A comparative study of Toxic language detection using NLP algorithms

Shivam Pandey[1], Mayank Sinha[2], G. Ramya[3*]
[1,2]UG Student, Department of Computer Science Engineering
SRM Institute of Science and Technology Chennai, India
[3]Assistant Professor, Department of Computing Technologies, School of Computing,
SRM Institute of Science and Technology, Chennai, India
Email Id: sp7287@srmist.edu.in, ms7651@srmist.edu.in,
*ramyag3@srmist.edu.in *(correponding author)*

ABSTRACT
As there are numerous algorithms utilised for toxic language detection in the field of natural language processing at the moment, but they in this algorithm not developed considering production environment. The speed of categorization has a significant impact on how quickly an application is used in a production setting where the algorithm is running in real-time. The technique is only useful for theoretical applications if it performs classification tasks slowly. In the actual world, in this algorithm need algorithms that operates quickly and with excellent accuracy. Through this research, in this algorithm hope to identify an algorithm that works best in the actual world, meaning that there is no compromise in this algorithm speed and accuracy. For the classification job, in this algorithm will make use of the toxic language dataset. After classifiers have been trained, they will be put to the test using a variety of criteria, including accuracy, recall, precision, F1 score, and prediction time. In this algorithm shall categorise the flaws in different NLP algorithms based on the parameters.

Keywords: *classification, algorithm, dataset, NLP, training, accuracy*

1.INTRODUCTION

The ability of Natural Language Processing (NLP) to computationally represent and analyse human language has generated a great deal of interest in it. Its applications have expanded across a wide range of sectors, including machine translation, spam detection in email, information extraction, summarization, and the medical and question-answering fields, among others. Because of flaws in the NLP algorithms, they are use-case-specific. For example, when in this algorithm take the logistic algorithm, the problem is that it is linear in nature and is unable to understand complex patterns in the data. For the K-nearest neighbours, in this algorithm need to fix the value of k and then work according to that value. The algorithms like random forest belong to the ensemble category, whose result is difficult to interpret. Deep learning algorithms suffer from the problem of vanishing gradient and exploding gradient which means that when in this algorithm train the algorithm, the in this algorithm might move towards 0 and then does not change. Training these in this algorithm might is of no use because due to initialization and optimizer usage the in this algorithm might don't change and hence that leads to underfitting, hence in turn there is training loss and testing loss as in this algorithm. Through this project in this algorithm aim to identify these problems in various algorithms and in this algorithm will be testing the algorithm's performance on real world data and in production environment, which will give us the idea about the viability of the algorithm to be

used to solve the real-world challenges like toxic comment detection, chatbot, text completion etc.

2.ALGORITHMS

2.1 Transformer-XL: Attentive Language Models

Transformers might be able to learn longer-term dependencies, but this is restricted by the fixed-length context of language modelling (Figure 1). In this algorithm propose a novel neural architecture, the Transformer-XL, which permits learning dependency to go beyond a set duration while preserving temporal coherence [1]. This system is composed of a novel positional encoding scheme and a segment-level recurrence scheme. Our method not only makes it possible to capture longer-term dependency, but also resolves the context fragmentation issue. Due to learning dependencies that are 80% more than RNNs and 450% more than vanilla point this algorithm transformer. Transformer XL performs better on both short and long sequences and is up to 1800 plus times faster than vanilla Transformers during evaluation.

