FIND: Fake Information and News Detections using Deep Learning

Abhishek Verma, Vanshika Mittal, Suma Dawn
Department of CSE/IT, Jaypee Institute of Information Technology, Noida, India
{abhishekverma1610, mvanshika25, suma.dawn}@gmail.com

***Abstract*— Fake news detection is very difficult while its spread is simple and has vast repercussions. To tackle this problem we propose a model which detects fake information and news with the help of Deep Learning and Natural Language Processing. A Deep Neural Network on self scraped data set is trained and by using Natural Language Processing the correlation of words in respective documents is found and these correlations serve as initial weights for the deep neural network which predicts a binary label to detect whether the news is fake or not. In this work we have successfully used Recurrent Neural Network and Long Short-Term Memories and Grated Recurrent Units to test for classification. Tensorboard is used for implementation of the proposed framework and provide visualizations for the neural network. Confusion matrix and classification reports show that accuracy score of upto 94 percent can be achieved using LSTM model. The tradeoff is the increased time requirement. Since the fake news can be established based on the learning model, a good training set is mandatory. The results show that the proposed framework is able to adequately present accurate result.**

Keywords—Fake News, Machine Learning, Deep Learning, Web Scraping, Neural Network, RNN, LSTM, Adam Optimizer.

I. Introduction

Fake news is a false piece of information. In today's world, with the increase in spread of fake news from social media and various other sources it is becoming very important to be able to categorize between real news and fake news. Fake news is a major factor in inciting riots, carnage, mob lynchings and other social-economic disturbances. Any piece of Fake news can be created to deliberately misinform or deceive readers, promote a biased point of view, particular cause or agenda, and even for the entertainment. Fake news can be promoted by unauthenticated user ID, social media, printing of fake news in newspapers maybe due to political pressure and many more. Spreading of fake news can cause discontent among people, riots, and even cause loss of trust between two people and even nations. It also has a huge possibility to exploit general public thinking in a completely different manner. News and media coverage gets hugely distorted due to initialization and spread of fake news. Where news can be a boon, fake news is a bane to the society. These days there are many anti-social elements, which may instigate and propagate fake news that may cause societal instability in many layers of social-economic-political-cultural aspects. Many instances of religion clash have been attributed to such fake news.

However, the distinction between a genuine news item and a fake one is very difficult. Its study hence plays a very important role in today's standing. Stopping the propagation of fake news has taken a huge leap due to many advertisements by the Government of India. However, its identification still remains a task filled with discrepancies and many dependencies. An example of this: Facebook aggregated and presented huge collection of fake news related narrations about the last U.S. presidential elections, in 2016. These were discussed in many media outlets of extensive and major impact. (Source: Buzzfeed)[18].

The most difficult part comes in the terms that human eye and comprehension cannot distinguish clearly between real and fake news as they are usually interwoven. One study found that 75% of the time people find fake news somewhat or very accurate. Fake news circulation on social media, these days, have taken a hit due to continued government advertisements and increased concern on the spread and negative effects of fake news. In the previous year, in January 2017, a German official even accepted the dilemma and citing the spread of fake news. They stated that "they are dealing with a phenomenon of dimension never seen before" referring to the spread of fake news [19]. So, detection of fake news becomes most crucial part we need to do in order to have sorted lives.

In this work, we have proposed a framework to distinguish between real and fake news articles. The dataset has been gathered from authentic sources and used for training and testing purpose. Long Short-Term Memories and Grated Recurrent Units model is used to classification purpose. Data is scraped from trusted sites, and is preprocessed before applying NLP. NLTK is used for performing stopword removal, lemmatization with customized part of speech tagging and making embeddings. Word Index of the tokenized dataset is proposed to be initial weights for the ANN. We have used RNN with LSTM units for creating and updating this neural network. After training the NN, we will have word embeddings for all the n-grams given the training dataset. Loss depiction is also done. Basic loss was based on Binary Cross Entropy Loss. This was reduced using Adam Optimizer. The results of the model give adequate output in terms of epoch requirement, confusion matrix results and classification report. The system has presented upto 94% accuracy which is higher and better than comparable existing models generically used for fake news deduction.

