Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

An Integrated Approach for Implementing AI Enabled Text Data Classification

A. Sandhya [1], Dr. J. Visumathi [2]
[1]Research Scholar,
Sathyabama Institute of Science & Technology, Chennai-600100
TamilNadu, sandhyalagar@gmail.com
[2] Professor, Dept of CSE,
Veltech Rangarajan Dr Sagunthala R&D Institute of Science and Technology,
Chennai-600100, TamilNadu
jsvisu@gmail.com

Article Info
Page Number: 670 - 681
Publication Issue:
Vol 71 No. 3s2 (2022)

Abstract
In this internet world, most of the enterprises run their business via web. Huge volume of data is being aggregated in the web repository on daily basis. The data is available in the web is in the form of text, images and numerical. Out of these three categories of data, text data is available abundantly and it is of more importance. The available text data must be classified and converted into useful data insights. This paper elaborates how to make efficient use of the available text data using various Artificial Intelligence Techniques. The paper also discusses on applying various hierarchy of AI techniques like Machine Learning, Neural Network and Deep Learning. The difference among each technique with its architecture, performance and accuracy is also discussed here.

Keywords: - Text classification, AI techniques, Machine Learning, Neural Network, Deep Learning Techniques.

Biographical notes: A.Sandhya has completed the Bachelor's degree in Computer Science and Engineering from SASTRA Deemed University, Master degree in Information Technology from Sathyabama University, Chennai, India. She is persuing her doctorate in the field of Big Data Analytics in Sathyabama University. She has 7 years of Industry experience and 7 years of teaching experience. She has published more than 5 papers in conferences and journals. Her current areas of interest include Bigdata, Artificial Intelligence and Deep learning.

Dr. J.Visumathi has completed the Bachelor's degree in Computer Science and Engineering from Manonmaniam Sundaranar University, Master and Ph.D. degree in Computer Science and Engineering from Sathyabama University, Chennai, India. She is working as Professor in the department of Computer Science and Engineering, Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, India. She has 20 years of teaching experience. She has published more than 75 papers in conferences and journals. Her current areas of interest include Network Security, Data mining, Bigdata, Cloud Computing and Artificial Intelligence.

Article History
Article Received*: 28 April 2022*
Revised*: 15 May 2022*
Accepted*: 20 June 2022*
Publication*: 21 July 2022*

I. INTRODUCTION AND RELATED WORK

The implementation of the AI techniques depends on various factors like the type of input data, volume of input, learning types such as supervised or unsupervised, nature of input data like linear or non-linear, type of problem like either classification or clustering and so on. Based on the above mentioned factors, any one of the suitable AI techniques like either Machine learning algorithms or Deep learning techniques will be chosen and implemented with its Framework. The basic implementation of AI is with machine learning algorithms and accurate results can be obtained by choosing the suitable algorithm for the appropriate problem statement. The summary of implementing the suitable algorithm based on the types of learning (unsupervised or supervised) and the type of data (categorical and continuous) is listed below:

Table 1: Learning Types Vs Algorithms

	Unsupervised	**Supervised**
Continuous	For Clustering and Dimension Reduction: - SVD - PCA - K-Means	1. Regression - Linear - Polynomial 2. Decision Trees 3. Random Forest
Categorical	- Apriori - FP-Growth - Hidden Markov model	For Classification: - KNN - Trees - Logistic regression - Naive-Bayes - SVM

Moreover before implementing any of the ML Algorithms, the optimal prediction and accuracy depends on an important factor known as Learning Type [5,7]. To conclude on the ML implementation, if the data is of Discrete, classification or categorization can be done in Supervised Learning and clustering is done using unsupervised learning algorithms. For the continuous data, Regression is preferred using supervised learning and Dimensionality reduction is done using unsupervised learning algorithms. As the data increases abundantly each and every second, there exist the need to upgrade the AI Techniques, leading to neural networks and Deep Learning Techniques.

