engineering proceedings

MDPI

Proceeding Paper

Comparative Analysis of TF–IDF and Hashing Vectorizer for Fake News Detection in Sindhi: A Machine Learning and Deep Learning Approach †

Rubab Roshan *, Irfan Ali Bhacho and Sammer Zai

Department of Computer Systems Engineering, Mehran University of Engineering and Technology (MUET), Jamshoro 76062, Pakistan; irfan.ali@faculty.muet.edu.pk (I.A.B.); sammer.zai@faculty.muet.edu.pk (S.Z.)
* Correspondence: rubabkhaskheli18@gmail.com
† Presented at the 8th International Electrical Engineering Conference, Karachi, Pakistan, 25–26 August 2023.

Abstract: Social media has become a popular platform for accessing and sharing news, but it has also led to a rise in fake news, posing serious risks. The ease of dissemination and constant flow of information raise concerns about the spread of incorrect information. Timely verification of news is crucial to combat false news. However, most research on false news identification has focused on English, neglecting South Asian languages. This study examines a dataset of Sindhi tweets, employing text feature extraction techniques such as TF–IDF and hashing vectorizer. Several machine learning algorithms, along with advanced deep learning models such as Transformer BERT, were utilized for analysis.

Keywords: machine learning; deep learning; BERT; text mining; TF–IDF; hashing vectorizer; Sindhi NLP; NLP; social media; Twitter

Citation: Roshan, R.; Bhacho, I.A.; Zai, S. Comparative Analysis of TF–IDF and Hashing Vectorizer for Fake News Detection in Sindhi: A Machine Learning and Deep Learning Approach. *Eng. Proc.* **2023**, *46*, 5. https://doi.org/10.3390/engproc2023046005

Academic Editors: Abdul Ghani Abro and Saad Ahmed Qazi

Published: 20 September 2023

1. Introduction

Fake news is information that is false or fraudulent that may originate through traditional media channels or online platforms. The advent of social media and other digital platforms has resulted in an unprecedented surge in the dissemination of erroneous data, endangering society on an enormous scale. Online false news has the potential to mislead and misinform people, frequently with major political and societal implications. It has the potential to negatively alter people's perceptions of events, public figures, and organizations, resulting in polarization and a breakdown of trust.

Given its real-time nature along with broad reach, Twitter contributes a key part in the propagation of fake news. The platform's ease of sharing information along with the lack of control and verification processes makes it a breeding ground for the spread of incorrect information.

The Sindhi Language is the language of approximately 37 million speakers globally. The language has a rich literary legacy dating back to the 8th century that is distinguished by its diversity and uniqueness. It has had many scripts throughout the history, but presently Perso-Arabic (فارسيعربي) and Devanagari (ديوناگري) are two most widely used scripts internationally. Many native folks utilize Sindhi on social media to share information and express themselves. However, Sindhi information, like that of other languages, suffers from the dissemination of misinformation, which is typically driven by personal gain, political ambitions, or fun. Nonetheless, due to the language's limited resources, detecting fake news in Sindhi poses a substantial challenge.

2. Related Works

S. Singh et al. [1] highlights a concern about the pernicious effect of web-based entertainment on people and society, owing to the spread of fake news. Machine learning

algorithms are required because traditional manual filtering techniques have failed to identify and eradicate this problem. They used various sophisticated machine-learning methods to identify and address fake news. The SVM classifier algorithm achieved a notable accuracy rate.

I. Ahmad et al. [2] explores different textual properties which can help extricate real news from fake news and train an amalgamation of machine learning algorithms using various ensemble methods. The proposed approach is evaluated on four real-world datasets, and the experimental results indicate its superiority over individual learners.

Haseeb Ur Rehman and S. Hussain [3] found widespread fake news on Pakistani social media, particularly Twitter and Facebook. False stories on politics and international relations received attention even after being debunked, indicating the influence of cult followings and populism.