978-1-7281-3591-5/19/$31.00 ©2019 IEEE

The next section gives the current state of art. Section III gives a description of the proposed method and its algorithm. The details of the proposed methodology are discussed in section IV, followed by result and experimentation discussed in section V. Section VI concludes the paper

II. Current State of Art

There exist a few models for detection of Fake News [12] – [17]. In the paper, by Bajaj [14], the author approaches the problem from a purely NLP perspective. A comparison from multiple different models implementation is done and an analysis is presented. A Convolution Network is investigated with a new design that incorporates an "attention-like" mechanism and several architectures are explored. The author compared models of Logistic Regression, Two-layer Feed-forward Neural Network, Recurrent Network, Long Short-Term Memories, Grated Recurrent Units, Bidirectional RNN with LSTMs, Convolution Neural Network (CNN) with Max Pooling, Attention-Augmented CNN and the observations and results are noted. According to author's observation, the RNN architecture with GRUs outperformed one with LSTM cells. This holds despite the fact that a positive bias was added to the LSTM's forget gate.

Author Gilda [16] applied Term Frequency-Inverse Document Frequency, a method commonly applied in Document analysis. This method consisted of bi-grams and probabilistic context free grammar (PCFG) detection to a collection of almost 11000+ narratives. He tested dataset on other classifiers too including, Random Forest, Support Vector Machines, Bounded Decision, Stochastic Gradient Descent, and others. His application of Term Frequency-Inverse Document Frequency of bi-gram, input to Stochastic Gradient Descent classifier model determined fake source with almost 77% accuracy.

In the paper, [17], authors Singhania, Fernandex and Rao, presented an automated detector using deep learning methods were employed. It had a three level hierarchical attention network (3HAN). This resulted in a fast and quite accurate detection of fake news articles. 3HAN was used for creating a news article with the help of 3 vectors, each for words, sentences and headlines and processes the input news article in a bottom-up manner. The headline, few words and sentences are distinguishing features of the articles and are relatively of more importance than the rest to which 3HAN gives the required attention with the help of 3 layers. Authors observed almost 96.5% accuracy on a sizeable real-world dataset. 3HAN gives an understandable output via attention parameter values given to various sections of the articles.

III. Algorithm Description

A. Recurrent Neural Network

Recurrent Neural Network (RNN) is a variety of Artificial Neural Network (ANN) in which the nodes are connected to form a sequential directed graph. RNNs can use their memory to process a sequential input. They are powerful and robust neural networks. Also, they are equipped with an internal representation that acts as memory or storage. This allows them to be widely used for ordered dataset such as text, audio, speech, video, etc.

B. Long Short - Term Memories and Grated Recurrent Units

Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM) Units are units of an RNN. This consists of cells, input, output, and forget gates. Cells are used to retain memories and gates are used for controlling the information flow in and out of these individual cells.

LSTM networks allow solving the problem of vanishing /exploding gradient. Exploding gradient problem can be easily fixed by gradient clipping by limiting the factor. Vanishing gradient descent problem is faced when output is generated by passing input to a large number of hidden layers. The problem occurs because output of RNN layer is decided by back propagation on all its initial values resulting in multiple weight updates. So the "earliest" memory is "forgotten". This issue solved by LSTM and GRU units which make use of gates.

IV. FIND Framework: Proposed Approach

The dataset of news articles needs to be made computationally acceptable to the algorithms and this is where the python's Natural Language Toolkit (NLTK) comes into play. To understand the data, sentiment analysis was used,. Sentiment, inherently congregate the use of natural language, text analysis, text classification and other processes. It is used for identification, extraction, quantification and understanding the affective conditions & parameters and subjective content related to the case study. Textual data comprises of a lot of useless words which are not related to the context nor can they be of help in classifying text contexts of the paragraph but helps us humans understand the proper meaning (words like "this", "that", "of", "or", "and", "the"). These are called stop words. Our stopwords corpus consists of python's NLTK library's stopwords, python's string library's punctuations and various other lists of such words gathered manually. The news article dataset comprises of paragraphs where each paragraph is split into words (word tokenized) and iterated through to remove all such stop words. Then the words are lemmatized (stemmed as in 'happier', 'happiness' and 'happy' are stemmed to their "origin" word 'happy'). But words like 'superb' and 'awesome' cannot be lemmatized to a common stem. So we computed the Part of Speech tag (noun, adjective, verb, etc.) on the word and custom lemmatized them. This took around 64 minutes for 72,000 items. The year, month, heading and the news article were then combined in a format to be fed into the artificial neural network.

