Urdu-news

[2] https://newspaper.readthedocs.io/en/latest/

3.3.1 Lemmatization

Lemmatization is a technique that reduces inflectional forms and sometimes derivationally related forms of a word to a common base form. It is carried out by using the vocabulary of a particular language and it returns the dictionary form of a word which is known as a lemma. For instance, a lemmatization process reduces the inflections, "am", "are", and "is", to the base form, "be". Sometimes the concept is interrelated or misunderstood with stemming, which is used to collapse the derivationally related words. Lemmatization on the datasets was performed by using the Stanza library of Python[3] which supports multiple languages including Urdu. Applying this technique helped the model to work efficiently, improve accuracy, and fester accurate results by decreasing the noise of the data, and using the contexts of the words in the dictionary form.

3.3.2 Tokenization (n-grams)

An n-gram is the sequence of n-items. In other words, it is the combination of adjacent words where n represents the number of items for example unigram represents 1 item, bigram represents 2 items, and so on. In this study, N-gram features are used to build fake news detection models such as character n-grams, and word n-grams. These character and word n-grams divide the words into characters and sentences u into words respectively, which help the model identify the trend and likeability of the proceeding word or characters. This way, the model of Natural Language Processing can identify the upcoming characters or words and their correspondence to specific positive or negative results. In other words, by using tokenization, the model can be trained on sets of corresponding characters and words to better identify or predict the similarities for test datasets, which leads to better identification of the subclass of the under-test item.

3.3.2.1 Word n-grams

Word n-grams represent the n number of words sequence, for example, consider the sentence “I am a Data Scientist”. For word unigram, it will be divided into, “I”, “am”, “a”, “Data”, and “Scientist”. During the experimentations, 1 and 2 sequences of words, namely word unigrams and word bigrams on the data were implemented and the results were examined for all the supervised learning algorithms.

3.3.2.2 Character n-grams

Similar to the word n-grams, the character n-grams represent the number of character sequences, for example, for the word “Data”, the character unigrams would be “D”, “a”, “t”, and “a”. In this study, researchers used 1,2,3,4,5,6-character n-grams individually as well as merged. These character n-grams were then tested against each classification algorithm accompanied by the word n-grams. The combination of different words n-grams and characters n-grams helped to get deeper insights into the data and suitable combinations for accomplishing the goal.

3.3.3 Classification models

In this research, multiple classification models and techniques were considered and their performance on the given word n-grams and character n-grams were examined for the detection of the authenticity of the news [1]. The classifiers include Multinomial Naive Bayes (MNB) [16], Bernoulli Naive Bayes (BNB) [16], Support Vector Machines (SVM) [17], Logistic Regression (LR) [21], Random Forests (RF) [22], Decision Tree (DT) [24], AdaBoost (AB), and XGBoost Classifier. The models have been described briefly as under.

3.3.3.1 Multinomial Naive Bayes (MNB)

It is a widely used classification model. Given a set of labeled data, the model often uses a parameter learning method called Frequency Estimate (FE), which computes appropriate frequencies from the data and calculates the probabilities of the words. The model is efficient for text classification and easy to implement.

3.3.3.2 Support Vector Machines (SVM)

An SVM classifier creates a maximum margin hyperplane that lies in transformed input space and splits the example classes while maximizing the distance to the nearest cleanly split examples. The parameters of the solution hyperplane are derived from a quadratic programming optimization problem [17].

[3] https://stanfordnlp.github.io/stanza/

3.3.3.3 Logistic Regression (LR)

Logistic regression is a predictive analysis like all regression analyses having the dichotomous (binary) dependent variable. It is used to describe data and explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval, or ratio-level independent variables [21]. It is more like a linear regression with added complexities of the cost function, which is known as sigmoid or logistic function.

3.3.3.4 Decision Tree (DT)

It is a supervised machine learning algorithm, which is used to classify the given record or to predict the outcome of regression problem. Generally, features in the data are placed on the non-leaf nodes while the branches contain the decision criteria. Every leaf node is a possible outcome of the problem [23]. Decision tree analysis is a divide-and-conquer approach to classification. They can be used to discover features and extract patterns in large databases that are important for discrimination and predictive modeling. Decision trees have an established foundation in both the machine learning and artificial intelligence literature and are slowly developing a niche in both the chemical and biochemical sciences [24].

3.3.3.5 Random Forests (RF)

Random forest is an ensemble of classifying and predictive machine learning algorithms used to solve more complex problems. It is a forest of many decision trees using bagging or bootstrap methods of aggregation. Random decision forests easily adapt to nonlinearities found in the data and therefore tend to predict better than linear regression. More specifically, ensemble learning algorithms like random forests are well suited for medium to large datasets.