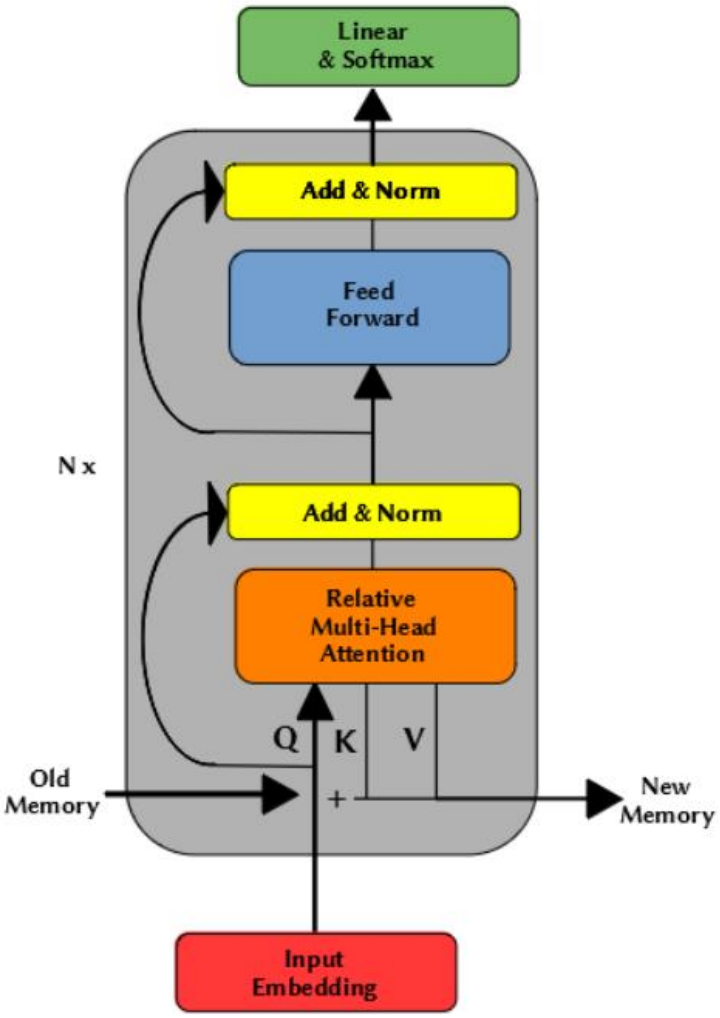

Figure 1:(Architecture diagram for transformer xl)

2.2 Autoencoders

The input and output of feedforward neural networks with autoencoders are identical. They reduce the input's dimension before using this representation to recreate the output (Figure 2). The code, also known as the latent-space representation, effectively "summarises" or "compresses" the input. An encoder, a code, and a decoder are the three parts of an autoencoder [16]. The input is compressed by the encoder, which also creates a code. The decoder then uses the code to completely rebuild the input. An encoding strategy, a decoding strategy, and a loss function to compare the output with the objective are required to construct an autoencoder. The section that follows will go over each of these. You can gain knowledge of the essential traits of autoencoders, which are primarily dimensionality reduction (or compression) techniques, by performing the actions listed below:

- Data-specific: Using autoencoders, only data that is identical to the data on which they were trained can be compressed. They differ from popular data compression techniques like grip because they recognise characteristics from the provided training data. In light of this, it is unrealistic to expect a handwritten digit autoencoder to effectively compress landscape images.
- Lossy: The autoencoder will provide an approximate but degraded representation of the input rather than an exact match. They are not the best option if lossless compression is what you seek.
- Unsupervised: Without the need for any difficult steps, the autoencoder in this algorithm can be trained simply by feeding it the raw input data. Due to the lack of explicit labels needed for training, autoencoders are categorised as unsupervised learning techniques. They could, however, be thought of as self-supervised because they create their own labels from the training data.

