Fake News Detection in Dravidian Languages Using Transformers and Ensembles

Rajalakshmi Sivanaiah, Angel Deborah S, Thirumurugan RA, Vishal Muralidharan, and Ananya Raman

Department of Computer Science and Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai, Tamil Nadu, India
{rajalakshmis, angeldeborahs}@ssn.edu.in
{thirumurugan2310277, vishal2310253, ananya2310278}@ssn.edu.in

Abstract. False information and misleading content can have disastrous consequences, even posing threats to lives. One particularly pervasive form of this is fake news, which has surged with the growth of the social media. The rapid increase in the spread of misleading information is alarming, as it travels much faster than real news since the source of the fake news cannot be identified, leading to potentially devastating outcomes. Preventing the spread of misleading information is crucial to mitigating these dangers. This paper shows how Ensemble techniques and BERT models can be employed to detect fake news for languages with limited resources, including Tamil, Malayalam, Telugu, and Kannada . The experimentation shows that BERT-base-multilingual-cased and Stacking Techniques reach the highest accuracies of approx. 96% and 90% respectively for all the languages

Keywords: Fake news · Resource-scarce Languages · Ensemble Techniques · BERT · Dravidian Languages

1 Introduction

In this rapidly growing age and time of technology where everyone from toddlers to aged people have access to basic technology and more importantly, the internet, communication has become almost instant and information is being shared and received within a second. Even though messages can be sent and have instant communication, it's legitimacy is seldom questioned. This causes a lot of misinformation to be spread and people believing it blindly as they just read the gist of the news and ignore the description given to it. According to a 2020 study by Microsoft, India has one of the highest rates of exposure to fake news, with 64 percent of Indians reporting they had encountered false information online. The same people continue to spread this information to other people of their community, family, friends and more which causes a huge and rapid misinformation spread. WhatsApp, being one of the most popular messaging apps in India, has been a significant medium for the spread of misinformation. The app has been used to circulate false information during elections, communal riots, and health crises.

The worse part of this is that many news channels, social media platforms, influential people and internet pages as they want to spread their own opinions or ideologies to others or portray the same fake news as they don't cross-check the integrity of the news that they receive and focus more on bringing 'exclusive' news as soon as possible to their audience and th1is causes more people to trust in the news they are seeing. A 2020 study by the Oxford Internet Institute found that even political parties in India were involved in organized social media manipulation, using bots and fake accounts to spread misinformation and influence public opinion.

Many resolutions, rules, policies, organisations, methods and techniques are being used by the government, the social media platforms and all other internet-using entities to maintain the integrity of data as much as possible. In India specifically, there have been a rise in the number of fact-checking organizations, such as Alt News, Boom, and Factly, which work to debunk misinformation. The Indian government has also recently introduced the IT Rules (2021) which require social media platforms to take responsibility for the content shared on their platforms and they must remove illegal or harmful content within 36 hours of receiving a government or court order.

Machine Learning is increasingly used in India to combat misinformation by detecting fake news through linguistic analysis, identifying bots and coordinated campaigns. They also go one step ahead and verify even images and videos. These AI-driven tools are essential for managing the vast and diverse digital landscape, particularly across the diversity of the nation which uses multiple languages.

2 Related Works

The goal is to identify fake information on the internet, with a special focus on Dravidian languages like Tamil, Telugu, Malayalam, and Kannada. To accomplish this, the proposed approach involves a fake news detection model that utilizes Ensemble techniques alongside the BERT model.

The use of Ensemble techniques for fake news detection is thoroughly examined in [1], achieving an accuracy of around 0.94. Similarly, the BERT-base-multilingual model was employed to detect fake tweets in a COVID-19 dataset involving Indian languages like Hindi and Bengali, attaining F1 scores of 0.81 and 0.78, respectively [9]. Different machine learning techniques for detecting fake news in English are explored in [11] and [22].

The invention of the BERT [5] model in 2018 led to shocking results in various word processing tasks within the Natural Language Processing (NLP) field. The applications of BERT in NLP tasks is given in detail in [13]. The implementation of BERT to identify fake news is explored in [8].

Interpreting BERT-based models in fake news identification is also explored in [24], where the authors propose a model framework that focuses on attention weights (what words the model focuses on in the input text, thus giving them higher attention scores). This helps in the analysis of the model's decision-making process. The approach helps address the interpretability challenge of complex models like BERT, making it easier to understand and trust their outputs in the context of misinformation identification. . Similar to Dravidian languages, Arabic is also a low-resource language, with limited datasets and tools available digitally. The paper [23] experimented with building a jointBERT application which detected fake news for Arabic data, in which Named Entity Recognition(NER) and Relative Features Classification (RFC) were used as shared parameters. There was also a focus on determining the degree of fakeness of news. Tests were carried out using current Arabic misinformation identification techniques, such as AraBERT, AraGPT2, and Qarib on four real-world Arabic fake news datasets.