We used FastText to make word embeddings of our dataset. FastText was proposed by Facebook in 2016. FastText breaks words into several n-grams (sub-words) rather than feeding each individual work into Neural Network. Example: the tri-grams for the word "delhi" is "del","elh" and "lhi". The sum of all these n-grams will be the word embedding vector for "delhi". Given the training dataset, we can get word embeddings for all the n-grams after training the Neural Network. This gives rare words the chance to be properly represented since it is highly probable that we find their n-grams in other words.

We make use of keras's pre-processing text's tokenizer. The dataset was tokenized (Text corpus is vectorized by turning each text into either a vector where the coefficient for each token could be binary, based on word count or tf-idf, or into a sequence of integers (where each integer is the index of a token in dictionary)) and its word index (dictionary mappings of words (str) to their rank/index (int) which is only set after tokenizer was trained on the text) was used to make a matrix of embedded words (integer matrix) which is to be used as the default weights for the neural network.

The dataset was padded so that each news article had the same effective length. This was fed into the RNN along with the LSTM units. The LSTM-RNN consists of input layer. As, each word needs to be represented by a unique integer and for that input data needs to be integer encoded, hence a further set of embedding layers are also used. At each update, during training time, dropout randomly sets a fraction rate of input units to 0 which helps prevent overfitting. Instead of individual elements, 1D drops entire 1D feature maps. Other layers include spatial dropout 1D layer, LSTM layer, dense layer (with activation RELU), dropout layer and the output dense, Sigmoid, layer with Adam optimizer. Adam optimization algorithm iteratively updates the network parameter weights based on certain training data. Then binary cross-entropy loss is calculated. Since, there are 2 classes, cross-entropy can be calculated as:

$$\text{Loss} = -(y*\log(p)+(1-y)*\log(1-p))$$

where y is the true output and p is the predicted output). We use Tensorboard to provide visualizations for the neural network. Below shown is the module diagram of the proposed framework, Figure 1.

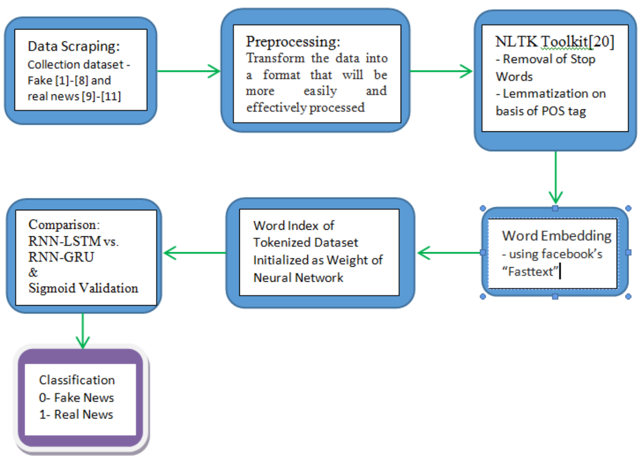


Fig. 1. Modular Diagram of the proposed Framework for FIND.

The neural net architecture is shown in Figure 2 (shown after the reference section due to space constrains).

V. Experimentation and Results

The dataset was self accumulated from web scraping using beautiful soup in python. The dataset is specific to Indian news and may contain traces of International news also. The dataset was aggregated from the sites [1] – [11], [19]. The investigation work was implemented and trained using Google's Colab service. A drawback of using this service was that - after every 12hrs Disk, RAM, VRAM, CPU cache etc., data that was on our allotted virtual machine would get erased. Hence, certain modifications were done so that the data can be stored in a usable form.

Hardware specifications of the systems used include: GPU: 1xTesla K80, having 2496 CUDA cores, compute 3.7, 12GB (11.439GB Usable) GDDR5 VRAM; CPU: 1xsingle core hyper threaded i.e., 1 core with 2 threads Xeon Processors @2.3Ghz (No Turbo Boost), 45MB Cache; RAM: ~12.6 GB Available and Disk: ~33 GB.

