MOCK-UP DETECTION OF COVID NEWS IN TWITTER USING OPCNN

Neetu Reji[1], Dr. Smitha C Thomas [2], Helen Susan Bejoy[3]

[1] M. Tech Student, Computer Science and Engineering, APJ Abdul Kalam Technological University, Kerala, India
[2] Asst. Professor, Computer Science and Engineering, Mount Zion College of Engineering, Kadammanitta, Kerala, India
[3] B.tech Student, Computer Science And Engineering, Mar Baselios College Of Engineering And Technology , Thiruvananthapuram

---***---

Abstract - The misinformation spread in the news platform is a severe danger to social cohesiveness and well-being since it increases political polarization and people's distrust of their leaders. Thus, fake news is a phenomenon that is having a significant impact on our social lives, particularly in politics. This paper proposes novel approaches based on Machine Learning (ML) and Deep Learning (DL) for the fake news detection system to address this phenomenon. The main aim of this paper is to find the optimal model that obtains high accuracy performance. Therefore, we propose an optimized Convolutional Neural Network model to detect fake news (OPCNN-FAKE).

***Key Words*:** Machine Learning, Deep Learning

1.INTRODUCTION

Sensational news are created and spread through social media to achieve intended end. On the other hand, it may also involve narration of a true fact however being deliberately exaggerated. Fake news: President Trump Asked What He Would Do If He Were To Catch The Coronavirus. Real news:States reported 630 deaths. We are still seeing a solid national decline. Death reporting lags approximately 28 days from symptom onset according to CDC models that consider lags in symptoms time in hospital and the death reporting process. The purpose of this project is not to decide the reader the data is fake or real, but alert them they need to use extra scrutiny for the data and also compare 4 algorithms and find the best accurate algorithm Detection of fake news online is important in today's society as fresh news content is rapidly being produced as a result of the abundance of technology that is present. In the world of false news, there are seven main categories and within each category, the piece of fake news content can be visual- and/or linguistic-based. In order to detect fake news, both linguistic and non-linguistic cues can be analyzed using several methods. With the increase in the use of social media many people consumes news from social media instead of traditional news media. However this social media has also being used to spread fake news which has negative impacts on individual users or the community. This project is an idea of a more accurate fake news detection algorithm. It is important that we have some mechanism for detecting fake news or it is an awareness that not everything we read on social media may be true. From this project people can make more informed decisions. With the rapid advances in Artificial Intelligence (AI),a significant number of experiments are being undertaken to tackle issues that were never addressed in the framework of computer science, such as fake news detection . Automatic detection approaches based on Machine Learning (ML) have been studied to combat the emergence and dissemination of false news. The majority of fake news detection systems utilize ML approaches to help consumers in filtering the content they are seeing and determining if a given news piece is misleading or not. Deep Learning (DL) techniques recent accomplishments in difficult natural language processing tasks make them viable for detecting fake news effectively and efficiently. Creating automatic, trustworthy, and accurate systems for identifying fake news on social media is a trending topic of research. The process of determining if a certain news item on any field, from any social media domain, is purposefully or inadvertently misleading might be characterized as fake news detection. Convolutional Neural Network (CNN) has been prominent in many fields with the best performance, including computer vision, smart building structures, and natural language processing. CNN uses convolution layers, pooling layers, and fully connected layers to extract more features with high-level and low-level features. Therefore, we have proposed an Optimal CNN model for Fake news detection (OPCNN-Fake) that can extract high-level and low-level features from the dataset to detect fake news, and it has registered the best performance compared with others models

2. Existing System

They conducted experiments using 5 million postings that were collected from Twitter and Sina Weibo microblogs. They made a comparison between DT, RF, SVM, LSTM and Gated Recurrent Unit (GRU), and RNN. On the same dataset, another study developed a hybrid DL model. A model which includes three modules: Capture, Score, and Integrate (CSI). The capture module has used LSTM and RNN to extract from particular article mundane patterns of user activity. Score module has used a fully connected neural network layer to capture characteristics from users' behavior. Both models have integrated with the third model to classify articles as fake or not. Shu et al. released the Fake Newsnet dataset and applied different algorithms to a dataset: SVM, LR, NB, and CNN. Salem, et al. used the FA-KES dataset that comprises news events around the Syrian war. There are 804 news articles in the collection, 376 of which are fraudulent. Semi-supervised with a fact-checking labeling approach were used to annotations dataset. However, the dataset can be used to train machine learning models for detecting fake news. Portal. introduced Declare, an end-to-end neural network model for debunking fake news and fraudulent claims. To support or reject a claim, it uses evidence and counter-evidence gathered from the internet. The authors trained a bi-directional LSTM model with at least four different datasets and achieved an overall classification accuracy of 80%.

