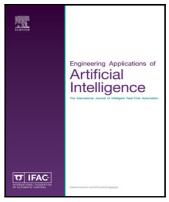
Engineering Applications of Artificial Intelligence 119 (2023) 105731

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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



Hindi fake news detection using transformer ensembles

Amit Praseed [a,*], Jelwin Rodrigues [b], P. Santhi Thilagam [b]

[a] Department of Computer Science and Engineering, Indian Institute of Information Technology Sri City, Chittoor, India
[b] Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, India

ARTICLE INFO

Keywords:
Fake news
Transformer
Hindi fake news
mBERT
ELECTRA
XLM-RoBERTa
Ensemble

ABSTRACT

In the past few decades, due to the growth of social networking sites such as Whatsapp and Facebook, information distribution has been at a level never seen before. Knowing the integrity of information has been a long-standing problem, even more so for the regional languages. Regional languages, such as Hindi, raise challenging problems for fake news detection as they tend to be resource constrained. This limits the amount of data available to efficiently train models for these languages. Most of the existing techniques to detect fake news is targeted towards the English language or involves the manual translation of the language to the English language and then proceeding with Deep Learning methods. Pre-trained transformer based models such as BERT are fine-tuned for the task of fake news detection and are commonly employed for detecting fake news. Other pre-trained transformer models, such as ELECTRA and RoBERTa have also been shown to be able to detect fake news in multiple languages after suitable fine-tuning. In this work, we propose a method for detecting fake news in resource constrained languages such as Hindi more efficiently by using an ensemble of pre-trained transformer models, all of which are individually fine-tuned for the task of fake news detection. We demonstrate that the use of such a transformer ensemble consisting of XLM-RoBERTa, mBERT and ELECTRA is able to improve the efficiency of fake news detection in Hindi by overcoming the drawbacks of individual transformer models.

1. Introduction

The Internet has revolutionized every facet of modern life, but one area that has changed beyond recognition is communication. With the advent of social media platforms like Whatsapp, Twitter and Facebook, the process of information sharing has been altered dramatically (Hanna et al., 2011). It now takes a fraction of a second to share a message, a piece of news or a viral article. While these platforms have existed for nearly a decade, the rise of Internet penetration has led to a dramatic change in how these platforms are being used. Evidence suggests that these social media platforms might be slowly replacing the traditional media. People prefer to read the news using online content from various websites rather than traditional news organizations because it takes less time, is less expensive, and the news can be easily shared and discussed among friends (Nelson and Lei, 2018).

While social media platforms enable the faster spread of information, they also have some disadvantages. The decentralized nature of social media means that there is often no central authority to verify the veracity of the information being shared. This, in turn, has led to a steep increase in the number of fake news articles being circulated through social media. Fake news articles or messages could be shared unknowingly, unintentionally or sometimes with the intent of spreading disinformation. It is often very difficult for individuals to verify the authenticity of a message, and they end up passing it on to others without regard to its veracity. In addition, it has been observed that old news gets circulated a lot, misleading and creating panic among people (Flintham et al., 2018). Thus, the need of the hour is to have efficient mechanisms in place to identify and curb the spread of fake news.

There are many existing measures to tackle fake news. Websites like www.factcheck.org and www.politifact.com (Fridkin et al., 2015) provide updates related to fake news articles being circulated through social media. However, this solution is limited to certain types of news, and users often have to perform manual verification of news articles by browsing through these websites. Significant research is being done to identify fake news by observing the style of writing itself. Several deep learning techniques, notably those based on Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM) and Transformers, have been shown to be extremely efficient in detecting fake news (Zhou and Zafarani, 2020).

However, a large number of these approaches are targetted towards the English language. This implies that most of these mechanisms cannot be used to identify fake news being circulated in regional languages such as Hindi, despite it being the third most spoken language globally. Some proposed solutions involve converting regional languages into

* Corresponding author.
E-mail address: amitpraseed@gmail.com (A. Praseed).

https://doi.org/10.1016/j.engappai.2022.105731
Received 28 February 2022; Received in revised form 1 November 2022; Accepted 11 December 2022
Available online 21 December 2022
0952-1976/

the English Language using a translator and then using the suggested solutions above to classify it into fake or factual news. Even though we can obtain good results using this, the approach is not always accurate (Saghayan et al., 2021).

Transformer based models offer an alternate solution. The use of transfer learning using pre-trained transformer bases, such as BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018), have been proved successful in a wide variety of applications, including fake news detection. Though the traditional BERT model is pre-trained on the English language, a multilingual variant of BERT, called mBERT, was developed in 2019. mBERT provides sentence representations for 104 languages (including Hindi), which are useful for many multi-lingual tasks. mBERT has been used successfully for fake news detection in the Hindi language. RoBERTa (Robustly Optimized BERT Pre-training Approach) (Liu et al., 2019) is a more advanced variation of BERT, pre-trained over a much larger corpus with more optimizations. A multi-lingual variation of RoBERTa, called as XLM-RoBERTa is also available, and has been demonstrated to be considerably effective in a variety of natural language processing tasks. One drawback of using BERT (or mBERT) is that the model tends to be correspondingly huge. Google developed a different transformer base called ELECTRA (Pre-training Text Encoders as Discriminators Rather Than Generators) (Clark et al., 2020), which could potentially serve as a replacement for BERT. ELECTRA is designed to have a much smaller model than BERT, thereby significantly reducing resource requirements. However, despite the multilingual nature of these transformer models, the representative power of the languages involved varies, depending upon the pre-training data used. Transformer models are usually trained on articles from Wikipedia, and a language is considered a high or low resource based on the size of Wikipedia in that language. Wu and Dredze (2020) introduced a measure of resource size, called as Wikisize, which is the log of the corpus size in MB. As an example, for the mBERT transformer model, English has a Wikisize of 14, while Hindi has a Wikisize of 7, which makes Hindi a resource constrained language.

Majority of the existing works on fake news detection in Hindi employ one of these pre-trained transformer models (with or without additional steps). However, the selection of the transformer models itself has an impact on the detection process due to the fact that different transformer models are pre-trained on different datasets using different modelling techniques. Due to this fact, they also exhibit subtle differences in the way they encode data and classify data points. Even when fine-tuned using the same dataset with the same training parameters, the transformer models are expected to show discrepancies in their outputs. In other words, certain transformer models may perform well for certain types of data points, while others may perform poorly. As has been demonstrated in multiple problems of machine learning and NLP, ensembling several diverse models is usually an effective way to improve performance. This is especially true in the case of resource constrained languages such as Hindi, where the differences in the language representation can overcome the comparative lack of pre-training data.

In this work, we present a mechanism for Hindi fake news detection using an ensemble of pre-trained transformer models. More precisely, the pre-trained transformer models – XLM-RoBERTa, mBERT and ELECTRA – are individually fine-tuned for the task of fake news detection in Hindi. The classification results from each of these transformer models are combined using a majority voting strategy. We demonstrate that this model provides a better performance than the separate transformer models and is able to detect fake news in Hindi with a high degree of accuracy. To the best of our knowledge, this is the first work that uses an ensemble of pre-trained transformer models for identifying fake news in Hindi.

Our contributions in this paper are:

- We present an approach for detecting fake news in Hindi using an ensemble of pre-trained transformer models – XLM-RoBERTa, mBERT and ELECTRA – which are separately fine-tuned for the task of Hindi fake news detection
- We demonstrate that the proposed ensemble performs better than the individual transformer models
- We also demonstrate that the proposed ensemble of pre-trained transformer models performs extremely well in detecting fake news in the Hindi language

2. Related work

2.1. Language models and transformers

Language models are one of the most crucial components of Natural Language Processing (NLP). Language models provide the necessary context required to distinguish between words and phrases in text and/or speech. The success of most NLP tasks are governed by how effectively they are able to model the sequential context in languages. Recurrent Neural Networks (RNN) have been widely used for this purpose, but the sequential nature of these models made it difficult to scale. The transformer architecture (Vaswani et al., 2017) replaces RNN cells with self-attention and fully connected layers. They are able to capture long range dependencies in languages and are hence extremely effective in NLP tasks. In simple terms, a transformer comprises of two parts — an encoder and a decoder. The encoder part is responsible for converting (or encoding) the input into an intermediate language model and the decoder is responsible for converting (or decoding) the intermediate language model into the output. In a real world context, a transformer could be used to convert a piece of information from one language to another, by means of the intermediate language model representation.