II. METHODOLOGY USED AND TESTING TECHNIQUES

The advanced AI techniques emerged with the implementation of neurons under the domain of neural network (NN) and Deep learning (DL). To define a neural network, it is a system that works similar to the functions of neurons present in the human brains. NN performs various computation tasks faster. Moreover NN is a method to achieve Deep Learning [1]. NN architecture consists of three layers namely input layer, hidden layer and output layer as shown in Figure 1. NN acts as a great tool in various applications like pattern recognition, machine translation, prediction and optimization. The perceptron algorithm is used in NN for binary classification. The other algorithm

is Feed Forward Neural Network. From this algorithm various advanced algorithms were evolved and developed in the Deep Learning Framework.

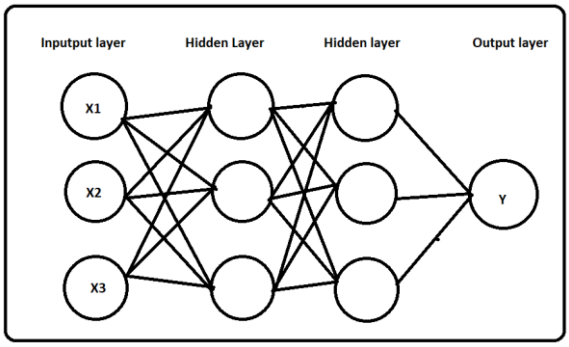


Figure 1: Neural Network – Architecture

Deep Learning is a special type of ML that replicates the learning approaches used by humans to obtain the knowledge. DL helps to improve the computer systems with experience and data. DL has various architectures which are based on artificial neural networks. The DL architectures that support supervised learning are Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN) and Encoder-Decoder Architectures. The DL algorithms that are suitable for Unsupervised Learning are Auto encoder and Generative Adversarial Networks. The algorithm preferred for Reinforcement learning are Networks for learning Actions, Values and policies. Based on the type of input such as Text, image and numerical, appropriate framework is selected and implemented to attain the optimal solution with maximum accuracy.

Based on the type of data acquired for processing through the proposed framework, the testing is carried out. The accuracy of the model is evolved based on various hyper parameters like batch size, number of epochs, learning rate and number of hidden layers. The following scenario is tested to derive at the accuracy of the model.

Scenario 1: Vary the batch size to find model performance.

Scenario 2: Derive the model accuracy based on the epoch number.

Scenario 3: Compare the model performance for various learning rate.

Since, we deal with text data; we evaluate the model performance by varying the different batch size and conclude how the model performs while processing more records. Similarly learning rates and number hidden layers are also experimented with different values. The next section describes in detail, how the DL framework is implemented for the classification of input text data.

III. Deep Learning for Text classification –Stepwise

Sentiment Analysis is a type of Text classification, in which prediction is done for various problems like movie reviews, email classification as spam or not, tweet sentiment prediction and so on. Deep Learning methods provides very accurate results for Text classification problems. The steps involved in text classification using Deep Convolution Neural Network are described in this section.

To perform the text classification using CNN, following steps must be implemented one by one.

1. Dataset Preparation and cleaning
2. Data preparation (Train and Test)

3. Train using Word Embedding
4. Apply CNN model and output layer

3.1 Dataset Preparation and cleaning:

The dataset is obtained from real time streams or from standard repository for the implementation. The dataset must comprise of the mixture of positive and negative values or tweets or reviews based on the application chosen. More the dataset is cleaned and ready, the model yields accurate results. Major data cleaning activities includes (i) all text in lowercase (ii) removing the punctuations (iii) removal of words with length less than one character. (iv) one sentence in a line in input text dataset.

3.2 Data preparation (Train and Test):

In this step the data is split into training and testing. The ratio of training and testing split can be 80: 20 or 90: 10 based on the application. Various operations are performed on the training data set and the suitable model is trained. Then the reserved testing data is verified with the trained model and the accuracy is measured. In the scenario of text classification, as a first step, the vocabulary must be constructed based on the training dataset. Since huge number of vocabulary obtained, the tokens with minimal occurrences can be removed from the list of vocabulary.

3.3 Train using Word Embedding:

The core part of Text classification starts from this module. In Deep Learning model the input need to be converted into vectors that can be understood by the neurons. Word Embedding is a method to represent each and every word in the vocabulary into vector in high dimensional space [2, 3, 4]. Also, the word embedding acts as a part of fitting to a neural network model. The words with similar meanings are represented closer in the vector space. This approach works more perfectly than the classical approach like bag of words.

