AUTOMATED MULTI-MODEL FAKE NEWS CLASSIFIER

[1]BHANU JOSHI, [2]SARAH VEJLANI, [3]SIDDHANT DUBEY, [4]NEELAM PHADNIS

[1]UG Student, [2]UG Student, [3]UG Student, [4]Assistant Professor
[1]Computer Engineering,
[1]Shree L.R. Tiwari College of Engineering, Thane, India
[1]jbhanu718@gmail.com, [2]sarahvejlani@gmail.com, [3]siddhantd965@gmail.com, [4]kulkarnineelam@gmail.com

Abstract

The widespread increase in fake news, whether created by humans or machines, has a negative impact on society and individuals on both a political and social level. The rapid rotation of news in the age of social media makes it difficult to assess its authenticity quickly. As a result, automated fake news identification tools have become a necessity. To solve the aforementioned problem, a hybrid Neural Network architecture is used, which incorporates the capabilities of CNN and LSTM, as well as two separate dimensionality reduction methods, PCA and Chi-Square.

We'll use data from the Fake News Challenges (FNC) website, which includes four different forms of stances: agree, disagree, discuss, and unrelated. The aim of this study is to figure out what a news article's body is in relation to its headline using different deep learning and ML models.

***Index Terms* - Fake News, Machine Learning, Natural Language Processing, Deep Learning, Multi-Layer Perceptron, Long Short-Term Memory**

I. INTRODUCTION

The problem of "fake news" has recently emerged as a possible challenge to high-quality journalism and informed public debate. "Fake News" is a term that refers to false news or propaganda that is disseminated by conventional media outlets such as newspapers and television, as well as non-traditional outlets such as social media. The general motivation for disseminating such information is to deceive readers, damage the credibility of any person, or profit from sensationalism. It is widely regarded as one of the most serious challenges to freedom, free speech, and the Western order [3]. Fake news is becoming more popular on social media sites like Twitter and Facebook [2]. These outlets provide a forum for the general public to express their thoughts and beliefs in an unfiltered and uncensored manner. As opposed to direct views from media outlets' websites, certain news stories hosted or shared on social media platforms receive more views. According to a study of the speed at which fake news spreads on Twitter, false information spreads six times faster than true information [3]. Inaccurate news has a variety of negative consequences, including convincing people that Hillary Clinton had an alien baby, attempting to persuade readers that President Trump is seeking to repeal the first amendment, and mob killings in India as a result of a false rumor spread on WhatsApp. The problem of "fake news" has recently emerged as a possible challenge to high-quality journalism and informed public debate.

We're doing "stance detection" in this project, which means determining whether a news headline "agrees" with, "disagrees" with, "discusses," or is unrelated to a specific news item, to make it easier for journalists and others to locate and investigate potential stories. The method of automatically detecting the relationship between two pieces of text is known as stance detection. Artificial Intelligence (AI) and Natural Language Processing (NLP) tools hold a lot of promise for researchers who want to develop systems that can detect fake news automatically. Detecting fake news, on the other hand, is a difficult job because it necessitates models that summaries the news and equate it to the real news in order to identify it as fake. Furthermore, comparing proposed news to original news is a difficult task because it is extremely subjective and opinionated. The focus of our research will be on stance identification, which is a different way to detect fake news. In this report, we look at how to predict a person's stance based on a news article and a news headline. The stances between them can be classified as 'agree,' 'disagree,' 'discuss,' or 'unrelated,' depending on how similar the news story material and headlines are.

II. LITERATURE REVIEW

In NLP, stance detection is a well-known and well-researched activity. It is classified as deciding whether the audience is for, against, or neutral on the goal based on the text. Many tasks, such as fake news detection, assertion validation, and argument search, depend on stance detection. Previous fake news detection research based on target-specific stance prediction, which involves determining the stance of a text entity related to a subject or a named entity. Goal-specific stance prediction for tweets (where tweets are the text and the target is a single stance) and online discussions is used in several studies. Such target-specific approaches rely on structural, linguistic, and lexical characteristics.

The issue is one of "stance detection," which entails contrasting a headline with the body of text from a news article to see if there is a relationship (if any). There are four different classifications that can be used:

- The title and the article text are in sync.
- The text of the article contradicts the headline.
- The headline is discussed in the article text, but no stance is taken on it.
- The article text has nothing to do with the headline (i.e., it doesn't deal with the same subject).

To solve the aforementioned problem, a hybrid Neural Network architecture is used, which incorporates the capabilities of CNN and LSTM, as well as two separate dimensionality reduction methods, PCA and Chi-Square.

In today's digital world, where there are thousands of knowledges sharing channels through which false news or disinformation can spread, the pervasive issue of fake news is extremely difficult to combat. It has become a bigger problem as AI advances, bringing with it artificial

bots that can be used to build and distribute false news [1]. The situation is critical because many people believe everything they read on the internet, and those who are inexperienced or new to digital technology are vulnerable to being duped.
Fraud is another problem that may arise as a result of spam or malicious emails and texts. As a result, it is convincing enough to accept the issue and take on the task of reducing violence, civil instability, and sorrow, as well as thwarting attempts to spread false news. Text, or natural language, is a difficult medium to process due to a variety of linguistic characteristics and styles such as sarcasm, metaphors, and so on. There are also thousands of spoken languages, each with its own grammar, script, and syntax. Natural language processing is a subset of artificial intelligence that includes methods for analyzing text, creating models, and making predictions. The aim of this project is to develop a framework or model that can use historical data to predict whether a news story is false or not.