In a wide variety of applications, generation of a language model is a crucial and time consuming task, as opposed to using the language model to identify whether an piece of text belongs to the language or not. Most of the labelled text datasets tend to be comparatively small, and training deep neural networks using them often result in overfitting. This issue can be solved using the concept of Transfer Learning (Pan and Yang, 2009), which allows a deep neural network trained on a large dataset (such as the Wikipedia corpus) to be used to perform similar tasks on a different dataset. Such deep learning models are called as pre-trained models, and have been used in various image processing tasks for a long time.

The concept of transfer learning became popular in NLP only in 2018 with the introduction of transformer based models such as BERT. These models take the entire input sequence as input unlike RNNs, which means that transformer based models can be parallelized using GPUs effectively. These models are already pre-trained with a large amount of data and can be used for different NLP tasks by fine tuning the model. Fine tuning often involves freezing one part of the network and training the other layers. In some situations, it might be even be sufficient to use the entire architecture as such and only train the output layer used for classification.

2.2. Existing transformer based models

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) is a method of pre-training language representations for certain NLP tasks that helps us obtain state-of-the-art results. BERT uses a Masked Language Model (MLM) pre-training objective, which randomly masks some of the tokens from the input. The training objective is to predict the original vocabulary using information present in its context. The MLM objective enables the representation to combine both the left and the right context, which allows BERT to perform efficiently as a deeply bidirectional transformer. BERT also

employs a next sentence prediction task which pre-trains joint text-pair representations. BERT is extensively used for various NLP tasks, including the detection of fake news. This is accomplished by adding a classifier head on top of the BERT base. A multilingual variation of BERT, called mBERT, is useful for multilingual classification tasks and can be used for 104 different languages.

RoBERTa (Robustly optimized BERT pretraining approach) (Liu et al., 2019) is an improvement over the BERT model, and has been shown to outperform BERT in a number of NLP tasks. RoBERTa accomplishes this improvement in performance by training the model for a longer duration, using more data and with bigger batches. RoBERTa also removes the next sentence prediction objective and dynamically alters the masking pattern applied to the training data. These improvements enable RoBERTa to outperform BERT in a wide variety of NLP tasks. A multilingual variant of RoBERTa, called XLM-RoBERTa, is also available, and has been trained on 100 different languages.

Although the MLM models produce good results when transferred to downstream NLP tasks, they usually require large amounts of computation to be effective. ELECTRA uses a different approach called as replaced token detection. Instead of masking the input, this approach works by replacing some tokens with suitable alternatives. ELECTRA (Clark et al., 2020) basically trains a discriminative model that tries to predict whether each token in the corrupted input was replaced or not. This approach is comparatively sample-efficient and uses fewer computational resources than MLM models. Similar to BERT and RoBERTa, a multilingual variant of ELECTRA is also available.

2.3. Natural language processing (NLP) in Hindi

Performing any NLP task on resource constrained languages such as Hindi is a challenging task (Ray et al., 2019; Srivastava et al., 2011; Mehta et al., 2021). One of the major challenges encountered by any researcher in this area is the lack of effective tools specifically targetting Hindi. This necessitates multiple workarounds, such as translating the text to English and then performing the NLP task. However, this approach has limited effectiveness and applicability, which forces researchers to look for other avenues.

Despite the challenges, considerable research has gone into NLP tasks in Hindi. Ray et al. (2019) discussed the major challenges faced in developing topic models for Hindi. They applied Latent Semantic Indexing (LSI), Nonnegative Matrix Factorization (NMF), and Latent Dirichlet Allocation (LDA) algorithms for topic modelling in Hindi and provided mechanisms for visualizing the results. Sharma et al. (2015b) presented a survey on Word Sense Disambiguation techniques in Hindi and observed that the difficulty in knowledge acquisition and lack of properly tagged sense data are major deterrents in this area of research. Kulkarni and Rodd (2021) observed a similar trend in the area of sentiment analysis in Hindi. Despite the fact that important research has gone into this area (Sharma et al., 2015a; Sharma and Moh, 2016), limited or non-availability of relevant resources is a major hindrance to research. They observed that new large datasets should be developed and already existing ones should be expanded for smooth research in this area, along with the introduction of optimized lexical and linguistic resources.

2.4. Fake news detection in Hindi

Comparatively fewer work has been done on fake news detection in Hindi. The majority of these research works have used pre-trained transformer models after fine-tuning. Fake news detection in Hindi is usually taken up in existing research works as a subtask of hostility detection, and use the CONSTRAINT2021 dataset (Bhardwaj et al., 2020). Bhardwaj et al. (2020) achieved an F1 score of 0.68 on this database using a simple BERT architecture, and this is often treated as the baseline performance on this dataset. Other research works in this domain have followed a similar approach and often use transformer models in combination with other machine learning algorithms for detecting fake news. Kamal et al. (2021) used pre-trained HindiBert, IndicBert, and HindiBerta models for the task of hostility detection in Hindi and were able to achieve an F1 score of 0.77. Gupta et al. (2021) used an ensemble of FastText, HindiBERT and BERT to achieve an F1 score of 0.77 on the task of Hindi fake news detection. Shekhar et al. (2021) used a combination of BERT, ANN and XGBoost and achieved an F1 score of 0.81 on Hindi fake news detection.

Kar et al. (2020) used the mBERT model for multilingual text classification on an annotated dataset of Hindi and Bengali tweets for Fake News Detection. This model reached an 0.89 F-1 score for fake news detection and surpassed some English language results, thus implying that models trained on multiple Indic languages perform better as they share similar syntactic constructs. Badam et al. (2022) curated a collection of fake news articles in Hindi and performed preliminary analysis of these articles using popular machine learning algorithms.

2.5. Need for transformer ensembles

Although most of the existing research works on fake news detection in Hindi use pre-trained transformer models, they are limited by the fact that they use a single transformer model. This is especially true in the case of resource constrained languages such as Hindi. Incorporating multiple transformer models which operate in slightly different ways, and are pre-trained on different datasets will improve the predictive capability of the model considerably. In addition, the transformer models used in the ensemble should also be different enough to improve the predictive capability. Table 1 describes the number of correct predictions made by the individual transformer models. The columns marked with R, B and E correspond to the number of instances exclusively identified correctly by XLM-RoBERTa, mBERT and ELECTRA respectively. Columns RB, BE and RE denotes the number of instances correctly identified by XLM-RoBERTa and mBERT, mBERT and ELECTRA and XLM-RoBERTa and ELECTRA respectively. The column marked as RBE denotes the number of instances identified by all three transformer models, and the column marked as 'None' denotes the number of data instances that were not detected by any of the transformer models. It can be seen that just over 50% of the fake news articles can be identified by all three transformer models. This clearly indicates the differences in the performances of the transformer models and underlines the need for a transformer ensemble. The use of ensembling techniques has been demonstrated to have a visible improvement in performance for various NLP tasks such as Named Entity Recognition (Saha and Ekbal, 2013; Ekbal and Saha, 2011). In particular, ensembling techniques improve the performance of NLP tasks on resource constrained languages (Sharif and Hoque, 2022; Meetei et al., 2021). At the same time, it is important to note that not all ensembles can improve efficiency. The models that are part of the ensemble need to be diverse enough to collectively perform better than their parts (Kioutsioukis and Galmarini, 2014; Brown and Kuncheva, 2010).

