

Sentiment Analysis of COVID19 Reviews Using Hierarchical Version of d-RNN

Arindam Chaudhuri

Samsung R & D Institute Delhi India,
NMIMS University Mumbai,
India

arindamphdthesis@gmail.com

Abstract. In recent years understanding person's sentiments for catastrophic events has been a major subject of research. In recent times COVID19 has raised psychological issues in people's minds across world. Sentiment analysis has played significant role in analysing reviews across wide array of real-life situations. With constant development of deep learning based language models, this has become an active investigation area. With COVID19 pandemic different countries have faced several peaks resulting in lockdowns. During this time people have placed their sentiments in social media. As review data corpora grows it becomes necessary to develop robust sentiment analysis models capable of extracting people's viewpoints and sentiments. In this paper, we present a computational framework which uses deep learning based language models through delayed recurrent neural networks (d-RNN) and hierarchical version of d-RNN (Hd-RNN) for sentiment analysis catering to rise of COVID19 cases in different parts of India. Sentiments are reviewed considering time window spread across 2020 and 2021. Multi-label sentiment classification is used where more than one sentiment are expressed at once. Both d-RNN and Hd-RNN are optimized by fine tuning different network parameters and compared with BERT variants, LSTM as well as traditional methods. The methods are evaluated with highly skewed data as well as using precision, recall and F1 scores. The results on experimental datasets indicate superiority of Hd-RNN considering other techniques.

Keywords. Sentiment analysis, viewpoints, sentiments, RNN, d-RNN, BERT, Hd-RNN.

1 Introduction

Coronavirus 2019 (COVID19) [1, 2, 3, 4] is a global pandemic since past two years. It has almost ruined mankind with catastrophic implications [5]

having major impact on global economy. This has led to unprecedented rise in unemployment, psychological issues and depression of people around world. The abrupt social, economic and travel changes have motivated research in several domains [6, 7, 8, 9]. In India COVID19 has adversely affected its economy in past two years [10]. It has battered Indian economy in two major COVID19 waves as shown in Fig. 1 which country has seen.

As of January 2022 considering official figures, India stands at second highest number of confirmed cases in world after United States and third highest number of deaths after United States and Brazil. However it is assumed that there has been high degree of under reporting in COVID19 cases.

During this phase of 2 years there has been unprecedented growth of social media usage such as Twitter by people where they have expressed several concerns related to their living conditions, psychology [11, 12, 13, 14] and mental health [15]. This data has been used for research in behavioural sciences [16]. It has also been used for personality prediction [11, 12] as well as understanding trends and backgrounds of users online [17, 18].

In view of this research on sentiment analysis of COVID19 reviews has become centrestage of attention for social media analytics. Sentiment analysis involves natural language processing (NLP) [19, 20, 21] in order to extract systematically affective states, attitudes and opinions of individuals or social groups [22, 23, 24] across various domains such as politics, sociology, business intelligence, etc. [25].



Fig. 1. COVID19 has battered Indian economy

Classification of sentiments within any specific domain involves any event, item, topic, product, application or others. The importance of sentiment analysis lies in many tasks such as digital mental health [26], recommendation systems [27] as well as intelligent cognitive assistants [28].

The major prima face in using sentiment analysis is towards text classification having polarities [29]. Polarity comes in several forms including sentiment class having several values such as very positive, positive, neutral, negative and very negative. However, polarity values and classes differ from one application to another. Sentiment expressions with happy, sad, angry etc emotions alongwith sentiment classes can be analyzed through sentiment analysis.

Text classification can be either subjective or objective [30] in nature. Sentiment analysis approaches are basically classified as three major groups [29] viz (a) lexicon-based approach (b) machine learning approach and (c) hybrid approach. In lexicon-based approaches, polarity is extracted with respect to predefined lexicon or dictionary. Sentiment analysis model is trained using dataset commonly known as corpus. In machine learning approaches, algorithm is trained considering given dataset in order to build classification model which extracts sentiment polarity of given text. Hybrid approach is most suitable for real life applications which combine both lexicon-based and machine learning methods to perform sentiment analysis activity.

One of the commonly used machine learning approaches is artificial neural networks (ANN) [31, 32]. Closely associated with ANN are deep learning networks [33, 34, 35].

In recent past both these methods have been used in abundance in sentiment analysis research [36]. Deep learning networks are more powerful than ANNs. They are blessed with better data representation capabilities with good features and multiple representation levels than ANNs. This allows different computer recognition tasks including text mining, image processing and pattern recognition to be performed more smoothly and efficiently. Deep learning networks allow representation of words considering textual data and produce word embedding which could be used by different machine learning approaches.

Several types of deep learning networks have been used for NLP such as feedforward neural networks, convolutional neural networks (CNN) and recurrent neural networks (RNN) [33, 34]. Deep learning networks have outperformed other machine learning methods in several NLP tasks including machine translation and named-entity recognition [37, 38]. Deep learning-based approaches for sentiment analysis are further classified in accordance to deep learning networks being used. There are basically three different categories viz: (a) feed forward neural network-based sentiment analysis, (b) CNN based sentiment analysis, and (c) RNN based sentiment analysis approaches. The majority of sentiment analysis approaches in previous few decades have been focused towards identification of text polarity. Deep learning models have played a significant role in forecasting COVID19 infection trends [39] in various parts of world.

In this research work, novel sentiment analysis methods have been used considering state-of-the-art methods towards understanding people's behavior arising from COVID19.

This work provides motivational direction towards making society aware of the fact that dissemination of useless information in a sensitive topic like COVID19 is harmful to mankind. It considers psychological well-being with respect to constant rise in number of active COVID19 cases during peak periods.

Considering abundant corpus of text data available for this research, here computational modeling and machine learning methods [20, 21, 36] have been primarily used. However, there have been number of challenges arising due to testing and reporting of COVID19 cases [39].

It has been observed that an appreciable amount of available reported data is plagued with false figures. This issue is addressed by re-collecting that portion of data from other reliable sources [39].

The computational framework highlights multi-label sentiment classification with more than one sentiment expressed at once. In order to achieve this deep learning based language models considering delayed recurrent neural networks (d-RNN) and hierarchical version of d-RNN (Hd-RNN) are used to address rise of COVID19 cases in different parts of India.

All sentiments are reviewed with respect to time window spread across January 2020 and June 2021 when India witnessed most active cases.

Both d-RNN and Hd-RNN are optimized by fine tuning different network parameters. The experimental hypothesis is made stronger by performing comparative performance analysis of d-RNN and Hd-RNN models with some traditional machine learning models like naive bayes (NB), support vector machine (SVM), k-nearest neighbor (kNN), random forest (RF), gradient boosting (GB), ada boost (AB) and decision trees (DT) [36].

We use different variants of BERT [36] in order to evaluate test datasets with prediction validation. LSTM [20, 21] and BD-LSTM [20, 21] models are also used in comparing performance of d-RNN and Hd-RNN.

The results on experimental datasets highlight Hd-RNN's superiority with respect to other techniques.

The novelty of this work is attributed through following points: (a) Preparation and analysis of COVID19 sentiment data spanning across time period of 1.5 years, (b) Computational framework comprising of novel RNN models such as d-RNN and Hd-RNN, (c) Comparative analysis with other models relevant to COVID19, (d) All methods are evaluated with highly skewed data where precision, recall and F1 scores are used.

This paper is organized as follows. In section 2 work related to sentiment analysis is presented. The computational methodology is discussed in section 3. In section 4 experiments and results are highlighted. Finally, conclusion is given in section 5.

2 Related Work

The process of feature extraction from text for NLP related tasks is known as word embedding [40, 41, 42]. Some of the common examples of word embedding include sentiment analysis. As such significant amount of research work has been performed on sentiment analysis using machine learning. It is basically obtained using methods where words or phrases from vocabulary are mapped to real number vectors. It involves mathematical embedding from large corpus with multi-dimensions per word to a vector space. The lower dimension here is used by machine learning or deep learning models for text classification [40].

Basic word embedding methods such as bag-of-words (BOW) [43] and term frequency inverse document frequency (TF-IDF) [44] do not have context awareness and semantic information in embedding. This problem is related to skip-grams [45] which use n-grams involving bigrams and tri-grams to develop word embedding. This allow adjacent word token sequences which needs to be skipped [46].

There has been considerable progress in word embedding and language models since last few decades. In [47] word2vec embedding is proposed which uses feedforward neural network model. They learn association between words from text dataset which detects synonymous words or suggest additional words with respect to a partial sentence.

It uses continuous bag-of-words (CBOW) or continuous skip-gram model architectures in order to produce distributed words' representation. The method creates large vector which represent each unique word in corpus. Here semantic information and relation between words are preserved. For any two sentences which do not have much common words, their semantic similarity is captured using word2vec [47].

However, word2vec does not well represent word context. To obtain vector representations for words, in GloVe [48] words are mapped into meaningful space where word distance is related to semantic similarity. It uses matrix factorization in order to construct large matrix of co-occurrence information. This results in representation which shows linear substructures of word vector space.

With top list words embedding feature vectors match within certain distance measures.

The relations between words such as synonyms, company-product relations etc can be found using GloVe. A gender-neutral GloVe has been proposed [49] where it has gender biased information.

In [50] word embedding methods evaluation is provided methods including GloVe [54], skip-gram and continuous space language models (CSLM) [51]. It has been observed that in all language related tasks skip-gram and GloVe have outperformed CSLM.

In [52], an evaluation of word embedding methods have been performed for biomedical text analysis applications. Here it has been observed that word embedding trained from clinical notes and literature better captured word semantics.

In [53], classification of twitter data using machine learning is performed with 89.47% accuracy.

In [54], sentiment analysis on Uri attack has been performed in order to mine emotions and polarity on Twitter data. The dataset comprised of about 5000 tweets. The experimental results showed Uri attack disgusted 94.3% of individuals.

In [55], sentiment analysis using machine learning for business intelligence has performed.

In [56], analytical categorization and evaluation of prevalent testing techniques and deployment of sentiment analysis machine learning techniques on different applications have been performed.

In [57], Naïve Bayes and OneR have been used for sentiment analysis.

Another work from [58] also uses various machine learning algorithms such as naïve bayes, support vector machine (SVM), logistic regression, decision trees, k-nearest neighbor and random forest for sentiment analysis.

In [59], distribution of COVID19 vaccines implies an urgent need to track and understand public opinion on an ongoing basis to establish baseline vaccine confidence levels in order to detect early confidence loss warnings.