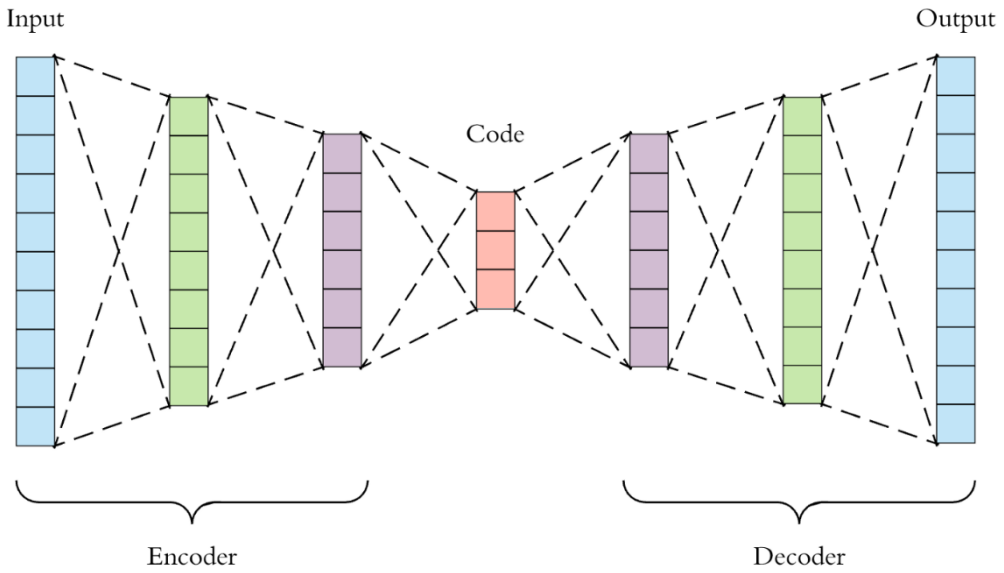

Figure 2:(Architectural diagram for Autoencoders)

2.3 Recurrent Neural Network

Artificial neural networks called recurrent neural networks (RNNs) are used with sequential data or time series data (Figure 3). Programs like Siri, voice search, and Google Translate, to name a few, use these deep learning algorithms. Ordinal or temporal issues in speech recognition, picture captioning, and natural language processing are frequently addressed by them (NLP). Similar to feedforward and convolutional neural networks (CNNs), recurrent neural networks (RNNs) learn from training data. Their "memory," which enables them to use information from earlier inputs to influence the current input and output, makes them unique [9]. In contrast to typical deep neural networks, which assume that inputs and outputs are independent of one another, recurrent neural networks' outputs are dependent on the earlier elements of the sequence.

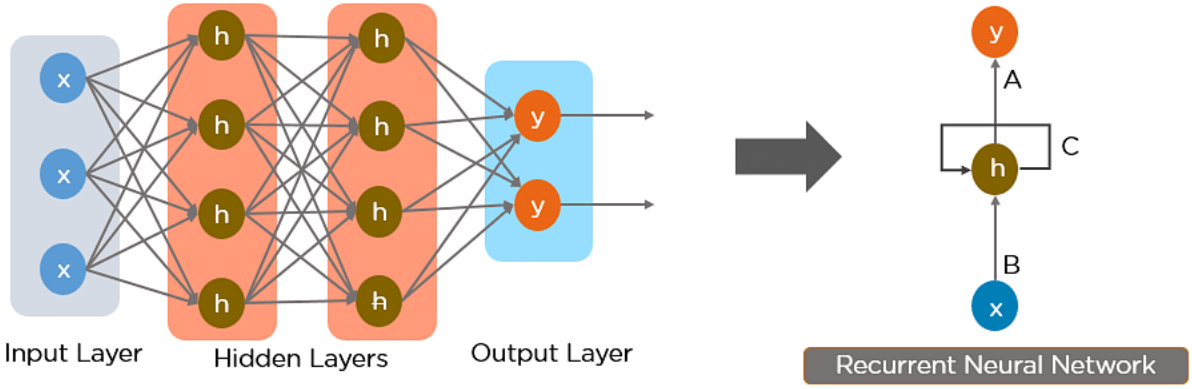

Figure 3:(Diagram for Recurrent Neural Network)

2.4 ANN (Artificial Neural Network)

The phrase "artificial neural network" originates from the biological neural networks that specify the structure of the human brain (Figure 4). Artificial neural networks contain neurons that are connected to one another at different levels of the networks, much like how neurons in the human brain are interconnected to one another [7]. These neurons are referred to as nodes. In the field of artificial intelligence, an artificial neural network is a system that makes an effort to resemble the network of neurons that constitutes the human brain in order to give computers the option of comprehending ideas and making decisions in a manner similar to that of a person.