To detect fake news, the authors of proposed a Deep Convolutional Neural Network (FNDNet) to learn the discriminatory features for fake news detection. Furthermore, the authors introduced a hybrid deep learning model that blends CNN and RNN. for the same aim of detecting fake news articles, the authors of presented CNN and LSTM to categorize fake news to produce significant results. Also, the authors of developed multi-level CNN, which incorporated local and global convolutional features to collect semantic information from article texts efficiently. Also, the authors of focused on the substance news piece and the presence of echo chambers in the social network. Table. summaries the comparison of the existing works and our proposed work.

3. IDENTIFICATION OF PROBLEM DOMAIN

With the increase in the use of social media many people consumes news from social media instead of traditional news media. However this social media has also being used to spread fake news which has negative impacts on individual users or the community. This project is an idea of a more accurate fake news detection algorithm. It is important that we have some mechanism for detecting fake news or it is an awareness that not everything we read on social media may be true. From this project people can make more informed decisions.

In this, we explored the fake news problem by reviewing existing literature in two phases characterization and detection. In the characterization phase, they introduced the basic concepts and principles of fake news in both traditional media and social media and detection phase they reviewed existing fake news detection approaches from a data mining perspective including feature extraction and model construction. In this, there are three sorts of fake news, each in contrast to genuine serious reporting, and weighs their pros and cons as a corpus for text analytics and predictive modeling. Filtering, vetting, and verifying online information continues to be essential in a library and knowledge science (LIS) because the lines between traditional news and online information are blurring. Here an innovative hybrid approach that mixes linguistic with network-based behavioral data. It is not a straight forward problem here propose operational guidelines for a fake news detecting system. Here the authors detect fake news using only text features that can be generated regardless of the source platform and are the most independent of the language as possible. They carried out experiments from five data sets, comprising both texts and social media posts in three language groups: Germanic, Latin, and Slavic, and got competitive results when they compared to benchmarks. Here the results obtained through a custom set of features and with other popular techniques when dealing with natural language processing, such as bag-of-words and Word2Vec. The authors come up with a solution that users to detect and filter out sites containing false and misleading information. We use simple and punctiliously selected features of the title and post to accurately identify fake posts. By using logistic classifier this experimental result shows 99.4\% accuracy. In this the authors use deep learning techniques to tackle the problem of the detection of fake news from the text. They trained different neural network models on data containing the full text of the analyzed articles. This models were trained using a labeled dataset of fake and real news. When comparing the evaluation metrics, most of the models gained consistent performance; however, convolutional and LSTM models proved to be the most effective. They use a deep learning classification model to learn the invisible contextual patterns of the content and produce benchmark results on this data set and their model was able to achieve state of the art ROC score of 97%.

4. PROPOSED SYSYEM ARCHITECTURE

The features are fed into the hybrid Neural Network architecture consisting of CNN, RNN LSTM layers and BERT. The proposed approach produces higher predictive performance when compared to the traditional deep learning models. To analyze the relationship, four data models are developed. In the first model, all the features are used without preprocessing for classification. In the second model, the non-reduced features set is used after preprocessing. Model third and fourth is developed by using dimensionality reduction. After the features are selected by any of the four models discussed above, the selected features are fed to the CNN-LSTM architecture. The first layer of the model is the embedding layer that accepts the input headlines and article bodies and converts each word into a vector of size 100. The number of features is 5000, thus, this layer will output a matrix of size $5000 * 100$. The output matrix will contain weights that we get through matrix multiplication, to produces a vector for each word. These vectors are passed to the CNN layer to extract contextual features. The output of the CNN layer is fed into LSTM and then passed to a fully connected dense layer to produce a single stance as final output.