A. Dimensionality Reduction: -

In text categorization, there are two methods for reducing dimensionality: feature extraction and feature selection. The most important and appropriate features are retained in feature selection processes, while the remaining features are discarded [61]. In function extraction methods, on the other hand, the original vector space is transformed into a new vector space with unique properties. In the new vector space, the features are reduced [32].

Reducing features has the advantage of lowering processing speed, which leads to improved efficiency [62]. The results of text classification are greatly influenced by feature reduction.
As a result, selecting the appropriate selection algorithm to minimize dimensions is critical. Some popular feature reduction algorithms include Information Gain (IG), Mutual Information (1v1I) [63], Gini Coefficient (GI), Term Frequency-Inverse Document Frequency (TF-IDF) [64], Principal Component Analysis (PCA), and Chi-Square Statistics (CHI). PCA and Chi-square are two-dimensionality reduction methods that are used in tandem with deep learning models to increase the scalability of the text classifier [72].

B. Principal Component Analysis (PCA): -

Principal Component Analysis (PCA) is a commonly used technique for reducing the dimensions of a feature set using a linear transformation. The resulting dataset is simplified, but it maintains the original data set's characteristics [35]. The new dataset can have the same number of features as the original dataset or less. The principal components are calculated using the covariance matrix. These elements are mentioned in order of decreasing importance [65]. If the original matrix has 'a' dimensions and 'b' observations, and the dimensionality needs to be reduced to a 't' dimensional subspace, then the following equation can be used to transform it.

$$Y = (EZX) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

E a*t is the projection matrix, which includes ‘t' eigen vectors corresponding to ‘t' highest eigen values, and X a*b is the mean centred data matrix in the above equation [72].

C. Chi-Square: -

One of the most effective feature selection algorithms is Chi-Square Statistics [66]. It's made to see if there's a connection between categorical variables. It's used to measure the lack of independence between a and b, as well as to judge extremeness by comparing it to a chi-square distribution of one degree of freedom [67],[63]. Chi-square is used for two types of tests: test for freedom and test for goodness of fit. For feature selection, a test for independence is used, and the target label's reliance on the feature is investigated (s). The Chi-square test looks at the relationship between the features.

The features that have a connection are retained, while the rest are discarded. The chi-square is measured separately for each function in relation to the target class, and its significance is determined using a predefined threshold (which is 0.05 commonly). The lesser the importance of the function, the higher the chi-square value. Similarly, the smaller the chi-square value, the more important the function. Many studies have shown that using chi-square for feature reduction in text categorization improves performance [62], [66]. [72].

D. Activation Function, Max-Pooling and Dropout: -

The ReLu activation function is added to each CNN neuron's output. This activation layer is used to transform every negative value to zero and to demonstrate network nonlinearity. The feature has no impact on the CNN layer's output shape, because it's the same as the input shape. After passing through the ReLu activation mechanism, the value of each neuron is fed into a 1-D max-pooling sheet. This layer selects the maximum value obtained in each kernel to transform the input of each kernel size into a single output. This reduces the size of input features for subsequent layers and prevents overfitting. In our example, the pool size p is 4, so the performance of this layer will reduce the features by the kernel/pool size (p). For the entire network model, the dropout rate D is 0:2. Another way to minimise overfitting is to use the dropout layer, which removes input with values less than the dropout rate. Since no value is lower than 0:2 in the FNC-1 dataset, the output of the dropout layer is the same as the input passed to it [72].

E. Input and Convolution Layer: -

Text Sequence 'a' includes 'w' entries in the dataset. Each entry 'w' is represented by a d-dimensional dense vector. The dimensions of input ‘a’ function map is d *w. We use Keras tokenizer to tokenize the headline and body texts in the first stage. The tokens are then transformed into word-vectors by the Keras embedding layer, which employs word2vec word embedding. The word vectors obtained from the word embedding layer are fed as input to the convolution layer in models one and two. Important features are derived from PCA and Chi-square first for models three and four, on the other hand. The embedding layer then converts these features into word-vectors. Finally, the convolution layer receives these word vectors. Convolution layers are used to extract a particular semantic or structural feature from a given input matrix. CNN neurons n receives each word vector. We can get a variety of features by applying filters of various sizes. On each word embedding e, multiple filters f with different kernel sizes c are applied, and the output is (c *e). In our work, the kernel size is 5, therefore, the filter of size 64 wil***l create 5-word combinations*** [72].

***F.* LSTM: -**

The next layer is LSTM, which has 100 units. We must create a long chain-like sequence structure for our data while keeping track of previous inputs. The LSTM is the best option for this task because it has three gates: input gate i_k, output gate o_k, and forget gate f_k. Based on the dropout value, these gates determine which information is essential for classification and which information is forgettable. Previous input is saved in cell memory block c_K, which is needed for prediction. There are numerous LSTM variants available, but the one we used in our model is as follows.

$$i_k = \sigma\,(w_i s_k + V_i h_k\text{-}1 + b_i) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

$$f_k = \sigma\,(W_f s_k + V_f h_k\text{-}1 + b_f) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

$$o_k = \sigma\,(W_o s_k + V_o h_k\text{-}1 + b_o) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

$c_k = \tanh(W_c x_k + V_c h_k - 1 + b_c)$ …………………………… (5)

where s is the skth vector representation's input sequence (s1, s2, s3, …, Sn). The weights associated with each matrix product are W and V. h represents the hidden state for time phase k1, where sk represents the input at that time and b represents the bias vector. At phase k -1, C is the cell memory block, which is modified each time. Both 100 units in the LSTM layer's output are connected to every unit in the dense layer [72].