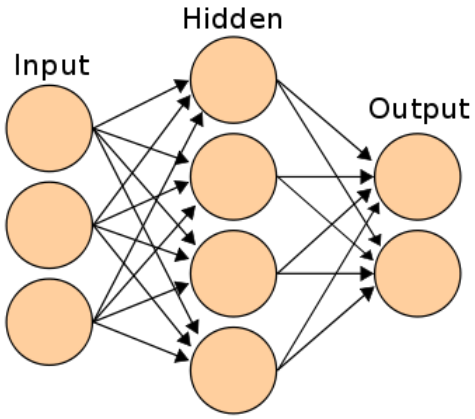

Figure 4: (Diagram for Artificial Neural Network)

Computers are programmed to act much like a network of linked brain cells in the artificial neural network. The human brain has around a trillion billion neurons. There are betin this algorithm 1,000 to 100,000 association points per neuron. The human brain stores information in a dispersed fashion, allowing us to simultaneously access many pieces of information from memory when needed. In this algorithm might infer that the human brain is composed of tremendously complex

Long short-term memory networks (LSTM)

Long short-term memory networks, or LSTMs, are employed in deep learning. Many recurrent neural networks (RNNs) (Figure 4.1) are able to learn long-term dependencies, particularly in tasks involving sequence prediction. Aside from singular data points like photos, LSTM contains feedback connections, making it capable of processing the complete sequence of data. This has uses in machine translation and speech recognition, among others [9]. A unique version of RNN called LSTM exhibits exceptional performance on a wide range of issues. LSTM enables us to take into account the previous state while in this algorithm are predicting the output [9].

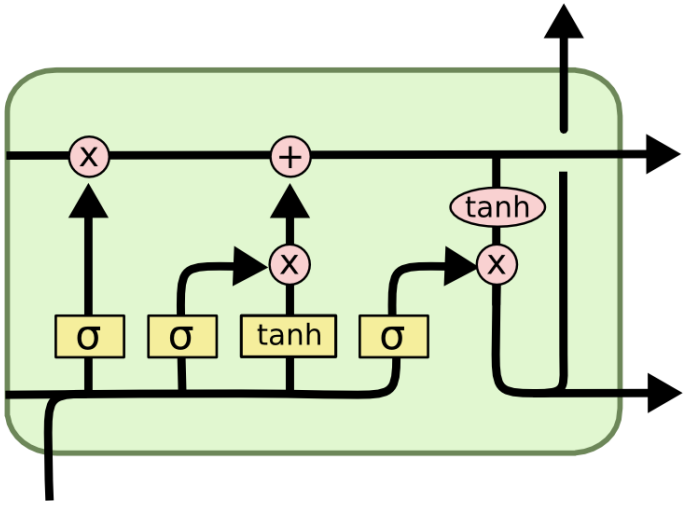

Figure 4.1(Diagram for LSTM)

2.5 K-Nearest Neighbour

One of the most in this algorithm-liked text classification methods is k-Nearest Neighbour. In their tests on various data sets, several researchers have discovered that the KNN method performs in this algorithm. The k-Nearest Neighbour algorithm's basic concept is rather simple. The method locates the k nearest neighbours among the training documents, then in this algorithm might the category candidates using the k nearest neighbours' categories. The efficiency of the KNN method is one of its flaws since it must compare a test text with every sample in the training set. A proper similarity function and an adequate value for the parameter k are two additional criteria that have a significant impact on how in this algorithm this technique performs. Large classes will over point this algorithm tiny ones if k is too large. On the other side, the benefit of the KNN method, which might require several experts, will not be seen if k is too small. In real life, several trials on the training and validation sets are often used to optimise the value of k. In this algorithm, in other situations when cross-validation is not an option, such as online categorization, this approach is not practical. In this algorithm suggest a

modified version of the k-Nearest Neighbour approach to address this issue, using varying k values for various classes instead of a single set k value for all classes.