An initial experimentation was done using GRU model and LSTM model. The comparison is showcased in Table 1. The specifications include using epoch count of upto 4, average timing per epoch, confusion matrix, classification report, and accuracy score. As was found by experimentation, LSTM gave better accuracy but with higher time requirement.

TABLE I: Comparison between GRU and LSTM model on a dataset of 15,000.

GRU	**LSTM**
mode1.fit(train_seq_x, ytrain,epochs=4) Epoch 1/4 14532/14532 [==] 995s 68ms/step - loss: 1.3480 Epoch 2/4 14532/14532 [==] 993s 68ms/step – loss: 0.5156 Epoch 3/4 14532/14532 [==] 993s 68ms/step – loss: 0.3387 Epoch 4/4 14532/14532 [==] 994s 68ms/step – loss: 0.2092	mode1.fit(train_seq_x, ytrain,epochs=4) Epoch 1/4 14532/14532 [==] 1291s 89ms/step loss: 0.3117 Epoch 2/4 14532/14532 [==] 1292s 89ms/step loss: 0.0722 Epoch 3/4 14532/14532 [==] 1292s 89ms/step loss: 0.0246 Epoch 4/4 14532/14532 [==] 1294s 89ms/step loss: 0.0112
Time – 1000s per epoch	Time — 1300 s per epoch
confusion matrix: [[2852 214] [290 2873]]	confusion matrix: [[2844 222] [130 3033]]
classification report: Precision recall fl-score support 0 0.91 0.93 0.92 3066 1 0.93 0.91 0.92 3163 avg/total 0.92 0.92 0.92 6229 0.9190881361374217	classification report : precision recall fl-score support 0 0.96 0.93 0.94 3066 1 0.93 0.96 0.95 3163 avg/total 0.94 0.94 0.94 6229 0.9434901268261359
Accuracy score: 92%	Accuracy score: 94 %

Confusion matrix is of the form as shown in Table 2. 0 is negative and 1 is positive. The following terms were used: True Negative (TN), i.e., the prediction was negative and test cases, too, were actually negative; True Positive (TP) i.e., the prediction was positive and test cases, too, were actually positive; False Negative (FN) i.e., the prediction was negative, but the test cases were actually positive; and finally, False Positive (FP), i.e., the prediction was positive, but the test cases were actually negative. The accuracy was depicted as given the formula:

Accuracy = (TP + TN) / Total

TABLE II: CONFUSION MATRIX

	Predicted 0	Predicted 1
True 0	TN	FP
True 1	FN	TP

Classification report has the form, as explained in table 3. It lists the precision factor, recall value, F1 score and support of a class value. Together, this gives a good indication of the reliability of the performing algorithm w.r.t class prediction.

TABLE III: EXPLANATION OF CLASSIFICATION REPORT

Precision of a class	Recall of a class	F1-Score of a class	Support of a class
how many of a class were predicted correctly / how many of the same class we totally predicted	how many of a class were predicted correctly / how many of the same class are actually present	Harmonic mean of precision and recall values	How many actual values of the class we had

Further, RNN with LSTM units trained on train : test ratio of 70:30 was used.

These are on 50,000 dataset test_size=0.3
Splitting x and y into xtrain, ytrain, xtest, ytest using train_test_split taking test_size as 0.3 .
Total Size of training data: 50104
Total Size of testing data: 21474

Dataset containing Real news for training: 36419 Dataset containing Fake news for training: 13685

Dataset containing Real news for testing: 15548
Dataset containing Fake news for testing: 5926

Modeling Specifications used:
mode1.fit (train_seq_x, ytrain,epochs = 1, ca1lbacks = [tensorboard])
Epoch 1/1
50104/50104 [======] - 4884s 97ms/step - loss: 0.1165
Sigmoid values are converted to binary value with a threshold of 0.5

The confusion matrix and related classification report is shown below.

Confusion Matrix: [[5784 142]
[63 15485]]

Classification Report:

	Precision	Recall	f1-score	Support
0	0.99	0.98	0.98	5926
0	0.99	1	0.99	15548
avg total	0.99	0.99	0.99	21474

Accuracy Score: 0.9904535717611996

Once the neural net is trained on all entries and neural net is saved to be deployed. To check the working of algorithm all entries are predicted and classification algorithms seem to be robust. The same model is trained on same dataset with two epochs to see that the model is able to sustain the load.