3. Methodology

3.1. Pre-processing

The input text needs to be pre-processed before it can be fed to any transformer model. The pre-processing steps eliminate unwanted and meaningless characters and words, thus making it much more easier for the transformer models to encode and extract meaning from the text. The pre-processing steps used in the proposed architecture are explained below, and are similar to the steps followed in most of the existing research works (Gupta et al., 2021; Kamal et al., 2021):

Table 1
Performance of individual transformer models.

Class	R	B	E	RB	BE	RE	RBE	None	Total
Fake	16	16	8	42	11	10	168	29	300
Non-fake	11	3	21	32	37	24	1203	22	1353

3.1.1. Sentence segmentation and tokenization

Sentence Segmentation involves dividing a string of written language into sub-components. Depending upon the scenario, this sub-component could be paragraphs, sentences or phrases. In this work, the text document is divided into sentences. In Hindi, a sentence is segmented whenever we observe purna viram (।) or question mark (?) The sentence must be further divided into various tokens of words, and corresponding spaces, commas, and other special symbols must be removed.

3.1.2. Stemming

Stemming is the process of reducing a word to its word root. We maintain a list of possible suffixes of the word, and when there is a match for any of the values of this list, the suffix part is removed. For example, say we have these three words in our dataset, पहली, पहला, पहले, and all these three words have the same root word that is the root is पहल. The stemming process helps us to get more similar words and thus improve overall accuracy.

3.1.3. Stop words removal

Stop words are frequent, evenly distributed, function words in any document corpus that do not add meaning to the text content. Over 260 stop words provided by data Mendeleev website, sarai.net, were collected and used. Some of the stop words is as follows:

- मुझको
- हमारा
- इसे
- उन्होने
- मेरा
- अपना
- उसके
- अपने
- मुझको
- आप
- उसकी
- जो
- मेरा
- आपका
- यह
- किसे
- हमने
- वह
- इसके
- किसको

3.1.4. Tokenization

Tokenization converts the words in the dataset to numeric tokens which can be easily interpreted by the model. The transformers library in PyTorch has built-in facilities to tokenize the input text. As an example, after tokenization, a sample input text 'विपक्ष कांग्रेस पार्ट सरकार स्थित स्पष्ट कह दूसर ओर सरकार कह स्थित उसक नज़र' gets tokenized to [101, 53836, 94464, 22965, 95490, 567, 17277, 13043, 39425, 24573, 95129, 547, 16080, 52254, 44239, 39425, 547, 16080, 24573, 34731, 566, 58246, 102]. The input is converted to a fixed length by truncating or padding wherever necessary. Attention masks are used so that the model can differentiate between the real token and padded token.

3.2. Model architecture

The fundamental idea of our work is the use of a transformer ensemble consisting of three pre-trained and fine-tuned transformer models - XLM-RoBERTa, mBERT and ELECTRA for the task of fake news detection. Existing research works have used a single transformer model or have used a combination of transformer models and other machine learning techniques. Compared to these approaches, the use of a transformer ensemble gives greater predictive power to the model as a whole. Despite RoBERTa being a more optimized version of BERT, there have been numerous research works demonstrating that the two models

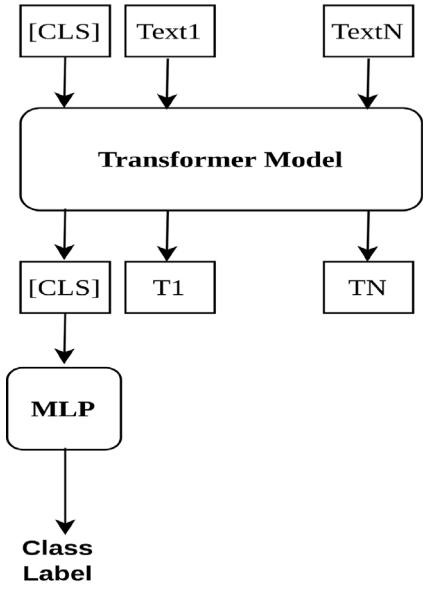


Fig. 1. Generalized architecture of a transformer component.

can be combined in ensembles to obtain better performance. Briskilal and Subalalitha (2022) used an ensemble of BERT and RoBERTa for classifying idioms and Nozza (2022) used an ensemble of BERT, RoBERTa, and HateBERT for detecting homophobia and transphobia in text. For this reason, we have chosen to include XLM-RoBERTa and mBERT in our ensemble. The inclusion of the ELECTRA model further enhances the capability of the model because ELECTRA and BERT (and by extension RoBERTa) work in fundamentally different ways.

3.2.1. Fine tuning individual transformer components

A generalized diagram representing the individual transformer components used in our work is shown in Fig. 1. The input to each transformer model consists of a sequence of tokens (Text1... TextN) and an initial classification token [CLS] which will be used for prediction. Each token has a corresponding embedding, a segment embedding that identifies each sentence, and a position embedding to distinguish the position of each token. The output consists of a sequence of tokens (T1... TN) which can be used for question answering tasks, along with a [CLS] token that now contains the classification information. For the task of fake news detection, this is the output that is focused on. The information contained in the [CLS] token is usually fed to a Softmax classification layer to get the final class output.

We use the pre-trained versions of all three transformer models and perform further fine-tuning of these models using our dataset. The hyperparameters were set using inputs from existing literature, which were further verified using 5-fold cross validation. For example, the number of hidden layers used in each of these transformer models was limited to 6, instead of the default 12. This was done to reduce overfitting and improve the performance of the model (Sajjad et al., 2020). This performance improvement was exhibited by XLM-ROBERTa, mBERT and ELECTRA, which is why the number of hidden layers was limited to 6 in this work. For training the model, the AdamW optimizer with a learning rate of $2e{-}5$ is used as this configuration provided the best results during cross validation. For most transformer based architectures, the Adam optimizer performs well (Kingma and Ba,

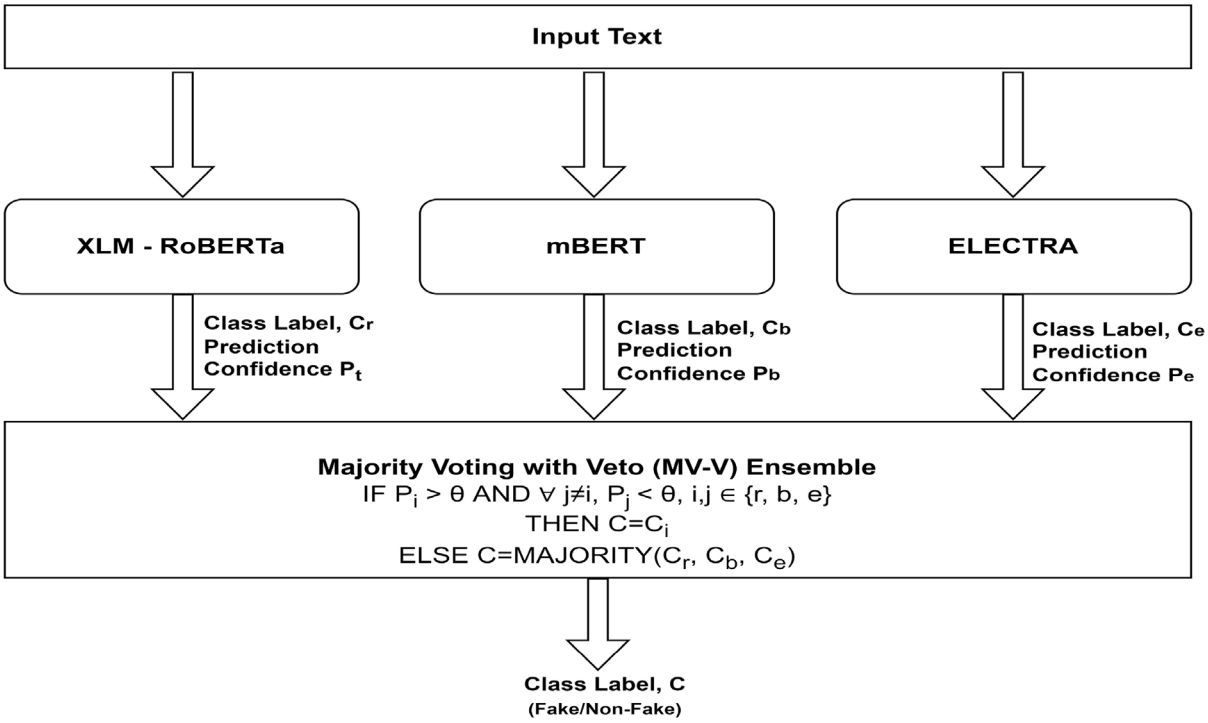


Fig. 2. Proposed architecture.