On the output of each CNN neuron, the ReLu activation function is applied. The purpose of using this activation layer is to convert any negative value to zero and to show non-linearity in the network. The function does not affect the output shape of the CNN layer, thus, it is the same as input shape. The value of each neuron, after passing through the ReLu activation function is then fed to a 1-D max-pooling layer. This layer converts the input of each kernel size into one output by selecting the maximum value obtained in each kernel. This will greatly reduce the size of input features for the next layers and will avoid overfitting. The pool size p in our case is 4 thus, the output of this layer will reduce the features by kernel/pool size (p). The dropout rate D for the whole network model is 0.2. The dropout layer is another way to reduce overfitting by dropping the input with values less than the dropout rate. In the FNC-1 dataset, the output of the dropout layer is the same as input passed to it because no value is lower than 0.2.

Convolutional Neural Networks are also known as CNN is a deep learning algorithm. convolutional layers are basic buliding block of a convolutional network and it is made up of neurons that have learnable weights and biases. A CNN consists of an input, output layer and multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers.

RNN is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feed forward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs.

LSTM is an RNN architecture which is used in the field of deep learning. In this we calculate the Memory units and create the models. We provide a threshold values of range from 0 to 1. After creating the model we create three layers of memory unit and calculate its maximum value. The calculation is based on the memory units

BERT: In the pre-training stage autoencoding-stype reconstruction, ie word tokenizer, where an word is tokenized into discrete visual tokens. In this pre-training part each words has two views, i.e., word patches, and visual tokens. We randomly mask some proportion of word and replace them with a special mask embedding. Then the patches are fed to a backbone vision Transformer. Based on the encoding vectors of the words, the pre training stage predict the visual tokens of the original words. Also, we randomly mask some proportion of word patches, and feed the corrupted input to Transformer. The model learns to recover the visual tokens of the original word, instead of the raw pixels of masked patches. We perform fine-tune the pretrained BERT and self-supervised learning on two downstream tasks, i.e., word classification, and semantic segmentation.

In BERT, randomly N% of the tokens was masked but in BEiT (BERT Pre-Training of Word Transformers) , word patch(token) is masked as a block rather than masking randomly. So overall 40% of the masks needs to be masked but in blocks. Minimum of 16 patches constitute a block and aspect-ratio for the block is also chosen randomly.

- Word is divided into grids(token).
- Blocks of token are masked randomly.
- Flatten the word patch into a vector.
- Positional embeddings and embeddings are learned for the patches.
- Now these embeddings are passed through BERT like architecture.
- For masked part, model has to predict word token.
- These tokens come from word tokenizer. Finally, word data can be reconstructed using tokens.

OPCNN is stands for Optimized CNN. Its application is text classification. OPCNN is a text representation technique like word embedding. BERT tokenizer is used to create text classification model. OPCNN is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. In order to use OPCNN text embedding as input to train text classification model we need to tokenize our text reviews. To tokenize our text we use OPCNN Tokenizer.

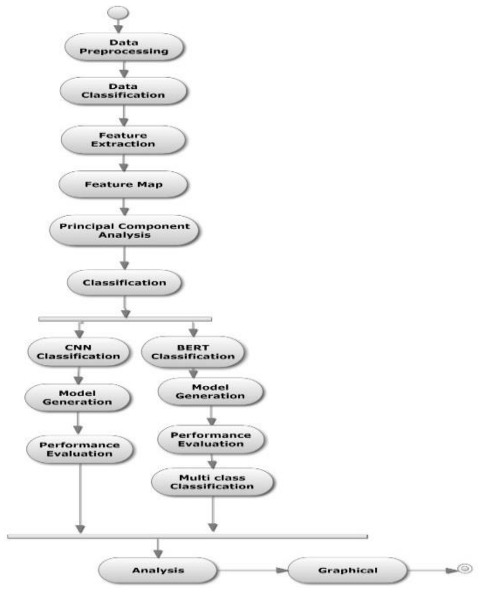

Fig -1: Proposed System Architecture

5. MODULES

- Data Preprocessing
- Feature Extraction
- Data Model Generation (Machine Learning)
- Data Analysis