In [2], the authors address the challenge of misinformation identification in Dravidian languages using Google's MuRIL (Multilingual Representations for Indian Languages) model.

Research presented in [19] evaluates Tamil, Malayalam, Telugu and Kannada fake news datasets using four transformer models: XLM-RoBERTa, mBERT, MuRIL, and IndicBERT with MuRIL outperforming the others in accuracy.

The identification of misinformation using deep learning techniques is discussed in [14], while the use of CNN-RNN-based approaches for misinformation identification is presented in [16].

The paper [12] focuses on identifying misinformation in resource-scarce languages, with a particular emphasis on Malayalam. It introduces a novel framework that integrates Sanskrit-transliterated Subword2vec embeddings, subword tokenization, and a robust Bidirectional Long Short-Term Memory (BiLSTM) architecture.

Finally, [21] utilizes machine learning algorithms like Decision Trees, Logistic Regression, and Random Forest to identify misinformation.

3 Fake News Dataset

The main target of the model for fake news detection is to accurately distinguish between fake and real information present. The input data has been divided into 2 parts, Training and Testing Data. The Training Data is utilized by the machine learning model to learn and build a reference for the model. Then the model derives prediction outcomes on the Testing data with all the learnings from the Training Data. The Dataset used is DFND: Dravidian Fake News Dataset [18] which contains both fake and real news texts in 4 Dravidian languages, Tamil, Telugu, Malayalam and Kannada.

Table 1. Real and Fake News Data Distribution

Language	Real	Fake
Tamil	3195	3086
Malayalam	4398	3913
Kannada	3058	3220
Telugu	3245	3236

4 Methodology

The methodology takes into consideration various machine learning models to facilitate identification of fake information. The Ensemble technique uses Decision Trees, Logistic Regression, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) methods which are a few renowned classifiers. So the Ensemble technique is used to improve the variance and bias. Multiple steps as mentioned in Figure 1 were carried out during the text processing to classify it.