The word embedding has three main algorithms namely (i) Embedding layer (ii) Word2Vec (iii) GloVe. Any of the suitable algorithms is chosen to form the network. In the learned vector space, the position of the word is known as embedding. The embedding layer algorithm is user defined based on the specific dataset and conditions [11, 12]. Whereas Word2Vec and GloVe are pre trained model that can be used for our dataset. Embedding layer is the first hidden layer and the steps followed in embedding layer are listed below:

1. Each input word is mapped by unique integer.
2. Specify the input dimension (size of vocabulary), output dimension (size of vector space) and input length.
3. Weights are learned and added to this layer.
4. The embedding layer produces output, which is 2D vector, where each word in the input document is mapped with the embedding.
5. Dense layer can be connected directly to the embedding layer by converting 2D matrix output into 1D vector by Flatten layer.

By now the trained model learned perfectly the trained dataset. The working flow of word embedding is shown below in Figure 2.

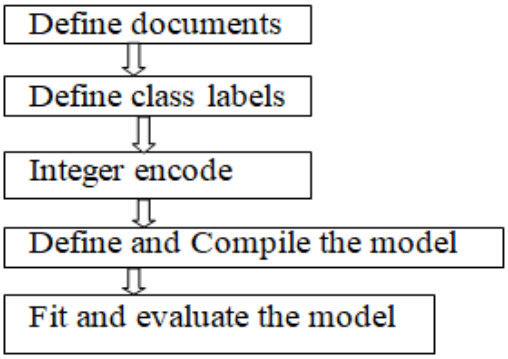


Figure 2: Workflow of Word Embedding

3.4 Apply CNN model:

The next step is to define the neural network model. CNN had proved its accuracy in many text classification case studies and it is confirmed as the best model. The CNN architecture model has three layers namely (i) convolution layer (ii) Pooling layer (iii) Fully connected layer. Convolution layer detects the various features from the input. These features are extracted by applying filter or kernel to the input. Then convolution function is applied to it. Convolution function is a simple matrix multiplication. Now the appropriate activation functions are applied before passing to next layer. This activation function in neural network is responsible to transfer the weighted input node. Various activation functions available in neural networks are listed in Table 2.

Table 2: Activation Functions in Neural networks

Function Name	Equation
Sigmoid	$\sigma(x) = \frac{1}{1 + e^{-x}}$
tanh	$tanh(x)$
ReLU	$max(0, x)$
Leaky ReLU	$max(0.1x, x)$

Among the various activation functions listed in Table 2, rectified linear activation function represented as ReLU acts as the best activation function for most neural networks. ReLU is easier for the model to train and also it achieves good performance. ReLU has major advantages like (i) make the model to learn faster. (ii) Acts as default activation function in case of developing the multilayer perceptron and CNN. (iii) Trains deeper networks efficiently [10]. The next layer is the pooling layer. The main role done by the pooling layer is reducing the output received by the convolution layer. The output is reduced by half when compared with the convolution layer output. The last layer is the fully connected layer. The 2D output is converted into one dimensional long vector, which represents the features extracted by the CNN. Multilayer perceptron layers are used for interrupting the features of CNN [6, 8, 9]. The sigmoid activation function is used in the output layer to produce binary classification results as spam or not, positive review or negative review and so on.

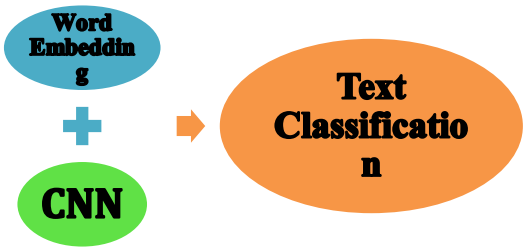
Thus the model is run with the training dataset which yields maximum accuracy and thereby tested with reserved testing dataset. The results obtained by the model are of high percentage in accuracy.

IV. PARAMETERS IMPROVISING RESULTS ON TEXT CLASSIFICATION

To achieve accurate results in text classification using deep learning models few practices and techniques need to be followed and is discussed in this section.

