International Journal of Modern Agriculture
ISSN: 2305-7246
Volume 10 Issue 2, 2021

Speech Segmentation with Neural Networks to Facilitate Vernacular Translation

Namratha Karanth[1], Jaideep Francis Reddy[2], Deepak Choudhary[3], Stive Hassan[4], Manjunath. R. Kounte[5]
[1,2,3,4,5] School of Electronics and Communication Engineering, REVA University, India.
[1]email: the.namratha.karanth@gmail.com
[2]email: jaideepfrancis999@gmail.com
[3]email: deepakchoudhary1299@gmail.com
[4]email: stive.lf@gmail.com
[5]email: manjunath.kounte@gmail.com

ABSTRACT

The concept of using Artificial Intelligence (AI) in improving the lifestyle of society emphasizes on the fact that it should reach and benefit every community; In compliance to this, overcoming language barriers is still prevalent in many countries. Kannada is the spoken language of more than 50 million people in the world (2011). However, dialects of Kannada are diverse and vary according to geographic distributions making current machine translation research work difficult and less attended to. This research ventures to construct English – Kannada neural machine translation systems using Machine Learning techniques. The Corpus in our Model is trained using Kannada & Coastal Kannada dictionaries, Wikipedia & news articles. In this paper, we've implemented sequence-to-sequence learning using encoder-decoder LSTM (Long Short-Term Memory) model. We also review the significance of the usage of attention mechanism as it is an added strategy to improve the performance of the system. Developing a translation system to overcome linguistic barriers is the main goal of this research.

Keywords: Artificial Intelligence, Neural Machine Translation (NMT), Machine Learning, Natural language processing, English, Kannada, Coastal Kannada, Machine Translation.

1. INTRODUCTION

In a rich and diverse country like India, language barriers are common and difficult to overcome; the languages differ geographically leading to a strenuous way of communication. Automatic or machine translation is known as one of the most challenging Artificial Intelligence tasks given the fluidity of languages. Artificial Intelligence is a technique which enables computers to mimic human behavior, this gives us a variety of real-time human dependent tasks to be independently powered by machines. Enabling such systems on targeted applications can ease the lifestyle and move us forward to a developed society. Upliftment of rural areas by making it easier to overcome

language barriers is the aim of this paper. This research aims at building a more progressive Machine Translation System for Kannada. \

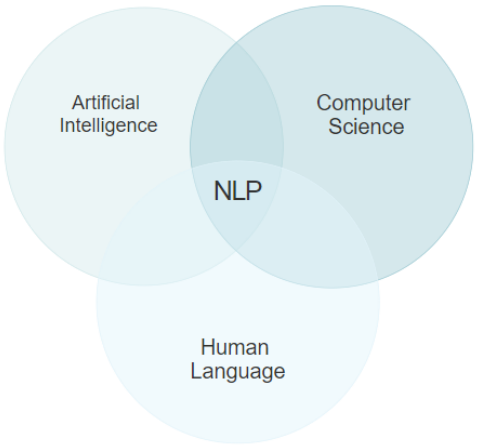

Fig. 1: NLP Venn Diagram

Speech segmentation plays an important role for an interpreter in a real-world scenario without a third person/middle-men manipulation, saving the user from many uncomfortable situations. Thus, development in effective translation becomes a demanding research area. Usage of machine translation [1] comes in very handy except when it comes to discriminable languages because of comparatively a smaller section of users of a particular dialect.

Neural Machine Translation (NMT) is a mode for machine translation using a neural network [1] that does computation to convert sequences of symbols from one language to sequence of symbols in another language. The key here is the data-driven approach, requiring the corpus of the source language to the target language. Sequence-to-sequence learning (Seq2Seq) is about training models to translate sequences from one group (e.g., sentences in English) to sequences in another group using RNNs (Recurrent neural network). RNNs in its most basic form consists of an input layer, a hidden layer and an output layer [2]. The input layer acquires the input, the hidden layer activations are applied and then we finally obtain the output. So, a recurrent neuron stores the state of a previous input and combines with the current input thereby preserving some relationship of the current input with the previous input [2][11].

A RNN layer (or stack thereof) behaves as an "encoder": It processes the input sequence and retrieves its own internal state. We scrap the outputs of the encoder RNN, only reclaiming the state. This state will render as the "context", or "conditioning", of the decoder in the succeeding step. Another RNN layer (or stack thereof) acts as "decoder": it is trained to speculate the adjacent characters of the target sequence, given prior characters of the target sequence.