Research on public attitudes with facebook and twitter towards COVID19 vaccinations uses Artificial Intelligence analysis [60] in UK and US.

In [61], a study is performed on social network analysis of COVID19 sentiments using machine learning methods with twitter data. This study

creates sentiment analysis through large number of tweets. The results are categorized considering consumers' viewpoint into positive and negative with respect to tweets [62].

Another significant work on sentiment analysis for COVID19 and infectious diseases can be found [63].

In [64], public sentiments on COVID19 using machine learning for tweet classification is performed where classification methods like naïve bayes and logistic regression are used.

LSTM and BERT language models have also provided promising results [65, 66] for sentiment analysis. Both LSTM and BERT belong to family of RNNs [67, 68].

LSTMs are characterised by two important gates viz forgot and output gates which makes them efficient sentiment analysis models. LSTM has feedback relation which helps in processing both single and sequential data points. Many versions of LSTM have been developed till date. For smaller datasets bidirectional LSTM models produce better results than BERT models.

These models are trained in lesser time than their pre-trained counterparts [69]. In LSTM are that words are passed in and generated sequentially. For capturing true meaning of words even bi-directional LSTMs do not have good performance.

These issues are resolved by BERT efficiently [70]. BERT is based on transformers where each output element is connected to each input element. They have been unveiled by Google in 2017. BERT architecture proved to be game changer in NLP which allow transfer learning usage in several tasks. It uses adjacent text in order to assist machines in interpretation of ambiguous language in text.

Some recently proposed significant sentiment analysis approaches are presented in Table 1.

In [71], feed forward neural network is used on Arabic tweets with accuracy and precision of 90% and 93.7% respectively.

Appreciable results are available in [72] where CNN is used on movie review datasets.

In [73] and [74], CNN have been used on SentiStrength (text) and SentiBank (visual) as well as manually annotated datasets with accuracy values of 79% and 95% respectively. With 8,000

Table 1. Recently proposed significant sentiment analysis approaches

Paper	Year	ANN Models	Datasets	Results
[75]	2020	CNN	8000 comments and posts	Accuracy 90.9%
[83]	2020	word embedding	Tweets	Accuracy 62.8%
[76]	2019	CNN and LSTM	Lithuanian Internet comments	Accuracy 70.6%
[80]	2019	LSTM	Militarylife PTT	Accuracy 85.4% F1-Score 88.41%
[81]	2019	LSTM	Roman Urdu datasets	Accuracy 95.2%
[74]	2018	CNN	Manually annotated dataset	Accuracy 95%
[79]	2018	LSTM	504 news headlines and 675 microblog messages	
[78]	2018	RNN	Twitter posts and news headlines in financial domain	
[71]	2017	Feed Forward	Arabic tweets	Accuracy 90% Precision 93.7%
[77]	2017	RNN	Amazon health product reviews, SST-1 and SST-2	GRU is best (Accuracy)
[72]	2016	CNN	Movie reviews and IMDB	
[73]	2015	CNN	SentiStrength (text) SentiBank (visual)	Accuracy 79%
[82]	2015	word embedding	SemEval 2013	
[84]	2015	auto-encoder	Arabic Tree Bank	Accuracy 73.5%

comments and posts CNN [75] have produced accuracy of 90.9%.

In [76] CNN and LSTM models on Lithuanian internet comments have provided an accuracy of 70.6%. In [77] RNNs with Amazon health product reviews, SST-1 and SST-2 considering accuracy have given best results for GRU. Another significant results can be found in [78] and [79] with RNNs.

In both of these Works datasets considered constituted new headlines with microblog messages as well as twitter posts. In [80] LSTM on Militarylife PTT datasets have produced accuracy and F1 Score of 85.4% and 85.4% respectively. In [81] LSTM on roman urdu datasets have produced accuracy of 95.2%. In [82] and [83] word embedding have been used on SemEval 2013 and tweets. In [84] auto-encoders have been used on Arabic tree bank.

3 Computational Methodology

The computational methodology for sentiment analysis of COVID19 reviews is presented in this

section. The section starts with discussion on COVID19 datasets. This is followed by brief introduction on RNN and delayed RNN. Based on RNN and d-RNN, Hd-RNN is presented.

Next an approximation of hierarchical delayed RNN with hierarchical bidirectional RNN is performed. The section concludes with twitter-based sentiment analysis framework for COVID19 in India.

3.1 Datasets

Due to non-availability of any standard COVID19 datasets, in this research we have developed datasets considering various factors. Some of the significant factors considered while preparing this dataset include time window, geographical region, age group, gender, testing frequency and COVID19 vaccine taken. The datasets are prepared considering tweet data available from Twitter with respect to abovementioned factors.

The prepared datasets are validated against available benchmark datasets [36]. The time window here spans from January 2020 to June

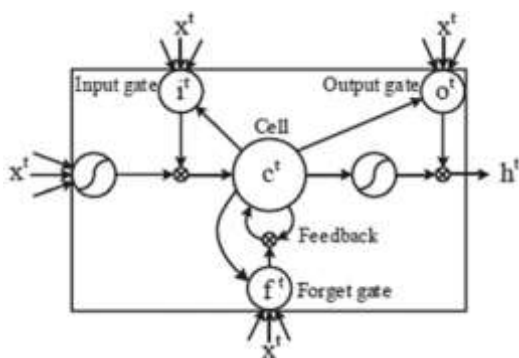


Fig. 2. LSTM memory block with single cell [20]

2021 when there were most active cases were detected in India. It is during this period India had two COVID19 waves, first one from March 2020 to September 2020 and second one from February 2021 to May 2021.

The geographical region included those states where most COVID19 cases are detected during stated time window.

In view of this, 5 top Indian states viz Maharashtra, Delhi, Karnataka, Kerala and Tamil Nadu are considered. The age groups are divided into 5 intervals viz 0-20, 20-40, 40-60, 60-80 and 80+. The gender considered as male and female. The testing frequency involves COVID19 tests performed per day by both government and non-government testing agencies. The COVID19 vaccine taken highlights those people in abovementioned states who have been vaccinated.

3.2 Recurrent Neural Networks

RNNs are an important category of deep learning networks [85] with infinite impulse response. The different computational units of RNNs are connected together to form directed circle. As a result of this, these networks create an internal state which highlights dynamic temporal behavior.

The arbitrary input sequences are processed through internal memory. This makes them readily applicable for non-segmented handwritten or speech recognition tasks. They are basically Turing complete [86] which provides capability to run arbitrary programs in order to process arbitrary input sequences.

Since RNNs can be trained in either supervised or unsupervised manner they are more efficient than traditional ANNs and SVMs [85, 86]. RNNs learn intrinsic characteristics about data without target vector's help. This learning capability is stored as network weights. The network's unsupervised training has similar input as target units.

In deep learning network's architecture is optimized through several routines. The network is treated as directed graph where different hidden units are connected to each other. Each hidden layer in network is non-linear combination of layers.

This is because combination of outputs from all previous units works with their activation functions. Each hidden layer becomes optimally weighted and non-linear, when optimization routine is applied to network. Each hidden layer becomes low dimensional projection of below layer, when each sequential hidden layer has fewer units than one below it.

The recurrent structure of network allows modeling of contextual information for temporal sequences. Due to issues of vanishing gradients and error blowing up problems [86], it is very difficult to train these networks with commonly used activation functions.

This is addressed through LSTM architecture [85, 86] which replaces non-linear units in traditional RNNs. LSTM memory block with single cell is highlighted in Fig. 2. It has one self-connected memory cell and three multiplicative units viz input, forget and output gates. These gates store and access long range temporal sequence based contextual information. The activations of memory cell and three gates are available in [85]. It has been shown that topological enhancements RNNs increases their expressive power and representation capacity [87].

The two most common enhancement strategies are: (a) stacked RNNs which increases learning non-linear functions capacity and (b) bidirectional processing which uses acausal information in sequence. The basic mathematical background to increase network's depth for single layer RNN is presented here. In view of this, let us consider an input sequence $\{y_p\}_{p=1, \dots, P}$, $y_p \in \mathbb{R}^m$ such that single layer RNN is specified as:

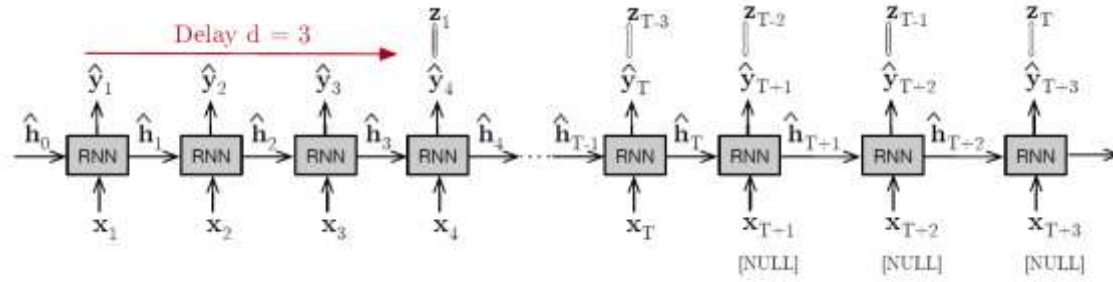


Fig. 3. d-RNN with sequence of T elements [87]

$$\hat{s}_p = g(\hat{W}_y y_p + \hat{W}_s \hat{s}_{p-1} + \hat{b}_s), \quad (1)$$

$$\hat{t}_p = h(\hat{W}_o \hat{s}_p + \hat{b}_o). \quad (2)$$

Here $g(\cdot)$ and $h(\cdot)$ are element-wise activation functions, $\hat{s}_p \in \mathbb{R}^n$ represent hidden state at timestamp p with n units and $\hat{t}_p \in \mathbb{R}^n$ represent network outputs.

The parameters include input weights $\hat{W}_y$, recurrent weights $\hat{W}_s$, bias term $\hat{b}_s$, output weights $\hat{W}_o$, bias term $\hat{b}_o$ with initial state as $\hat{b}_o$. The depth in RNNs is basically provided by stacked recurrent units [87]. With respect to equations (1) and (2) a stacked RNN with j layers are represented as:

$$s_p^{(1)} = g(W_y^{(1)} y_p + W_s^{(1)} s_{p-1}^{(1)} + b_s^{(1)}), i = 1, \quad (3)$$

$$s_p^{(i)} = g(W_y^{(i)} s_p^{(i-1)} + W_s^{(i)} s_{p-1}^{(i)} + b_s^{(i)}), \quad (4)$$

$$i = 2, \dots, j,$$

$$t_p = h(W_o s_p^{(j)} + b_o). \quad (5)$$

Here activation function and parametrization abide by single layer RNN. The weights and bias terms for each layer j are represented as $W_y^{(i)}$, $W_s^{(i)}$ and $b_s^{(i)}$. For this layer hidden state at timestamp p is $s_p^{(i)}$. Corresponding to j layers, stacked RNN has initial hidden state vectors as $s_0^{(1)}, \dots, s_p^{(j)}$.