4.1 Scenario 1: The advantage of Word Embedding over Traditional models like One-hot Representation:

The most of the Natural Language Processing tasks that deals with text data uses many of the traditional methods to handle text data. The most common methods are term frequency (TF), inverse document frequency (IDF) and one hot representation. The traditional methods perform the numeric representation for the words given in the document. It does not deal with relevant words during representations. While using word embedding algorithm like Word2Vec, the more similar words are represented with closer value in the vector space. This ensures the text classification to be more effective. To attain the best results for any of the text or document classification problems it is proposed to choose word embedding along with CNN.



The word2vec helps to attain the semantic nature of text representation, whereas the traditional approaches provide the syntactic representations. The comparison is shown in the figure 3.

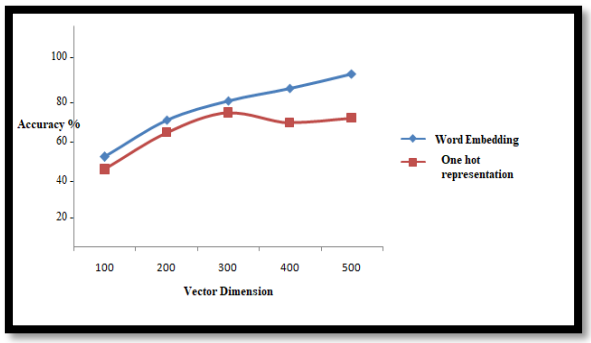


Figure 3: word embedding and one hot method - Accuracy comparison

The horizontal axis denotes the dimension of the vector. The vertical axis represents the accuracy. While using word embedding algorithms, the words are represented in the vector space in most appropriate value even for huge datasets. Where as in the traditional method, the consistency in vector notation is not reached. This concludes that the use of word2vec embedding drastically will improve the accuracy in classification.

4.2 Scenario 2: The use of single layer in CNN for optimal results:

While using more number of layers in the architecture of CNN, the information loss occurs when the data is transferred from one layer to the other. The structure of the data also may change and leads to reduced accuracy in classification. Therefore, CNN Architecture with Single layer yields appropriate results in the text classification scenarios. Implement various sized kernel across the filters. By doing so word representations are grouped by different scales. Kernel size can be 3 or 4 or 5 and the number of filters may range till 100. Hence the model accuracy will not improve if we increase the number of convolution layers. Model with minimum layer is more than enough to attain good accuracy. The figure 4 shows the accuracy value of the model based on the number of convolutional layer (CL) used in the model. The horizontal axis denotes the volume of data and the vertical axis represents the accuracy value in percentage. The model performs high accuracy value while using two convolutional layers. As the number of convolutional layers increases in the model, the accuracy value is reducing. This will eventually end in poor classification.

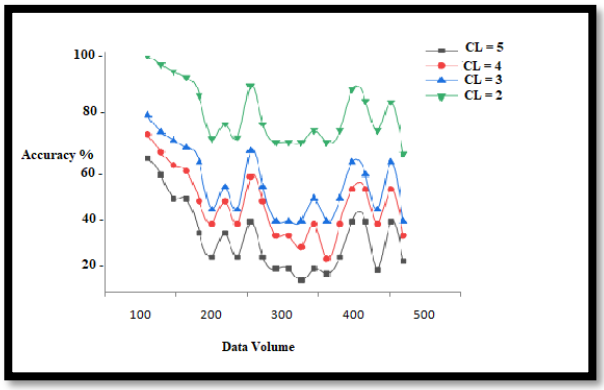


Figure 4: - Accuracy compared for various number of convolution layers.

Hence for efficient classification model, it is best option to reduce the number of convolutional layers and attain the maximum accuracy. The model with one layer with different filters will give more accurate classification. The implementation results can vary from scenario to scenario by doing the change in hyper parameters. These hyper parameters play a vital role while tuning the neural network for text or document classification problem [13].

4.3 Scenario 3: The implementation of character level CNN before word level CNN:

The most efficient parameter to attain accuracy in text classification model is to perform the character level CNN and then to apply word CNN. The character level CNN gives more accurate representations and helps to attain the accuracy value at faster rate. From figure 5, we can see the flow of linking between the character embedding and word embedding. Initially, the text input consists of group of words. After pre-processing, each word are iterated. For each of the words in the text corpus, every character of the word is processed to generate the character embedding vectors. While all the characters of the word are processed completely, then the word embedding vectors are created. The character embedding vectors generation before the word embedding helps to attain accurate vector representations for each and every character, which in turn reflects in the word vector space.