However, traditional models are limited by a fixed-length input sequence where output must be the same length. Such must be combined with an effective model like sequence-to-sequence learning using encoder decoder architecture with an attention mechanism that allows variable

length input and output sequences.

2. LITERATURE SURVEY

In present age, the Dravidian languages form a well-knit group by themselves and unlike the Aryan, the Austric and the Sino- Tibetan dialects they have no connection outside the Indian mainland. The Dravidian languages are categorized into two major groups: (i) The North Dravidian Languages: Telugu and a number of other languages such as various Gondi dialects, Kuruth, Maler and a few others are included in this group. (ii) South Dravidian Languages: This group of languages includes Kannada, Tamil and Malayalam, etc.

Kannada is the State Language of Karnataka, is amongst the most ancient Dravidian languages and is spoken in its various dialects by more than 45 million people around the world. The language has existed for about 2500 years. Kannada is a distinctly inflected language with three genders (masculine, feminine, neutral or common) and two numbers (singular, plural). It is articulated for gender, number and tense in the number of other things. Kannada has held the Classical Language status since November 1, 2008. The rich and profuse heritage of the language drives for further exposure of the less explored parts of Karnataka and bridges the gap caused by semantic and dialect barriers.

There is an acute distinction between the spoken and written forms of the Kannada. Spoken Kannada tends to vary in accordance with the region. The ethnologue identifies around twenty different dialects of Kannada. Here we are focusing on the most common spoken Karnataka dialect for our machine translation model.

This research proposes to build a translation model in order to uplift the rural Karnataka language translation that can benefit the people and tourists as human interpreters are always not available. Here, we deal with inadequate resources for building our translation model, we are aiming to transliterate a huge amount of Kannada dialect dictionaries and Wikipedia articles to a raw corpus. The most used Kannada dialect and its English phrases are composed in addition to the Kannada to English corpus. This is a first attempt to analyze and extract data from the region-specific dictionaries and web content.

3. DATA PREPARATION

The Coastal Kannada here has the basic foundation of generic Kannada and is also aided by the coastal Kannada words/sentences. The reason being, the generic Kannada is spoken in all parts of Karnataka and every town has its own linguistic tone and words mixed to the generic base. Same applies to coastal Kannada here.

Hence, the corpus was completely constructed from the base using web scraping and data feeding methods. Although, the ratio of generic Kannada to coastal Kannada ratio is comparatively very less in our built corpus, it’s the first corpus in the research area. This issue is because of lack of

online data, since there weren't enough resources to continue building the corpus. The corpus can be further expanded in future works.

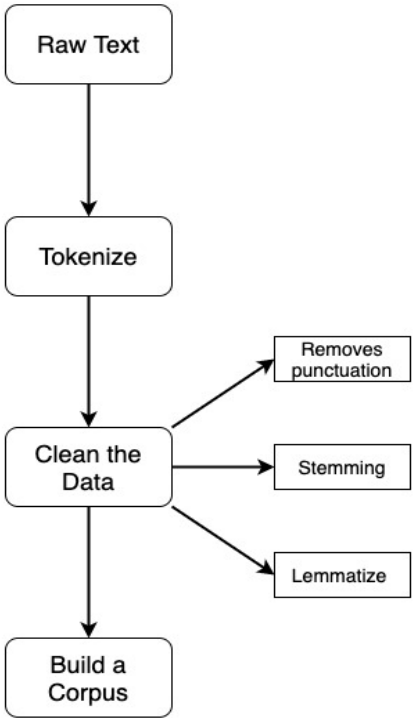

Fig. 2: Data Preparation Flow Diagram

3.1. RAW TEXT

We decided to opt for creating our own corpus that is comparatively smaller and reduces the time complexity, using the Web Scraping techniques, which included Wikipedia/news articles and dictionaries [10]. We generated random sentences and their parallel meanings in Kannada to first create a raw text. Now our goal is to form a sequence of characters that forms a search pattern, that will organize the raw text into an understandable format. This is where tokenization comes into the picture.

3.2. TOKENIZE

Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as words or terms. Each of the smaller units are called tokens. It essentially tells the model what to look at, by breaking the input sequence into chunks of tokens. The first thing to do is to create values for our start of the sentence, end of the sentence, and sentence padding special tokens.