***G.* Dense: -**

The proposed model's final layer is a densely connected completely connected layer that produces a single output. A SoftMax activation feature follows this sheet. For multi-class grouping, SoftMax activation is used. Since our dataset comprises four classes, we used SoftMax activation (agree, disagree, discuss, and unrelated). For research purposes, we used Adam as the optimizer. The number of epochs is set to 50, and the batch size used in research is 32 [72].

EXISTING SYSTEM: -

1. Team Athene (Approach)

Throughout the market, we experimented with a variety of approaches, and our model has progressed through a series of phases. Part of the changes show ideas from the fake news challenge slack's discussion of methods. We began by using standard classifiers such as SVM and XGBoost, and then added features such as paraphrase detection, emotion lexica, and others. The methods' performance did not substantially improve over the reported baseline, so we moved on to more advanced techniques [48].
Athene UKP Lab – MLP - ensemble-81.97(Accuracy)

III. Automated Multi-Model Fake News Classifier

The main goal of Automated Multi-Model Fake News Classifier is to identify fake news, which is a standard text classification issue with a simple solution. Building a model that can distinguish between "true" and "fake" news in four categories ("i.e., support, disagree, unrelated, discussed") is needed.

There will be two phases to the project. 1)Pre-processing Dataset and implementing featured vector using TFID Vector Node, Cosine Similarity, Word2Vec, and other methods by the end of the first step
The implementation of the deep learning method (CNN) and computing the result using SoftMax Mapping would be the project's second step.

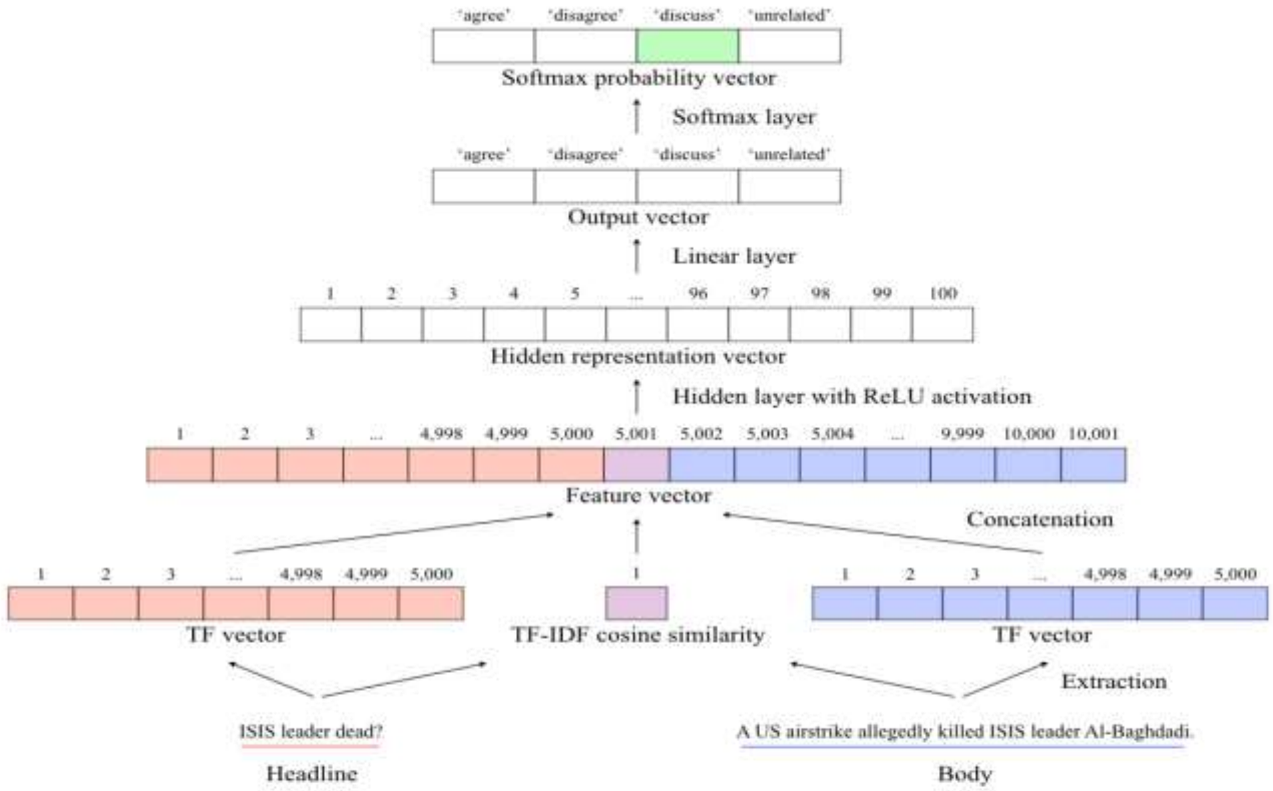

fig. 1. automated multi-model fake news classifier

This research makes the most significant contribution by proposing feature reduction strategies in conjunction with hybrid deep learning models involving two neural network layers, namely CNN and LSTM. When compared to conventional deep learning models, the proposed approach produces better predictive results. Four data models are created to evaluate the relationship. In the first model, all features are used for classification without any preprocessing. After preprocessing, the non-reduced features collection is used in the second model. Using dimensionality reduction techniques such as PCA and Chi-square, the third and fourth models are developed.