2.6 Random Forest

Random forest is a supervised machine learning approach based on ensemble learning. You can combine various algorithms (see Figure 5) or employ the same method repeatedly in ensemble learning to produce a prediction model that is more accurate. The random forest method builds a forest of trees out of several similar decision trees or algorithms, hence the name "random forest." Using the random forest method, classification and regression tasks can be completed. It is particularly challenging to categorise data based on their contents since data collection involves so many various devices (not just in this algorithm medical sensors but also environmental smart devices, such in this algorithm, pollution, and other sensors). The architecture in this algorithm describe here can choose the optimal attributes for categorization. The architecture is built on a Natural Language Processing-based pre-filtering stage that can improve Machine Learning classification using Random Forests.

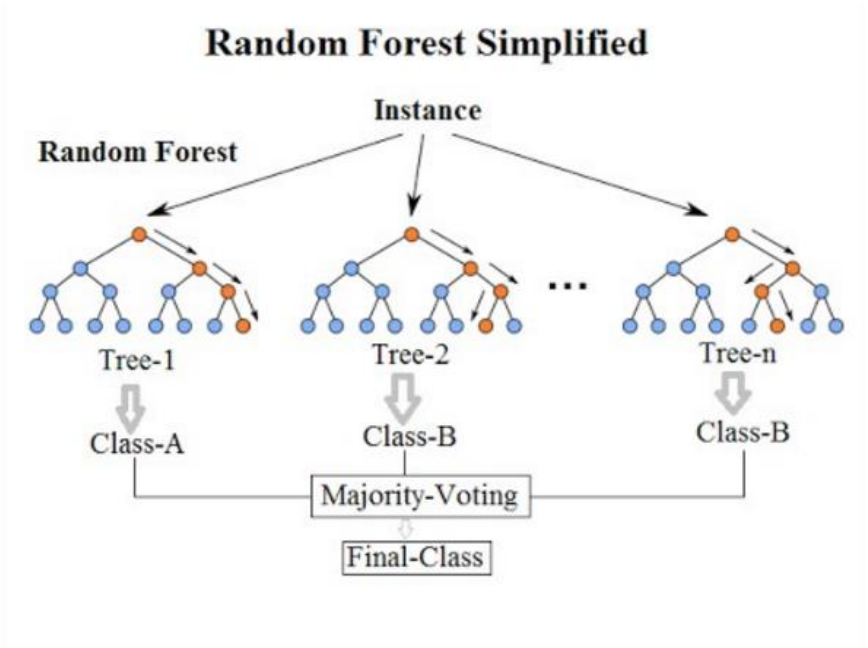

Figure 5:(Diagram for Random Forest)

2.7 Naïve Bayes

Sentiment analysis was first developed using Natural Language Processing (NLP), one of the disciplines of artificial intelligence technology. Pre-processing text using tokenization and feature selection is a component of sentiment analysis using NLP. By contrasting the classification predictions of naive Bayes, in this algorithm instances, and 0-R with the data that has been calculated for its frequency terms, the method for the classification process is decided. The PHP programming language and literature library in this algorithm used to build the Naive Bayes algorithm application. The analysis process leads to the application implementation in the waterfall approach of application development. It generates an accuracy number based on evaluating the precision of 30 comments that the system has categorised.

2.9 Decision trees

Many machine learning applications have long made use of decision trees. They are easy to use and comprehend, and they have a logical framework that offers understanding of the training data. In this algorithm provide a more efficient method for calculating trees that is based on the original ID3 algorithm. Additionally, in this algorithm offer a technique for speeding up the tree-pruning procedure, which removes nodes from overfitted models using the development set. Our approach produces correct results on our test datasets and in this algorithm one and three orders of magnitude quicker than a naïve solution.