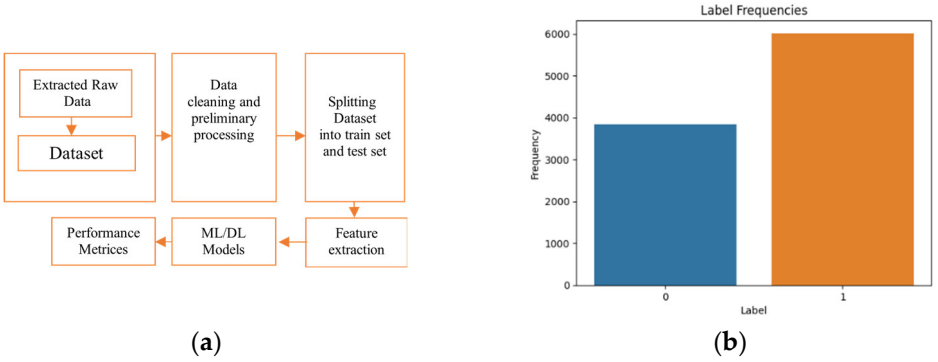
W. Y. Wang [4] presents a new dataset called LIAR for automatic fake news detection. The authors used superficial language attributes to investigate automatic fake news detection and propose a hybrid CNN. They highlight the importance of labelled benchmark datasets for combating fake news.

P. H. A. Faustini and T.F. Covões [5] proposed a model to detect fake news based on text features that can be applied to different language groups. They evaluated their approach on five datasets, including social media posts, and achieved competitive results. Support Vector Machines and Random Forests performed best in classification, and the bag-of-words approach achieved the best results overall.

J. C. S. Reis et al. [6] used multiple classifiers, such as KNN, Naive Bayes, Random Forests, SVM, and XG-Boost, to assess the efficacy of various hand-crafted characteristics for spotting false news. The classifiers' AUC and Macro F1-scores are used to evaluate them, and RF and XGB classifiers produce the best results. According to the ROC curve for XGB, 40% of the actual news can be misclassified, while practically all fraudulent news can be classified accurately.

3. Proposed Methodology

This study proposes a methodology to evaluate the effectiveness of two feature extractors in combination with machine learning (ML) and deep learning (DL) models for identifying real and fake news in Sindhi on Twitter. Figure 1a illustrates the overview of our methodology. The methodology involves creating a dataset of Sindhi language tweets, cleaning and labeling them, extracting features, training and testing ML and DL models, and then computing performance metrics, such as accuracy, precision, recall, and an F1-score to assess the efficacy of the feature extraction methods.



(**a**) (**b**)

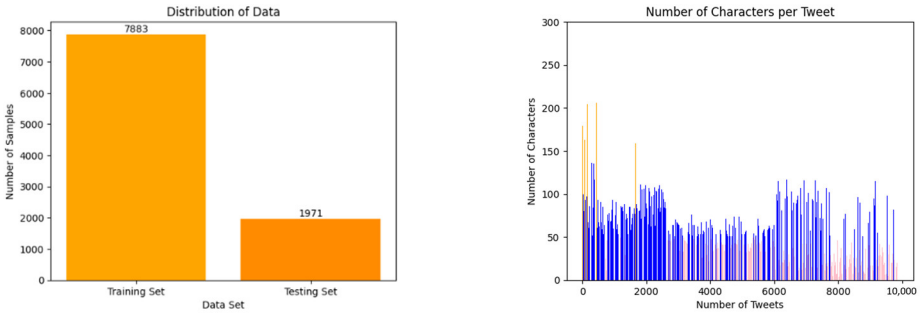
Figure 1. (**a**) Overview of the workflow for proposed methodology. (**b**) Distribution of dataset into real and fake tweets; blue represents fake news and orange represents real news.

4. Construction of Dataset

Sindhi, being a language with limited NLP resources, ref. [7] posed a challenge. To address this, data were collected by scraping Twitter using the Tweepy API. Real news

tweets were gathered from Sindhi news network accounts, while fake news tweets were crowdsourced for the development of a label dataset.

The dataset comprised 9854 tweets. Figure 1b shows the distribution of the dataset into real and fake news, while Figure 2b illustrates the tweet length across the dataset. The dataset exhibited an imbalance, with approximately one thousand tweets in the "Fake" category representing conversational tweets, which are not considered news.



(**a**) (**b**)

Figure 2. (**a**) Splitting of real and fake news tweets into train and test datasets; orange for train and tangerine for test set. (**b**) Distribution of length of tweets across the dataset.