The model's first layer is the embedding layer, which takes the input headlines and article bodies and transforms each word into a 100-pixel vector. Since there are 5000 features, this layer can generate a 5000 100 matrix. The weights obtained through matrix multiplication will be included in the output matrix, resulting in a vector for each term. The CNN layer uses these vectors to extract contextual attributes. The CNN layer's output is fed into an LSTM, which is then passed on to a completely connected dense layer, which produces a single stance as the final output. The proposed model is educated and evaluated on small batches of size 32, as shown in Fig. 1.

- *Agree*
    - There is a relation between headline and body of the article
- *Disagree*

- There is no relation between headline and article body

- *Discuss*
  - There is Little bit of match between headline and article body, taking it as a neutral
- *Unrelated*
  - The topic discussed in headline and body are completely different.

A. Application Architecture

The tokens of article w(i) are fed one-by-one into the encoder (a single-layer bidirectional LSTM), producing a sequence of encoder hidden states h(i). On each step t, the decoder (a single-layer unidirectional LSTM) receives the word embedding of the previous word (while training, this is the previous word of the reference summary; at test time it is previous word emitted by the decoder state's(t)).
The attention distribution at is calculated as in Bahdanau et al. (2015):

$$e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{\mathrm{attn}})$$
$$a^t = \mathrm{softmax}(e^t)$$

fig. 2. SoftMax of Output

The attention distribution is used to produce a weighted sum of the encoder hidden states, known as the context vector. The context vector, is concatenated with the decoder state s(t) and fed through two linear layers to produce the vocabulary distribution P(vocab) and provides us with our final distribution from which to predict words w:

P(w) = P[vocab(w)]

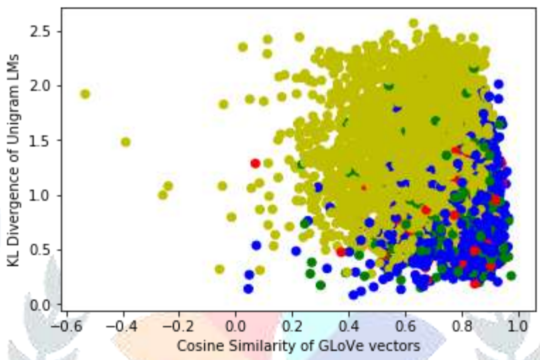

fig. 3. relation between kl and cosine similarity

A.1. Model 1:

Hidden representation vector of dense layer is feed by feature vector consisting of tf-idf, cosine similarity, and kl divergence, with *relu* activation function, which is forwarded to Linear dense layer in order to SoftMax the output vector, to probability vector.
Summary of dense model layer is given below:

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 32)	128
dense_2 (Dense)	(None, 128)	4224
dense_3 (Dense)	(None, 128)	16512
dense_4 (Dense)	(None, 128)	16512
dense_5 (Dense)	(None, 128)	16512
dense_6 (Dense)	(None, 64)	8256
dense_7 (Dense)	(None, 64)	4160
dense_8 (Dense)	(None, 32)	2080
dense_9 (Dense)	(None, 32)	1056
dense_10 (Dense)	(None, 16)	528
dense_11 (Dense)	(None, 16)	272
dense_12 (Dense)	(None, 4)	68

Total params: 70,308
Trainable params: 70,308
Non-trainable params: 0

fig. 4. dense model summary

A.2. Model 2

Padded sequence of article body and headline is feed to model, to SoftMax the output vector, to probability vector, with sigmoid and relu activation function.
Summary of Model is given below:

Layer (type)	Output Shape	Param #	Connected to
input_body (InputLayer)	(None, 64)	0	
input_headline (InputLayer)	(None, 64)	0	
embedding_8 (Embedding)	(None, 64, 50)	1161300	input_body[0][0]
embedding_7 (Embedding)	(None, 64, 50)	1161300	input_headline[0][0]
dense_10 (Dense)	(None, 64, 16)	816	embedding_8[0][0]
dense_9 (Dense)	(None, 64, 16)	816	embedding_7[0][0]
concatenate_4 (Concatenate)	(None, 64, 32)	0	dense_10[0][0] dense_9[0][0]
flatten_3 (Flatten)	(None, 2048)	0	concatenate_4[0][0]
dense_11 (Dense)	(None, 4)	8196	flatten_3[0][0]

Total params: 2,332,428
Trainable params: 2,332,428
Non-trainable params: 0

fig. 5. embedding model summary

A.3. Model 3

Padded sequence of article body and headline is feed to LSTM model, to SoftMax the output vector, to probability vector, with sigmoid and relu activation function.