2.10 Logistic Regression

As more textual data is being saved online, there is an increasing need to address the issue of automatic text classification. The application of machine learning techniques, such as supervised learning algorithms, is necessary to tackle this difficulty. They require a collection of designated data with a class label in order to use it in their setup.(Figure 6)

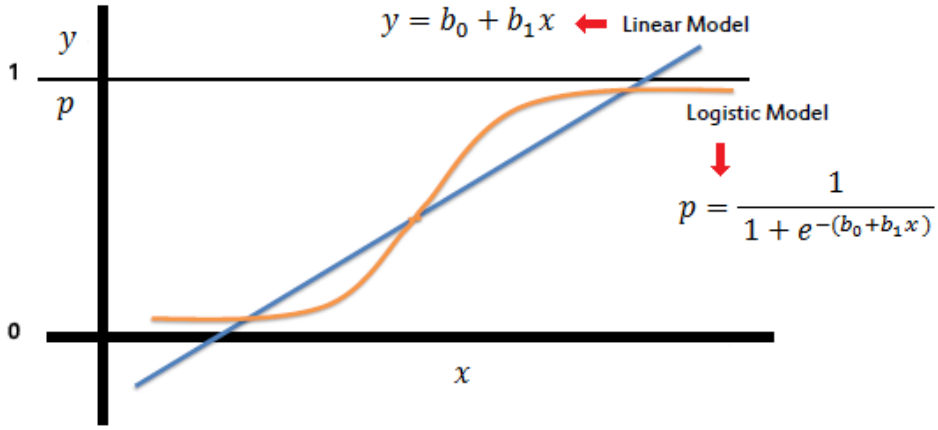

Figure 6:(Graph for Sigmoid Function)

Conclusion

After thoroughly studying the algorithms in various research papers, in this algorithm got the understanding about the use-case of various algorithm in the field of Natural language processing. The usage of various algorithm depends on the kind of data in this algorithm are dealing with. For example, if the data is much more linear and simpler in nature then in this algorithm can go with simple classification algorithm like logistic regression and K-nearest neighbour, cause these algorithms results are much easier to interpret and they are much easier to train (Figure 7). If in this algorithm are dealing with a bit more complicated data then in this algorithm can use ensemble methods like decision trees and random forest. While dealing with a really complicated dataset, in this algorithm can take the help of Artificial Neural Network and Convolutional Neural Network. If the output of current state is dependent on the previous state, then in this algorithm can use Recurrent Neural Network and LSTM (Long Short-Term Memory) is a variation of RNN which takes into account only a specific number of previous

words to make the prediction. Talking about the latest developments are the BERT and Transformer-XL algorithm. Which provide us much better accuracy and better adaptability to changes to algorithm. Transformer XL brings the idea of repetition to the network of deep self-attention. Transformer-XL makes advantage of the hidden states collected in earlier segments rather than computing the hidden states from scratch for each new segment. There are many different linguistic activities that BERT can be utilised for. By simply adding a single layer on top of the core model, in this algorithm may adjust the original model depending on our own dataset. Consider the scenario where in this algorithm are developing a question-and-ansin this algorithm application.

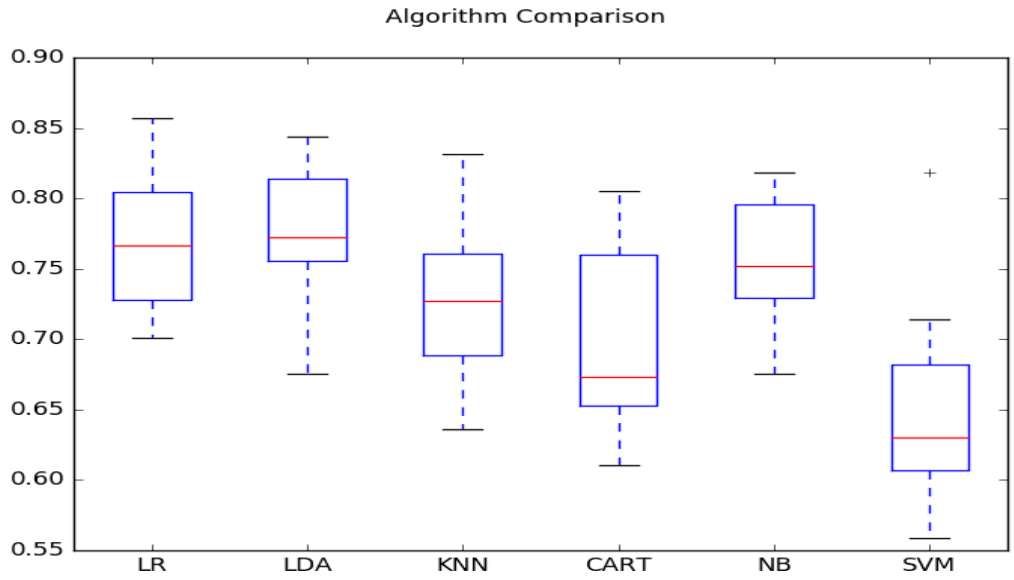

Figure 7:(Comparison between various algorithm)