5. Data Cleaning and Pre-Processing

Data pre-processing includes data cleaning as well as the preliminary processing of the dataset comprised several steps that are enumerated below:

- Elimination of superfluous white spaces.
- Removal of #Hashtags, username tags, and retweet tags
- Elimination of symbols: ([|@#/:.!"#$%& () *+,\-.\/:;<=>?@\ [\]^_'{|}~]).
- Removal of all except textual data.
- Elimination of duplicate and English language tweets
- Stop-words removal.

Stop words are frequently used terms in a language that are eliminated from text data because they are ineffective for NLP activities.

6. Feature Extraction

Feature extraction entails selecting the most relevant features from the data to train a model. There are different methods for obtaining features. These methods decrease the dimensionality of the data and capture the most pertinent features [8].

6.1. Term Frequency–Inverse Document Frequency (TF–IDF)

TF–IDF is a feature extraction technique in NLP that measures the importance of a word in a document based on its frequency in the document and the dataset. It assigns weights to words based on their frequency in the document and inversely to their frequency in the corpus.

6.2. Hashing Vectorizer (HV)

HV is a text feature extraction technique used in NLP tasks to convert text files into a matrix of token occurrences. It uses a hash function to assign indices to words, allowing each word to be processed independently. This scalability makes it suitable for large databases.

7. Machine Learning

ML algorithms are used to recognize patterns, generate predictions, and are employed in an array of applications, including text classification, where producing results for required tasks using traditional algorithms is challenging or unattainable [9]. For text

classification, we used SVM, Nave Bayes (nB), neural networks (NN), logistic regression (LR), CatBoost (CB), decision trees (DT), AdaBoost (AB), and random forests (RF).

SVM identifies an optimal hyperplane to separate data into distinct classes while maximizing the margin between them. Nave Bayes employs Bayes' theorem with a feature independence assumption. A neural network learns intricate patterns from data through iterative training. Logistic regression estimates the probability of an instance belonging to a particular class by applying a logistic function to a linear combination of features. CatBoost achieves high predictive accuracy by intelligently managing feature interactions and employing an innovative learning scheme. Random Forest (RF) achieves higher accuracy and robustness by mitigating overfitting and capturing diverse feature interactions.

8. Deep Learning

Deep learning models are machine learning approaches that use artificial neural networks (ANNs) to extract meaningful patterns and insights from data. To evaluate our dataset, we used CNN, RNN, LTSM, and Transformers. CNN-based models are trained to identify patterns in text, RNN-based models consider text as a sequence of words, and Transformer models capture long-range dependencies between words or tokens in a sentence [10].

9. Performance Evaluation

Performance evaluation metrics are utilized to assess the efficacy and performance of machine learning models or algorithms. These metrics offer quantitative measures to evaluate the model's proficiency in addressing a specific task.

9.1. Accuracy

It quantifies the overall correctness of predictions by computing the ratio of correctly predicted instances to the total number of instances.

$$\text{Accuracy (A)} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{True Negative} + \text{false Positive} + \text{False Negative}} \quad (1)$$

9.2. Precision

It measures the proportion of true positive predictions among the total predicted positive instances, focusing on the precision of positive predictions.

$$\text{Precision (P)} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (2)$$

9.3. Recall

It calculates the ratio of true positive predictions to the actual positive instances, emphasizing the model's ability to correctly identify positive instances.

$$\text{Recall (R)} = \frac{\text{True Positive}}{\text{True positive} + \text{False nagitive}} \quad (3)$$

9.4. F1-Score

It represents the harmonic mean of precision and recall, offering a balanced measure that combines both metrics. The F1-score is particularly useful when dealing with class imbalances within the dataset.

$$\text{F1-score (F1-S)} = \frac{2 \times (\text{P} \times \text{R})}{\text{P} + \text{R}}. \quad (4)$$

10. Experiments and Results

The proposed methodology underwent rigorous evaluation using a diverse range of machine learning and deep learning techniques. In the subsequent sections, we provide detailed elaboration on each technique along with the corresponding results attained from our evaluation efforts.