3.3 Basic d-RNN

An alternative means to increase RNNs' depth is to consider time within single layer RNN. Single layer RNNs are restricted considering number of non-

linearities applied to recent inputs. [87] have addressed this restriction by adding intermediate non-linearities between input elements. Here computational steps are added between elements in sequence which increases runtime complexity. The delayed recurrent neural networks (d-RNNs) addresses this by increasing effective depth through introduction of delay between input and output.

The d-RNN can be defined as single layer RNN such that for any input respective output is obtained d timesteps later as shown in Fig. 3. Here d is network's delay. The initial hidden state for d-RNN is initialized in similar manner as an RNN. Delaying output requires special considerations on data that differ slightly from RNN. Input sequences need to have $P + d$ elements.

Depending on task being solved this can be achieved by adding null input element or including d additional elements in input sequence. When doing forward pass over d-RNN for inference, outputs from $p = 1$ to d are discarded as output appears after a delay. The output sequence has P elements. Training loss is computed by comparing expected output for input with delay factor. Thus, gradients are backpropagated only from delayed outputs.

Another RNN very similar in structure to single layer RNN is stacked RNN where additional in between layer connections are placed which adds depth in network. Any stacked RNN can be configured into a single-layer d-RNN which produces exact hidden states and output sequences. The depth from in between layer connections are replaced with temporal depth applied through output delays. Considering above equations parameters of single layer RNN using

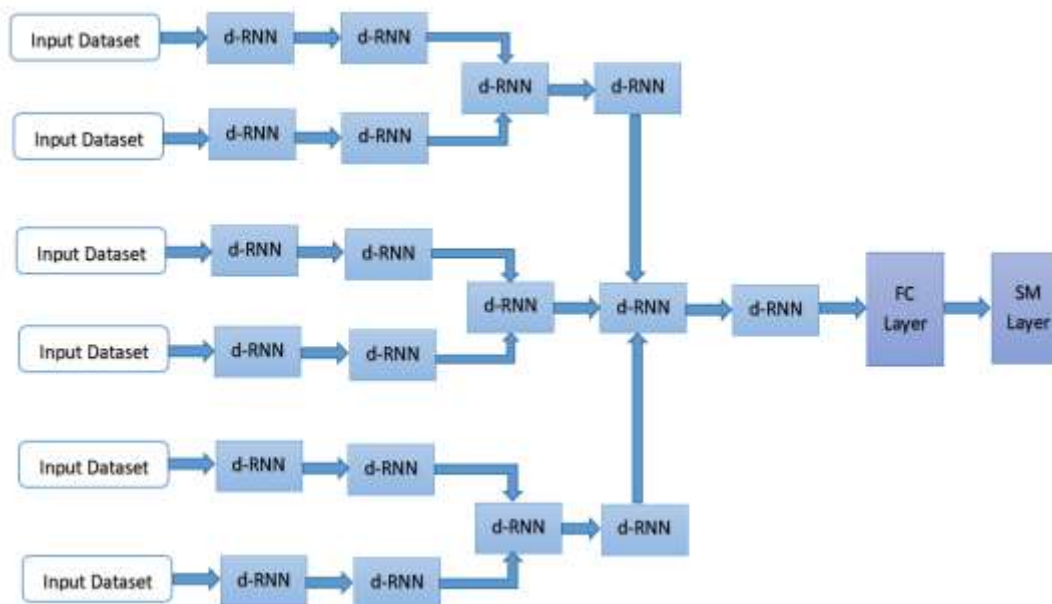


Fig. 4. Schematic representation of architecture of Hd-RNN [36]

weights and bias terms of k -layer stacked RNN as highlighted in [87]. It is observed from [87] that stacked RNN's each layer is converted into group of units in single layer RNN.

The structure of recurrent weight matrix $\hat{W}_h$ considers hidden state to act as buffer. Here each group of units receive inputs from itself and previous group.

This buffering mechanism processes information which eventually arrives at output after $k - 1$ timesteps. The model achieved is d-RNN with delay $k - 1$ and sparsely constrained weights. It is to be noted that d-RNN performs identical computations as stacked RNN by maintaining depth in layers for depth in time.

It has been proved that d-RNN parameterized by above equations is exactly equivalent to stacked RNN in above equations. This proposition can be extended towards recurrent cells with more complexity. A k -layer stacked RNNs can be represented as single layer d-RNN. The d-RNN in its weight matrices has specific sparsity structure which is not present in generic RNN or d-RNN. As such stacked RNN and d-RNN with sparsely constrained weights models are equivalent in

nature. They can be interchanged using weight matrix definitions in above equations.

3.4 Hierarchical d-RNN

Considering RNN and d-RNN in previous sections, we now discuss Hd-RNN [36] for semantic analysis of COVID19 reviews. The better approximation of network topologies like stacked RNN, bidirectional RNN and stacked bidirectional RNN by d-RNN with faster runtimes [36] serve major motivation. Hd-RNN differs from its non-hierarchical counterparts with respect to better classification accuracy taking similarities and running time parameters as data corpus grows [36].

A schematic representation of architecture of Hd-RNN is shown in Fig. 4. Here, d-RNN is used to model temporal sequences in COVID19 reviews. The results from d-RNN are combined together to form Hd-RNN. Hd-RNN comprises of 7 layers $d\text{-RNN}_1 \rightarrow d\text{-RNN}_2 \rightarrow d\text{-RNN}_3 \rightarrow d\text{-RNN}_4 \rightarrow d\text{-RNN}_5 \rightarrow fc \rightarrow sm$.

Here, $d\text{-RNN}_i$ with $(i = 1, 2, 3, 4, 5)$ depict layers with d-RNN nodes, fc is fully connected layer and sm is softmax layer. Each layer in Hd-RNN constitutes classifier's hierarchy and addresses

classification tasks [36] which plays a vital role in network's success.

In order to recover any single hierarchy, we can run split d-RNN on small subset of reviews having few words [36]. This helps in computation of seed classification value.

The input dataset subsets are developed randomly which is initiated at layer d-RNN₁. With initial classification value remaining part of data corpus is placed into seed class for which average similarity is present.

This leads to classification of entire dataset using only similarities to words with respect to small subset. This process is applied recursively to each class such that Hd-RNN is build up considering only small similarities fraction. The classification process continues till d-RNN₅.

This recursive phase has no measurements between classes at previous split. This results in robust version of Hd-RNN which aligns its measurements ms in order to resolve higher class structure resolution. The pseudo code for Hd-RNN is highlighted in Algorithm below.

Algorithm: HdRNN ($dRNN, ms, \{x_i\}_{i=1}^{WR_j}, CS_j$)

if $WR_j < ms$ **then return** $\{x_i\}_{i=1}^{WR_j}$

 Select $W \subseteq \{x_i\}_{i=1}^{WR_j}$ of size w uniformly at random

$C'_1, \dots, C'_{CS_j} \rightarrow dRNN(W, CS_j)$

 Set $C_1 \leftarrow C'_1, \dots, C_{WR_j} \leftarrow C'_{WR_j}$

for $x_i \in \{x_i\}_{i=1}^{WR_j} \setminus W$ **do**

$\forall s \in [CS_j], \alpha_s \leftarrow \frac{1}{|C'_j|} \sum_{x_h \in C'_j} S(x_i, x_h)$

$C_{\text{argmax}_{s \in [CS_j]} \alpha_s} \leftarrow C_{\text{argmax}_{s \in [CS_j]} \alpha_s} \cup \{x_i\}$

endfor

output $\{C_s, \text{HdRNN}(dRNN, ms, C_s, CS_j)\}_{j=1}^{CS_j}$

Hd-RNN is specified with respect to success probability of success in recovering true hierarchy CS^* , measurement ms and runtime complexity. Certain restrictions are placed on similarity function SM such that similarities agree with hierarchy up to some random noise:

P1 For each $x_i \in CS_j \in CS^*$ and $i \neq j$ we have:

$$\min_{x_p \in CS_j} \mathbb{E} \mathbb{P}[S(x_i, x_p)] - \min_{x_p \in CS_j'} \mathbb{E} \mathbb{P}[S(x_i, x_p)] \geq \delta \geq 0.$$

Here expectations are taken with respect to possible noise on SM .

P2 For each $x_i \in CS_j$, a set of W_j words of size w_j drawn uniformly from CS_j satisfies:

$$\text{Prob} \left(\min_{x_p \in CS_j} \mathbb{E} \mathbb{P}[S(x_i, x_p)] - \sum_{x_p \in W_j} \frac{S(x_i, x_p)}{w_j} > \epsilon \right) \leq 2e^{\left\{ \frac{-2w_j \epsilon^2}{\sigma^2} \right\}}.$$

Here $\sigma^2 \geq 0$ parameterizes noise on similarity function SM . Similarly set W_j of size w_j drawn uniformly from class CS_j with $i \neq j$ satisfies:

$$\text{Prob} \left(\sum_{x_p \in W_j} \frac{S(x_i, x_p)}{w_j} - \min_{x_p \in C_j} \mathbb{E} \mathbb{P}[S(x_i, x_p)] > \epsilon \right) \leq 2e^{\left\{ \frac{-2w_j \epsilon^2}{\sigma^2} \right\}}.$$

The condition **P1** highlights similarity from word y_i to its class should have expectation larger than similarity from same word in other class. This relates towards tighter classification condition [36] with lesser stringency than earlier results. The condition **P2** highlights within-and-between-class similarities which concentrate away from each other. This condition is satisfied when similarities remain constant in expectation perturbed with respect to any subgaussian noise.

Considering feature learning d-RNN extracts temporal features for sentiment data sequences. After obtaining sentiment sequence features, fully connected layer fc and softmax layer sm performs classification. This architecture addresses vanishing gradient problem [36]. Network neurons are adopted in last recurrent layer d-RNN₅. The first four d-RNN layers use tanh activation function. This is trade-off between improving representation ability and avoiding any over fitting. The number of weights in network is more than in tanh neuron.