Modeling Specifications:
model fit(train_seq_x, Y,epochs=2,callbacks=[tensorboard])
Epoch 1/2
71578/71578 [=====] - 7058s 99ms/step - loss: 0.4206
Epoch 2/2
71578/71578 [=====] - 7009s 98ms/step - loss: 0.1028
Sigmoid values are converted to binary value with a threshold of 0.5

Confusion Matrix: [[19540 71]
[327 51640]]

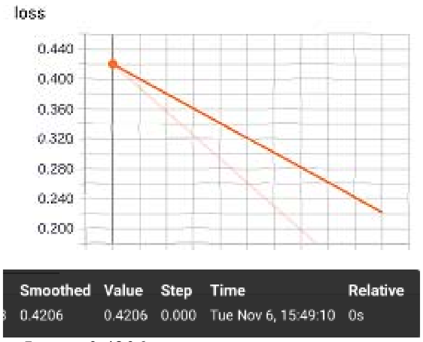
Classification Report:

	Precision	Recall	f1-score	support
0	0.98	1	0.99	19611
0	1	0.99	1	51967
avg total	0.99	0.99	0.99	71578

Accuracy Score : 0.9944396322892509

Loss: For loss of this model Refer Table 4. Binary Cross Entropy Loss and its minimized version using Adam Optimizer is depicted in Figures 6(a) and 6(b) respectively.

TABLE IV: LOSS DEPICTION IN THE MODEL. (A) SHOWS THE BINARY CROSS ENTROPY LOSS. (B) SHOWS THE BINARY CROSS ENTROPY LOSS AFTER USING ADAM OPTIMIZER.



Loss = 0.4206
(a)

Loss = 0.1028
(b)

Example test cases: Two sample test cases are shown in table, Table 5, containing a real and fake news item. The classifier is able to classify these correctly.

TABLE V: SAMPLE TEST DATA

Test Cases	**Test Data Values**	**Output**
"Customs officials bust drug racket" Seize 12 kg of methaqualone at international airport Customs officials at Kempegowda International Airport seized 12.9 kg of the banned drug methaqualone on Wednesday. Its value in the international market is around Rs. 6.45 crore. According to Customs officials, the consignment was to be shipped to Kuala Lumpur in Malaysia by an export firm based in Chennai, Tamil Nadu. The goods were declared as 'palm sugar' for personal use, with a declared value of Rs. 6,000. Methaqualone is a sedative and hypnotic medication. Officials believe that the consignment is part of an international drug racket. Drugs concealed in packets of palm sugar. The drugs were packed in plastic packets, which were inserted in larger packets of palm sugar. Officials of the "Courier Section of Air Cargo" detected the drugs while assessing the consignment. Acting on a hunch that something was not quite right with the consignment, they opened a few packets and found smaller pouches within filled with a white crystalline powder, which was later identified as methaqualone. The consignment was booked by the exporter through a courier service in Chennai and sent to the Bengaluru hub for shipment to Kuala Lumpur. In the first week of January 2018, ketamine worth over Rs. 1 crore, hidden in sweet packages, was seized. That consignment, too, was marked for Kuala Lumpur, by an exporter in Chennai. "There are glaring similarities in the two cases. In both cases, the courier company is the same. The Customs Intelligence Unit of Bangalore Air Cargo is investigating the case," an official said.	1	1
"PCB mulls over legalizing match fixing, to offer it as service" Article taken from [1]	0	0

Extended list of stop words
Stop words= list of punctuations + stop words from nltk toolkit

Word tokenizing
txt= word_tokenize(doc)
append txt to a list . Let the list be named as documents.
The output for this stage is the list of all the words in the narration.

Lemmatization and stop word removal
For each word in documents if it is not a stop word then that words tag is taken from pos_tag and word is lemmatized. Then, this collection of words is appended to documents. This is shown in Table 6.

TABLE VI: OUTPUT AFTER LEMMATIZATION AND STOP WORD REMOVAL ON DATA.