3.3.3.6 AdaBoost (AB)

It is a boosting technique of machine learning which follows the sequence apply, boosts the previously learned model, and adapts. It repetitively follows this sequence and gets better and better results each time.

3.3.3.7 XGBoost Classifier.

XG Boost is a boosted version of the gradient boosting framework machine learning algorithm which is based on the decision tree. Each tree in the XG boost model boosts attributes that lead to the misclassification of the previous tree. The flexibility and speed of this technique provide it an edge over many algorithms in terms of efficiency, and validity.

3.3.4 Vectorizer

The tf-idf vectorizer was implemented on the data. A tf-idf vectorizer is a widely accepted technique for text vectorization, for Bag of Words representation. Each document is represented as a vector and the terms in the document are represented by the fields. In tf-idf, tf represents the number of times a term has appeared in a document and the idf denotes how informative the term is. The higher the repetition of a term across the documents the lesser informative it is considered to be. The tf-idf of a term is calculated as

$$TF.IDF = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \qquad Eq.1$$

Where $tf_{i,\,j}$ = Number of occurrences of i in document j

df_i = Number of documents containing i
N = Total number of documents

3.4 Evaluation

As mentioned previously, multiple classification algorithms along with the combination of word and character n-grams were applied to the data, and their effects were studied. Test score (Accuracy), F1 Real, F1 Fake, Precision, and recall. Out of these parameters, accuracy (Test Score) was the main metric that denotes the efficiency of the classifier along with the combination of the word and character n-grams. As the previously conducted studies on fake news detection also judge the efficacy of the algorithms based on the test score, hence the results of the proposed approach can be compared to them.

The F1 score is called the harmonic mean of precision and recall. Precision and recall are metrics of performance more suitable for imbalanced data because they allow taking into account the type of errors (false positives or false negatives) that the model makes

The accuracy score can be calculated by:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \qquad Eq.2$$

4. Result Analysis

This section tends to describe the results of all the experimented tools and techniques, and their effect on the accuracy of the model. More than 30 test experiments were performed which catered to almost all the combinations of preprocessing, word and character n-grams, and classified learning algorithms. For each step, the efficacy on the model was compiled, and the next approach was decided based on the previous results. All the experimentations carried out on the data are explained as under.

4.1. Preprocessing

From the results of preprocessing, it could be found that removing the punctuation, English and Urdu numbers, the English words and characters, and stop words would be a bad decision as it led to lesser accuracy in the experiments. Further, after extracting the features, it was found that the fake to real news punctuation count ratio was around 1:2, as displayed in Table 2.

Target	Word Count	Unique word	Stop words	Mean word Length	Character Count	Punctuation count	English character count
0	355.66	178.42	174.24	3.74	1682.39	16.54	8.37
1	298.8	155.51	150.91	3.62	1374.28	8.9	1.55

Table 2. The features extraction of the real and fake news articles (Target = 0 (Real news), Target = 1 (Fake News)

The features extracted from the data proved to be very helpful in recognizing the authenticity of the news for example the significant difference in the punctuation count and English characters count, as in fig 2. Also, the number of stop words in the real news articles was around 17.1% more than that of the fake news articles. Removing the punctuations or these characters would have blurred the difference between the two.

4.2 Lemmatization

After performing the lemmatization, the results were studied by checking the accuracy of multiple classifiers, and it was found that lemmatization affected negatively the accuracy of the models. So, after the confirmation, the dataset was not lemmatized for further evaluation as it would have deteriorated the model's efficiency to find out the authenticity of the articles in the dataset.

4.3 Tokenization

Classifier	Word Grams	Char Grams	Total Features	Accuracy (TestScore)	F1 Real	F1 Fake
AdaBoost	0	2,3,4,5	317276	0.83	0.86	0.78
RandomForest	0	2,3,4,5	317276	0.82	0.86	0.75
RandomForest	1	2, 3, 4, 5	327987	0.81	0.85	0.71
AdaBoost	1, 2	3, 4	199709	0.79	0.82	0.74
AdaBoost	1	3, 4, 5	327987	0.78	0.82	0.72
AdaBoost	1	2, 3, 4, 5	327987	0.77	0.8	0.72
RandomForest	1	3, 4, 5	327987	0.74	0.8	0.62
AdaBoost	1	0	11569	0.74	0.78	0.66
RandomForest	1, 2	3, 4	199709	0.74	0.8	0.62
RandomForest	1	0	11569	0.72	0.79	0.56
RandomForest	1, 2	0	78712	0.72	0.79	0.57
AdaBoost	1, 2	0	78712	0.71	0.77	0.62
NaiveBayes	1	0	11569	0.59	0.74	0.03
NaiveBayes	1	3, 4, 5	327987	0.58	0.73	0.00
NaiveBayes	0	2,3,4,5	317276	0.58	0.73	0.00
NaiveBayes	1	2, 3, 4, 5	327987	0.58	0.73	0.00
NaiveBayes	1, 2	0	78712	0.58	0.73	0.00
NaiveBayes	1, 2	3, 4	199709	0.58	0.73	0.00