When we tokenize text, we need special tokens to delineate both the beginning and end of a sentence, as well as to pad the sentence (or some other text chunk) storage structures when

sentences are shorter than the maximum allowable space. Here, we have used "words2idx" to generate IDs for the tokens.

3.3. CLEANING THE DATA

After converting the data into tokens, we now must clean the text by removing the special characters, punctuations, lemmatize and organize it to a format which is easier for the model to understand. This is done by using RegEx, that finds all the words that match the search pattern passed on it and stores it in a list (i.e., the corpus). The goal of both stemming and lemmatization is to reduce inflectional forms and the derivationally related forms of a word to a common base form. For example, am, are, is ⇒ be.

3.4. BUILD A CORPUS

All the above methods and steps together constitute the formation of our corpus. Since our list is still a meaningless set of words to our model, we need to give context to the data present in the corpus, as the translation is not word-to-word but context-wise translation. It is necessary to perform pragmatic analysis later, this process is called word-embedding/ word vectorization.

Word vectorization is done by fitting the actual objects or events that exist in each context with object references obtained by the previous component. In simple words, by measuring the occurrence of the repetitive words, realizing similar words, grouping the words to map words or phrases from vocabulary on a corresponding vector of real numbers, it is used to find word predictions, word similarities/semantics [3][4][5][8].

4. THE PROPOSED METHODOLOGY

4.1. COMPONENTS OF MACHINE TRANSLATION

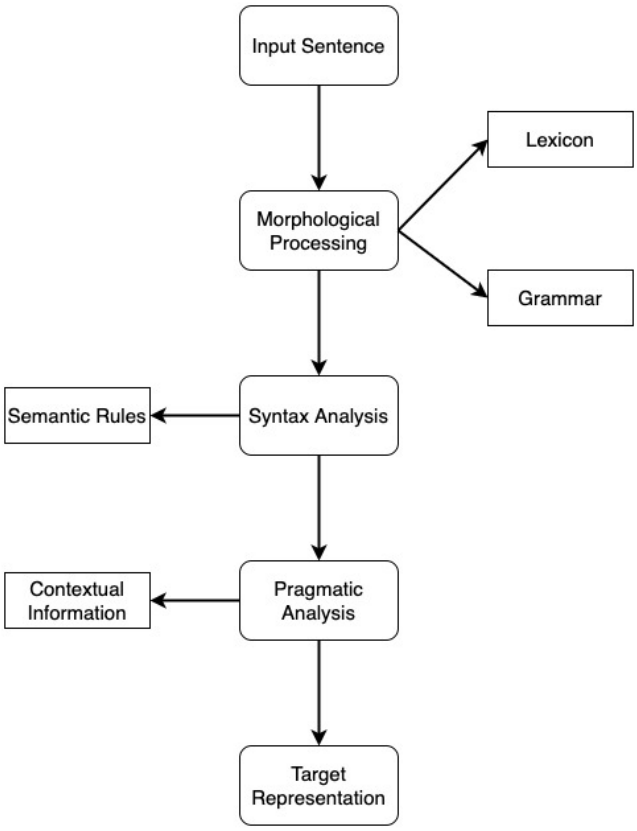

Fig. 3: Components of Neural Machine Translation

Input sequence is to be translated from the word/phrase sent in the source language which is then taken for morphological analysis. Morphology is Identification, analysis and description of the structure of a given language's morphemes and other linguistic units, such as root words, affixes, parts of speech, intonations and stresses or context.

The individual words are analyzed and semantic analysis is performed where the syntactic analyzer assigns meanings. Semantics focuses on the literal meaning of words, phrases and the sentences. This only abstracts the dictionary meaning from the given context.

Overall interpretation of the words/phrases has to be analyzed and this is done using pragmatic analysis. Here, the main focus is always on what was said, and that is reinterpreted on what it actually meant. The input sequence is now analyzed and is represented in the target language [6][7].

4.2. MODEL ARCHITECTURE

In this paper we will be realizing a word-level sequence-to-sequence model, processing the input word-by-word and yielding the output word-by-word on a smaller dataset. The Encoder-Decoder LSTM can be implemented directly using the Keras deep learning library. The Sequence-to-Sequence module (or) Encoder Decoder is implemented using two recurrent neural network that acts as an Encoder-Decoder pair that converts sequences from one domain to sequences in another

domain (i.e., sentences in English to Kannada) which will take the textual data and translate it to the target language that is Kannada and store that data, it is appropriate in our case where the length of the input sequence does not have the same length as the output data. Although a RNN can learn dependencies however, it can only learn about recent information.