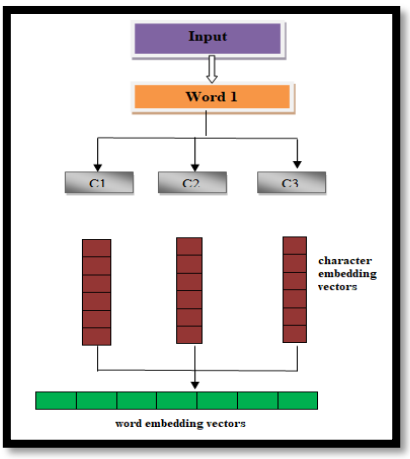


Figure 5: character embedding and word embedding vector flow

Thereby, while handling text classification, the results can be improved by implementing the character level CNN first rather than word level CNN. This model achieves success for problems with large corpus of words.

V. MODEL EVALUATION BASED ON HYPER PARAMETERS USED IN TEXT CLASSIFICATION

The text classification using embedding algorithm uses various hyper parameters like batch size, Epoch, Learning rate and number of hidden layers. The performance of the model is evaluated based on the functionality and behaviour of the hyper parameters used in the model.

5.1 Scenario 1: Deriving optimal Neural Network for text input:

The challenging activity in producing the accurate result depends on type of Neural Network (NN) model being used in the framework. The selection of NN model is directly proportional to the type of input dataset. Certain NN model that produced high accuracy result for test will not always yield accurate results for image dataset. Therefore, selection of the NN model based on the input dataset will help to attain most accurate classification output. In figure 6, the input text data is classified using various machine and deep learning models like support vector machine (SVM), XGBoost, Naive bayes, Term frequency (TF), Inverse document frequency (IDF), convolutional neural network (CNN) and Recurrent neural network (RNN).

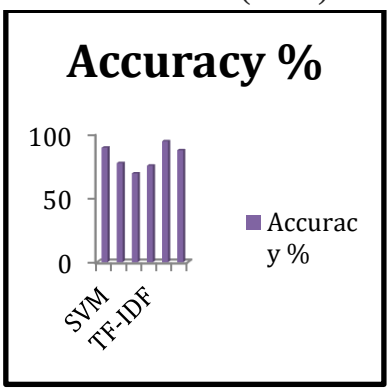


Figure 6: Accuracy tested for various ML/DL models

The results clearly conclude that the CNN model provides maximum accuracy for text classification when compared to other machine and deep learning models. Hence, CNN can be used for text classification instead of RNN.

5.2 Scenario 2: Fixing suitable learning rate to attain maximum accuracy

While training the model, we make sure to reduce the errors occurred during training. This is done by the parameter learning rate. The model is adjusted with certain values to see the error occurred. The error log and learning rate values are logged. At some point, the training error will start reducing and the value at which it is reduced is known as the good starting point of learning rate. This will continue for the while and the loss may increase. In that case, the in between values are termed to be best learning rates. These values were used for testing the model with different learning rates and derive the classification in a accurate way.

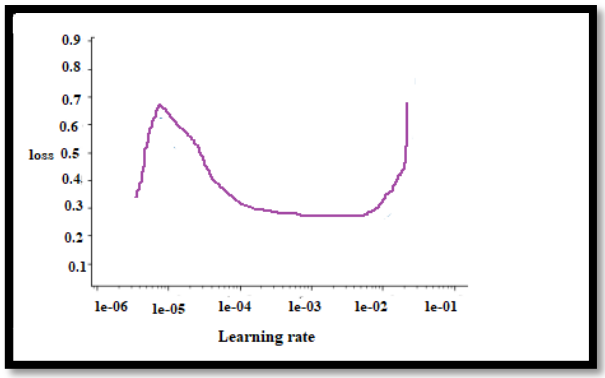


Figure 7: loss attained against various learning rates.