Table 3. Classification models and their performances based on different sets of word n-grams and character n-grams

Different sets of character and word n-grams were applied belonging to each binary classification algorithm. From the experiments, it was found that applying tokenization improved the model’s capability to confirm the real/fake nature of the given part of the data, and was counted as a positive development, as evident in Table 3.

4.4 Classification

The research found that the boosting classification algorithms performed the best as compared to other classifiers, with the XGBoost classifier outperforming all the classifiers with a combination of 1,2,3 word n-grams and 1,2,3,4 character n-grams and an accuracy of 91%, F1 Real Score and F1 fake score equals to 0.93 and 0.89 respectively. These scores are the highest scores achieved so far. AdaBoost classifier got the second position in achieving the goal. Multiple character n-gram and word n-gram were implemented on the XGBoost classifier. The top five best-performing word n-gram and character n-gram features combination are represented in table 4.

Classifier	**Word Grams**	**Char Grams**	**Total Features**	**Accuracy (Test Score)**	**F1 Real**	**F1 Fake**
XGB	1,2,3	1,2,3,4	376318	0.91	0.93	0.89
XGB	1	1,2,3,4	151945	0.89	0.9	0.85
XGB	0	1,2,3	19756	0.8	0.84	0.74
XGB	1,2	1,2,3,4	181254	0.8	0.83	0.75
XGB	0	1,2	1539	0.79	0.83	0.72

Table 4. Word n-grams, character n-grams, and the relevant score in metrics for XGBoost Classifier

5. Conclusion and Future Work

Fake news is a major problem in today’s world. The detection of fake news is a promising task for all languages to help people be aware of the truth. Fake news dispersal is a serious issue that needs to be addressed and tackled in many languages. Researchers have proposed different data mining techniques for detection of the fake news in multiple international languages. But, there is a large room for improvement and creativity in the Urdu language which is spoken by a large community. The availability of very few previous studies for fake news detection in the Urdu language is a major limitation in conducting this research. Hence this study also adopts the methods and techniques previously applied for fake news detection in other renowned resource-rich languages.

This research contributed to filling this gap to the best possibilities by classifying the news articles of the first-ever labeled dataset in the Urdu language. The study applied Multinomial Naive Bayes (MNB), Bernoulli Naive Bayes (BNB), Support Vector Machines (SVM), Logistic Regression (LR), Random Forests (RF), Decision Tree (DT), AdaBoost (AB), and XGBoost classifiers along with the combination of different sets of the word and character n-grams. The proposed approach in this study achieved 91% accuracy, the best accuracy with the highest scores so far using the XGBoost classifier with the combination of word unigram and 1-4 character n-grams.

This study attempts to fulfill the gap in Urdu fake news detection to the best possibilities and create room for further research on feature extraction techniques and classifiers. The boosting algorithms and n-grams used in this study can be researched further to enhance the accuracy of the model. Apart from this, the enhancement of the dataset can be a major add-on that will help to deeply learn the characteristics and the writing patterns of authentic and unauthentic writings. This way, it will be easier to distinguish between the truth and the lie.

References

Amjad, M., Sidorov, G., Zhila, A., Gómez-Adorno, H., Voronkov, I., & Gelbukh, A. (2020). “Bend the truth”: Benchmark dataset for fake news detection in Urdu language and its evaluation. Journal of Intelligent & Fuzzy Systems, 1–13. doi:10.3233/jifs-179905

Destiny Apuke, O., & Omar, B. (2020). Fake News and COVID-19: Modelling the Predictors of Fake News Sharing Among Social Media Users. Telematics and Informatics, 101475. doi:10.1016/j.tele.2020.101475

Amjad, M., Sidorov, G., & Zhila, A. (2020). Data Augmentation using Machine Translation for Fake News Detection inthe Urdu Language. Proceedings of the 12th Conference on Language Resources and Evaluation.