LSTM can help solve this problem as it can understand the context along with recent dependency. Hence, LSTMs are a special kind of RNN where understanding context can help to be useful [9]. LSTM (Long- Short Memory Cells) networks are like RNNs with one major difference that the hidden layer updates are replaced by the memory cells. This makes them better at finding and exposing long range dependencies in data which is imperative for sentence structures.

4.3. IMPLEMENTATION

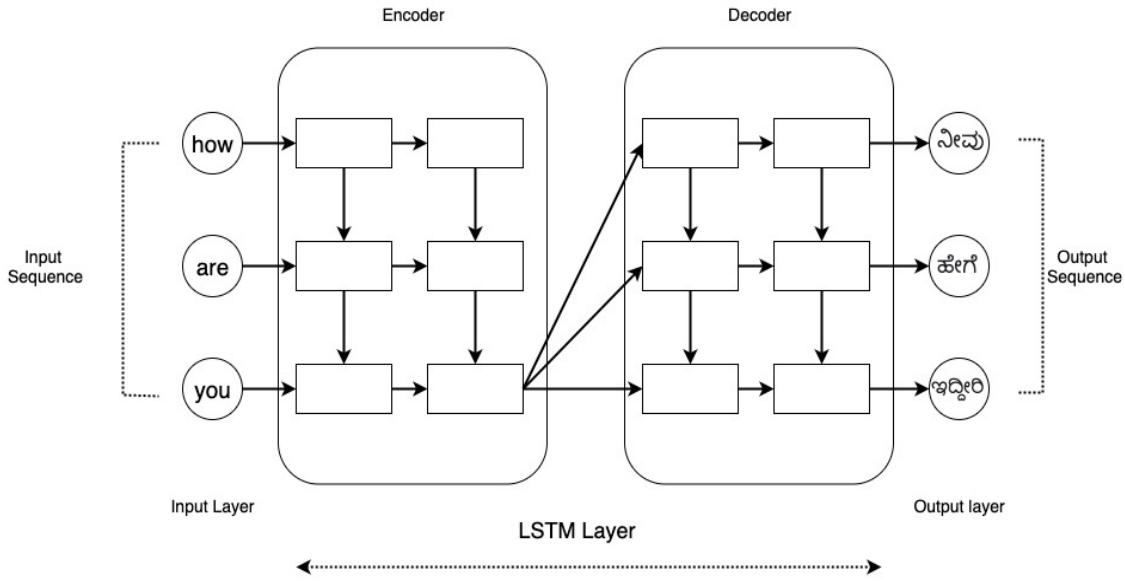

Fig. 4: Encoder – Decoder Layer

4.3.1. ENCODER

Encoding means converting data in a required format. We convert a sequence of words in English to a two-dimensional vector, this two-dimensional vector is also known as hidden state. The encoder maps the variable-size input sequence to a fixed-size vector which intends on condensing the information for all input elements in an effort to aid the decoder produce reliable predictions.

The input sequence is a cluster of every single word from the question. Each word is characterized as x_i where i is the order of that word. The hidden states h_i are determined using the formula:

(1): $h_t = f\left(W^{(hh)}.h_{t-1} + W^{(hx)}.x_t\right)$

One or more layers of LSTM can be implemented in the encoder model and the decoder model. The number of memory cells in this layer defines the length of the fixed-sized vector.

4.3.2.DECODER

The decoder maps the vector representation back to the variable-sized target sequence. In layman's terms, we will convert the two-dimensional vector into the output sequence, the Kannada sentence. It is a stack of various periodic units where each predicts an output y_t at a time step t.

Each recurrent unit accepts a hidden state from the previous unit and produces an output as well as its own hidden state. Here, each word is characterized as y_i where i is the order of that word.

(2): $h_t = f\left(W^{(hh)}.h_{t-1}\right)$

The output y_t at time step t is determined with the formula:

(3): $y_t = softmax(W^s.h_t)$

We evaluate the outputs by making use of the hidden state at the current time step along with the corresponding weight W(S). SoftMax is used to create a probability vector which will aid us in determining the final output.

4.3.3.ATTENTION NETWORK

We evaluate the outputs by making use of the hidden state at the current time step along with the corresponding weight W(S). SoftMax is used to create a probability vector which will aid us in determining the final output.