Output for 1st article	Output for 2st article
['Customs', 'official', 'bust', 'drug', 'racket', 'Seize', '12', 'kg', 'methaqualone', 'international', …………………………………]	['PCB', 'mull', 'legalize', 'match', 'fix', 'offer', 'service', 'Lahore', 'Pakistan', 'After', ……………………………………]

Prediction

For both preprocessed testing data result is predicted.
If the predicted value>0.5
Classified as 1 i.e. real
Else
Classified as 0 i.e. fake

For our testing data given as input (Refer Table 4.) to the model, output is [1 0].

VI. CONCLUSION

The spread of fake news can adversely affect our lives. Hence, in order to detect fake news we have proposed a computational model. A huge dataset was gathered by scraping a lot of trusted sites. Data is preprocessed and NLP is applied consisting of methods such as stop word removal, lemmatization with customized part of speech tagging and making embeddings. Word Index of the tokenized dataset is initialized as weights for the ANN. We have used RNN with LSTM units for creating and updating this neural network. After training the NN, we will have word embeddings for all the n-grams given the training dataset. Adam is used for optimization instead of classical stochastic gradient descent to update weights and binary cross-entropy loss. In accordance with the observations, the RNN architecture with LSTM outperformed the implementation with GRU. Further, Adam Optimizer was used to reduce the loss carried by the model.

We experimented to figure out the best model on a dataset of 20,000 news articles. LSTM model showed an accuracy of 94.3%, time of 1300 s/epoch and loss of 0.209 whereas GRU model has 91.9% accuracy, time of 1000 s/epoch and loss 0.011.

We use the better model as depicted by the accuracy results. The train : test ratio of 0.7:0.3 on 72,000 news articles, was used to get an accuracy of 99.04% with time of 4900 s/epoch and loss 0.1165. We further trained the model on all 72,000 articles with time of 7050 s/epoch and loss 0.1028.

In future, we plan to host the algorithm on a server which automatically scrapes news and keeps retraining itself along with an interactive user interface.

REFERENCES

[1] Faking News URL: http://www.fakingnews.com/. Accessed on 21st, Jan 2019.

[2] The Indian Satire URL: http://www.theindiasatire.com/. Accessed on 21st, Jan 2019.

[3] Teekhi Mirchi URL: http://teekhimirchi.in/, Accessed on 21st, Jan 2019.

[4] The Unreal Times URL: , Accessed on 21st, Jan 2019.

[5] The Spoof URL:https://www.thespoof.com/, Accessed on 21st, Jan 2019.

[6] Fun Funny Khez URL: http://www.funfunnykhez.com/, Accessed on 21st, Jan 2019.

[7] Dapaan URL: http://dapaan.com/, Accessed on 21st, Jan 2019.

[8] Farzi News URL: http://farzinews.com/, Accessed on 21st, Jan 2019.

[9] The Hindu URL: https://www.thehindu.com/, Accessed on 21st, Jan 2019.

[10] The Times of India URL: https://timesofindia.indiatimes.com/, Accessed on 21st, Jan 2019.

[11] Rising Kashmir URL: http://www.risingkashmir.com/, Accessed on 21st, Jan 2019.

[12] Ruchansky, N., Seo, S., & Liu, Y. (2017, November). CSI: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 797-806). ACM.

[13] Zhang, J., Cui, L., Fu, Y., & Gouza, F. B. (2018). Fake News Detection with Deep Diffusive Network Model. *arXiv preprint arXiv:1805.08751*.

[14] Bajaj, S. (2017). The Pope Has a New Baby!. *Fake News Detection Using Deep Learning, Stanford*.

[15] Trovati, M., Hill, R., & Bessis, N. (2015, November). A Non-genuine Message Detection Method Based on Unstructured Datasets. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2015 10th International Conference on* (pp. 597-600). IEEE.

[16] Gilda, S. (2017, December). Evaluating machine learning algorithms for fake news detection. In *Research and Development (SCOReD), 2017 IEEE 15th Student Conference on* (pp. 110-115). IEEE.

[17] Singhania, S., Fernandez, N., & Rao, S. (2017, November). 3HAN: A Deep Neural Network for Fake News Detection. In *International Conference on Neural Information Processing*(pp. 572-581). Springer, Cham.