Summary of Model is given below:

Layer (type)	Output Shape	Param #	Connected to
input_headline (InputLayer)	(None, 16)	0	
input_body (InputLayer)	(None, 48)	0	
embedding_9 (Embedding)	(None, 16, 50)	162750	input_headline[0][0]
embedding_10 (Embedding)	(None, 48, 50)	1152250	input_body[0][0]
concatenate_5 (Concatenate)	(None, 64, 50)	0	embedding_9[0][0] embedding_10[0][0]
lstm_2 (LSTM)	(None, 64)	29440	concatenate_5[0][0]
dropout_2 (Dropout)	(None, 64)	0	lstm_2[0][0]
dense_12 (Dense)	(None, 64)	4160	dropout_2[0][0]
dense_13 (Dense)	(None, 4)	260	dense_12[0][0]

Total params: 1,348,860
Trainable params: 33,860
Non-trainable params: 1,315,000

fig. 6. lstm model summary

A.4. Model 4-15

Ten different Machine learning classification model is feed by feature vector of article body and headline to get the probability output vector. Below are the ten different ML classification model used:

1. Logistic Regression
2. K Nearest Classifier
3. Support Vector Machine Classification
4. Quadratic Discriminant Analysis
5. Random Forest Classifier
6. Adaboost Classifier
7. SGD Classifier
8. Decision Tree Classifier
9. XG Boost Classifier
10. Linear Discriminant Analysis

Graphical User Interface for classification of fake news

The system is designed for client to classify any news is fake or fact in four Stance. By deploying such a web application on servers or cloud, client gets the flexibility to classify any news anywhere.

Following are screenshots of the screens designed for the user interface to classify fake news.

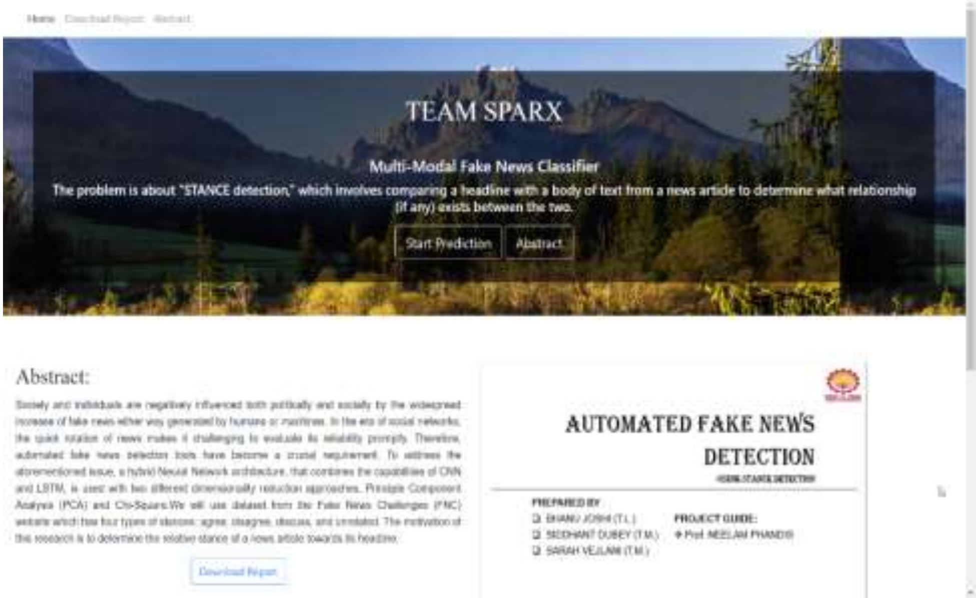

fig. 7. home section

fig. 8. prediction section

Model	Result	Accuracy
Proposed Model	discuss	63.01293989154356
LSTM Model	unrelated	92.71811842918396
Sequential Model	unrelated	15.676026078412201
ML SVM	discuss	NA
ML AdaBoost	discuss	NA
ML Decision Tree	discuss	NA
ML K-Nearest	discuss	NA
ML Logistic Regression	discuss	NA
ML Linear Descriminant	discuss	NA
ML Quadratic Regression	disagree	NA
ML Random Forest	discuss	NA
ML SGD	discuss	NA
ML XGBoost	discuss	NA

fig. 9. result section

IV. Results

The experimental results show that, the implemented system designed for Multi Model fake news Classifier, has provided significant results in classifying a fake news in four stances (i.e., agree, disagree, discuss, unrelated).

Following fig shows the comparison of different ML model

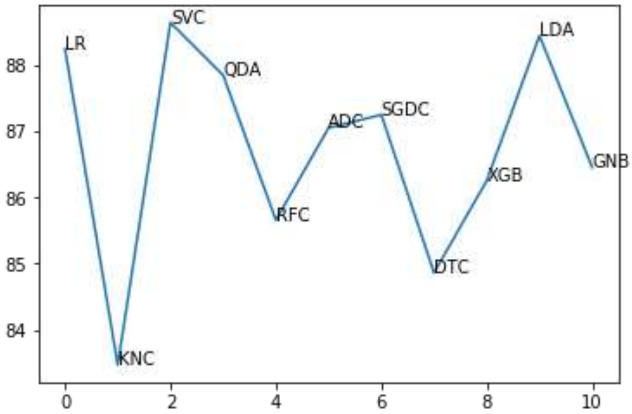

fig. 10. result comparison of ml model

Following table shows the comparison of different model to classify fake news over the dataset of 50,000 news.

Table 1 :results of multi model to classify news

Model	Model Name	Accuracy (in %)
Model 1	Simple Linear Dense Model	87.5
Model 2	Embedding Dense Model	86.34
Model 3	LSTM Dense Model	73.20
Model 4	Logistic Regression	88.2
Model 5	K Nearest Classifier	83.5
Model 6	Support Vector Machine Classification	88.6
Model 7	Quadratic Discriminant Analysis	87.8
Model 8	SGD Classifier	87.3
Model 9	Decision Tree Classifier	84.9
Model 10	XG Boost Classifier	86.3
Model 11	Linear Discriminant Analysis	88.4
Model 12	Random Forest Classifier	85.7
Model 13	Adaboost Classifier	87.1

V. Conclusion And Future Work

This paper presented the results of various model to detect/classify the fake news. The work presented here is a novel on this subject domain because it shows the results of a comprehensive research project that started with quality implementation of the classification model. The work presented in this paper is also promising, because it is demonstrating an effective level of machine learning and deep learning techniques to classify a fake news.