The above method works effectively for meagre sequences. However, as the size of the sequence expands, a single vector becomes a bottleneck and it gets problematic to encapsulate long sequences into a single vector. The attention mechanism is used here to retain those intermediate encoder states and utilize all the states in order to compose the context vectors needed by the decoder to yield the output sequence.

Attention mechanism is simply giving the network access to its internal memory, which is the hidden state of the encoder. In this interpretation, instead of choosing what to "attend" to, the network chooses what to retrieve from memory. Unlike typical memory, the memory access mechanism here is soft, which means that the network retrieves a weighted combination of all memory locations, not a value from a single discrete location. Making the memory access soft has the benefit that we can easily train the network.

The above strategies that are implemented in the model made the loss function very less yielding to an overall higher translation accuracy of the NMT system as seen in Fig. 5.

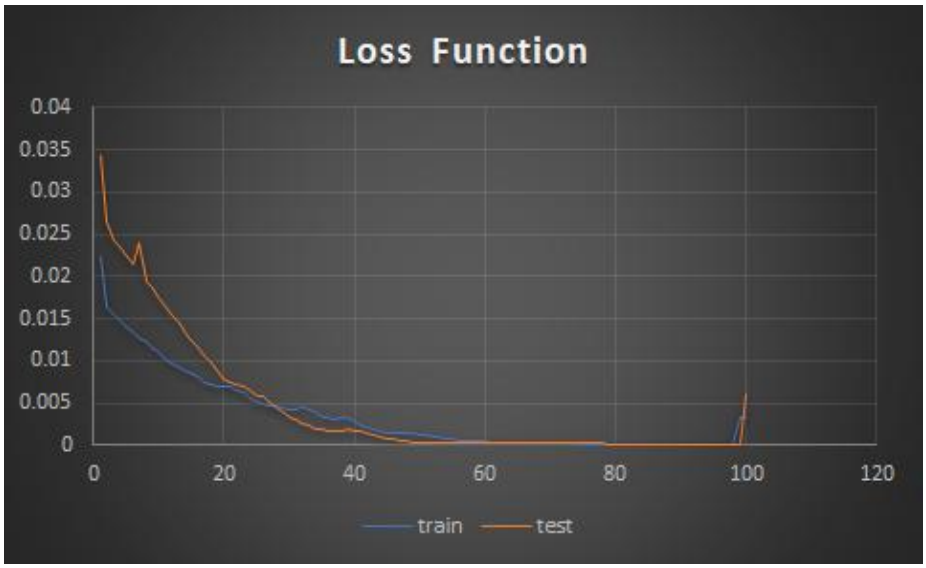

Fig. 5. Loss Analysis Function

5. CONCLUSION

Table. 1. Obtained Results from the Neural Translator

English	**Kannada**
Don't talk	ಮಾತನಾಡಬೇಡ . ಮಾತಾಡ ’ಬಾರದು .
For ladies only	ಹಂಗಸರಗಮಾತರ .
What are you doing?	ನೀನುಏನುಮಾಡುತ್ತಿರುವೆ . ಎಂಥಮಾಡ್ತಿದ್ಯ .
What is that?	ಅದುಎಂಥದು . ಏನದು

All the samples enlisted above are taken from the output produced by the machine translation system developed by us. The performance of the system was evaluated with a set of 3000 sentences which are carefully chosen in the corpus from various articles and dictionaries. The sentences are translated using the NMT (neural machine translation) with LSTMs and the attention mechanism added much improvement in the performance. The goal of this research is to develop an instant text communication tool that can translate English to Kannada. The corpus was updated continuously as we gathered more data from many resources which increased the quality of the output with many trials. This yielded us 87.5% accuracy with the validation data. This

accuracy is also obtained due the fact that only simple conversational data was added/used in the corpus. Addition of Complex sentences to the corpus could be the future scope of this paper. However, this will also mean a decrease in the accuracy.

The motivation behind this project is to reduce the unavailability of translation systems in rural areas that could promote development in economic aspects of the rich cultural diversity present in such areas. As we could not find previous research in the domain related to anything other than the generic Kannada language, this research aims to make a contribution towards building an efficient corpus.

The corpus in this paper took extensive research (i.e., taking inputs from native people via forms and dictionaries) which still results in a smaller ratio of coastal Kannada words/sentences compared to the Kannada words; hence more contributions to our current corpus could result in an augmentation for a much-improved translation.