The network can be overfitted with limited data training sequences. Specifically, when CS is known and constant across splits in hierarchy, above assumptions are practically violated. This is resolved by fine tuning this algorithm with heuristics. The eigengap is employed where CS is chosen such that eigenvalues gap of Laplacian is large. All subsampled words in data are discarded with low degree when restricted to sample which removes underrepresented sample classes.

In averaging phase if words in data are not similar to any represented class, new word class is developed.

3.5 Approximation with Hierarchical Bidirectional Recurrent Neural Networks

It is very well known that d-RNN can have equivalent structure as stacked RNN when its weight matrices are constrained. If these constraints are ignored, d-RNN peeks at future inputs. It computes delayed output considering time using also inputs which are beyond specified timestep.

An analogous idea has been used as benchmark for bidirectional recurrent neural networks (BRNN) [36, 88]. It has been shown that BRNN are superior to d-RNN considering relatively simple problems. However, it is not clear that this comparison holds true for problems requiring more non-linear solutions. The similar proposition holds for hierarchical bidirectional recurrent neural networks (HBRNN) [36].

If a recurrent network computes its output for specified time by exploiting future input elements, what are necessary conditions in order to approximate its BRNN and HBRNN. Moreover, can d-RNN and Hd-RNN have similar results. And with these conditions, is it more beneficial to use d-RNN and Hd-RNN instead of BRNN and HBRNN. There are number of non-linear transformations [36] where each network applies to any input element prior to computation of output at initial timestep. Only past inputs are processed by generic RNN where number of non-linearities decrease when inputs are close to initial timestep.

BRNN has similar behavior considering causal inputs. For acausal inputs it is symmetrically augmented. For casual inputs Hd-RNN has similar behavior with higher number of non-linearities. This remains for first d acausal inputs with non-linearities decreasing. For Hd-RNN to have at least similar number of non-linearities as BRNN for every sequential element, a delay is required which is twice as sequence length. Hd-RNN can superceed BRNN when non-linear influence of nearby acausal inputs on learned function is superior than farther elements. When Hd-RNN is used in order to approximate BRNN, it also decreases computational cost. For length of considerable sequence stacked BRNN computes both forward and backward RNNs for each layer prior to computation of next layer. As a result of this synchronization parallelization is not allowed which

increases runtime. Forward passes for d-RNN takes additional steps, but synchronization does not affect it. In highly parallel hardware, runtime of k layer stacked BRNN is at least k times slower than d-RNN or Hd-RNN. In lines with d-RNN, Hd-RNN can also be used in critical output values in near real time applications [89, 90].

3.6 Twitter Based Sentiment Analysis Framework for COVID19 in India

In this study a novel framework is presented which uses twitter information in order to understand public behavior in India during COVID19 pandemic. This research addresses sentiments of people in different parts of India during outbreak of pandemic in 2020 and 2021.

Considering datasets defined in section 3.1 our analysis is mainly concentrated in states of Maharashtra, Delhi, Karnataka, Kerala and Tamil Nadu where maximum number of COVID19 active cases are detected. Fig. 5 highlights major components of this framework. It is to be noted that framework is attributed with multi-label classification with multiple outcomes. The social media language has been rapidly evolving. As a result of this special phrases, emotion symbols and abbreviations are present in tweets. These are transformed for building language models [36]. The languages predominant in stated 5 Indian states are Marathi, Hindi, Kannada, Malayalam and Tamil which are used in combination with English. Hence, a transformation is also performed for certain words, emotions and character symbols highlighted in these stated languages.

The computational framework for sentiment analysis involves following steps: (a) tweet extraction, (b) tweet pre-processing, (c) model development with training, and (d) prediction. Tweet extraction involves processing on COVID19 dataset mentioned in section 3.1. During this process special symbols and abbreviations present in tweets are transformed towards building language model. Few such instances are represented in Table 2.

It is to be noted that transformation is performed for specific words, emotions and character symbols which are represented in Marathi, Hindi, Kannada, Malayalam and Tamil.

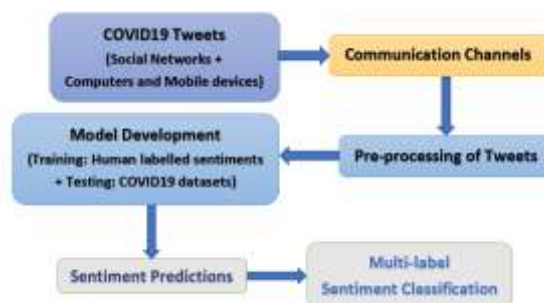


Fig. 5. Twitter based sentiment analysis framework for COVID19 [36]

Table 2. Instances of special symbols and abbreviations present in tweets

Original Phrases	Transformed Word
Facemasks	face masks
Socialdistancing	social distancing
😊	smiling faces
🛏	beds
🧴	sanitisors
😉	winks

The experimental dataset features 17 different sentiments [36], which are labelled by a group of 1000 experts for 100,000,000 tweets during COVID19 in 2020 and 2021. In tweet pre-processing each word is converted into its corresponding GLoVe vector.

Here, each word is converted into a 500-dimensional vector. The main reason behind selection of GLoVe embedding involves good results it has shown for sentiment analysis [48].

From each word, GloVe vector is passed towards respective d-RNN and Hd-RNN models for training. The trained models are evaluated initially. After successful evaluation they are used for COVID19 sentiment analysis considering test data from states of Maharashtra, Delhi, Karnataka, Kerala and Tamilnadu. The trained models are used for classification of 17 sentiments. The evaluation metric is so chosen such that it evaluates trained models in best possible way [36].

Hence, it is required that metric captures correct loss [36] arising from misclassification and gives best representative score. As such classification here is of multi-label in nature and is based on

binary cross entropy loss, hamming loss, jaccard coefficient score, label ranking average precision score and F1 score [36]. Binary cross entropy loss represents softmax activation alongwith cross entropy loss.

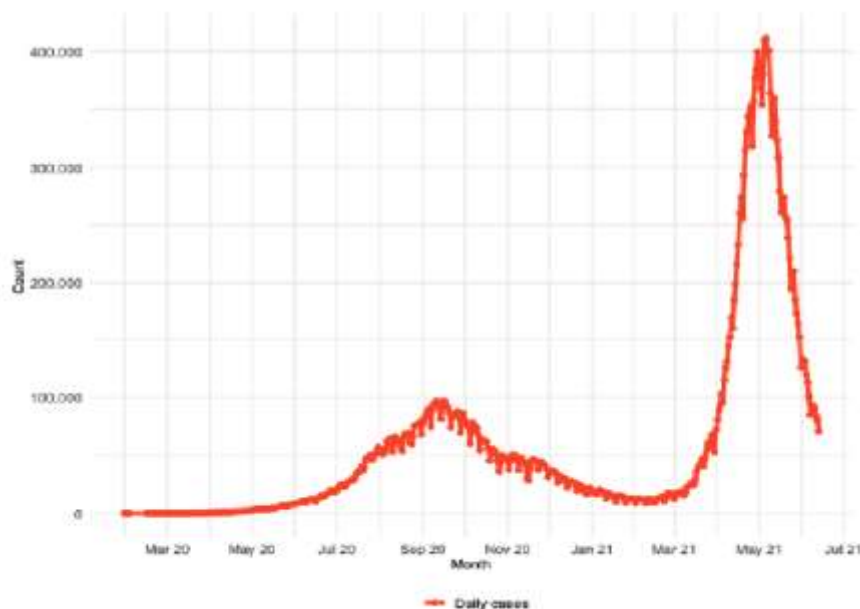
Hamming loss generates bit string of class labels using XOR between actual and predicted labels and averages with respect to dataset instances. Jaccard coefficient score measures overlap between actual and predicted labels with respect to similarity and diversity attributes. Label ranking average precision score calculates percentage of higher ranked labels which resemble true labels considering given samples. F1 score highlights balance between precision and recall measures.

Since classification here is of multi-label in nature, combination of two variants of F1 score viz F1 macro and F1 micro are used towards evaluation of training models.

Considering experimental framework highlighted in Fig. 5 multi-label classification is performed with respect to d-RNN and hierarchical d-RNN models. We use 80:10:10 ratio of dataset

Table 3. Analysis of training performance for d-RNN, Hd-RNN and BERT for COVID19 datasets

Metric used	d-RNN	Hd-RNN	BERT
Binary Cross Entropy Loss	0.279	0.260	0.370
Hamming Loss	0.160	0.155	0.145
Jaccard Score	0.435	0.437	0.530
Label Ranking Precision Score	0.507	0.517	0.779
F1 Macro	0.449	0.460	0.570
F1 Micro	0.500	0.505	0.589

**Fig. 6.** India with first major peak (mid-September 2020) and second major peak (mid-May 2021) [36]

for training, validation and testing. The aforementioned models are trained using COVID19 dataset [36] stated in section 3.1. This dataset has 100,000,000 tweets collected between March 2020 to June 2021.

In d-RNN and Hd-RNN hyperparameters are determined considering experiments performed. GloVe embedding uses word vector of size 500 in order to provide data representation [48].

A dropout regularization probability of 0.75 is used for d-RNN and hierarchical d-RNN models which feature 500 input units, two layers with 128 and 64 hidden units and an output layer with 17 units for sentiment classification. Here, BERT model is used as benchmark to validate results

obtained from d-RNN and hierarchical d-RNN models. The main reason for considering BERT models lies in success obtained by using these models in sentiment analysis [36]. In BERT default hyperparameters are used. For BERT base uncased model learning rate is tuned.

BERT architecture has dropout layer and linear activation layer with 17 outputs corresponding to 17 sentiments. Table 3 shows model training results for 10 experiments with different initial weights and biases for respective models with different performance metrics. In India given huge population with large number of densely populated cities [36, 39] COVID19 management is plagued with major challenges.