Data Preprocessing

The data set we used in this study are open source and available from SHARED TASK CONSTRAINT @ 2021. The data includes both real and fake news article from various domains. The real news articles contain the true description of real world events while the fake news website contains claims that are not aligned with facts. The data set is called the "COVID19 Fake News Detection in English" illustrated in figure 1 which contain real and fake news which are extracted from various social media platforms. This task focuses on the detection of COVID-19 related fake news. The file contains 6420 records and three columns: Id, tweet and label. The tweet column contains text for the real and fake news. The label column is the target column can have two values i.e. "fake" and "real" which makes the problem a binary classification problem

Pre-processing is a data mining technique that transforms incomplete and inconsistent raw data into a machine-understandable format. Several tasks for texts pre-processing were performed on dataset. In order to perform these tasks, NLP techniques such as the conversion of the texts' characters to lowercase letters, stop words removal, stemming, and tokenization was applied, with the application of algorithms available in Keres's library. Stop words are very common words that exist in the text and have very minor importance in terms of features and are irrelevant for this work e,g 'of', 'the', 'and', 'an', etc. By removing the stop words, we reduce the processing time and save space otherwise taken by meaningless words mentioned above. In the text, words having similar meanings can occur more than once e.g. games and games. If so, reducing the words to a common basic form is very effective. This process is known as stemming and it is performed with the open-source implementation of the NLTK's Porter stemmer algorithm. The tokenizer function from Keras's library was used to split each headline into a vector of words. Once the preprocessing is done, we use word embedding (word2vec) to map word/text to a list of vectors. Finally, a dictionary of the 5, 000 uni-gram words of headlines and article bodies is created. The length of all the headlines is set to the maximum length of the headline. The headlines with a length smaller than maximum length are zero-padded.