[18] Probe reveals stunning stats about fake election headlines on Facebook. URL: https://www.cbsnews.com/news/facebook-fake-election-news-more-popular-than-real-news-buzzfeed-investigation/ Accessed on 21st, Jan 2019.

[19] Germany investigating unprecedented spread of fake news online. URL: https://www.theguardian.com/world/2017/jan/09/germany-investigating-spread-fake-news-online-russia-election. Accessed on 21st, Jan 2019.

[20] Natural Language Toolkit. URL: https://www.nltk.org/. Accessed on 21st, Jan 2019.

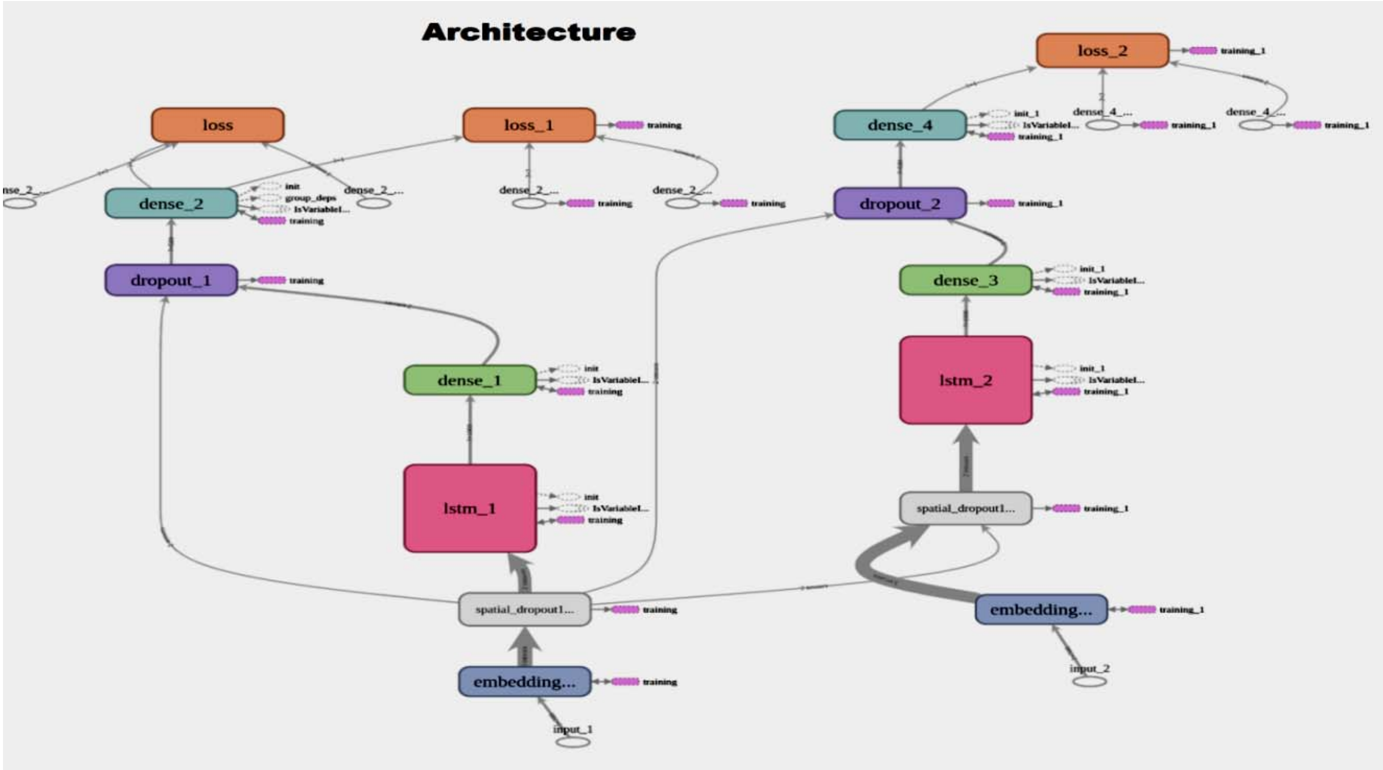


Fig. 2. Neural Net Architecture using Tensorboard