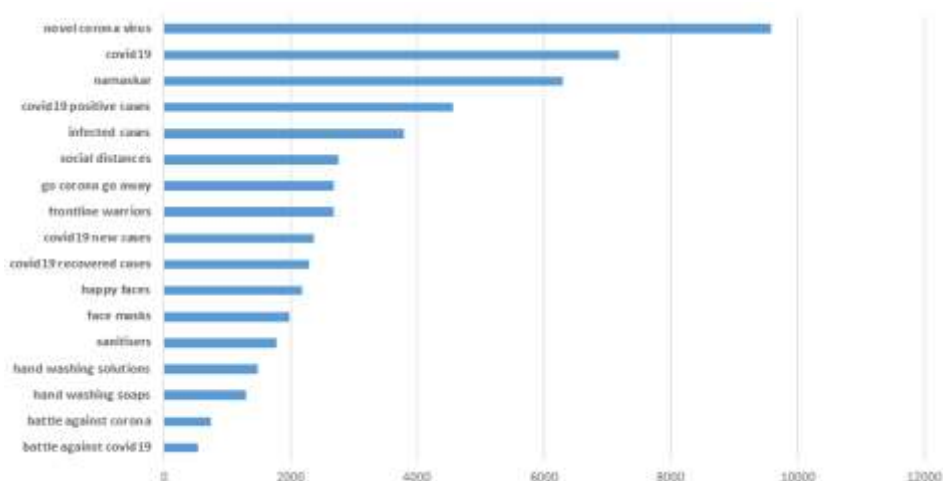


Fig. 7. Bi-grams for cases prevalent in India

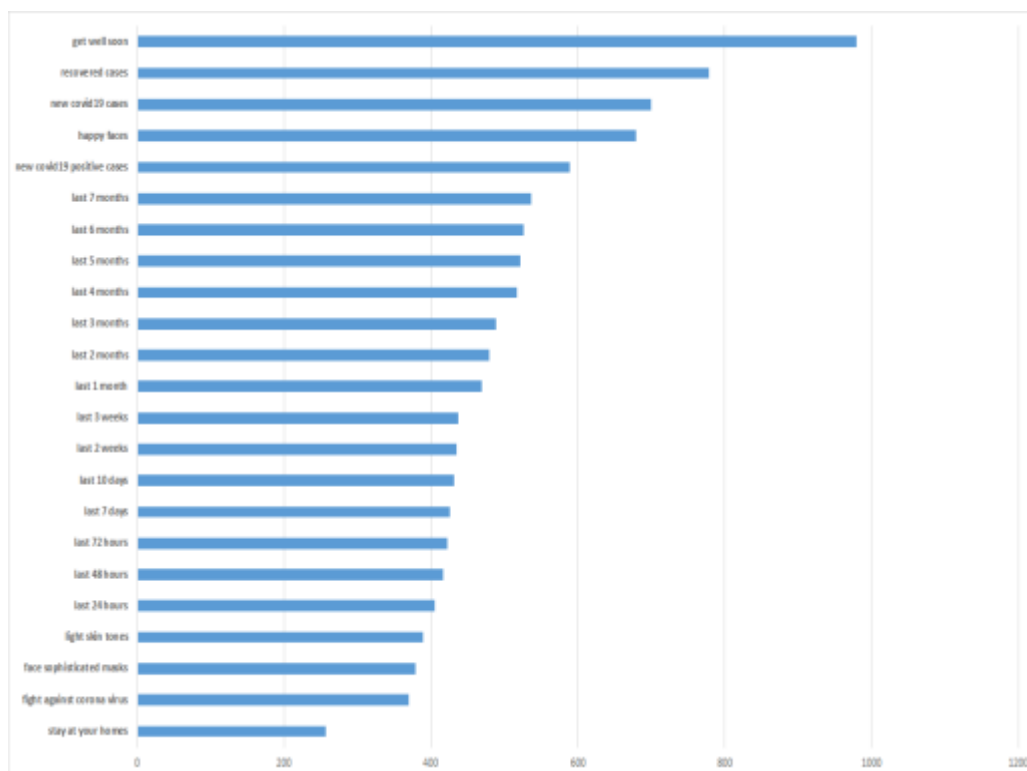


Fig. 8. Tri-grams for cases prevalent in India

The first COVID19 case in India was detected on 30th January 2020. Thereafter India started lockdown as a result of which situation gradually changed. On 22nd March 2021 India had more than

11.6 million confirmed cases with more than 160,000 deaths. This made India third highest with confirmed cases after United States and Brazil. India was 8th in world with more than 300,000

Table 4. Certain situations of tweets which are captured in most prominent bi-grams

Month	Tweets	Bi-gram
May 2020	#lockdown times#.....	pointing up
July 2020	face masks mandatory.....	pointing up
September 2020	positive cases increasing....	Pointing up
March 2021	vaccination necessary.....	with folded hands
April 2021	#lockdown times#.....	pointing up
May 2021	vaccination mandatory.....	with folded hands
June 2021	partial lockdown.....	Pointing up

Table 5. Certain situations of tweets which are captured in most prominent tri-grams

Month	Tweets	Tri-gram
September 2020	positive cases increasing....	Pointing up
March 2021	vaccination necessary.....	with folded hands
April 2021	#lockdown times#.....	pointing up
May 2021	vaccination mandatory.....	with folded hands
June 2021	partial lockdown.....	Pointing up

Confident	2379	237	175	235	269	300	389	89	160	400	989
Grateful	237	507	35	17	49	35	35	37	75	99	100
Understand	175	35	400	22	57	79	45	7	7	35	69
Not Confident	235	17	22	1435	269	279	437	100	70	270	579
Worrying	269	49	57	269	1700	375	469	96	165	389	537
Not Happy	300	35	79	279	375	2233	737	70	189	300	750
Angry	389	35	45	437	469	737	3596	269	179	550	1437
Refusal	89	37	7	100	96	70	269	637	60	289	189
Official Document	160	75	7	70	165	189	179	60	1232	287	99
Surprised	400	99	35	270	389	300	550	289	287	2035	735
Amusing	989	100	69	579	537	750	1437	189	99	735	4579

Fig. 9. Heatmap depicting occurrences of given sentiment with respect to remaining sentiments for tweets from training dataset [36]

active cases prior to second wave. India witnessed its first major peak around middle of September 2020 with close to 100,000 cases daily. This gradually decreased to around 11,000 cases daily by end of January 2021. In March 2021 India witnessed its second peak when cases began rising faster and by 22nd March, 2021 India had 47,000 daily new active cases [39].

In first six months, states of Maharashtra, Delhi and Tamil Nadu led COVID19 infections [39] with

city of Mumbai having highest number of active cases. In later half of 2021, Delhi cases reduced but still remained in leading 8 states [39]. In 2021, Maharashtra continued as state of highest infections and in March, 2021 it featured more than half of new active cases on a weekly basis. Delhi continued with less than a thousand daily active cases. Alongwith this Karnataka, Kerala and Tamilnadu continued to show good number of daily active cases.

Table 6. Comparative analysis of k -fold cross validation and accuracy of all models [36]

Models	k -fold (1)	k -fold (2)	k -fold (3)	k -fold (4)	k -fold (5)	test level
NB	0.79	0.80	0.82	0.79	0.84	0.84
SVM	0.82	0.84	0.85	0.87	0.87	0.87
kNN	0.75	0.76	0.77	0.78	0.79	0.79
RF	0.79	0.77	0.78	0.80	0.79	0.80
GB	0.84	0.85	0.87	0.84	0.85	0.87
AB	0.85	0.84	0.82	0.84	0.85	0.85
DT	0.77	0.78	0.79	0.80	0.80	0.80
d-RNN	0.95	0.96	0.97	0.96	0.96	0.96
Hd-RNN	0.96	0.97	0.98	0.99	0.97	0.99

Table 7. Comparative analysis of k -fold cross validation and F1 Score of all models [36]

Models	k -fold (1)	k -fold (2)	k -fold (3)	k -fold (4)	k -fold (5)	test level
NB	0.80	0.80	0.82	0.84	0.85	0.85
SVM	0.84	0.85	0.87	0.88	0.89	0.89
kNN	0.77	0.78	0.79	0.80	0.80	0.80
RF	0.80	0.79	0.79	0.82	0.80	0.82
GB	0.85	0.87	0.87	0.88	0.85	0.88
AB	0.87	0.85	0.84	0.85	0.87	0.87
DT	0.79	0.79	0.82	0.82	0.82	0.82
d-RNN	0.96	0.97	0.97	0.97	0.97	0.97
Hd-RNN	0.97	0.98	0.98	0.99	0.99	0.99

Table 8. Comparative analysis of k -fold cross validation and precision of all models [36]

Models	k -fold (1)	k -fold (2)	k -fold (3)	k -fold (4)	k -fold (5)	test level
NB	0.82	0.84	0.84	0.80	0.85	0.85
SVM	0.84	0.85	0.87	0.88	0.89	0.89
kNN	0.77	0.78	0.79	0.80	0.80	0.80
RF	0.82	0.79	0.80	0.82	0.80	0.82
GB	0.85	0.87	0.88	0.85	0.88	0.88
AB	0.87	0.85	0.84	0.85	0.87	0.87
DT	0.79	0.80	0.80	0.80	0.80	0.80
d-RNN	0.96	0.97	0.98	0.97	0.98	0.98
Hd-RNN	0.98	0.98	0.98	0.99	0.99	0.99

Table 9. Comparative analysis of k -fold cross validation and recall of all models [36]

Models	k -fold (1)	k -fold (2)	k -fold (3)	k -fold (4)	k -fold (5)	test level
NB	0.79	0.79	0.80	0.82	0.84	0.84
SVM	0.82	0.84	0.85	0.86	0.88	0.88
kNN	0.76	0.77	0.78	0.79	0.79	0.79
RF	0.79	0.78	0.78	0.80	0.79	0.80
GB	0.84	0.85	0.85	0.87	0.84	0.87
AB	0.85	0.84	0.82	0.84	0.87	0.87
DT	0.78	0.78	0.80	0.80	0.79	0.79
d-RNN	0.95	0.96	0.96	0.96	0.96	0.96
Hd-RNN	0.96	0.97	0.97	0.98	0.98	0.98

Table 10. Comparative analysis of BERT variants, LSTM and BD-LSTM with d-RNN and Hd-RNN models (L: hidden layers, H: hidden size, A: attention heads) [36]

BERT/Other Models	Accuracy	Precision	Recall	F1 Score
L-2 H-128 A-2	0.61	0.60	0.57	0.61
L-2 H-256 A-4	0.82	0.80	0.77	0.79
L-2 H-512 A-8	0.84	0.82	0.85	0.85
L-2 H-768 A-12	0.84	0.80	0.85	0.85
L-4 H-128 A-2	0.80	0.79	0.78	0.80
L-4 H-256 A-4	0.82	0.84	0.85	0.84
L-4 H-512 A-8	0.84	0.82	0.80	0.82
L-4 H-768 A-12	0.87	0.84	0.85	0.84
L-6 H-128 A-2	0.82	0.80	0.84	0.82
L-6 H-256 A-4	0.82	0.84	0.82	0.82
L-6 H-512 A-8	0.85	0.82	0.82	0.82
L-6 H-768 A-12	0.87	0.85	0.84	0.84
L-8 H-128 A-2	0.80	0.80	0.82	0.82
L-8 H-256 A-4	0.82	0.84	0.82	0.82
L-8 H-512 A-8	0.85	0.82	0.82	0.82
L-10 H-128 A-2	0.82	0.80	0.84	0.82
L-10 H-256 A-4	0.84	0.80	0.85	0.85
L-10 H-512 A-8	0.84	0.82	0.84	0.85
L-12 H-128 A-2	0.82	0.80	0.87	0.87
L-12 H-256 A-4	0.85	0.80	0.80	0.85
L-12 H-512 A-8	0.89	0.84	0.89	0.89
LSTM	0.90	0.93	0.94	0.95
BD-LSTM	0.93	0.94	0.95	0.96
d-RNN	0.96	0.95	0.97	0.97
Hd-RNN	0.99	0.96	0.95	0.99

We have applied COVID19 datasets from different parts of India including nationwide COVID19 sentiments with 5 major states having majority of COVID19 cases.