2014). AdamW is a modification of Adam with correct weight decay and has been shown to have a better generalization capability than Adam, and is hence suitable for transformer models (Loshchilov and Hutter, 2017). Each of these transformer models were fine-tuned with our dataset for 5 epochs using a batch size of 32 which provided the best results.

In addition, the default Softmax classification head was removed in favour of a multi layer perceptron (MLP) (Su et al., 2019). The MLP consists of 2 hidden layers, each having 2 neurons with a ReLU activation function. The use of the MLP instead of the Softmax layer for classification also provides a small improvement in performance for each individual transformer model. The choice of the 2×2 hidden layer architecture and the ReLU activation function were verified through experiments. The output from the transformer models are used as inputs to train the MLP classification heads. For training the MLP classification head, we used the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LM-BFGS) solver. The selection was made from both a practical and theoretical perspective. From a theoretical perspective, the LM-BFGS solver tends to converge faster and perform better on smaller datasets. Additionally, we experimented with SGD and Adam optimizers, but we observed a better performance with the LM-BFGS optimizer. The learning rate of the MLP was set to $1e{-}5$. The results of the cross validation which support the choice of the hyperparameters used in our work is given in Section 4.3.1.

3.2.2. Ensembling the transformer models

The class predictions of each individual transformer model is combined using an ensembling technique. In order to select the best ensembling technique for this task, we have experimented with multiple ensembling techniques like Multi Layer Perceptron, SVM, Decision Trees etc. We have also experimented with two voting techniques — Majority Voting (MV) and Majority Voting with Veto capability (MV-V) (More and Gaikwad, 2016; Shahzad and Lavesson, 2013). In the Majority Voting technique, the class label which is predicted by two out of the three transformer models is chosen as the final prediction. In the MV-V technique, a transformer prediction with a confidence exceeding a particular threshold θ is allowed to veto the predictions of the other two transformer models if they do not have a similar confidence level.

We decided to use the MV-V ensembling mechanism as it demonstrated the best performance. The ensemble model receives the predictions and the confidence values of these predictions from all three transformer models. In most cases, the final prediction is made by majority voting between the three models. However, a model is capable of vetoing the majority voting if it predicts the class of the input text with a confidence greater than a threshold θ, but the predictions of the other two classifiers does not have the same level of confidence. In such cases, the prediction of the model which acquires the veto power is considered to be the final prediction. Fig. 2 provides an illustration of the proposed architecture. The class labels output by XLM-RoBERTa, mBERT and ELECTRA are C_r, C_b and C_e respectively and their prediction confidences are P_r, P_b and P_e respectively. The MV-V ensembling mechanism can be described as a set of rules as follows:

$$IF\ P_i \geq \theta\ AND\ \forall j \neq i\ P_j < \theta,\ i,j \in \{r,b,e\}\ THEN\ C = C_i$$

$$ELSE\ C = MAJORITY(C_r, C_b, C_e)$$

3.3. Training phase

The tokenized dataset is divided into training and testing datasets, and the training dataset is fed into the three transformer models for training. The weights of these transformer models are pre-trained with a large corpus, and it is only necessary to fine tune the model with the dataset for the task of Hindi fake news detection. For training the model, the AdamW optimizer with a learning rate of $2e{-}5$ is used. The default classification head, the Softmax layer, has been replaced by a multi layer perceptron with 2 hidden layers with two nodes each. The output from the transformer models are used as inputs to train the MLP classification heads. For training the MLP classification head, we used the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LM-BFGS) solver. The learning rate of the MLP was set to $1e{-}5$. During the training phase, only the transformer models and the classification heads are trained and the Majority Voting Ensemble is not used. The training was performed for 5 epochs as no further improvement in performance was observed by training for higher epochs.

Table 2
Details of dataset used.

	CONSTRAINT2021					Dataset used	
	Fake	Hate	Offense	Defame	Non hostile	Fake	Non-fake
Training	1144	792	742	564	3050	1144	4584
Validation	160	103	110	77	435	160	435
Testing	300	237	219	169	873	300	1353

Table 3
Performance of XLM-RoBERTa model with different number of hidden layers.

Hidden layers	F1 score						MCC					
	1	2	3	4	5	Average	1	2	3	4	5	Average
4	0.73	0.72	0.72	0.73	0.77	0.734	0.67	0.66	0.66	0.67	0.72	0.676
6	**0.77**	**0.74**	**0.73**	**0.76**	**0.77**	**0.754**	**0.71**	**0.68**	**0.68**	**0.71**	**0.72**	**0.7**
8	0.74	0.74	0.71	0.74	0.77	0.74	0.68	0.68	0.66	0.68	0.72	0.684
10	0.77	0.74	0.72	0.74	0.76	0.746	0.71	0.68	0.66	0.68	0.71	0.688
12	0.78	0.75	0.72	0.72	0.76	0.746	0.73	0.69	0.66	0.66	0.71	0.69

3.4. Testing phase

After the model is trained and the weights have converged, the model can be used for testing input text. The input text is pre-processed and tokenized similar to the training data. It is then fed to the three transformer based models for classification. The transformer models provide their outputs to the classification heads, which in turn, predict the class of the test data. The output predictions of all three transformer models are fed to an ensemble classifier which outputs the final class of the input text. The model is evaluated on the metrics of accuracy, precision, recall and F1 score. In addition, we also use the Matthews Correlation Coefficient (MCC) to quantify model performance (Chicco and Jurman, 2020; Boughorbel et al., 2017). The MCC provides a much better estimate of model performance than the F1 score as it takes into account the number of true negatives as well.

4. Results and discussion

4.1. Experimental setup

We have used the Colab environment to conduct our experiments with a Tesla K80 GPU with 16 GB RAM. The testbed was written in the Python language with the help of the PyTorch framework. The pre-trained transformer models were taken from the HuggingFace library. The MLP was implemented using the Scikit-learn python library.

4.2. Datasets used

There is a lack of available datasets for fake news detection in Hindi. The most popular dataset for this task is the CONSTRAINT2021 dataset (Bhardwaj et al., 2020). This dataset consists of a total of 8192 online posts, out of which 4358 posts are non-hostile, while the remaining 3834 posts display some nature of hostility. The hostile posts are further sub-classified into defamatory posts, fake news, hate speech and offensive posts. Despite not being a dataset focused exclusively on the task of fake news detection, the CONSTRAINT2021 dataset has been used extensively for this task. This can be done by considering a fine grained approach to classification wherein the objective is to distinguish between different different hostility classes. Alternately it is also possible to use a One-vs-All approach wherein all the posts which are not labelled as fake news in the dataset are labelled as being factual news. In this work, we have followed the One-vs-All approach. Table 2 provides a description of the CONSTRAINT2021 dataset and how the dataset was used in our work.

4.3. Results

4.3.1. Hyperparameter tuning of individual transformer models

In this section, the results of hyperparameter tuning for the individual transformer models are described. The hyperparameter tuning was performed by using 5-fold cross validation on the CONSTRAINT2021 dataset.