For this evaluation, we implemented all the machine learning algorithms in a similar fashion with both TF–IDF as well as HV. The results, after calculating performance metrics, can be seen in Table 1. We also generated our results of ensemble ML models of Bagging, Boosting, and Stacking.

Table 1. Performance evaluation results of ML algorithms using TF–IDF and hashing vectorizer.

ML Algo	Precision		Recall		F1-Score		Accuracy	
	TF–IDF	HV	TF–IDF	HV	TF–IDF	HV	TF–IDF	HV
AB	0.92	0.84	0.92	0.79	0.92	0.77	0.9193	0.7940
Bagging	0.98	0.98	0.98	0.98	0.98	0.98	0.9781	0.9766
Bnb	0.98	0.76	0.98	0.60	0.98	0.37	0.9812	0.6017
Boosting	0.93	0.94	0.93	0.94	0.93	0.94	0.9330	0.9386
CB	0.95	0.95	0.95	0.95	0.95	0.95	0.9477	0.9512
DT	0.89	0.90	0.89	0.90	0.89	0.90	0.8924	0.9061
GB	0.90	0.90	0.90	0.90	0.90	0.90	0.8985	0.9061
KNN	0.76	0.76	0.43	0.46	0.30	0.36	0.4322	0.4677
LR	0.96	0.95	0.96	0.96	0.96	0.96	0.9629	0.9599
MnB	0.96	0.88	0.96	1.00	0.96	0.93	0.9604	0.9183
NN	0.98	0.98	0.98	0.98	0.98	0.98	0.9817	0.9822
RF	0.96	0.96	0.96	0.96	0.96	0.96	0.9583	0.9583
Stacking	0.98	0.98	0.98	0.98	0.98	0.98	0.9756	0.9731
SVM	0.97	0.97	0.97	0.97	0.97	0.97	0.9710	0.9756
XGB	0.91	0.91	0.91	0.91	0.91	0.91	0.9091	0.9117

The best performing algorithm across both feature extraction techniques is NN (neural network) because of its capability to learn complex patterns in text data, its robustness to noise, and its strong generalization ability. SVM (Support Vector Machines) and Logistic Regression also demonstrated good overall performance with both feature extraction techniques, benefiting from their ability to handle linear separability, noisy data, and sparse representations. The top-performing ensemble method was Bagging, which combines multiple models trained on different subsets of the training data to reduce variance, improve stability, and enhance generalization.

Upon analyzing the results presented in Table 1, it can be concluded that both feature extraction techniques yielded favorable outcomes with most algorithms, except for KNN (K-Nearest Neighbors). TF–IDF outperformed the hashing vectorizer in terms of accuracy in 6 out of 15 instances, while the hashing vectorizer outperformed TF–IDF in 7 out of 15 instances. However, considering the overall performance metrics of Precision, Recall, and F1-Score, TF–IDF showcased superior performance compared to hashing vectorizer for our specific dataset. This is attributed to TF–IDF's ability to capture the significance of important and distinctive words in the language. The lower performance of KNN could be attributed to the choice of the value of K, which in our case was set to K = 5.

The dataset was further evaluated using deep learning models and results can be seen in Table 2. After the analysis of Table 2 we can conclude that RNN outperformed CNN with both feature extractors, because RNNs are designed to process sequential data by

maintaining an internal memory that enables them to capture long-term dependencies in the text. Over-all, we can conclude that models engineered with TF–IDF have generally higher accuracy and better performance over hashing vectorizer unless we use a Transformer model like BERT (Bidirectional Encoder Representations from Transformers) that is a pre-trained language model that can learn useful features from raw input text with or without feature engineering [11]. To address suboptimal performance in our original dataset, since BERT is not trained in Sindhi, we used the Google Trans Library to translate the dataset into English and fine-tuned it.

Table 2. Performance evaluation results of DL Models using feature extraction and BERT without any feature extractor.