Amjad, M., Sidorov, G., Zhila, A., Gelbukh, A., & Rosso, P. (2020). Overview of the Shared Task on Fake NewsDetection in Urdu at FIRE 2020. Center for Computing Research (CIC), Instituto Politécnico Nacional (IPN), Mexico.

Khiljia, A. F., Laskara, S. R., Pakraya, P., & Bandyopadhyaya, S. (2020). Urdu Fake News Detection using Generalized Autoregressors. aDepartment of Computer Science and Engineering, National Institute of Technology Silchar, Assam, India.

Jahangir, R. (2020, March 28). Dawn. Desi totkas and fake news — a guide to surviving the Covid-19 'infodemic' . Dawn. Retrieved from Dawn.com: https://www.dawn.com/news/1544256/desi-totkas-and-fake-news-a-guide-to-surviving-the-covid-19-infodemic

KAUSAR, S., TAHIR, B., & MEHMOOD, M. ,. (2020). ProSOUL: A Framework to Identify PropagandaFrom Online Urdu Content. IEEE Access.

V. P´erez-Rosas, B. Kleinberg, A. Lefevre and R. Mihalcea, Automatic Detection of Fake News, in: Proceedings of the 27th International Conference on Computational Linguistics, Association for Computational Linguistics, Santa Fe, New Mexico, USA, (2018), pp. 3391–3401.772https://www.aclweb.org/anthology/C18-1287.773[8]

M. Aldwairi and A. Alwahedi, Detecting Fake News in Social Media Networks,Procedia Computer Science141(2018), 215–222. https://linkinghub.elsevier.776com/retrieve/pii/S1877050918318210.

M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff and B. Stein, A Stylometric Inquiry into Hyperpartisan and Fake News, in:Proceedings of the 56th. Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, (2018), pp. 231–240.

K. Shu, A. Sliva, S. Wang, J. Tang and H. Liu, Fake News Detection on Social Media: A Data Mining Perspec-tive, ACM SIGKDD Explorations Newsletter19(1) (2017), 22–36.

J.P. Posadas-Dur´an, H. G´omez-Adorno, G. Sidorov and J. Jaime Moreno Escobar, Detection of Fake News in a New Corpus for the Spanish Language, Journal of Intelligent & Fuzzy Systems(2018).

W. Ferreira and A. Vlachos, Emergent: a Novel Data-Set for Stance Classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL' 2016, (2016), pp. 1163–1168.804.

P. Krejzl, B. Hourov´a and J. Steinberger, Stance Detection in Online Discussions,arXiv preprint arXiv:1701.00504806(2017)

Reis, J. C. S., Correia, A., Murai, F., Veloso, A., Benevenuto, F., & Cambria, E. (2019). Supervised Learning for Fake News Detection. IEEE Intelligent Systems, 34(2), 76–81. doi:10.1109/mis.2019.2899143

Wang, S., Jiang, L. & Li, C. Adapting naive Bayes tree for text classification. Knowl Inf Syst 44, 77–89 (2015). https://doi.org/10.1007/s10115-014-0746-y

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. IEEE Intelligent Systems and Their Applications, 13(4), 18–28. doi:10.1109/5254.708428

Shmilovici, A. (2009). Support Vector Machines. Data Mining and Knowledge Discovery Handbook, 231–247. doi:10.1007/978-0-387-09823-4_12

Dey, A., Jenamani, M., & Thakkar, J. J. (2017). Lexical TF-IDF: An n-gram Feature Space for Cross-Domain Classification of Sentiment Reviews. Pattern Recognition and Machine Intelligence, 380–386. doi:10.1007/978-3-319-69900-4_4

Bhattacharjee, U., P.K, S., & Desarkar, M. S. (2019). Term Specific TF-IDF Boosting for Detection of Rumours in Social Networks. 2019 11th International Conference on Communication Systems & Networks (COMSNETS). doi:10.1109/comsnets.2019.8711

Chao-Ying Joanne Peng , Kuk Lida Lee & Gary M. Ingersoll (2002) An Introduction to Logistic Regression Analysis and Reporting, The Journal of Educational Research, 96:1, 3-14, DOI: 10.1080/00220670209598786

Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. The Stata Journal: Promoting Communications on Statistics and Stata, 20(1), 3–29. doi:10.1177/1536867x20909688

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE Transactions on Systems, Man, and Cybernetics, 21(3), 660–674. doi:10.1109/21.97458

Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., & Brown, S. D. (2004). An introduction to decision tree modeling. Journal of Chemometrics, 18(6), 275–285. doi:10.1002/cem.