Table 3 describes the 5-fold cross validation results of the XLM-RoBERTa architecture for different number of hidden layers. It can be seen that the best performance in terms of F1 score and MCC are obtained when 6 hidden layers are used. Table 4 provides the 5-fold cross validation results for different batch sizes in XLM-RoBERTa. We have experimented with the popular batch sizes — 16, 32 and 64. It can be seen that the batch size of 32 provides the best results for our model. Additionally, Table 5 shows the effect of learning rate on the transformer models. As the learning rate of $2e{-}5$ showed the best performance, it has been used for our work. Similar results were observed for all transformer models.

Table 6 describes the performance of different MLP head architectures. We have experimented with 6 different architectures, two of which have one hidden layer and the remaining four having 2 hidden layers each. We observed that a 2×2 architecture, with 2 hidden layers having 2 nodes each gave the best result in terms of the average F1 and MCC scores, while providing lower variance. We refrained from testing with more complex architectures to avoid the possibility of overfitting. Table 7 describes the performance of the chosen 2×2 MLP head architecture with different activation functions — identity, sigmoid, tanh and ReLU. The activation functions demonstrate a similar performance with the ReLU activation function demonstrating a slightly better performance.

The use of an MLP head for classification instead of the default softmax classification head provides an improvement in the performance of the system in terms of both F1 score and MCC value. This is shown in Table 8 using 5-fold cross validation. Due to this improvement, we use an MLP architecture with a 2×2 hidden layer architecture instead of the default softmax head.

4.3.2. Ensemble selection

In this work, we have experimented with stacking and voting techniques for combining the results of the transformer models. In stacking, we have used three classifiers to combine the outputs generated by our transformer models — Multi Layer Perceptron (MLP), Support Vector Machines (SVM), Decision Trees (DT). For the MLP ensembling technique, we fed the results of the three transformer models to a neural network with a single hidden layer of two neurons each. The SVM ensemble utilizes an SVM classifier with an RBC kernel for performing the ensembling. The Decision Tree ensemble uses a Decision Tree constructed using the Gini Index as the attribute selection measure and

Table 4
Effect of batch size.

Batch size	F1 score							MCC						
	1	2	3	4	5	Avg	Std Dev.	1	2	3	4	5	Avg	Std. Dev.
16	0.77	0.74	0.74	0.74	0.75	0.748	0.0117	0.72	0.68	0.69	0.68	0.69	0.692	0.015
32	**0.77**	**0.74**	**0.74**	**0.75**	**0.76**	**0.752**	**0.012**	**0.72**	**0.68**	**0.69**	**0.69**	**0.71**	**0.698**	**0.015**
64	0.76	0.72	0.72	0.73	0.76	0.738	0.018	0.71	0.65	0.66	0.66	0.71	0.678	0.026

Table 5
Effect of learning rate.

Learning rate	F1 score						MCC					
	1	2	3	4	5	Avg	1	2	3	4	5	Avg
2e−4	0.68	0.68	0.69	0.63	0.7	0.676	0.6	0.6	0.61	0.54	0.63	0.596
2e−5	**0.77**	**0.74**	**0.73**	**0.76**	**0.77**	**0.754**	**0.71**	**0.68**	**0.68**	**0.71**	**0.72**	**0.7**
2e−6	0.71	0.68	0.69	0.68	0.74	0.7	0.64	0.6	0.61	0.6	0.68	0.626

Table 6
Performance of MLP Head with different architectures.

Hidden Layer Arch.	F1 score							MCC						
	1	2	3	4	5	Avg	Std Dev.	1	2	3	4	5	Avg	Std. Dev.
(2,)	0.78	0.71	0.73	0.73	0.77	0.744	0.027	0.73	0.65	0.67	0.67	0.71	0.686	0.029
(3,)	0.78	0.72	0.74	0.74	0.77	0.75	0.022	0.73	0.66	0.68	0.68	0.72	0.694	0.027
(2,2)	**0.75**	**0.76**	**0.73**	**0.74**	**0.77**	**0.75**	**0.014**	**0.69**	**0.7**	**0.67**	**0.68**	**0.71**	**0.69**	**0.014**
(2,3)	0.75	0.75	0.73	0.74	0.77	0.748	0.013	0.69	0.69	0.67	0.68	0.72	0.69	0.017
(3,2)	0.75	0.75	0.73	0.74	0.77	0.748	0.013	0.7	0.69	0.68	0.68	0.72	0.694	0.015
(3,3)	0.75	0.75	0.73	0.74	0.77	0.748	0.013	0.7	0.69	0.68	0.68	0.72	0.694	0.015

Table 7
Performance of different activation functions in the MLP head.

Activation function	F1 score							MCC						
	1	2	3	4	5	Avg	Std Dev.	1	2	3	4	5	Avg	Std. Dev.
Identity	0.75	0.75	0.73	0.74	0.77	0.748	0.0133	0.69	0.69	0.67	0.68	0.72	0.69	0.0167
Logistic Sigmoid	0.75	0.76	0.73	0.74	0.77	0.75	0.014	0.7	0.7	0.67	0.68	0.72	0.694	0.017
tanh	0.75	0.75	0.73	0.74	0.76	0.746	0.01	0.7	0.69	0.68	0.68	0.71	0.692	0.012
ReLU	**0.75**	**0.76**	**0.73**	**0.74**	**0.77**	**0.75**	**0.014**	**0.69**	**0.7**	**0.67**	**0.68**	**0.71**	**0.69**	**0.014**

Table 8
Effect of MLP head on fake news detection.

Model	Classif. head	F1 score						MCC					
		1	2	3	4	5	Avg	1	2	3	4	5	Avg
XLM-RoBERTa	Default	0.73	0.75	0.72	0.72	0.77	0.738	0.67	0.69	0.66	0.66	0.71	0.678
	MLP	**0.74**	**0.75**	**0.73**	**0.73**	**0.77**	**0.744**	**0.69**	**0.69**	**0.67**	**0.67**	**0.72**	**0.688**
mBERT	Default	0.75	0.71	0.72	0.72	0.75	0.73	0.69	0.64	0.65	0.65	0.69	0.664
	MLP	**0.75**	**0.7**	**0.72**	**0.73**	**0.76**	**0.732**	**0.69**	**0.63**	**0.66**	**0.67**	**0.7**	**0.67**
ELECTRA	Default	0.7	0.67	0.67	0.65	0.71	0.68	0.63	0.6	0.61	0.57	0.64	0.61
	MLP	**0.7**	**0.68**	**0.67**	**0.66**	**0.71**	**0.684**	**0.64**	**0.61**	**0.6**	**0.58**	**0.65**	**0.616**

Table 9
Performance summary of transformer models using 5 fold cross validation.

Parameter	ELECTRA		mBERT		XLM-RoBERTa		Ensemble	
	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Accuracy	0.882	0.00836	0.894	0.0167	0.898	0.0083	0.904	0.011
Recall	0.604	0.0296	0.72	0.037	0.688	0.042	0.69	0.025
Precision	0.756	0.032	0.736	0.035	0.778	0.0083	0.804	0.0167
F1 score	0.672	0.0192	0.728	0.0327	0.73	0.00707	0.742	0.0178
MCC	0.61	0.0255	0.664	0.0397	0.666	0.026	0.688	0.0258

performs the ensembling. These three classifiers were trained using the outputs generated by the transformer models on the training data.

We have also experimented with two voting techniques — Majority Voting (MV) and Majority Voting with Veto capability (MV-V) (More and Gaikwad, 2016; Shahzad and Lavesson, 2013). In the Majority Voting technique, the class label which is predicted by two out of the three transformer models is chosen as the final prediction. In the MV-V technique, a transformer prediction with a confidence exceeding a

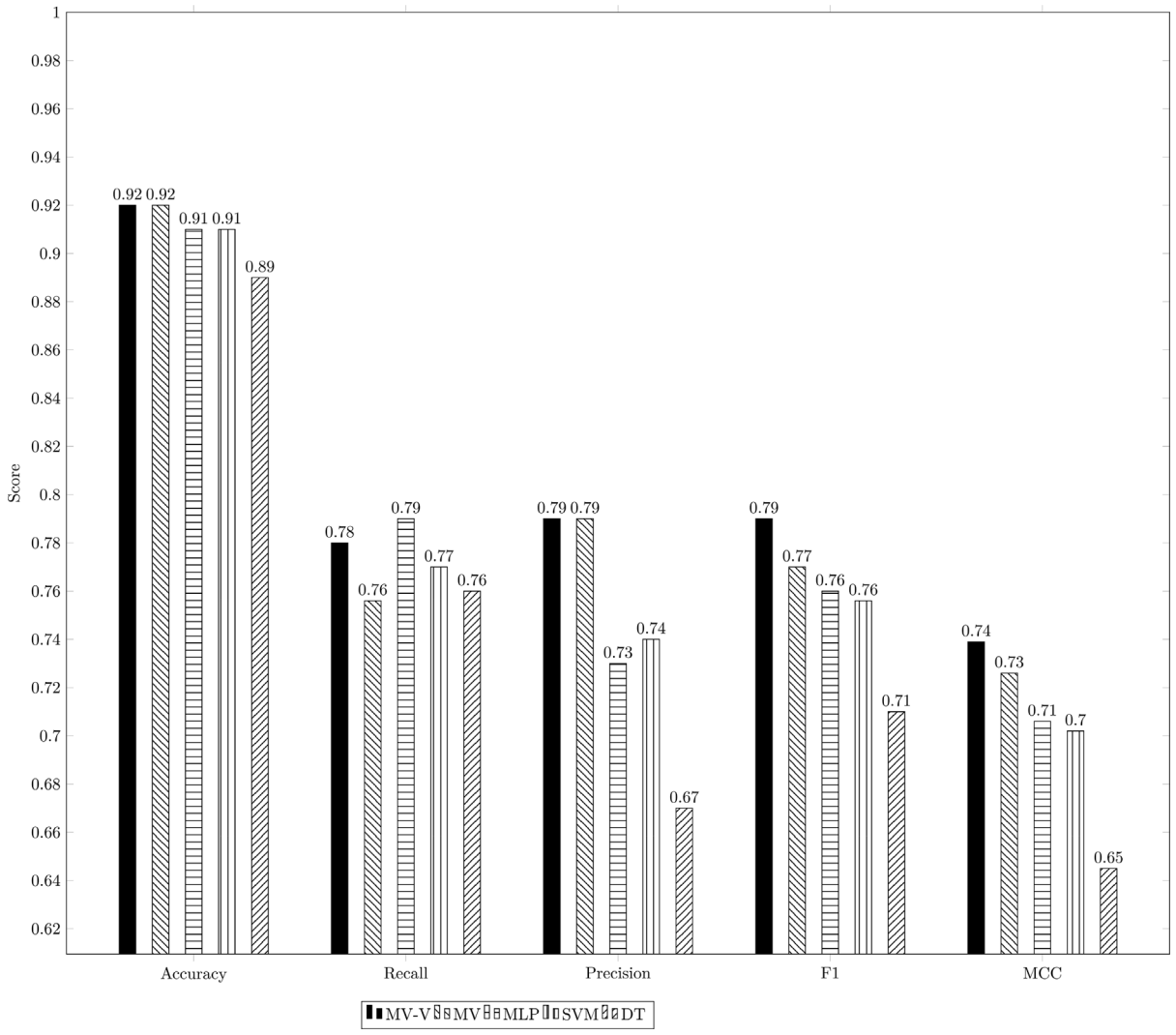


Fig. 3. Performance metrics of the model for different ensembling techniques.

particular threshold θ is allowed to veto the predictions of the other two transformer models if they do not have a similar confidence level. In this work, we have used a θ value of 0.9.

Fig. 3 shows the performance of the model using different ensembling techniques. It can be seen that the MV-V ensembling technique performs better. A possible reason for this is that the use of stacking further complicates the model and leads to overfitting. Hence, in this work, we have used a Majority Voting technique with Veto capability for combining the results of the three transformer models while avoiding overfitting.

4.3.3. Performance of the model

In order to effectively evaluate the performance of the proposed model, we used 5-fold cross validation technique. The entire dataset was divided into 5 non-overlapping equal parts, and for each test run, one part was selected as the testing dataset, while the other 4 parts were jointly used as the training data. Table 9 shows the performance of the ensemble model along with the three individual transformer based models on the CONSTRAINT2021 dataset for the task of fake news detection. The results in Table 9 are the average results obtained from the 5-fold cross validation performed on the CONSTRAINT2021 dataset.

The comparison has been made on five metrics — accuracy, precision, recall, F1 score and the Matthews Correlation Coefficient (MCC). Arguably, MCC is the most suitable metric to represent the model as it is a combination of all four parameters in the confusion matrix. An MCC score of 1 shows that the model predicted all the test data correctly, while a score of −1 shows that the model predicts all the data incorrectly. An MCC score of 0 shows that the model performs similar to a coin flip, and classifies half of the data correctly and the other half incorrectly. An MCC score close to 1 is, therefore, the target for binary classification tasks.

ELECTRA gives an MCC value of 0.61, which is considerably good for a simple model as demonstrated here. This clearly indicates that ELECTRA is suitable for use for the task of fake news detection in Hindi. However, ELECTRA is easily outperformed by mBERT which gives an MCC value of 0.664. This is due to the fact that mBERT has a much larger model and consequently, has a much greater expressive power. Additionally, mBERT is trained on a much larger corpus than ELECTRA. XLM-RoBERTa outperforms mBERT by a small margin, and demonstrates an MCC value of 0.668. Once again, this is due to the fact that XLM-RoBERTa is trained on a much larger corpus and is a heavily optimized version of BERT.

The ensemble model outperforms all of the transformer models and gives an MCC score of 0.688. It also outperforms the individual transformer models in all the other metrics like accuracy, precision, recall and F1 score. This clearly demonstrates that an ensemble model of transformer models is able to detect fake news in Hindi at a reasonably high level.

Table 10
Performance comparison — ELECTRA with ensemble.

Parameter	Model	Fold - 1	Fold - 2	Fold - 3	Fold - 4	Fold - 5	t-Score	*p*-Value	Significant(Yes/No)
Accuracy	ELECTRA	0.87	0.89	0.89	0.88	0.88	5.8797	0.002	Yes
	Ensemble	0.9	0.92	0.91	0.9	0.89			
Recall	ELECTRA	0.61	0.65	0.6	0.57	0.59	7.483	0.00085	Yes
	Ensemble	0.69	0.71	0.72	0.66	0.67			
Precision	ELECTRA	0.73	0.74	0.78	0.73	0.8	3.1379	0.017	Yes
	Ensemble	0.79	0.83	0.81	0.79	0.8			
F1 score	ELECTRA	0.67	0.69	0.68	0.64	0.68	12.78	0.000108	Yes
	Ensemble	0.74	0.76	0.76	0.72	0.73			
MCC	ELECTRA	0.6	0.63	0.62	0.57	0.63	8.045	0.000648	Yes
	Ensemble	0.68	0.72	0.71	0.66	0.67			

Table 11
Performance comparison — mBERT with ensemble.

Parameter	Model	Fold - 1	Fold - 2	Fold - 3	Fold - 4	Fold - 5	t-Score	*p*-Value	Significant(Yes/No)
Accuracy	mBERT	0.89	0.92	0.9	0.88	0.88	3.1623	0.017	Yes
	Ensemble	0.9	0.92	0.91	0.9	0.89			
Recall	mBERT	0.75	0.75	0.73	0.66	0.71	−2.74	0.974	No
	Ensemble	0.69	0.71	0.72	0.66	0.67			
Precision	mBERT	0.74	0.79	0.74	0.71	0.7	6.3688	0.00155	Yes
	Ensemble	0.79	0.83	0.81	0.79	0.8			
F1 score	mBERT	0.74	0.77	0.73	0.68	0.72	1.5097	0.1028	No
	Ensemble	0.74	0.76	0.76	0.72	0.73			
MCC	mBERT	0.67	0.72	0.67	0.61	0.65	2.588	0.0304	Yes
	Ensemble	0.68	0.72	0.71	0.66	0.67			

Table 12
Performance comparison — XLM-RoBERTa with ensemble.

Parameter	Model	Fold - 1	Fold - 2	Fold - 3	Fold - 4	Fold - 5	t-Score	*p*-Value	Significant(Yes/No)
Accuracy	XLM-RoBERTa	0.89	0.91	0.9	0.9	0.89	2.4495	0.035	Yes
	Ensemble	0.9	0.92	0.91	0.9	0.89			
Recall	XLM-RoBERTa	0.67	0.78	0.7	0.69	0.7	1.516	0.44	No
	Ensemble	0.69	0.71	0.72	0.66	0.67			
Precision	XLM-RoBERTa	0.77	0.79	0.77	0.78	0.78	4.33	0.00616	Yes
	Ensemble	0.79	0.83	0.81	0.79	0.8			
F1 score	XLM-RoBERTa	0.72	0.73	0.73	0.73	0.74	0.3714	0.3646	No
	Ensemble	0.74	0.76	0.76	0.72	0.73			
MCC	XLM-RoBERTa	0.62	0.68	0.68	0.67	0.68	1.5795	0.095	No
	Ensemble	0.68	0.72	0.71	0.66	0.67			

Table 13
Sample contingency table for McNemar's test.

	Model 2 Correct	Model 2 Wrong
Model 1 Correct	n_{11}	n_{10}
Model 1 Wrong	n_{01}	n_{00}

For example, one of the statements in our test set was फटा दूध पीने से इम्युनिटी बढ़ती है और ऐसा करना कोरोना से लड़ने के लिए फायदेमंद है, which spreads misinformation about how spoilt milk can help build immunity against Covid-19. This statement was reported as Fake by all three of the transformer models, and consequently by the ensemble. Another statement, कांग्रेस पार्टी की राष्ट्रीय संवादाता प्रियंका चतुर्वेदी ने ये बयान दिया जिस फौजी ने बेकसूरआतंकवादी को घसीटा उसको फांसी नहीं हुई तो देश में आग लगा देंगे, which is fake news, but is voted by XLM-RoBERTa as Non-Fake. Both mBERT and ELECTRA predict the text as Fake, and our ensemble predicts it as Fake. An approach relying on purely XLM-RoBERTa would have predicted the text as Non-Fake, adding to the count of false negatives. This example illustrates the advantage of using multiple transformer models for predicting fake news.

Tables 10–12 give a detailed comparison of the performance of the three transformer models – XLM-RoBERTa, mBERT and ELECTRA – with the ensemble model for each of the 5 folds in the cross validation experiments. A statistical right tailed t-test was also performed to analyse if the results obtained are statistically significant. The results of the t-test are also given in Tables 10–12. It shows that the ensemble model outperforms the individual transformer models and the improved performance is statistically significant in majority of the instances.

4.4. Comparison with existing research works

We have compared the performance of our model with three research works which use the same CONSTRAINT2021 dataset. Since these research works have not used the MCC score for evaluation, instead relying on the F1 score, we have used the F1 score for the comparison. The baseline model was described by Bhardwaj et al. (2020), the research work R1 denotes the work done by Kamal et al. (2021), and R2 represents the work done by Gupta et al. (2021). Fig. 4 provides a comparison between these existing research works with our proposed mechanism. The F1 scores reported by these research works were based on the training–testing split provided in the CONSTRAINT2021 dataset. To provide a more effective comparison, the F1 scores quoted for our work were also computed on the same split. Fig. 4 clearly shows that our proposed mechanism outperforms all three of these research works with respect to the F1 score. This proves that transformer ensembles are capable of detecting Hindi fake news with a considerable level of efficiency.

Table 14
McNemar's test results.

	Ensemble correct	Ensemble wrong	Chi Square value	*p*-Value
XLM-RoBERTa Correct	1456	20	1.653	0.198
XLM-RoBERTa Wrong	29	133		
mBERT Correct	1438	30	3.753	0.0527
mBERT Wrong	47	123		
ELECTRA Correct	1420	30	12.895	0.00033
ELECTRA Wrong	65	123		

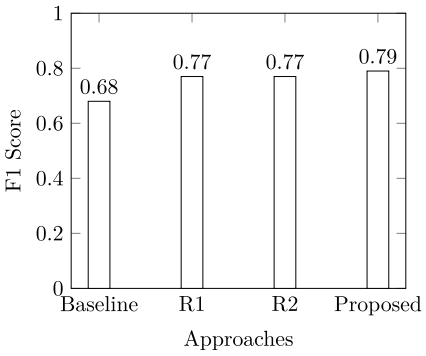


Fig. 4. Comparison with existing works.

4.5. McNemar's test

In order to show that the results obtained using the ensemble model are statistically significant, we use the McNemar's Test. McNemar's test is a nonparametric statistical test for paired nominal data. McNemar's test can be used to compare the predictive accuracy of two machine learning models. The test has also been used extensively in the domain of fake news detection for comparing two models (Jiang et al., 2021; Giachanou et al., 2020). The test is based on a 2×2 contingency table of the two model's predictions on the test dataset, similar to the table shown in Table 13. McNemar's chi-squared test statistic is computed as

$$\chi^2 = \frac{(n_{01} - n_{10})^2}{n_{10} + n_{01}} \tag{1}$$

Contingency matrices for the individual transformer based classifiers and the ensemble model are given in Table 14. These results are the consolidated results from the 5 fold cross validation. After setting a significance threshold α and computing the test statistic, the p-value is computed, which is the probability of observing this empirical chi-squared value. If the p-value is lower than our chosen significance level, there is a statistically significant difference between the models. The results of McNemar's Test is significant with $\alpha = 0.05$ for ELECTRA, and at $\alpha = 0.1$ for mBERT. For XLM-RoBERTa, the results are significant at a lower confidence level. However, the test does show significance for majority of the individual folds even for XLM-RoBERTa. Table 14 further underlines the fact that the performance difference between the ensemble model and the individual transformer models is statistically significant.

5. Conclusion

In this work, we undertake the critical and relevant problem of detection of fake Hindi news. Due to the rise of users in social media, it is necessary to keep the circulating information accurate. The complexity of fake news detection is even more prominent in a resource constrained language such as Hindi which is still extensively unexplored. This work presents a mechanism of identifying Hindi fake news using an ensemble of pre-trained transformer models. The use of multiple transformer models – each working on a slightly different principle and pretrained on different datasets – improves the overall performance of the system by compensating for the errors that occur in a single transformer model, especially for resource constrained languages like Hindi. For this purpose, three pre-trained transformer models – ELECTRA, mBERT and XLM-RoBERTa – were fine-tuned for the task of fake news detection. The results from the individual transformer models are combined with a majority voting ensemble with veto capability. We demonstrate that the ensemble model is capable of detecting Hindi fake news with an accuracy of over 90%. In addition, the model also has high values of precision, recall, F1 score and MCC score, which makes it suitable for Hindi fake news detection. This also provides additional proof that ensembling can improve the performance of NLP tasks, including fake news detection, on resource constrained languages.