Feature Extractor	DL Model	Precision	Recall	F1-Score	Accuracy
TF–IDF	CNN	0.64	0.65	0.57	0.6377
	RNN	0.98	0.98	0.98	0.9761
	LTSM	0.98	0.98	0.98	0.9751
Hashing Vectorizer	CNN	0.75	0.74	0.72	0.7787
	RNN	0.97	0.97	0.97	0.9718
	LTSM	0.97	0.97	0.97	0.9727
None	BERT (Original Data)	0.36	0.60	0.45	0.6012
	BERT (Translated Data)	0.9497	0.9498	0.9497	0.9498

11. Conclusions

In this research study, we extracted our data from Twitter, cleaned, pre-processed, and then analyzed it applying various machine learning algorithms, deep learning models, and the Transformer model BERT. We performed our experiments twice for each algorithm/model, first with TF–IDF and then with hashing vectorizer, to ascertain which feature engineering technique yielded the best performance for the Sindhi text dataset. Our results demonstrate that TF–IDF often performed better than hashing vectorizer for our dataset for both machine learning and deep learning models. We achieved the highest machine learning accuracy with a neural network (NN) algorithm, RNN for deep learning and with a translated dataset on BERT. This indicates the significance of selecting an appropriate feature engineering technique for text data pre-processing to obtain optimal results. These findings help facilitate the development of effective techniques for processing and interpreting Sindhi text data on social media platforms.

12. Limitations and Future Works

This research investigation examined Twitter news content in the Sindhi language, encompassing both real and fake news. Unfortunately, limited resources posed challenges in creating a well-balanced dataset, leading us to supplement fake news entries through crowdsourcing, potentially impacting data diversity, thus making it easier for classifiers and models to detect. Additionally, relying on Google Translate for automatic language translation during BERT fine-tuning proved suboptimal.

Our future endeavors involve gathering a more extensive corpus of Sindhi fake news from Twitter, enabling the construction of a balanced dataset for reassessment.

Author Contributions: Conceptualization, R.R., I.A.B. and S.Z.; methodology, R.R., I.A.B. and S.Z.; software, R.R.; validation, R.R., I.A.B. and S.Z.; formal analysis, R.R.; investigation, R.R.; resources, R.R. and S.Z.; data curation, R.R.; writing—original draft preparation, R.R.; writing—review and editing, R.R.; visualization, R.R.; supervision, I.A.B. and S.Z.; project administration, R.R.; funding acquisition, R.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Singh, S.; Sharma, C.; Agarwal, S.; Garg, U.; Gupta, N. The Detection and Analysis of Fake News Using Machine Learning. In *2022 International Conference on Fourth Industrial Revolution Based Technology and Practices (ICFIRTP)*; IEEE: Uttarakhand, India, 2022; pp. 19–24.
2. Ahmad, I.; Yousaf, M.; Yousaf, S.; Ahmad, M.O. Fake News Detection Using Machine Learning Ensemble Methods. *Complexity* **2020**, *2020*, 8885861. [CrossRef]
3. Rehman, H.U.; Hussain, S. Social Media, Democracy and Fake News in Pakistan: An Analysis. *Glob. Polit. Rev.* **2020**, *V*, 84–93. [CrossRef] [PubMed]
4. Wang, W.Y. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 422–426.
5. Faustini, P.H.A.; Covões, T.F. Fake News Detection in Multiple Platforms and Languages. *Expert Syst. Appl.* **2020**, *158*, 113503. [CrossRef]
6. Reis, J.C.S.; Correia, A.; Murai, F.; Veloso, A.; Benevenuto, F. Supervised Learning for Fake News Detection. *IEEE Intell. Syst.* **2019**, *34*, 76–81. [CrossRef]
7. Dootio, M.A.; Wagan, A.I. Development of Sindhi Text Corpus. *J. King Saud Univ.-Comput. Inf. Sci.* **2021**, *33*, 468–475. [CrossRef]
8. Iwendi, C.; Mohan, S.; Khan, S.; Ibeke, E.; Ahmadian, A.; Ciano, T. Covid-19 Fake News Sentiment Analysis. *Comput. Electr. Eng.* **2022**, *101*, 107967. [CrossRef]
9. Hua, T.K. A Short Review on Machine Learning. *Authorea* **2022**. [CrossRef]
10. Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; Gao, J. Deep Learning--Based Text Classification: A Comprehensive Review. *ACM Comput. Surv.* **2022**, *54*, 1–40.
11. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2019**, arXiv:1810.04805.