The trends have shown that these 5 states had two major peaks followed by several minor ones. However, India had first major peak around mid-September, 2020 and second major peak around mid-May, 2021 as shown in Fig. 6 [36].

4 Experiments and Results

Considering computational methodology in section 3, in this section experiments and results are presented.

4.1 Analysis of COVID19 Results

The test dataset [36, 39] contains COVID19 tweets between March 2020 to June 2021.

It comprised of more than 750,000 tweets from India. Five more datasets are generated considering states of Maharashtra, Delhi, Karnataka, Kerala and Tamilnadu with around 22,000 tweets each.

It is observed that number of tweets in India follows identical trend as number of COVID19 cases increase till July, 2020 after which number of tweets decline.

Again, number of COVID19 cases increase till April, 2021 after which number of tweets decline. There is similar pattern for Maharashtra and Kerala.

Confident	55217	17235	2737	875	3522	375	4975	175	3132	5095	8686
Grateful	17235	17809	375	175	237	217	1479	79	1675	800	1789
Understand	2737	375	3799	219	190	179	96	7	9	7	61
Not Confident	875	175	219	7917	3569	1775	3707	175	170	289	589
Worrying	3522	237	190	3569	23707	379	475	196	175	489	673
Not Happy	375	217	179	1775	379	10179	1737	170	289	389	2750
Angry	4975	1479	96	3707	475	1737	45975	275	187	559	3437
Refusal	175	79	7	175	196	170	275	5596	280	389	489
Official Document	3132	1675	9	170	175	289	187	280	21770	387	599
Surprised	5095	800	7	289	489	389	559	389	387	37789	3735
Amusing	8686	1789	61	589	673	2750	3437	489	599	3735	37275
Confident Grateful Understand Not Confident Worrying Not Happy Angry Refusal Official Document Surprised Amusing											

Fig. 10. Heatmap showing number of occurrence of given sentiment with respect to remaining sentiments for India datasets using Hd-RNN [36]

Confident	5528	7237	2737	8735	3523	475	4979	275	3135	7095	8989
Grateful	7235	17809	2375	179	269	228	1979	179	2875	889	3789
Understand	2737	2375	3779	319	190	179	96	37	79	89	261
Not Confident	8735	179	319	8917	5569	1775	3709	275	179	3289	3589
Worrying	3523	269	190	5569	28707	879	475	196	275	3789	1673
Not Happy	475	228	179	1775	879	179	1937	170	289	489	3750
Angry	4979	1979	96	3709	475	1937	975	375	89	959	479
Refusal	275	179	37	275	196	170	375	559	599	389	489
Official Document	3135	2875	79	179	275	289	89	599	2770	987	3599
Surprised	7095	889	89	3289	3789	489	959	389	987	3789	3775
Amusing	8686	3789	261	3589	1673	3750	479	489	3599	3775	3779
Confident Grateful Understand Not Confident Worrying Not Happy Angry Refusal Official Document Surprised Amusing											

Fig. 11. Heatmap showing number of occurrence of given sentiment with respect to remaining sentiments for Maharashtra datasets using Hd-RNN [36]

Confident	617	1735	277	87	3522	379	975	175	3132	5099	8999
Grateful	1735	809	375	176	237	217	1479	179	3675	800	3789
Understand	277	375	799	279	190	179	96	37	95	79	2761
Not Confident	87	176	279	917	569	3775	3707	179	500	989	589
Worrying	3522	237	190	569	707	579	475	396	275	889	1673
Not Happy	379	217	179	3775	579	179	173	170	289	389	2750
Angry	975	1479	96	3707	475	173	7575	375	185	559	3437
Refusal	175	179	37	179	396	170	375	5796	980	989	589
Official Document	3132	3675	95	500	275	289	185	980	770	3787	3599
Surprised	5099	800	79	989	889	389	559	989	3787	789	3737
Amusing	8999	3789	2761	589	1673	2750	3437	589	3599	3737	275
Confident Grateful Understand Not Confident Worrying Not Happy Angry Refusal Official Document Surprised Amusing											

Fig. 12. Heatmap showing number of occurrence of given sentiment with respect to remaining sentiments for Delhi datasets using Hd-RNN [36]

For Delhi, Karnataka and Tamilnadu situation is slightly different as first peak was observed in July,

2020 with increasing tweets that declined afterwards and did not keep up with second peak

of cases in September, 2020. Again, number of COVID19 cases increase till April, 2021 after which number of tweets decline. This indicates that as more cases were observed in early months, there was much concern which eased before major peak was reached and number of tweets were drastically decreased. There could be signs of fear, depression and anxiety as tweets decreased drastically after July, 2020 with increasing cases. Fig. 7 and Fig. 8 show number of bi-grams and tri-grams prevalent in India. In bi-grams it is observed that novel corona virus is most used followed by covid19. Similarly, namaskar is most used followed by covid19 positive cases and infected cases. In order to provide better understanding of tweets some examples are highlighted as shown in Table 4. In case of tri-grams we can find more information in tweets as shown in Table 5.

Fig. 9 shows number of occurrences of a given sentiment in relation to rest of tweets sentiments in training datasets [36, 39]. Now we present results of COVID19 tweets prediction in India, Maharashtra, Delhi, Karnataka, Kerala and Tamilnadu by considering them as individual datasets. The dataset mentioned in Section 3.1 has been used here for training. Fig. 10 above presents distribution of sentiments predicted by Hd-RNN for respective datasets considering stated span of time. Hd-RNN has provided best results for training data. In Fig. 11 sentiments predicted are reviewed considering heatmap in order to examine number of occurrences of given sentiment with respect to rest sentiments. These heatmaps indicate how two sentiments have been expressed and provides more insights regarding positive and negative sentiments. Fig. 12 provides visualisation of tweets distribution with number of combination sentiments.

4.2 Validation of COVID19 Results

After an initial analysis of results in previous section, we present further validation of obtained results here. In order to achieve this cross validation is used. This is a resampling method which evaluates developed models on small data samples. The given data samples are divided into number of groups such that we have k -fold cross validation. It is used in order to estimate model's ability on unknown data. It provides lesser degree

of biasedness or positive estimation considering model's ability in comparison to other approaches. As such it considers simple training and testing splits.

The success of k -fold cross validation lies in choosing a set of split numbers instead of any specific split number. This helps in checking acceptability of dataset as well as addresses issues related to skewness in datasets.

Since COVID19 datasets have an uneven class distribution, most intuitive performance metric to be used here is precision. Closely associated with precision is recall performance metric which also works well here as well as F1 Score which is weighted average of precision and recall. Another important performance metric which could have been used here is accuracy but because datasets are not symmetric, it will work well here.

In order to make our experimental hypothesis stronger we now present comparative performance analysis of d-RNN and Hd-RNN models with some traditional machine learning models such as NB, SVM, kNN, RF, GB, AB and DT. Table 6 highlights k -fold cross validation and accuracy comparison of all these models. Table 7 presents k -fold cross validation and F1 Score comparison of all these models. Table 8 shows k -fold cross validation and precision comparison of all these models. Table 9 shows k -fold cross validation and recall comparison of all these models.

As BERT model has been used here for evaluating test datasets with prediction validation, we use different variants of BERT [36] as well as LSTM [36] and BD-LSTM [36] models as shown in Table 10 in order to compare performance of d-RNN and Hd-RNN models.

5 Conclusion

In this work we have presented sentiment analysis of COVID19 infections with d-RNN and Hd-RNN as major computational models. The time span considered is from January 2020 to June 2021. We considered tweets from different regions of India. We reviewed tweets from specific regions such as Maharashtra, Delhi, Karnataka, Kerala and Tamilnadu. The models are trained with COVID19 datasets hand labelled tweets. The majority of tweets have highlighted optimism, fear and

uncertainty during infections of COVID19 cases in India. There has been variability in number of tweets during peak of new cases.

The predictions indicate that although majority of population have been optimistic, a significant group has been disturbed by way pandemic was handled by Indian government.

This computational framework can be used for better COVID19 management in order to support cases of depression and mental health issues. The experimental results are validated considering various traditional machine learning as well as different variants of BERT models. The results with d-RNN and Hd-RNN highlight superiority of proposed methods.

The computational model can be used for different regions, countries, ethnic and social groups. This can also be extended to understand reactions towards vaccinations with rise of antivaccine sentiments given fear, insecurity and unpredictability of COVID19 situations. This computational framework incorporates topic modelling with sentiment analysis which provides more details during COVID19 cases with respect to various government rules and regulations. As concluding remarks we would like to mention that present Indian government has been unsuccessful in addressing stress and strain through which country's economy is passing through.