Future planned research efforts involve more research on feature vector to classify fake news more effectively among related category (i.e., agree & discuss)

References

[1] T. Mihaylov, G. Georgiev, and P. Nakov, “Finding opinion manipulation trolls in news community forums,” in Proceedings of the Nineteenth Conference on Computational Natural Language Learning. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 310–314. [Online]. Available: https://www.aclweb.org/anthology/K15-1032

[2] T. Mihaylov, I. Koychev, G. Georgiev, and P. Nakov, “Exposingpaid opinion manipulation trolls,” in Proceedings of the International Conference Recent Advances in Natural Language Processing. Hissar, Bulgaria: INCOMA Ltd. Shoumen, BULGARIA, Sep. 2015, pp. 443–450. [Online]. Available: https://www.aclweb.org/anthology/R15-1058

[3] T. Mihaylov and P. Nakov, “Hunting for troll comments in newscommunity forums,” Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016. [Online]. Available: http://dx.doi.org/10.18653/v1/P16-2065

[4] P. Bourgonje, J. Moreno Schneider, and G. Rehm, “From clickbait to fake news detection: An approach based on detecting the stance of headlines to articles,” in Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 84–89. [Online]. Available: https://www.aclweb.org/anthology/W17-4215

[5] S. Vosoughi, D. Roy, and S. Aral, “The spread of true and false news online,” Science, vol. 359, pp. 1146–1151, 03 2018.

[6] A. M. Michael Barthel and J. Holcomb, “Many americans believe fake news is sowing confusion,” https://www.journalism.org/2016/12/15/manyamericans- believe-fake-news-is-sowing-confusion/, 2016, online; accessed 29 September 2019.

[7] A. K. Chaudhry, “Stance detection for the fake news challenge : Identifying textual relationships with deep neural nets.”

[8] S. Chopra, “Towards automatic identification of fake news: Headlinearticle stance detection with lstm attention models,” 2017.

[9] G. Bhatt, A. Sharma, S. Sharma, A. Nagpal, B. Raman, and Mittal, “Combining neural, statistical and external features for fake news stance identification,” in Companion Proceedings of the The Web Conference 2018, ser. WWW ’18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 1353–1357. [Online]. Available: https://doi.org/10.1145/3184558.3191577

[10] L. Konstantinovskiy, O. Price, M. Babakar, and A. Zubiaga, “Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection,” ArXiv, vol. abs/1809.08193, 2018.