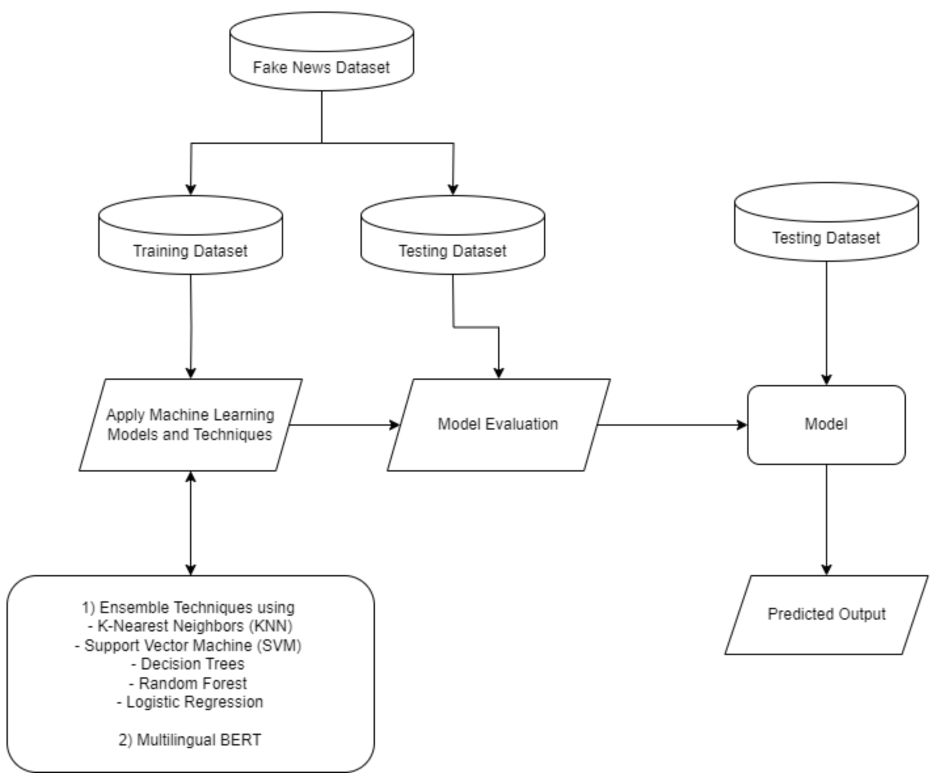

Fig. 1. Workflow chart

4.1 Ensemble Techniques

These methods [6] focus on combining many **trees base algorithms** to construct more effective **predictive performance**, in comparison to a single **tree base algorithm.** ***The main idea behind this model is to form a strong learner by grouping weaker learners, thus increasing the model accuracy.*** There are three main causes of difference in actual and predicted values when such AI techniques are used: **noise, variance, and bias.** Evaluation based on multiple models (ensemble) significantly reduces these (with the exception of noise, as it is an irreducible error).

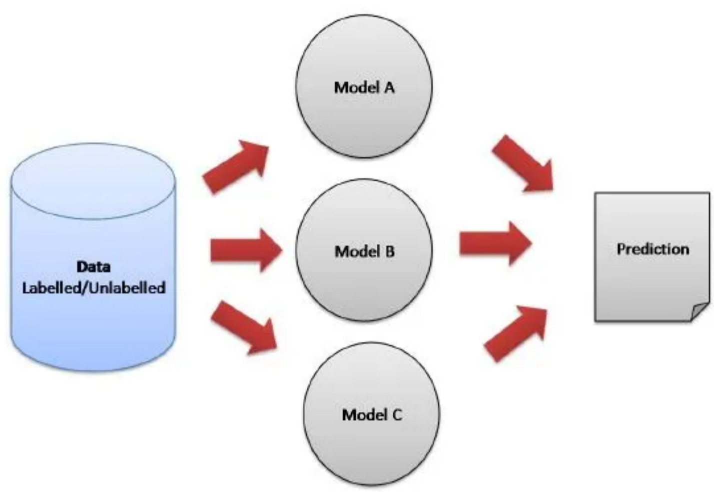

Fig. 2. Flow diagram of Ensemble machine learning method. Source: [7]

Multiple classifiers, such as Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, and Support Vector Machine (SVM) used Voting Classifier. Specifically, the type of voting used was **Soft Voting**. In soft voting, the classifier takes the predicted probabilities from each base classifier and takes the average of these values. The final outcome is the one with the highest average probability.

Based on usage, the Ensemble Algorithms can be categorised as:

Bagging (Bootstrap Aggregating): Bagging [3], or ***bootstrap aggregation***, generates and substitutes samples from the dataset. i.e, a particular sam-

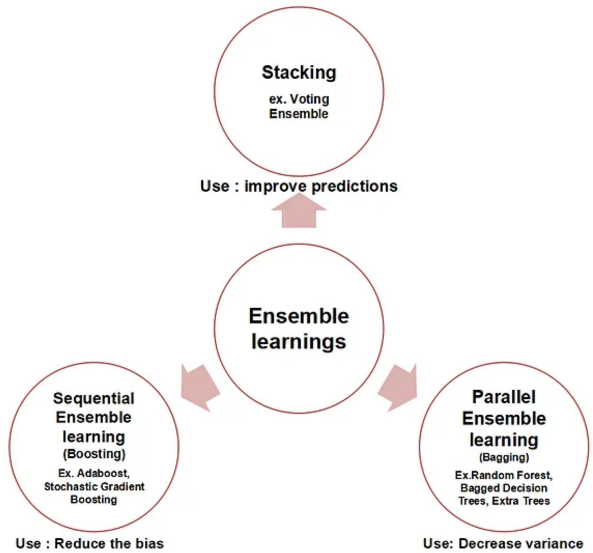

Fig. 3. Types of Ensemble Models and their uses

ple can have a single instance repeated multiple times. Bootstrapping increases the training data, which upon creation and implementation can be pushed into the classifier model. The prediction outcome is derived from the average of the predictive models that have been used.

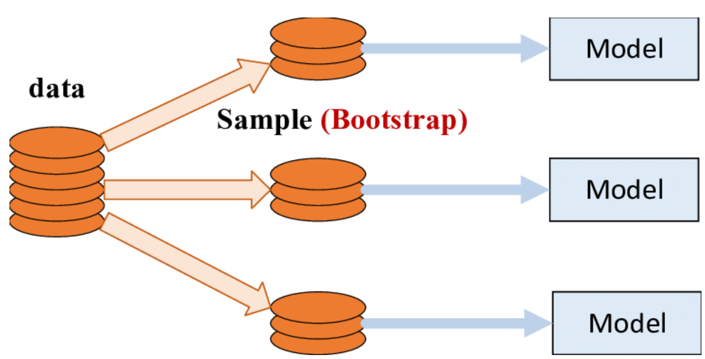

Fig. 4. Flow diagram of Bagging machine learning method. Source: [26]