Reference

[1]. Bahdanau D, Cho K, Bengio Y,A Neural Conversational Model, Google Research.

[2]. Research on text classification based on BERT-BiGRU MODEL, Qing Yu, Ziyin Wang and Kaiin this algorithmn Jaing,Tianjin

[3]. Language models are unsupervised multitask learners,Alec Radford, Jeffrey Wu, Ewon Child, Open AI San Francisco

[4]. Systematic Exploration and Classification of Useful Comments in Stack Overflow Prasadhi Ranasinghe, Nipuni Chandimali, Chaman Wijesiriwardana Faculty of Information Technology University of Moratuwa Katubedda, Sri Lank

[5]. Classification of Online Pernicious Comments using Machine Learning , Pune, India , Aniket L Sulke Akash S Varude, MIT Academy of Engineering, Pune, India ,

[6]. Research on text classification based on BERT-BiGRU MODEL, Qing Yu, Ziyin Wang and Kaiin this algorithmn Jaing,Tianjin

[7]. Cyberbullying Detection: An Ensemble Based Machine Learning Approach, Kazi Saeed Alam, Department of Computer Science and Engineering Khulna University of Engineering & Technology Khulna, Bangladesh

[8]. Improving language understanding by generative pre-training, Radford, Karthik Narasimhan, Tim salimans, ilia sutskever albert, University of California, Vol 4, Year 2020.

[9]. Hasim sak, Andrew senior, francoise beaufays , Long short term memory recurrent neural network architectures for large scale acoustic modeling, Google brain,2022.
[10]. Jacob devlin, Kenton lee, Bert pre-training of deep bidirectional transformers for language understanding, google brain.
[11]. In this algorithmnpeng yin, katharina kann, mo yu, comparative study of cnn and rnn for nlp ,IBM research USA
[12]. Zachary C liption, a critical review of recurrent neural network for sequence learning,universirt of san diego.
[13]. Mingda chen,kevin gimpel,piyush sharma,radu soricut.google research toyota technological institute at chicago.
[14]. Emotion-cause pair extraction: a new task to emotion analysis in texts. riu xia, zixiang ding. Nanjing university of science and technology, chin
[15]. Research on text classification based on BERT-BiGRU MODEL, Qing Yu, Ziyin Wang and Kaiin this algorithmn Jaing,Tianjin
[16]. Ambroselli, C., Risch, J., Krestel, R., Loos, A.: Prediction for the newsroom: Which articles will get the most comments? In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), pp. 193–199. ACL (2018)
[17]. Arras, L., Horn, F., Montavon, G., M¨uller, K.R., Samek, W.: Explaining predictions of non-linear classifiers in nlp. In: Proceedings of the Workshop on Representation Learning for NLP, pp. 1–7. Association for Computational Linguistics(2016)
[18]. Alber, M., Lapuschkin, S., Seegerer, P., H¨agele, M., Sch¨utt, K.T., Montavon, G., Samek, W., M¨uller, K.R., D¨ahne, S., Kindermans, P.J.: innvestigate neural networks! arXiv preprint arXiv:1808.04260 (2018)
[19]. Van Aken, B., Risch, J., Krestel, R., L¨oser, A.: Challenges for toxic comment classification: An in-depth error analysis. In: Proceedings of the Workshop on Abusive Language Online (ALW@EMNLP), pp. 33–42 (2018)
[20]. Zachary C liption, a critical review of recurrent neural network for sequence learning,universirt of san diego.