[11] D. Lazer, M. Baum, Y. Benkler, A. Berinsky, K. Greenhill, F. Menczer, M. Metzger, B. Nyhan, G. Pennycook, D. Rothschild, M. Schudson, S. Sloman, C. Sunstein, E. Thorson, D.Watts, and J. Zittrain, "The science of fake news," Science, vol. 359, pp. 1094–1096, 03 2018.
[12] L. Borges, B. Martins, and P. Calado, "Combining similarity features and deep representation learning for stance detection in the context of checking fake news," Journal of Data and Information Quality, vol. 11, no. 3, p. 1–26, Jul 2019. [Online]. Available: http://dx.doi.org/10.1145/3287763
[13] M. A. Walker, P. Anand, R. Abbott, and R. Grant, "Stance classification using dialogic properties of persuasion," in Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ser. NAACL HLT '12. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012, pp. 592–596. [Online]. Available: http://dl.acm.org/citation.cfm?id=2382029.2382124
[14] D. Sridhar, J. Foulds, B. Huang, L. Getoor, and M. Walker, "Joint models of disagreement and stance in online debate," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics
and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 116–125. [Online]. Available: https://www.aclweb.org/anthology/P15-1012 10 VOLUME 4, 2016 10.1109/ACCESS.2020.3019735, IEEE Access
[15] S. Somasundaran and J. Wiebe, "Recognizing stances in ideological on-line debates," in Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text. Los Angeles, CA: Association for Computational Linguistics, Jun. 2010, pp. 116–124. [Online]. Available: https://www.aclweb.org/anthology/W10-0214
[16] M. Lukasik, P. K. Srijith, D. Vu, K. Bontcheva, A. Zubiaga, and T. Cohn, "Hawkes processes for continuous time sequence classification: an application to rumour stance classification in twitter," in Proceedingsof the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 393–398. [Online]. Available: https://www.aclweb.org/anthology/P16-2064
[17] L. Derczynski, K. Bontcheva, M. Liakata, R. Procter, G. Wong Sak Hoi, and A. Zubiaga, "SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours," in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 69–76. [Online]. Available: https://www.aclweb.org/anthology/S17-2006
[18] C. Stab and I. Gurevych, "Parsing argumentation structures in persuasive essays," Computational Linguistics, vol. 43, no. 3, pp. 619–659, Sep. 2017. [Online]. Available: https://www.aclweb.org/anthology/J17-3005
[19] D. Pomerleau and Rao, "Exploring how artificial intelligence technologies could be leveraged to combat fake news," http://www.fakenewschallenge.org/, 2017, online; accessed 29 October 2019.
[20] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.
[21] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," Neural networks: the official journal of the International Neural Network Society, vol. 18, pp. 602–10, 07 2005.
[22] P. Neculoiu, M. Versteegh, and M. Rotaru, "Learning text similarity with siamese recurrent networks," 01 2016.
[23] H. He, K. Gimpel, and J. Lin, "Multi-perspective sentencesimilarity modeling with convolutional neural networks," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1576–1586. [Online]. Available: https://www.aclweb.org/anthology/D15-1181
[24] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors," 2015.
[25] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015. [Online]. Available: http://dx.doi.org/10.3115/v1/p15-1150
[26] L. Yang, Q. Ai, D. Spina, R.-C. Chen, L. Pang, W. B. Croft, J. Guo, and F. Scholer, "Beyond factoid qa: Effective methods for non-factoid answer sentence retrieval," in ECIR, 2016.
[27] Y. Yang, W.-t. Yih, and C. Meek, "WikiQA: A challenge dataset for opendomain question answering," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 2013–2018. [Online]. Available: https://www.aclweb.org/anthology/D15-1237
[28] Z. Imtiaz, M. Umer, M. Ahmad, S. Ullah, G. S. Choi, and A. Mehmood, "Duplicate questions pair detection using siamese malstm," IEEE Access, vol. 8, pp. 21 932–21 942, 2020.
[29] B. Ghanem, P. Rosso, and F. Rangel, "Stance detection in fake news a combined feature representation," in Proceedings of the First Workshop on Fact Extraction and VERification (FEVER). Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. [Online]. Available: https://www.aclweb.org/anthology/W18-5510
[30] Q. Zeng, "Neural stance detectors for fake news challenge," 2017.
[31] S. R. Pfohl, "Stance detection for the fake news challenge with attention and conditional encoding," 2017.
[32] D. A., A. I., and S. S., "A comparative study on using principle component analysis with different text classifiers," International Journal of Computer Applications, vol. 180, no. 31, p. 1–6, Apr 2018. [Online]. Available: http://dx.doi.org/10.5120/ijca2018916800
[33] S. Karamizadeh, S. Abdullah, A. Manaf, M. Zamani, and A. Hooman, "An overview of principal component analysis," Journal of Signal and Information Processing, 08 2013.
[34] M. Ahmad, A. M. Khan, J. A. Brown, S. Protasov, and A. M. Khattak, "Gait fingerprinting-based user identification on smartphones," in 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 3060–3067.
[35] S. Deegalla and H. Bostrom, "Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbour classification," in 2006 5th International Conference on Machine Learning and Applications (ICMLA'06), 2006, pp. 245–250.
[36] M. Ahmad, D. Ulhaq, Q. Mushtaq, and M. Sohaib, "A new statistical approach for band clustering and band selection using k-means clustering," International Journal of Engineering and Technology, vol. 3, pp. 606–614,
12 2011.
[37] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and Cherry, "SemEval-2016 task 6: Detecting stance in tweets," in Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 31–41. [Online]. Available: https://www.aclweb.org/anthology/S16-1003
[38] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the truth lies: Explaining the credibility of emerging claims on the web and social media," in Proceedings of the 26th International Conference on World Wide Web Companion, 04 2017.

[39] C. Stab, T. Miller, and I. Gurevych, "Cross-topic argument mining from heterogeneous sources using attention-based neural networks," CoRR, vol. abs/1802.05758, 2018. [Online]. Available: http://arxiv.org/abs/1802.05758

[40] I. Augenstein, T. Rocktäschel, A. Vlachos, and K. Bontcheva, "Stance detection with bidirectional conditional encoding," in Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 876–885. [Online]. Available: https://www.aclweb.org/anthology/D16-1084

[41] G. Zarrella and A. Marsh, "MITRE at semeval-2016 task 6: Transfer learning for stance detection," CoRR, vol. abs/1606.03784, 2016. [Online]. Available: http://arxiv.org/abs/1606.03784

[42] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the truth lies: Explaining the credibility of emerging claims on the web and social media," in WWW, 2017.

[43] O. Enayet and S. R. El-Beltagy, "NileTMRG at SemEval-2017 task 8: Determining rumour and veracity support for rumours on twitter." in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 470–474. [Online]. Available: https://www.aclweb.org/anthology/S17-2082

[44] S. Dungs, A. Aker, N. Fuhr, and K. Bontcheva, "Can rumour stance alone predict veracity?" in Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 3360–3370. [Online]. Available: https://www.aclweb.org/anthology/C18-1284

[45] X. Wang, C. Yu, S. Baumgartner, and F. Korn, "Relevant document discovery for fact-checking articles," in Companion Proceedings of the The Web Conference 2018, 2018, pp. 525–533.

[46] W. Ferreira and A. Vlachos, "Emergent: a novel data-set for stance classification," in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego, California: Association for Computational Linguistics, Jun. 2016, pp. 1163–1168. [Online]. Available: https://www.aclweb.org/anthology/N16-1138

[47] Baird Sean, Sibley Doug, and Pan Yuxi, "Talos Targets Disinformation with Fake News Challenge Victory," http://blog.talosintelligence.com/2017/06/talos-fake-news-challenge.html, 2017, online; accessed 29 October 2019.

[48] Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, and Felix Caspelherr, "Team Athene on the Fake News Challenge," https://medium.com/@andre134679/team-athene-on-the-fake-newschallenge- 28a5cf5e017b, 2017, online; accessed 29 October 2019.