References

1. Gorbalenya, A.E., Baker, S.C., Baric, R.S., De Groot, R.J., Drosten, C., Gulyaeva, A.A., Haagmans, B.L., Lauber, C., Leontovich, A.M., Neuman, B.W., Penzar, D., Perlman, S., Poon, L.L.M., Samborskiy, D.V., Sidorov, I.A., Sola, I., Ziebuhr, J. (2020). The species severe acute respiratory syndrome-related coronavirus: classifying 2019-nCoV and naming it SARS-CoV-2. *Nature Microbiology*, Vol. 5, No. 4, pp. 536–544. DOI: 10.1038/s41564-020-0695-z.
2. Monteil, V., Kwon, H., Prado, P., Hagelkrüys, A., Wimmer, R.A., Stahl, M., Leopoldi, A., Garreta, E., Hurtado del Pozo C., Prosper, F., Romero J.P., Wirnsberger, G., Zhang, H., Slutsky, A.S., Conder, R., Montserrat, N., Mirazimi, A., Penninger, J.M. (2020). Inhibition of SARSCoV-2 infections in engineered human tissues using clinical-grade soluble human ACE2. *Cell*, Vol. 181, No. 4, pp. 905–913. DOI: 10.1016/j.cell.2020.04.004.
3. WHO. (2020). Coronavirus disease 2019 (COVID-19): Situation report, 72. World Health Organization, April, 1, 2020.
4. Cucinotta, D., Vanelli, M. (2020). WHO declares COVID-19 a pandemic. *Acta Bio-medica: Atenei Parmensis*, Vol. 91, No. 1, pp. 157–160. DOI: 10.23750/abm.v91i1.9397.
5. Wikipedia. COVID19.
6. ILO, FAO, IFAD, WHO. (2020). Impact of COVID-19 on people's livelihoods, their health and our food systems. World Health Organization (WHO).
7. Siche, R. (2020). What is the impact of COVID-19 disease on agriculture? *Scientia Agropecuaria*, Vol. 11, No. 1, pp. 3–6. DOI: 10.17268/sci.agropecu.2020.01.00.
8. Richards, M., Anderson, M., Carter, P., Ebert, B. L., Mossialos, E. (2020). The impact of the COVID-19 pandemic on cancer care. *Nature Cancer*, Vol. 1, No. 6, pp. 565–567. DOI: 10.1038/s43018-020-0074-y.
9. Tiwari, A., Gupta, R., Chandra, R. (2021). Delhi air quality prediction using LSTM deep learning models with a focus on COVID-19 lockdown. DOI: 10.48550/arXiv.2102.10551.
10. Upadhyay, A. (2021). Impact of Covid-19 on Indian economy. *The Times of India*.
11. Golbeck, J., Robles, C., Edmondson, M., Turner, K. (2011). Predicting personality from twitter. *IEEE 3rd international conference on privacy, security, risk and trust and IEEE 3rd international conference on social computing*, pp. 149–156. DOI: 10.1109/PASSAT/SocialCom.2011.33.
12. Quercia, D., Kosinski, M., Stillwell, D., Crowcroft, J. (2011). Our twitter profiles, our selves: Predicting personality with twitter. *IEEE 3rd international conference on privacy, security, risk and trust and IEEE 3rd international conference on social computing*, pp. 180–185. DOI: 10.1109/PASSAT/SocialCom.2011.26.
13. Bittermann, A., Batzdorfer, V., Müller, S.M., Steinmetz, H. (2021). Mining twitter to detect hotspots in psychology. *Zeitschrift für Psychologie*, Vol. 229, No. 1, pp. 3–14. DOI: 10.1027/2151-2604/a000437.
14. Lin J. (2015). On building better mousetraps and understanding the human condition: Reflections on big data in the social sciences. *The ANNALS of the American Academy of Political and Social Science*, Vol. 659, No. 1, pp. 33–47. DOI: 10.1177/0002716215569174.

15. **Coppersmith, G., Dredze, M., Harman, C. (2014).** Quantifying mental health signals in Twitter. *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pp. 51–60.
16. **Murphy, S.C. (2017).** A hands-on guide to conducting psychological research on Twitter. *Social Psychological and Personality Science*, Vol. 8, No. 4, pp. 396–412. DOI: 10.1177/1948550617697178.
17. **Zhou, Y., Na, J.C. (2019).** A comparative analysis of Twitter users who tweeted on psychology and political science journal articles. *Online Information Review*, Vol. 43, No. 7, pp. 1188–1208. DOI: 10.1108/OIR-03-2019-0097.
18. **Wang, W., Hernandez, I., Newman, D.A., He, J., Bian, J. (2016).** Twitter analysis: Studying US weekly trends in work stress and emotion. *Applied Psychology*, Vol. 65, No. 2, pp. 355–378. DOI: 10.1111/apps.12065.
19. **Manning, C., Schutze, H. (1999).** *Foundations of statistical natural language processing*. MIT Press.
20. **Chaudhuri, A., Ghosh, S.K. (2016).** Sentiment analysis of customer reviews using robust hierarchical bidirectional recurrent neural network. *Artificial Intelligence Perspectives in Intelligent Systems*, Springer, Cham, Vol. 464, pp. 249–261. DOI: 10.1007/978-3-319-33625-1_23.
21. **Chaudhuri, A. (2019).** Visual and text sentiment analysis through hierarchical deep learning networks. *Springer Briefs in Computer Science*, Springer, pp. 1–98.
22. **Liu, B., Zhang, L. (2012).** A survey of opinion mining and sentiment analysis. *Mining Text Data*, Springer, pp. 415–463. DOI: 10.1007/978-1-4614-3223-4_13.
23. **Medhat, W., Hassan, A., Korashy, H. (2014).** Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, Vol. 5, No. 4, pp. 1093–1113. DOI: 10.1016/j.asej.2014.04.011.
24. **Hussein, D. (2018).** A survey on sentiment analysis challenges. *Journal of King Saud University – Engineering Sciences*, Vol. 30, No. 4, pp. 330–338. DOI: 10.1016/j.jksues.2016.04.002.
25. **Beigi, G., Hu, X., Maciejewski, R., Liu, H. (2016).** An overview of sentiment analysis in social media and its applications in disaster relief. *Sentiment Analysis and Ontology Engineering*, Springer, Cham, Vol. 639, pp. 313–340. DOI: 10.1007/978-3-319-30319-2_13.
26. **Drus, Z., Khalid, H. (2019).** Sentiment analysis in social media and its application: Systematic literature review. *Procedia Computer Science*, Vol. 161, pp. 707–714. DOI: 10.1016/j.procs.2019.11.174.
27. **Kolenik, T., Gams, M. (2021).** Intelligent cognitive assistants for attitude and behavior change support in mental health: State-of-the-art technical review. *Electronics*, Vol. 10, No. 11, pp. 1–34. DOI: 10.3390/electronics10111250.
28. **Alhijawi, B., Awajan, A. (2021).** Prediction of movie success using twitter temporal mining. *Proceedings of 6th International Congress on Information and Communication Technology, Lecture Notes in Networks and Systems*, Springer, Vol. 235, pp. 105–116. DOI: 10.1007/978-981-16-2377-6_12.
29. **Kolenik, T., Gams, M. (2021).** Persuasive technology for mental health: One step closer to (Mental Health Care) equality? *IEEE Technology and Society Magazine*, Vol. 40, No. 1, pp. 80–86. DOI: 10.1109/MTS.2021.3056288.
30. **Biltawi, M., Etaiwi, W., Tedmori, S., Hudaib, A., Awajan, A. (2016).** Sentiment classification techniques for Arabic language: A survey. *7th International Conference on Information and Communication Systems (ICICS)*, pp. 339–346. DOI: 10.1109/IACS.2016.7476075.
31. **Fang, X., Zhan, J. (2015).** Sentiment analysis using product review data. *Journal of Big Data*, Vol. 2, No. 5, pp. 1–14. DOI: <https://doi.org/10.1186/s40537-015-0015-2>.
32. **Haykin, S. (2008).** *Neural networks and learning machines*. 3rd ed. Prentice Hall, pp. 1–906.
33. **Moreno, L.M., Kalita, J. (2017).** Deep Learning applied to NLP. *arXiv e-prints* DOI: 10.48550/arXiv.1703.03091.
34. **Deng, L., Yu, D. (2014).** *Deep Learning: Methods and Applications*. Foundations and Trends in Signal Processing, Publishers Inc., Vol. 7, No. 3–4, pp. 197–387. DOI: 10.1561/20000000039.
35. **Young, T., Hazarika, D., Poria, S., Cambria, E. (2018).** Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, Vol. 13, No. 3, pp. 55–75. DOI: 10.1109/MCI.2018.2840738.
36. **Chaudhuri, A. (2021).** Sentiment analysis on COVID19 data in India using advanced machine learning methods. *Samsung R & D Institute Delhi, India, Technical Report, TR-8989*.
37. **Gholizadeh, S., Zhou, N. (2021).** Model explainability in deep learning based natural language. *Processing. arXiv:2106.07410:2106.07410*. DOI: 10.48550/arXiv.2106.07410.
38. **Deng, L., Liu, Y. (2018).** Deep learning in natural language processing. *Springer*, pp. 1–327.