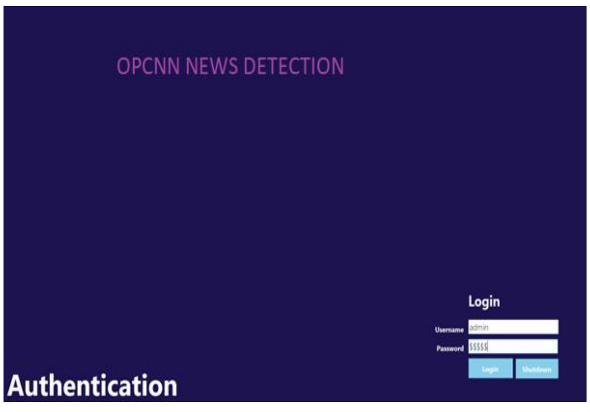

Fig -2: Home Page

Fig -3: Data upload for Preprocessing

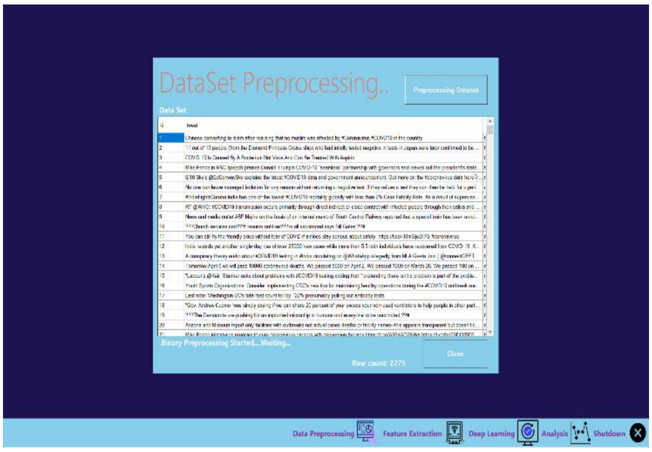

Fig -3: Data Preprocessing

Feature Extraction

There are two ways to perform dimensionality reduction in text categorization: feature extraction and feature selection. In feature selection methods, the most significant and relevant features are retained and the remaining features are discarded. in feature extraction methods, a new vector space with special characteristics is created by transforming the original vector space. The features are reduced in new vector space. Principal Component Analysis (PCA) is a widely used technique that uses a linear transformation to reduce the dimensions of a feature set. The resulting dataset is simplified but it retains the characteristics of the original data set. The new dataset might have an equal or lesser number of features than the original dataset. The covariance matrix is used to compute the principal components. These components are arranged in decreasing order of importance. Chi-Square Statistics is one of the most effective feature selection algorithms. It is designed for testing relationships between categorical variables. It is used to estimate the lack of independence between a and b as well as compare to the chi-square distribution with one degree of freedom to judge extremeness. Test for independence and test for goodness of fit are two types of tests for which Chi-square is used. For feature selection, test for independence is implemented and the dependency of target label is examined on feature(s). Chi-square investigates the correlation of the features. The feature having correlation are kept and the remaining features are discarded. For each feature, chi-square is calculated independently towards the target class and its significance is decided based on a predefined threshold (which is 0.05 commonly). The greater the value of chi-square, the lesser the significance of the feature.

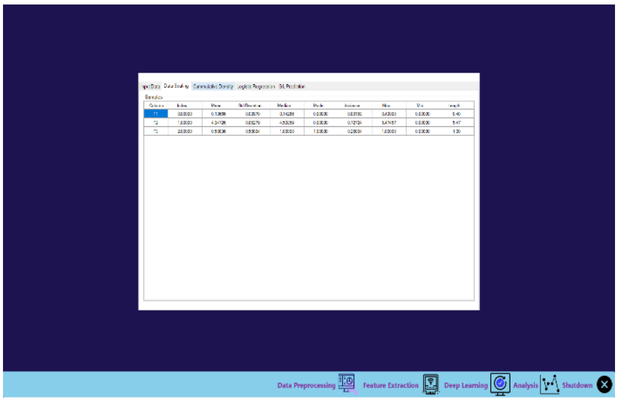

Fig -4: PCA ANALYSIS

Data Model Generation (Machine Learning)

The features are fed into the hybrid Neural Network architecture consisting of CNN, RNN LSTM layers. The proposed approach produces higher predictive performance when compared to the traditional deep learning models. To analyze the relationship, four data models are developed. In the first model, all the features are used without preprocessing for classification. In the second model, the non-reduced features set is used after preprocessing. Model third and fourth is developed by using dimensionality reduction. After the features are selected by any of the four models discussed above, the selected features are fed to the CNN-LSTM architecture. The first layer of the model is the embedding layer that accepts the input headlines and article bodies and converts each word into a vector of size 100. The number of features is 5000, thus, this layer will output a matrix of size $5000 * 100$. The output matrix will contain weights that we get through matrix multiplication, to produces a vector for each word. These vectors are passed to the CNN layer to extract contextual features. The output of the CNN layer is fed into LSTM and then passed to a fully connected dense layer to produce a single stance as final output.

Convolutional Neural Networks are also known as CNN is a deep learning algorithm. convolutional layers are basic building block of a convolutional network and it is made up of neurons that have learnable weights and biases. A CNN consists of an input, output layer and multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers.

RNN is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feed forward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs.

LSTM is an RNN architecture which is used in the field of deep learning. In this we calculate the Memory units and create the models. We provide a threshold value of range from 0 to 1. After creating the model we create three layers of memory unit and calculate its maximum value. The calculation is based on the memory units.

Data Analysis

To evaluate the performance of the models, we used:

A. Accuracy Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. if we have high accuracy then model is best. Yes, accuracy is a great measure but only when you have symmetric data sets where values of false positive and false negatives are almost same. Therefor to evaluate the performance of your model

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

B. Precision Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

C. Recall Recall is the ratio of correctly predicted positive observations to the all observations in actual class.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

D. Fscore F score is also called F1 score - F1 Score is the weighted average of Precision and Recall. It is used to evaluate binary classification systems, which classify examples into positive or negative.

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class.
True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.
False positives and false negatives, these values occur when your actual class contradicts with the predicted class.
False Positives (FP) – When actual class is no and predicted class is yes.
False Negatives (FN) – When actual class is yes but predicted class in n.