Implementing a speech recognition model to take speech input, process the audio and convert the audio to recognizable sentences that requires voice embeddings for our target language in real time is the future scope of this paper.

REFERENCES

[1] R. Vyas, K. Joshi, H. Sutar and T. P. Nagarhalli, "Real Time Machine Translation System for English to Indian language," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 838-842, doi: 10.1109/ICACCS48705.2020.9074265.

[2] Middi, Venkata Sai Rishita & Raju, Middi & Harris, Tanvir Ahmed. (2019). Machine translation using natural language processing. MATEC Web of Conferences. 277. 02004. 10.1051/matecconf/201927702004.

[3] Rajpirathap S, Sheeyam S, Umasuthan K and A. Chelvarajah, "Real-time direct translation system for Sinhala and Tamil languages," 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), 2015, pp. 1437-1443, doi: 10.15439/2015F113.

[4] R. Sunil, V. Jayan and V. K. Bhadran, "Preprocessors in NLP applications: In the context of English to Malayalam Machine Translation," 2012 Annual IEEE India Conference (INDICON), 2012, pp. 221-226, doi: 10.1109/INDCON.2012.6420619.

[5] P. Kumar, S. Srivastava and M. Joshi, "Syntax directed translator for English to Hindi language," 2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), 2015, pp. 455-459, doi: 10.1109/ICRCICN.2015.7434282.

[6] S. Nahar, M. N. Huda, M. Nur-E-Arefin and M. M. Rahman, "Evaluation of machine translation approaches to translate English to Bengali," 2017 20th International Conference of Computer and Information Technology (ICCIT), 2017, pp. 1-5, doi: 10.1109/ICCITECHN.2017.8281851.

[7] S. Saini and V. Sahula, "A Survey of Machine Translation Techniques and Systems for Indian Languages," 2015 IEEE International Conference on Computational Intelligence & Communication Technology, 2015, pp. 676-681, doi: 10.1109/CICT.2015.123.
[8] M. M. Kodabagi and S. A. Angadi, "A methodology for machine translation of simple sentences from Kannada to English language," 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), 2016, pp. 237-241, doi: 10.1109/IC3I.2016.7917967.
[9] Chandramma, P. Kumar Pareek, K. Swathi and P. Shetteppanavar, "An efficient machine translation model for Dravidian language," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 2101-2105, doi: 10.1109/RTEICT.2017.8256970.
[10] M. A. Hasan, F. Alam, S. A. Chowdhury and N. Khan, "Neural vs Statistical Machine Translation: Revisiting the Bangla-English Language Pair," 2019 International Conference on Bangla Speech and Language Processing (ICBSLP), 2019, pp. 1-5, doi: 10.1109/ICBSLP47725.2019.201502.
[11] T. Do, M. Utiyama and E. Sumita, "Machine translation from Japanese and French to Vietnamese, the difference among language families," 2015 International Conference on Asian Language Processing (IALP), 2015, pp. 17-20, doi: 10.1109/IALP.2015.7451521.
[12] M. R. Kounte, P. K. Tripathy, P. P. and H. Bajpai, "Analysis of Intelligent Machines using Deep learning and Natural Language Processing," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), 2020, pp. 956-960, doi: 10.1109/ICOEI48184.2020.9142886.
[13] ONLINE PASTRY SHOP WITH USER SERVICE RATING, V.SOWMITHA, M.MANORANJITH, P.KISHOR, S.VIGNESH, N.ARUNKUMAR, International Journal Of Advance Research In Science And Engineering http://www.ijarse.com IJARSE, Volume No. 10, Issue No. 04, April 2021 ISSN-2319-8354(E).
[14] Kounte, Manjunath R & Tripathy, Kumar & Bajpai, Harshit. (2020). Implementation of Brain-Machine Interface using Mind Wave Sensor. Procedia Computer Science. 171. 244-252. 10.1016/j.procs.2020.04.026.
[15] Kamble, Shridevi & Kounte, Manjunath R. (2020). Machine Learning Approach on Traffic Congestion Monitoring System in Internet of Vehicles. Procedia Computer Science. 171. 2235-2241. 10.1016/j.procs.2020.04.241.
[16] Naveen, Soumyalatha & Kounte, Manjunath R. (2022). Machine Learning at Resource Constraint Edge Device Using Bonsai Algorithm. 10.1109/ICAECC50550.2020.9339514.