CRediT authorship contribution statement

Amit Praseed: Conceptualization, Methodology, Software, Writing – original draft. **Jelwin Rodrigues:** Conceptualization, Methodology, Software. **P. Santhi Thilagam:** Conceptualization, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data used for this research is open source and is available publicly.

References

Badam, J., Bonagiri, A., Raju, K., Chakraborty, D., 2022. Aletheia: A fake news detection system for Hindi. In: 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD). In: CODS-COMAD 2022, Association for Computing Machinery, New York, NY, USA, pp. 255–259. http://dx.doi.org/10.1145/3493700.3493736.

Bhardwaj, M., Akhtar, M.S., Ekbal, A., Das, A., Chakraborty, T., 2020. Hostility detection dataset in Hindi. arXiv:2011.03588.

Boughorbel, S., Jarray, F., El-Anbari, M., 2017. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. PLoS One 12 (6), e0177678.

Briskilal, J., Subalalitha, C., 2022. An ensemble model for classifying idioms and literal texts using BERT and RoBERTa. Inf. Process. Manage. 59 (1), 102756. http://dx.doi.org/10.1016/j.ipm.2021.102756, URL https://www.sciencedirect.com/science/article/pii/S0306457321002375.

Brown, G., Kuncheva, L.I., 2010. "good" and "bad" diversity in majority vote ensembles. In: International Workshop on Multiple Classifier Systems. Springer, pp. 124–133.

Chicco, D., Jurman, G., 2020. The advantages of the matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics 21 (1), 1–13.

Clark, K., Luong, M.-T., Le, Q.V., Manning, C.D., 2020. Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Ekbal, A., Saha, S., 2011. A multiobjective simulated annealing approach for classifier ensemble: Named entity recognition in Indian languages as case studies. Expert Syst. Appl. 38 (12), 14760–14772.

Flintham, M., Karner, C., Bachour, K., Creswick, H., Gupta, N., Moran, S., 2018. Falling for fake news: investigating the consumption of news via social media. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. pp. 1–10.

Fridkin, K., Kenney, P.J., Wintersieck, A., 2015. Liar, liar, pants on fire: How fact-checking influences citizens' reactions to negative advertising. Political Commun. 32 (1), 127–151.

Giachanou, A., Zhang, G., Rosso, P., 2020. Multimodal multi-image fake news detection. In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics. DSAA, IEEE, pp. 647–654.

Gupta, A., Sukumaran, R., John, K., Teki, S., 2021. Hostility detection and covid-19 fake news detection in social media. arXiv preprint arXiv:2101.05953.

Hanna, R., Rohm, A., Crittenden, V.L., 2011. We're all connected: The power of the social media ecosystem. Bus. Horiz. 54 (3), 265–273.

Jiang, T., Li, J.P., Haq, A.U., Saboor, A., Ali, A., 2021. A novel stacking approach for accurate detection of fake news. IEEE Access 9, 22626–22639.

Kamal, O., Kumar, A., Vaidhya, T., 2021. Hostility detection in Hindi leveraging pre-trained language models. arXiv preprint arXiv:2101.05494.

Kar, D., Bhardwaj, M., Samanta, S., Azad, A.P., 2020. No rumours please! A multi-indic-lingual approach for COVID fake-tweet detection. arXiv preprint arXiv:2010.06906.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kioutsioukis, I., Galmarini, S., 2014. De praeceptis ferendis: good practice in multi-model ensembles. Atmos. Chem. Phys. 14 (21), 11791–11815.

Kulkarni, D.S., Rodd, S.S., 2021. Sentiment analysis in Hindi—A survey on the state-of-the-art techniques. Trans. Asian Low-Resource Lang. Inf. Process. 21 (1), 1–46.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Loshchilov, I., Hutter, F., 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101.

Meetei, L.S., Singh, T.D., Borgohain, S.K., Bandyopadhyay, S., 2021. Low resource language specific pre-processing and features for sentiment analysis task. Lang. Res. Eval. 55 (4), 947–969.

Mehta, M., Pandey, U., Chaudhary, Y., Sharma, R., Gill, I., Gupta, D., Khanna, A., 2021. Hindi text classification: A review. In: 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N). IEEE, pp. 839–843.

More, S.S., Gaikwad, P.P., 2016. Trust-based voting method for efficient malware detection. Procedia Comput. Sci. 79, 657–667.

Nelson, J.L., Lei, R.F., 2018. The effect of digital platforms on news audience behavior. Digit. J. 6 (5), 619–633.

Nozza, D., 2022. Nozza@LT-EDI-ACL2022: Ensemble modeling for homophobia and transphobia detection. In: Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion. Association for Computational Linguistics, Dublin, Ireland, pp. 258–264. http://dx.doi.org/10.18653/v1/2022.ltedi-1.37, URL https://aclanthology.org/2022.ltedi-1.37.

Pan, S.J., Yang, Q., 2009. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Ray, S.K., Ahmad, A., Kumar, C.A., 2019. Review and implementation of topic modeling in Hindi. Appl. Artif. Intell. 33 (11), 979–1007. http://dx.doi.org/10.1080/08839514.2019.1661576.

Saghayan, M.H., Ebrahimi, S.F., Bahrani, M., 2021. Exploring the impact of machine translation on fake news detection: A case study on Persian tweets about COVID-19. In: 2021 29th Iranian Conference on Electrical Engineering. ICEE IEEE, pp. 540–544.

Saha, S., Ekbal, A., 2013. Combining multiple classifiers using vote based classifier ensemble technique for named entity recognition. Data Knowl. Eng. 85 15–39.

Sajjad, H., Dalvi, F., Durrani, N., Nakov, P., 2020. On the effect of dropping layers of pre-trained transformer models. arXiv preprint arXiv:2004.03844.

Shahzad, R.K., Lavesson, N., 2013. Comparative analysis of voting schemes for ensemble-based malware detection. J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. 4 (1), 98–117.

Sharif, O., Hoque, M.M., 2022. Tackling cyber-aggression: Identification and fine-grained categorization of aggressive texts on social media using weighted ensemble of transformers. Neurocomputing 490, 462–481. http://dx.doi.org/10.1016/j.neucom.2021.12.022, URL https://www.sciencedirect.com/science/article/pii/S0925231221018567.

Sharma, Y., Mangat, V., Kaur, M., 2015a. A practical approach to sentiment analysis of Hindi tweets. In: 2015 1st International Conference on Next Generation Computing Technologies. NGCT, IEEE, pp. 677–680.

Sharma, P., Moh, T.-S., 2016. Prediction of Indian election using sentiment analysis on Hindi Twitter. In: 2016 IEEE International Conference on Big Data (Big Data). IEEE, pp. 1966–1971.

Sharma, D.K., et al., 2015b. A comparative analysis of Hindi word sense disambiguation and its approaches. In: International Conference on Computing, Communication & Automation. IEEE, pp. 314–321.

Shekhar, C., Bagla, B., Maurya, K.K., Desarkar, M.S., 2021. Walk in wild: An ensemble approach for hostility detection in Hindi posts. arXiv preprint arXiv:2101.06004.

Srivastava, S., Sanglikar, M., Kothari, D., 2011. Named entity recognition system for Hindi language: a hybrid approach. Int. J. Comput. Linguist. 2 (1), 10–23.

Su, T., Macdonald, C., Ounis, I., 2019. Ensembles of recurrent networks for classifying the relationship of fake news titles. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 893–896.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008.

Wu, S., Dredze, M., 2020. Are all languages created equal in multilingual BERT? In: Proceedings of the 5th Workshop on Representation Learning for NLP. Association for Computational Linguistics, Online, pp. 120–130. http://dx.doi.org/10.18653/v1/2020.repl4nlp-1.16, URL https://aclanthology.org/2020.repl4nlp-1.16.

Zhou, X., Zafarani, R., 2020. A survey of fake news: Fundamental theories, detection methods, and opportunities. ACM Comput. Surv. 53 (5), 1–40.