Fig -5: Fake News Detection

6. CONCLUSIONS

This study proposed a fake news stance detection model, based on the headline and the body of the news irrespective of the previous studies which only considered the individual sentences or phrases. The proposed model incorporates principal component analysis (PCA) and chi-square with CNN and LSTM, in which PCA and chi-square extract the quality features which are passed to the OPCNN model. First, we pass the non-reduced feature set with and without preprocessing to the neural network. Then the dimensionality reduction techniques are applied and the results are compared. PCA elevates the performance of the classifier for fake news detection as it removes the irrelevant, noisy, and redundant features from the feature vector. This process produces promising results by scoring up to 97.8% accuracy which is considerably better than the previous studies. With the increase in the popularity of social media, many people consumes news from social media instead of traditional news media. Social media has also been used to spread fake news with strong negative impacts on individual users or a community. The spread of fake news has raised concerns everywhere the planet recently. This fake news have severe consequences. The identification of fake news is vital. In this report, we propose deep learning models to spot covid-19 fake news and compare different four algorithms and find which one is the best. We used several preprocessing techniques. CNN, RNN, LSTM, OPCNN. The data set in this project focuses on the news about COVID-19. The developed system performance is comparable to that of humans in this task with an accuracy up to .96% and this study shows BERT is the most accurate algorithm compare to others.
Our future work entails:
(a) validate the performance of our proposed model on larger datasets,
(b) A tree-based learning may perform better than simple approaches,
(c) different textual features and their fusion shall be analyzed to boost the performance.
In future, we will use our proposed model to detect COVID-19 fake news. Also, we plan to apply multimodel-based methods with recently pre-trained word embeddings (i.e., Elmo, XLNet, etc.) to handle visual information like video and words. In addition, we may use knowledge-based and fact-based approaches to detect fake news. We will also expand our planned dataset to include data from additional languages.

REFERENCES

1. T. Mihaylov, G. Georgiev, and P. Nakov, ''Finding opinion manipulation trolls in news community forums,'' in Proc. 19th Conf. Comput. Natural Lang. Learn., Beijing, China, Jul. 2015, pp. 310–314. [Online]. Available: https://www.aclweb.org/anthology/K15-1032
2. S. Vosoughi, D. Roy, and S. Aral, ''The spread of true and false news online,'' Science, vol. 359, no. 6380, pp. 1146–1151, Mar. 2018.
3. A. K. Chaudhry, ''Stance detection for the fake news challenge: identifying textual relationships with deep neural nets,'' in Proc. Natural Lang. Process. Deep Learn., 2017, pp. 1–10.

4. S. Chopra, ''Towards automatic identification of fake news: Headlinearticle stance detection with LSTM attention models,'' Stanford Univ., Stanford, CA, USA, Tech. Rep., 2017.
5. L. Borges, B. Martins, and P. Calado, ''Combining similarity features and deep representation learning for stance detection in the context of checking fake news,'' J. Data Inf. Qual., vol. 11, no. 3, pp. 1–26, Jul. 2019, doi: 10.1145/3287763.
6. D. Sridhar, J. Foulds, B. Huang, L. Getoor, and M. Walker, ''Joint models of disagreement and stance in online debate,'' in Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process., vol. 1. Beijing, China, Jul. 2015, pp. 116–125. [Online]. Available: https://www.aclweb.org/anthology/P15-1012
7. L. Konstantinovskiy, O. Price, M. Babakar, and A. Zubiaga, ''Towards automated factchecking: Developing an annotation schema and benchmark for consistent automated claim detection,'' 2018, arXiv:1809.08193. [Online]. Available: http://arxiv.org/abs/1809.08193

BIOGRAPHIES

Neetu Reji, currently pursuing MTech degree in Computer Science and Engineering from APJ Abdul Kalam Technological University, Kerala, India at Mount ZionCollege of Engineering, Kadammanitta, Kerala, India.

Dr. Smita C Thomas received the MTech degree in Computer science and Engineering.. She is currently working as Assistant Professor in the Department of Computer science and Engineering at Mount Zion College of Engineering, Kadammanitta, Kerala, India.