39. **Chaudhuri, A., Ghosh, S.K. (2022).** COVID19 forecasting in India through deep learning models. Recent Advances in AI-enabled Automated Medical Diagnosis. Taylor and Francis [in press].
40. **Li, Y., Yang, T. (2018).** Word embedding for understanding natural language: A survey. **Srinivasan, S. (ed)** Guide to Big Data Applications, Springer, Cham, Vol. 26, pp. 83–104. DOI: 10.1007/978-3-319-53817-4_4.
41. **Kutuzov, A., Øvrelid, L., Szymanski, T., Velldal, E. (2018).** Diachronic word embeddings and semantic shifts: A survey. Proceedings of COLING 2018, DOI: 10.48550/arXiv.1806.03537.
42. **Ruder, S., Vulić, I., Søgaard, A. (2017).** A survey of cross-lingual word embedding models. Journal of Artificial Intelligence Research, DOI: 10.48550/arXiv.1706.04902.
43. **Zhang, Y., Jin, R., Zhou, Z.H. (2010).** Understanding bag-of-words model: A statistical framework. International Journal of Machine Learning and Cybernetics, Vol. 1, No. 1, pp. 43–52. DOI: 10.1007/s13042-010-0001-0.
44. **Ramos, J. (2003).** Using TF-IDF to determine word relevance in document queries. 1st Instructional Conference on Machine Learning, Vol. 242, No. 1, pp. 29–48.
45. **Goodman, J.T. (2001).** A bit of progress in language modeling. Computer Speech & Language, Vol. 15, No. 4, pp. 403–434. DOI: 10.1006/csla.2001.0174.
46. **Guthrie, D., Allison, B., Liu, W., Guthrie, L., Wilks, Y. (2006).** A closer look at skip-gram modelling. 5th International Conference on Language Resources and Evaluation, pp. 1222–1225.
47. **Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J. (2013).** Distributed representations of words and phrases and their compositionality. Computation and Language, DOI: 10.48550/arXiv.1310.4546.
48. **Pennington, J., Socher, R., Manning, C.D. (2014).** GloVe: Global vectors for word representation. Empirical Methods in Natural Language Processing, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.
49. **Zhao, J., Zhou, Y., Li, Z., Wang, W., Chang, K.W. (2018).** Learning gender-neutral word embeddings. EMNLP'18, DOI: 10.48550/arXiv.1809.01496.
50. **Ghannay, S., Estève, Y., Camelin, N., Deléglise, P. (2016).** Evaluation of acoustic word embeddings. 1st Workshop on Evaluating Vector-Space Representations for NLP, pp. 62–66.
51. **Schwenk, H. (2007).** Continuous space language models. Computer Speech & Language, Vol. 21, No. 3, pp. 492–518. DOI: 10.1016/j.csl.2006.09.003.
52. **Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., Liu, H. (2018).** A comparison of word embeddings for the biomedical natural language processing. Journal of Biomedical Informatics, Vol. 87, pp. 12–20. DOI: 10.1016/j.jbi.2018.09.008.
53. **Nareesh, A., Venkata Krishna, P. (2021).** An efficient approach for sentiment analysis using machine learning algorithm. Evolutionary Intelligence, Vol. 14, No. 2, pp. 725–731. DOI: 10.1007/s12065-020-00429-1.
54. **Kawade, D.R., Oza, K.S. (2017).** Sentiment analysis: Machine learning approach. International Journal of Engineering and Technology, Vol. 9, No. 3, pp. 2183–2186. DOI: 10.21817/ijet/2017/v9i3/1709030151.
55. **Chaturvedi, S., Mishra, V., Mishra, N. (2017).** Sentiment analysis using machine learning for business intelligence. IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, pp. 2162–2166. DOI: 10.1109/ICPSCI.2017.8392100.
56. **Shathik, A., Karani, K.P. (2020).** A literature review on application of sentiment analysis using machine learning techniques. International Journal of Applied Engineering and Management Letters, Vol. 4, No. 2, pp. 41–77. DOI: 10.5281/zenodo.3977576.
57. **Singh, J., Singh, G., Singh, R. (2017).** Optimization of sentiment analysis using machine learning classifiers. Human-centric Computing and Information Sciences, Vol. 7, No. 32, pp. 1–12. DOI: 10.1186/s13673-017-0116-3.
58. **Raza, H., Faizan, M., Hamza, A., Mushtaq, A., Akhtar, N. (2019).** Scientific text sentiment analysis using machine learning techniques. International Journal of Advanced Computer Science and Applications, Vol. 10, No. 12, pp. 157–165.
59. **De Figueiredo, A., Simas, C., Karafillakis, E., Paterson, P., Larson H. (2020).** Mapping global trends in vaccine confidence and investigating barriers to vaccine uptake: a large-scale retrospective temporal modelling study. The Lancet, Vol. 396, No. 10255, pp. 898–908. DOI: 10.1016/S0140-6736(20)31558-0.
60. **Hussain, A., Tahir, A., Hussain, Z., Sheikh, Z., Gogate, M., Dashtipour, Ali, A., Sheikh, A. (2021).** Artificial intelligence-enabled analysis of public attitudes on facebook and twitter towards COVID-19 vaccines in the United Kingdom and the United States: Observational Study. Journal of Medical Internet Research, Vol. 23, No. 4, pp. 1–10. DOI: 10.2196/26627.

61. **Hung, M., Lauren, E., Hon, E.S., Birmingham, W.C., Xu, J., Su, S., Hon, S.D., Park, J., Dang, P., Lipsky, M. (2020).** Social network analysis of COVID-19 sentiments: Application of Artificial Intelligence. *Journal of Medical Internet Research*, Vol. 22, No. 8, pp. 1–13. DOI: 10.2196/22590.
62. **Sarlan, A., Nadam, C., Basri, S. (2014).** Twitter sentiment analysis. 6th International Conference on Information Technology and Multimedia, pp. 212–216. DOI: 10.1109/ICIMU.2014.7066632.
63. **Alamoodi, A.H., Zaidan, B.B., Zaidan, A.A., Albahri, O.S., Mohammed, K.I., Malik, R.Q., Almahdi, E.M., Chyad, M.A., Tareq, Z., Albahri, A.S., Hameed, H., Alaa, M. (2021).** Sentiment analysis and its applications in fighting COVID-19 and infectious diseases: A systematic review. *Expert Systems with Applications*, Vol. 167, pp. 1–13. DOI: 10.1016/j.eswa.2020.114155.
64. **Samuel, J., Ali, G.G., Rahman, M., Esawi, E., Samuel, Y. (2020).** COVID-19 public sentiment insights and machine learning for tweets classification. *Information*, Vol. 11, No. 6. DOI: 10.3390/info11060314.
65. **Hochreiter, S., Schmidhuber, J. (1997).** Long short-term memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
66. **Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2018).** BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* 1810.04805. DOI: 10.48550/arXiv.1810.04805.
67. **Omlin, C.W., Giles, C.L. (1996).** Constructing deterministic finite-state automata in recurrent neural networks. *Journal of ACM*, Vol. 43, No. 6, pp. 937–972. DOI: 10.1145/235809.235811.
68. **Omlin, C.W., Giles, C.L. (1992).** Training second-order recurrent neural networks using hints. 9th International Conference on Machine Learning, pp. 361–366. DOI: 10.1016/B978-1-55860-247-2.50051-6.
69. **Schmidhuber, J. (2015).** Deep learning in neural networks: An overview. *Neural Networks*, Vol. 61, pp. 85–117. DOI: 10.1016/j.neunet.2014.09.003.
70. **Schuster, M., Paliwal, K.K. (1997).** Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, pp. 2673–2681. DOI: 10.1109/78.650093.
71. **Altaher, A. (2017).** Hybrid approach for sentiment analysis of Arabic tweets based on deep learning model and features weighting. *International Journal of Advanced and Applied Sciences*, Vol. 4, No. 8, pp. 43–49. DOI: 10.21833/ijaas.2017.08.007.
72. **Gao, Y., Rong, W., Shen, Y., Xiong, Z. (2016).** Convolutional neural network-based sentiment analysis using adaboost combination. *International Joint Conference on Neural Networks*, pp. 1333–1338. DOI: 10.1109/IJCNN.2016.7727352.
73. **Cai, G., Xia, B. (2015).** Convolutional neural networks for multimedia sentiment analysis. *Natural Language Processing and Chinese Computing*, *Lecture Notes in Computer Science*, Vol. 9362, pp. 159–167. DOI: 10.1007/978-3-319-25207-0_14.
74. **Rani, S., Kumar, P. (2018).** Deep learning based sentiment analysis using convolution neural networks. *Arabian Journal for Science and Engineering*, Vol. 44, No. 4, pp. 3305–3314. DOI: 10.1007/s13369-018-3500-z.
75. **Kumar, A., Srinivasan, K., Cheng, W.H., Zomaya, A.Y. (2020).** Hybrid context enriched deep learning model for fine-grained sentiment analysis in textual and visual semiotic modality social data. *Information Processing and Management*, Vol. 57, No. 1, pp. 102–141. DOI: 10.1016/j.ipm.2019.102141.
76. **Kapociute-Dzikiene, J., Damaševičius, R., Woźniak, M. (2019).** Sentiment Analysis of Lithuanian texts using traditional and deep learning approaches. *Computers*, Vol. 8, No. 4, pp. 1–16. DOI: 10.3390/computers8010004.
77. **Baktha, K., Tripathy, B.K. (2017).** Investigation of recurrent neural networks in the field of sentiment analysis. *International Conference on Communication and Signal Processing*, pp. 2047–2050. DOI: 10.1109/ICCSP.2017.8286763.
78. **Shijia, E., Yang, L., Zhang, M., Xiang, Y. (2018).** Aspect based financial sentiment analysis with deep neural networks. *The Web Conference*, pp. 1951–1954. DOI: 10.1145/3184558.3191825.
79. **Piao, G., Breslin, J.G. (2018).** Financial aspect and sentiment predictions with deep neural networks: An ensemble approach. *The Web Conference*, pp. 1973–1977. DOI: 10.1145/3184558.3191829.
80. **Chen, L.C., Lee, C.M., Chen, M.Y. (2019).** Exploration of social media for sentiment analysis using deep learning. *Soft Computing*, Vol. 24, No. 11, pp. 8187–8197. DOI: 10.1007/s00500-019-04402-8.
81. **Ghulam, H., Zeng, F., Li, W., Xiao, Y. (2019).** Deep learning-based sentiment analysis for Roman Urdu text. *Procedia Computer Science*, Vol. 147, pp. 131–135. DOI: 10.1016/j.procs.2019.01.202.
82. **Tang, D. (2015).** Sentiment-specific representation learning for document-level sentiment analysis. 8th ACM International Conference on Web Search and Data Mining, pp. 447–452. DOI: 10.1145/2684822.2697035.

83. Chakraborty, K., Bhatia, S., Bhattacharyya, S., Platos, J., Bag, R., Hassanien, A.E. (2020). Sentiment analysis of COVID-19 tweets by deep learning classifiers — A study to show how popularity is affecting accuracy in social media. *Applied Soft Computing*, Vol. 97, pp. 1–14. DOI: 10.1016/j.asoc.2020.106754.
84. Al Sallab, A., Baly, R., Badaro, G., Hajj, H., El Hajj, W., Shaban, K.B. (2015). Deep learning models for sentiment analysis in Arabic. 2nd Workshop on Arabic Natural Language Processing, pp. 9–17.
85. Heaton, J. (2015). Deep learning and neural networks. *Artificial Intelligence for Humans*, Vol. 3. Heaton Research, Inc.
86. Chaudhuri, A. (2015). Semantic analysis of customer reviews with machine learning methods. Samsung R & D Institute, Delhi, India, Technical Report, TR-3699.
87. Turek, J., Jain, S., Vo, V., Capota, M., Huth, A., Willke, T. (2019). Approximating stacked and bidirectional recurrent architectures with the delayed recurrent neural network. *Proceedings of Machine Learning Research*, DOI: 10.48550/arXiv.1909.00021.
88. Graves, A., Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, Vol. 18, No. 5-6, pp. 602–610. DOI: 10.1016/j.neunet.2005.06.042.
89. Guo, T., Xu, Z., Yao, X., Chen, H., Aberer, K., Funaya, K. (2016). Robust online time series prediction with recurrent neural networks. *IEEE International Conference on Data Science and Advanced Analytics*, pp. 816–825. DOI: 10.1109/DSAA.2016.92.
90. Arik, S.O., Chrzanowski, M., Coates, A., Damos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Ng, A., Raiman, J., Sengupta, S., Shoenybi, M. (2017). Deep voice: Real-time neural text-to-speech. 34th International Conference on Machine Learning, Vol. 70, pp. 195–204.

*Article received on 17/02/2022; accepted on 27/03/2022.
Corresponding author is Arindam Chaudhuri.*