[49] B. Riedel, I. Augenstein, G. P. Spithourakis, and S. Riedel, "A simple but tough-to-beat baseline for the fake news challenge stance detection task," CoRR, vol. abs/1707.03264, 2017. [Online]. Available: http://arxiv.org/abs/1707.03264

[50] J. Thorne, M. Chen, G. Myrianthous, J. Pu, X. Wang, and A. Vlachos, "Fake news stance detection using stacked ensemble of classifiers," in Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 80–83. [Online]. Available: https://www.aclweb.org/anthology/W17-4214

[51] Q. Zhang, E. Yilmaz, and S. Liang, "Ranking-based method for news stance detection," in Companion Proceedings of the The Web Conference 2018, ser. WWW '18. Republic and Canton of Geneva, CHE:

[52] International World Wide Web Conferences Steering Committee, 2018, p. 41–42. [Online]. Available: https://doi.org/10.1145/3184558.3186919

[53] J. Shang, T. Sun, J. Shen, X. Liu, A. Gruenheid, F. Korn, Á. D. Lelkes, C. Yu, and J. Han, "Investigating rumor news using agreementaware search," CoRR, vol. abs/1802.07398, 2018. [Online]. Available: http://arxiv.org/abs/1802.07398

[54] A. Hanselowski, A. PVS, B. Schiller, F. Caspelherr, D. Chaudhuri, C. M. Meyer, and I. Gurevych, "A retrospective analysis of the fake news challenge stance-detection task," in Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 1859–1874. [Online]. Available: https://www.aclweb.org/anthology/C18-1158

[55] M. Umer, S. Sadiq, M. Ahmad, S. Ullah, G. S. Choi, and A. Mehmood, "A novel stacked cnn for malarial parasite detection in thin blood smear images," IEEE Access, vol. 8, pp. 93 782–93 792, 2020.

[56] M. Mohtarami, R. Baly, J. R. Glass, P. Nakov, L. Màrquez, and Moschitti, "Automatic stance detection using end-to-end memory networks," CoRR, vol. abs/1804.07581, 2018. [Online]. Available: http://arxiv.org/abs/1804.07581

[57] C. Dulhanty, J. L. Deglint, I. B. Daya, and A. Wong, "Taking a stance on fake news: Towards automatic disinformation assessment via deep bidirectional transformer language models for stance detection," 2019.

[58] D. Pomerleau and Rao, "Fake news challenge dataset," http://www.fakenewschallenge.org/, 2017, online; accessed 29 October 2019.

[59] M. Ahmad, "A fast 3d cnn for hyperspectral image classification," arXiv preprint arXiv:2004.14152, 2020.

[60] M. Ahmad, D. Ihsan, and D. Ulhaq, "Linear unmixing and target detection of hyperspectral imagery using osp," 01 2011.

[61] M. Ahmad, M. A. Alqarni, A. M. Khan, R. Hussain, M. Mazzara, and S. Distefano, "Segmented and non-segmented stacked denoising autoencoder for hyperspectral band reduction," Optik - International Journal for Light and Electron Optics, vol. 180, pp. 370–378, Oct 2018.

[62] M. Ahmad, A. M. Khan, and R. Hussain, "Graph-based spatial–spectral feature learning for hyperspectral image classification," IET Image Processing, vol. 11, no. 12, pp. 1310–1316, 2017.

[63] P. Meesad, P. Boonrawd, and V. Nuipian, "A chi-square-test for word importance differentiation in text classification," in Proceedings of International Conference on Information and Electronics Engineering, 2011, pp. 110–114.

[64] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in ICML, 1997.

[65] S. M. H. Dadgar, M. S. Araghi, and M. M. Farahani, "A novel text mining approach based on tf-idf and support vector machine for news classification," in 2016 IEEE International Conference on Engineering and Technology (ICETECH). IEEE, 2016, pp. 112–116.

[66] D. Hou, J. Zhang, Z. Yang, S. Liu, P. Huang, and G. Zhang, "Distribution water quality anomaly detection from uv optical sensor monitoring data by integrating principal component analysis with chi-square distribution," Optics express, vol. 23, no. 13, p. 17487—17510, June 2015. [Online]. Available: https://doi.org/10.1364/OE.23.017487

[67] Y. Zhai, W. Song, X. Liu, L. Liu, and X. Zhao, "A chi-square statistics based feature selection method in text classification," in 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS). IEEE, 2018, pp. 160–163.

[68] X. Xia, D. Lo, W. Qiu, X. Wang, and B. Zhou, "Automated configuration bug report prediction using text mining," in 2014 IEEE 38th Annual Computer Software and Applications Conference, 2014, pp. 107–116.

[69] V. Slovikovskaya, "Transfer learning from transformers to fake news challenge stance detection (fnc-1) task," ArXiv, vol. abs/1910.14353, 2019.

[70] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," CoRR, vol. abs/1906.08237, 2019. [Online]. Available: http://arxiv.org/abs/1906.08237

[71] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," CoRR, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[72] Muhammad Umer, Zainab Imtiaz, Saleem Ullah, Arif Mehmood, Gyu Sang Choi and Byung-Won On. "Fake News Stance Detection Using Deep Learning Architecture(CNN-LSTM)" in 2020 IEEE Access PP(99):1-1 DOI:10.1109/ACCESS.2020.3019735 [Online] Available: https://www.researchgate.net/publication/343897727_Fake_news_stance_detection_using_deep_learning_architecture_CNN-LSTM