In figure 7, learning rate and loss are mapped. During training, the loss value is high initially and when the learning rate is increased, the loss value decreases and becomes steady for few ranges of learning rate. Again the loss increased. The steady state is the best learning rate since it produces minimal loss. Based on this training learning rate, the testing is performed. Since the learning rates at le-04, le-03 and le-02 has decrease in training loss, we compare how the loss occurs for different number of epoch. The range of epoch starts from minimum 10 to maximum 70.

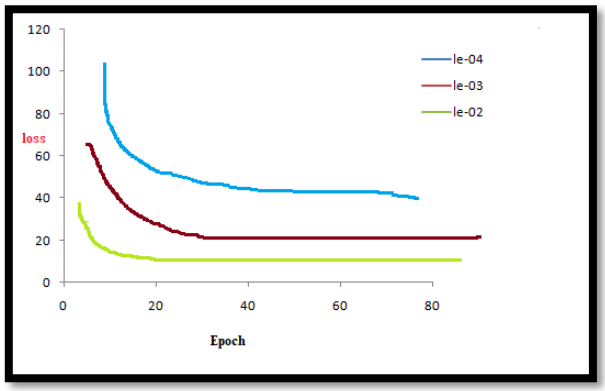


Figure 8: loss per epoch for various learning rates

From the figure 8, we can infer that as the learning rate and the epoch are increased, the loss is reduced. The model works steady with minimal loss at higher learning rates. Based on the epoch used in the model, the accuracy is evaluated for various learning rates in figure 9.

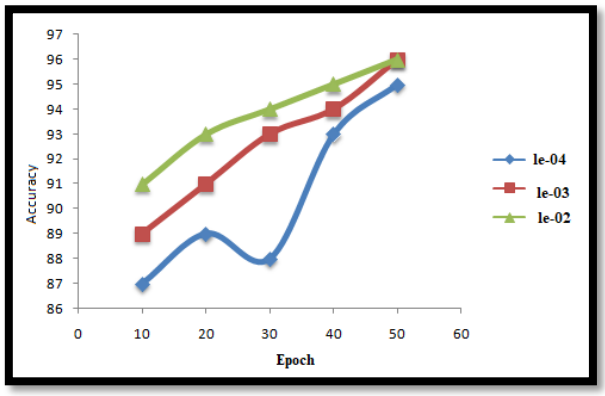


Figure 9: Accuracy per epoch for various learning rates

As the learning rate increases with the number of epoch, the models accuracy also improved to provide efficient classification results.

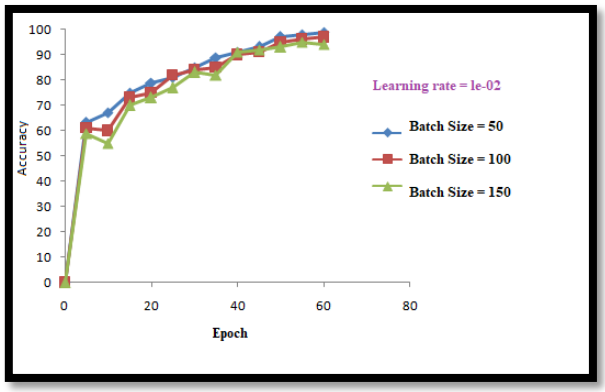


Figure 10: Accuracy per epoch for various batch size of a learning rate

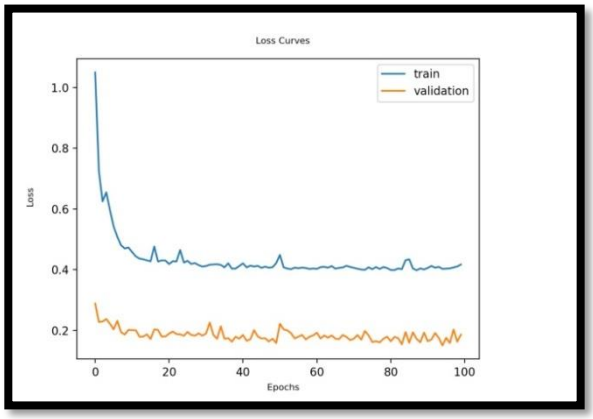


Figure 11: Loss Curves for Training and Validation data.