As in Figure 4, Bagging method can be broken down into three steps:

1. **Bootstrap Sampling**
2. **Aggregation (Model fit)**
3. **Combining different models and aggregating the Results**

Boosting: Boosting [17] is typically used for prevention of under-fitting (when a single model does not work well) and over-fitting (when the model does not produce efficient results on the evaluation data set). This is done by obtaining the strength of weak learners. Thus, multiple weaker machine learning models combine all their final outcome to make a "stronger model" based on input data.

Gradient Boosting Algorithm was used in the testing, **Gradient Boosting** is an ensemble technique similar to bagging but with a different approach. While bagging combines multiple models by averaging their predictions to reduce variance, gradient boosting focuses on reducing bias by sequentially building models that correct the errors of the preceding ones.

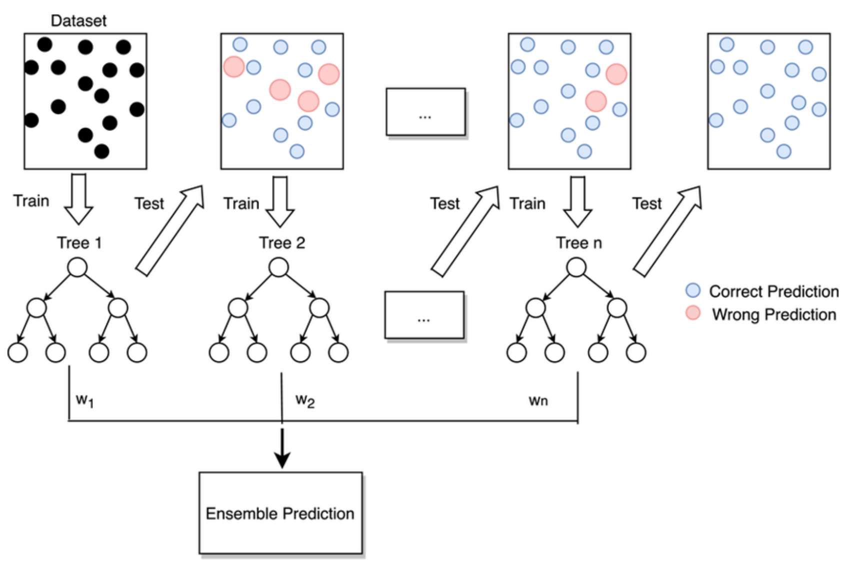

Fig. 5. Gradient Boosting machine learning method Flow Diagram. Source: [27]

Gradient boosting works through the following steps:

1. **The ensemble of classifiers consists of a group of weaker other classifiers.**

2. **In the next classifier, the weightages of the data points that have been predicted incorrectly are increased.**
3. **The output is determined taking into consideration the average of each individual data point's prediction.**

Meta Stacking: On the basis of **meta learning**, multiple machine learning algorithms are combined (or ***stacked***. Here, the base level algorithms train on a complete training data-set. The meta model trains based on these evaluations of the base level models. Hence, while bagging and boosting deals with bias and variance, stacking handles the accuracy of the prediction of the given method.

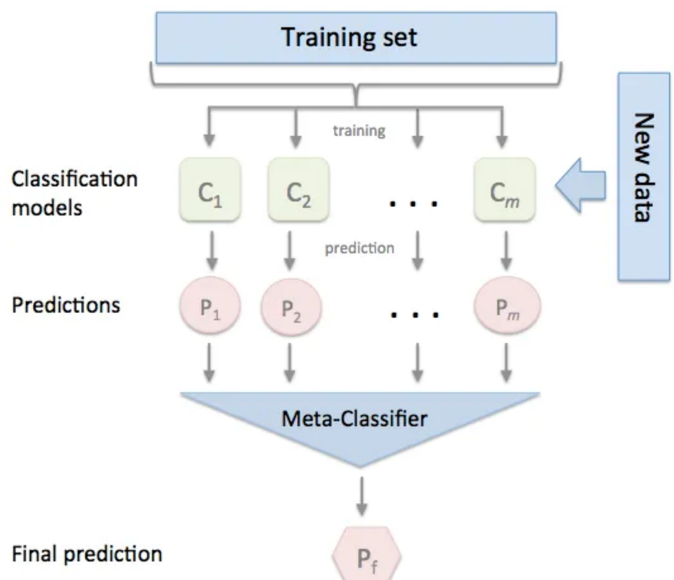

Fig. 6. Diagram depicting the flow of Stacking machine learning method. Source: [25]

In the given example, the dataset used for training trains multiple classifiers, denoted as **C1, C2, . . . Cm**, each producing predictions **P1, P2, . . . Pm**. By applying the technique of **Stacking**, the predictions can be aggregated either by ***averaging the results*** from the classifiers or by using a ***weighted sum*** approach. This process yields the final prediction **Pf**.

4.2 BERT-base Model

BERT, introduced by researchers at Google in 2018 [5], is a powerful language model that uses transformer architecture. Going beyond the earlier model architecture, such as LSTM and GRU, that were either unidirectional or sequentially bi-directional, BERT considers context from both past and future simultaneously. This is due to the innovative "attention mechanism," in which the model weighs the importance of words in a sentence when generating representations.

For the experimentation, **BERT-based-multilingual-cased** model was used, which includes 768 layers that are not seen by the user, 110 million variables for calculation and a dozen focus mechanisms. Pre-training of the system is done using data from the most-widely used 104 languages with the help of a masked language modeling (MLM) technique. The BERT model used is also programmed based on the following two language processing tasks:

- **Masked Language Model (MLM):** The bits of the input sentence is hidden, or masked. This masked sentence is then fed into the model, which predicts the words that have been masked. This process enables the model to develop bidirectional knowledge.

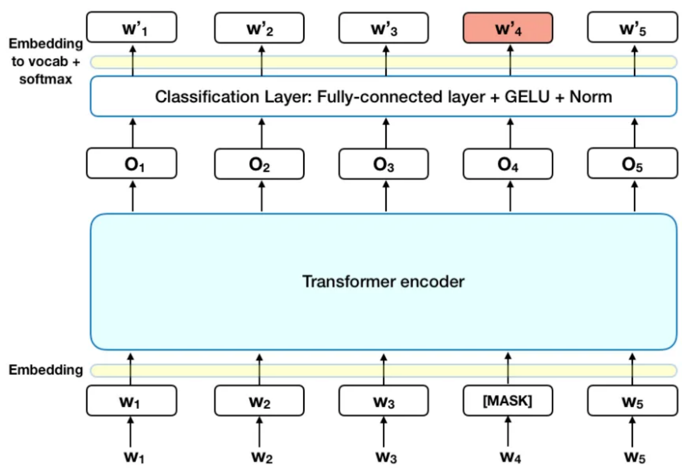

Fig. 7. Working of MLM Process Source: [4]

- **Next Sentence Prediction (NSP):** The model is presented with concatenated sentences and has to determine whether the sentences follow each other in the original order.

The process of building a BERT model involves two critical steps: pre-training and fine-tuning.

- **Pre-training** is the step where the training of BERT model is done on vast amounts of recorded data. As a result, it learns to predict masked words in a sentence (MLM task) and to predict if a sentence follows another one (NSP task). With this, a pre-trained language processing model is obtained, with

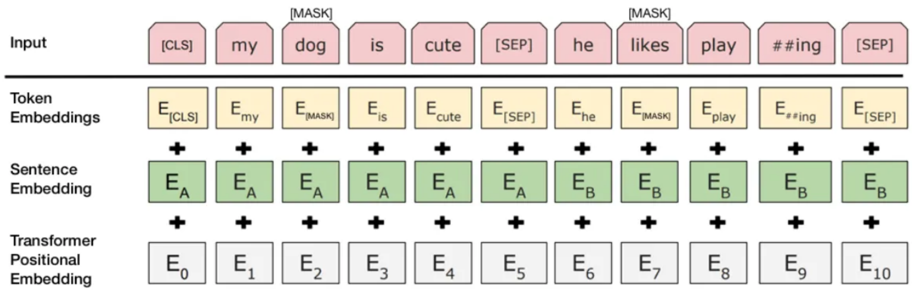

Fig. 8. Working of NSP Process Source: [10]

a general-purpose to gain an "understanding" of the language which has to be processed.

– **Fine-tuning** is where the previously trained BERT model is further trained and made more accurate on a specific task. The model already has a set of default learning parameters, and the entire model is specifically fine-tuned for a downstream task, allowing BERT to adjust its understanding of the language to the details of the task.

The Simple transformers library was utilized to implement the BERT model, which breaks down the creation of models from Hugging Face, an AI community that hosts up-to-date models.

5 Implementation

The goal of this work was to identify real and fake news in the dataset while improving model efficiency and accuracy. The dataset was divided such that 70% was used for training and 30% for testing. Both datasets were vectorized using TfidfVectorizer and Word2Vec to compare the accuracy scores.

Term Frequency-Inverse Document Frequency (TF-IDF) [20] measures how relevant a word is within a corpus or series of texts. A higher weightage is assigned to words that are more important (**Term Frequency**), while less weightage is given to words that are less important (**Inverse Document Frequency**).

Word2Vec [15] is a technique used to generate word embeddings, or numerical representations of words that capture their meanings in a vector space. This method transforms words into dense vectors of fixed dimensions, and hence, words that have similar meanings (or the same root words) will have vectors close to each other. Word2Vec operates in two main ways: **Continuous Bag of Words (CBOW)** and **Skip-gram**.

– **Continuous Bag of Words** predicts a target word based on its surrounding context words.

– **Skip-gram**, in contrast, predicts the surrounding context words using a given target word.

Through this process, Word2Vec captures the semantic relationships between words, assigning similar vectors to words that often appear in similar contexts within the text. This is captured in the form of scores, which are then converted into feature vectors. The model is trained with these vectors.

A **Voting Classifier** was employed that combines multiple classifiers, including Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, and Support Vector Machine (SVM). **Soft Voting** was the voting method chosen. In soft voting, the classifier aggregates the predicted probabilities from each base classifier and averages them. The final output is determined by the class with the **highest average probability**.

Soft Voting method was chosen as the Voting method for the following reasons:

– Soft voting utilizes the predicted probabilities from each base classifier rather than just the predicted class labels.
– By averaging the predicted probabilities, it reduces the impact of any individual classifier that might be making incorrect predictions, thereby leveraging the strengths of all classifiers involved.

For the **Meta Stacking technique**, the classifiers Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, and Support Vector Machine (SVM) served as the **Base Learners**. These individual classifiers made predictions independently on the same dataset. Their predictions were then used as input for the stacking process. Logistic Regression was chosen as the **Meta Learner**, which combines the predictions of the base learners and generates the final result of the stacking model.

Logistic Regression was chosen as the Meta Learner for the following reasons:

– Logistic Regression is a simple, linear model that efficiently combines the base models' predictions using a straightforward **linear combination**, making it easy to implement and interpret how each base learner contributes to the final output.
– Logistic Regression works effectively with the probabilistic outputs from the base learners. It can take these probabilities and linearly combine them to make a final decision, making it ideal for aggregating predictions from different classifiers that may have different strengths.

The **Simple transformers** library was utilized to implement a **bert-based-multilingual-cased** model, enhancing accuracy. This pre-trained model, compatible with Transformers, was utilized along with BERT's built-in tokenizer to efficiently convert text data into tokens. The model then processes these tokens to generate weightages for training and prediction tasks.

6 Results and Analysis

6.1 Performance Metric

The **Confusion Matrix** is a tool for assessing the performance of a classifier by comparing the actual labels to the predicted labels based on four key components:

- True Positives (TP) - predicted as positive and are actually positive
- True Negatives (TN) - predicted as negative and are actually negative
- False Positives (FP) - predicted as positive but are actually negative
- False Negatives (FN) - predicted as negative but are actually positive

The values in the matrix help visualize the model's performance and assess metrics like F1-score, precision, recall, and accuracy, which are crucial to understanding the overall effectiveness of the classifier.

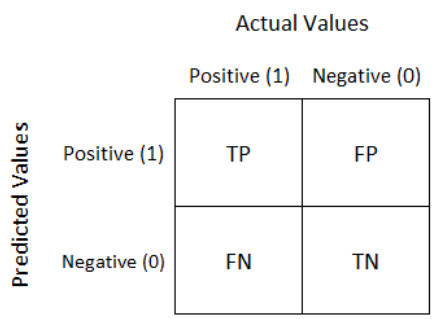

Fig. 9. Confusion Matrix

The classifiers were evaluated based on their accuracy scores. Accuracy in classification refers to the overall effectiveness of a model in correctly classifying all instances. The accuracy is determined using the following formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Equation 1: *Accuracy formula*

6.2 Scores and Findings

Table 2, **Table 3** and **Table 4** shows the accuracy scores of various machine learning models applied to text classification tasks in 4 South Indian Dravidian languages: Tamil, Malayalam, Telugu, and Kannada. The models were evaluated using two different vectorization techniques — Word2Vec (**Table 2:** 2) and TF-IDF (**Table 3:** 3) — along with the BERT-multilingual model (**Table 4:** 4).

The following analysis simplifies the performance of each model and compares their effectiveness across languages and vectorization techniques.

Table 2. Accuracy of Models with Word2Vec

Models	Tamil	Malayalam	Telugu	Kannada
SVM	0.6	0.72	0.68	0.67
Logistic Regression	0.63	0.72	0.72	0.66
KNN	0.69	0.82	0.7	0.69
Random Forest	0.69	0.82	0.7	0.69
Decision Tree	0.66	0.76	0.72	0.66
Voting Classifier	0.72	0.81	0.77	0.71
Bagging	0.7	0.79	0.79	0.7
Gradient Boosting	0.71	0.81	0.8	0.72
Meta Stacking	0.76	0.84	0.81	0.74

Table 3. Accuracy of Models with TF-IDF Vectorizer

Models	Tamil	Malayalam	Telugu	Kannada
SVM	0.86	0.89	0.87	0.8
Logistic Regression	0.85	0.88	0.86	0.8
KNN	0.53	0.57	0.53	0.71
Random Forest	0.87	0.89	0.86	0.79
Decision Tree	0.79	0.86	0.78	0.69
Voting Classifier	0.87	0.89	0.86	0.79
Bagging	0.84	0.87	0.83	0.73
Gradient Boosting	0.85	0.84	0.85	0.78
Meta Stacking	0.9	0.91	0.88	0.82

Table 4. Accuracy of BERT-base-multilingual-cased

Models	Tamil	Malayalam	Telugu	Kannada
BERT-multilingual	0.96	0.98	0.98	0.94

The results indicate that models generally perform more efficiently with TFIDF than Word2Vec, suggesting that TFIDF captures text features more effectively for this classification task. The BERT-multilingual model clearly outperforms other models, followed by Meta Stacking with TFIDF. Traditional models like SVM and Random Forest also perform well with TFIDF compared to Word2Vec.

Malayalam consistently achieves the highest accuracy score across almost all models and vectorization techniques, while Kannada generally has the lowest accuracy, which could be due to variations in the complexity of text features and scripts between the languages.

7 Conclusion

The analysis of various machine learning models for text classification tasks across South Indian languages has highlighted the significance of model choice and vectorization techniques. TFIDF outperformed Word2Vec in most cases, and pre-trained models like BERT showed substantial improvements. Further optimization through hyperparameter tuning and ensemble methods could enhance model performance, especially for languages like Kannada.

References

1. Ahmad, I., Yousaf, M., Yousaf, S., Ahmad, M.O.: Fake news detection using machine learning ensemble methods. Complexity **2020**(1), 8885861 (2020). https://doi.org/https://doi.org/10.1155/2020/8885861, https://onlinelibrary.wiley.com/doi/abs/10.1155/2020/8885861
2. Bala, A., Krishnamurthy, P.: AbhiPaw@ DravidianLangTech: Fake news detection in Dravidian languages using multilingual BERT. In: Chakravarthi, B.R., Priyadharshini, R., M, A.K., Thavareesan, S., Sherly, E. (eds.) Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages. pp. 235–238. INCOMA Ltd., Shoumen, Bulgaria, Varna, Bulgaria (Sep 2023), https://aclanthology.org/2023.dravidianlangtech-1.34
3. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (Aug 1996), https://doi.org/10.1007/BF00058655
4. Chowdhury, A., Das, A., Saha, S., Rahman, M., Hasan, T.: Sentiment analysis of covid-19 vaccination from survey responses in bangladesh. (04 2021). https://doi.org/10.21203/rs.3.rs-482293/v1
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019), https://arxiv.org/abs/1810.04805
6. Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. Frontiers of Computer Science **14**(2), 241–258 (Apr 2020). https://doi.org/10.1007/s11704-019-8208-z
7. Ganta, D.P., Gupta, H.D., Sheng, V.S.: Knowledge distillation via weighted ensemble of teaching assistants. In: 2021 IEEE International Conference on Big Knowledge (ICBK). pp. 30–37 (2021). https://doi.org/10.1109/ICKG52313.2021.00014
8. Jwa, H., Oh, D., Park, K., Kang, J.M., Lim, H.: exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). Applied Sciences **9**(19) (2019). https://doi.org/10.3390/app9194062
9. Kar, D., Bhardwaj, M., Samanta, S., Azad, A.P.: No rumours please! a multi-indic-lingual approach for covid fake-tweet detection. In: 2021 Grace Hopper Celebration India (GHCI). pp. 1–5 (2021). https://doi.org/10.1109/GHCI50508.2021.9514012

10. Karimi, A., Rossi, L., Prati, A.: Adversarial training for aspect-based sentiment analysis with bert. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 8797–8803 (2021). https://doi.org/10.1109/ICPR48806.2021.9412167
11. Khanam, Z., Alwasel, B.N., Sirafi, H., Rashid, M.: Fake news detection using machine learning approaches. IOP Conference Series: Materials Science and Engineering **1099**(1), 012040 (mar 2021). https://doi.org/10.1088/1757-899X/1099/1/012040
12. Kodali, R., Manukonda, D.: byteSizedLLM@DravidianLangTech 2024: Fake news detection in Dravidian languages - unleashing the power of custom subword tokenization with Subword2Vec and BiLSTM. In: Chakravarthi, B.R., Priyadharshini, R., Madasamy, A.K., Thavareesan, S., Sherly, E., Nadarajan, R., Ravikiran, M. (eds.) Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages. pp. 79–84. Association for Computational Linguistics, St. Julian's, Malta (Mar 2024), https://aclanthology.org/2024.dravidianlangtech-1.12
13. Koroteev, M.V.: Bert: A review of applications in natural language processing and understanding (2021), https://arxiv.org/abs/2103.11943
14. Kumar, S., Asthana, R., Upadhyay, S., Upreti, N., Akbar, M.: Fake news detection using deep learning models: A novel approach. Transactions on Emerging Telecommunications Technologies **31**(2), e3767 (2020). https://doi.org/https://doi.org/10.1002/ett.3767
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013), https://arxiv.org/abs/1301.3781
16. Nasir, J., Khan, O., Varlamis, I.: Fake news detection: A hybrid cnn-rnn based deep learning approach. International Journal of Information Management Data Insights **1**, 100007 (04 2021). https://doi.org/10.1016/j.jjimei.2020.100007
17. Natekin, A., Knoll, A.: Gradient boosting machines, a tutorial. Frontiers in Neurorobotics **7** (2013). https://doi.org/10.3389/fnbot.2013.00021
18. Raja, E., Soni, B., Borgohain, S.K.: Dfnd : Dravidian_ fake news data (2023). https://doi.org/10.21227/nj13-t949
19. Raja, E., Soni, B., Borgohain, S.K.: Fake news detection in dravidian languages using transformer models. In: Malhotra, R., Sumalatha, L., Yassin, S.M.W., Patgiri, R., Muppalaneni, N.B. (eds.) High Performance Computing, Smart Devices and Networks. pp. 515–523. Springer Nature Singapore, Singapore (2024)
20. Robertson, S.: Understanding inverse document frequency: On theoretical arguments for idf. Journal of Documentation - J DOC **60**, 503–520 (10 2004). https://doi.org/10.1108/00220410410560582
21. Shanmugavadivel, K., Subramanian, M., R, S., Sameer B, M., K, M.: Beyond tech@DravidianLangTech2024 : Fake news detection in Dravidian languages using machine learning. In: Chakravarthi, B.R., Priyadharshini, R., Madasamy, A.K., Thavareesan, S., Sherly, E., Nadarajan, R., Ravikiran, M. (eds.) Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages. pp. 124–128. Association for Computational Linguistics, St. Julian's, Malta (Mar 2024), https://aclanthology.org/2024.dravidianlangtech-1.20
22. Sharma, U., Saran, S., Patil, S.M.: Fake news detection using machine learning algorithms. International Journal of creative research thoughts (IJCRT) **8**(6), 509–518 (2020)
23. Shishah, W.: Jointbert for detecting arabic fake news. IEEE Access **10**, 71951–71960 (2022)
24. Szczepański, M., Pawlicki, M., Kozik, R., Choraś, M.: New explainability method for bert-based model in fake news detection. Scientific reports **11**(1), 23705 (2021)

25. Tran, T.: On some studies of fraud detection pipeline and related issues from the scope of ensemble learning and graph-based learning (2022), https://arxiv.org/abs/2205.04626
26. Wu, L.Y., Weng, S.S.: Ensemble learning models for food safety risk prediction. Sustainability **13**, 12291 (11 2021). https://doi.org/10.3390/su132112291
27. Zhang, T., Lin, W., Vogelmann, A., Zhang, M., Xie, S., Qin, Y., Golaz, J.: Improving convection trigger functions in deep convective parameterization schemes using machine learning. Journal of Advances in Modeling Earth Systems **13** (05 2021). https://doi.org/10.1029/2020MS002365