The model is also tested with fixed learning rate for various batch sizes on the validation dataset. This ensures to derive at the optimal learning rate and batch size to attain maximum accuracy in the

model. In figure 10, the learning rate le-02 is tested with batch size 50, 100 and 150. Similarly, different learning rate was tested with various batch sizes. Figure 11 describes the loss curves for training and validation data for the built model based on the above mentioned hyper parameters. The X axis denotes the number of epochs and Y axis represents the loss values. The result proves that the validation loss is minimal for the built model using word embedding algorithms and CNN for text classification problems.

VI. CONCLUSION

This paper described the exhausted study and approaches for handling the text classification using Advanced AI techniques mainly the Deep Learning algorithms. The paper also provides the different ways to improve the results of the text classification using CNN. The next level of improvement needs to be performed with hybrid model implementation. This hybrid approach is considered for further enhancement of this paper and will be implemented soon.

REFERENCES

[1] Conneau A, Schwenk H, Barrault L, Lecun Y (2017), "Very deep convolutional networks for text classification". In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics: Volume 1, Long papers, vol 1, pp 1107–1116

[2] Debora Nozza, Pikakshi Manchanda, Elisabetta Fersini, Matteo Palmonari, Enza Messina(2021), "LearningToAdapt with word embeddings: Domain adaptation of Named Entity Recognition systems",Information Processing & Management,Volume 58, Issue 3,2021, ISSN 0306-4573, https://doi.org/10.1016/j.ipm.2021.102537.

[3] Duyen Thi Do, Thanh Quynh Trang Le, Nguyen Quoc Khanh Le(2021), "Using deep neural networks and biological subwords to detect protein S-sulfenylation sites, Briefings in Bioinformatics", Volume 22, Issue 3, May 2021, bbaa128, https://doi.org/10.1093/bib/bbaa128

[4] Liang-Chih Yu, Jin Wang, K. Robert Lai and Xuejie Zhang(2017), "Refining Word Embeddings for Sentiment Analysis", Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 534–539, Copenhagen, Denmark, September 7–11, 2017. Association for Computational Linguistics.

[5] Saqib SM, Kundi FM, Ahmad S (2018) "Unsupervised learning method for sorting positive and negative reviews using LSI (latent semantic indexing) with automatic generated queries". Int J Comput Sci Network Secur 18(1):56–62

[6] Sangwan N., Bhatnagar V. (2021) "Optimized Text Classification Using Deep Learning." In: Goar V., Kuri M., Kumar R., Senjyu T. (eds) Advances in Information Communication Technology and Computing. Lecture Notes in Networks and Systems, vol 135. Springer, Singapore. https://doi.org/10.1007/978-981-15-5421-6_30

[7] Shervin Minaee, Nal Kalchbrenner, Erik Cambria, Narjes Nikzad,Meysam Chenaghlu, Jianfeng Gao(2021), "ACM Computing Surveys",Volume 54,Issue 3,May 2021 Article No.: 62pp 1–40https://doi.org/10.1145/3439726

[8] Sun J., Jin R., Ma X., Park J., Sohn K., Chung T. (2021), "Gated Convolutional Neural Networks for Text Classification". In: Park J.J., Fong S.J., Pan Y., Sung Y. (eds) Advances in Computer Science and Ubiquitous Computing. Lecture Notes in Electrical Engineering, vol 715. Springer, Singapore. https://doi.org/10.1007/978-981-15-9343-7_43

[9] Wang, S., Zhou, W. & Jiang, C(2020). "A survey of word embeddings based on deep learning". Computing 102, 717–740 (2020). https://doi.org/10.1007/s00607-019-00768-7.

[10] Xin Liu, Yanju Zhou, Zongrun Wang(2021), "Deep neural network-based recognition of entities in Chinese online medical inquiry texts, Future Generation Computer Systems", Volume 114, 2021, Pages 581-604, ISSN 0167-739X, https://doi.org/10.1016/j.future.2020.08.022.

[11] Zeng D, Liu K, Lai S, Zhou G, Zhao J (2014) "Relation classification via convolutional deep neural network." In: Proceedings of the 25th international conference on computational linguistics: technical papers, pp 2335–2344 (2014)

[12] Yoav Goldberg, "Neural Network Methods in Natural Language Processing"

[13] https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras