I. INTRODUCTION

Humans are subjective creatures and their opinions are important because they reflect their satisfaction with products, services and available technologies. A movie review is an article reflecting its writers' opinion about a certain movie and criticizing it positively or negatively which enables everyone to understand the overall idea of that movie and make the decision whether to watch it or not, and these reviews can affect the success or failure of a movie [4]. Therefore, a vital challenge is to be able to classify movies reviews to retrieve and analyze watchers more effectively. Movie reviews classification into positive or negative reviews are connected with words occurrences from the reviews text, and whether those words have been used before in a positive or a negative context. These factors help enhance the review understanding process using Sentiment Analysis, where it has become the gateway to understanding consumer needs. Sentiment analysis is concerned with identifying and categorizing opinions which are subjective impressions, not facts but usually expressed in a text and determining whether the writer's feelings, attitudes or emotions towards a particular topic are positive or negative [1]. The aim of this research is to classify movie reviews into positive and negative reviews using Linear Support Vector Classifier model and CNN and compare the discrepancies between the two models.

II. METHODOLOGY

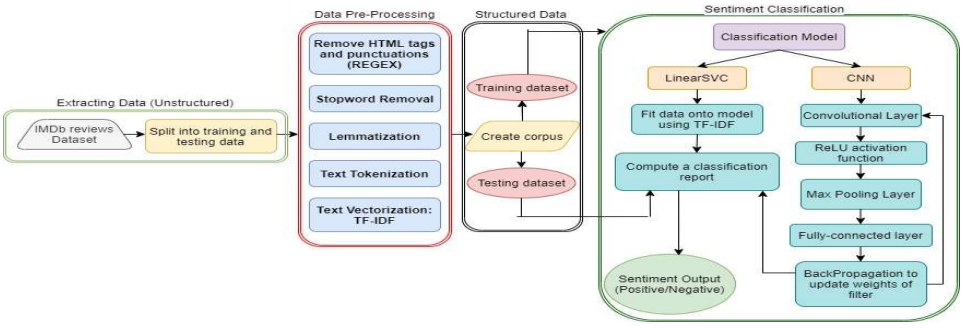

Figure 1: Block Diagram of IMDb movie reviews sentiment classification

The process of Sentiment Analysis and preprocessing the data steps includes Natural Language Processing (NLP) tasks or text cleaning techniques, tokenization, word filtering, lemmatization, vectorization and classification using Linear Support Vector Classifier (LinearSVC) model and Convolutional Neural Network (CNN) as shown in Figure 1 and Figure 2. The text cleaning techniques are used to eliminate unwanted information from the dataset (reviews) that does not affect the outcome of the predictions such as HTML tags like "
", punctuations and special characters such as "/" , and the removal of stopwords. Regular Expressions (Regex) which are sequence of characters that illustrates a search pattern to find and replace unwanted words were used [2]. Stopwords are words (English language words in this case) that do not add much meaning to a sentence and can be safely ignored without sacrificing the meaning of the review, words like 'are', 'the' does not provide any insights and does not affect the process of classification. Usually, these words are filtered out because they return a vast amount of unnecessary information. Text tokenization is a process of segmenting text into sentences into single words (tokens) by specifying the basic linguistic units such as words which will be later used for easy vectorization of the data.

For grammatical reasons, documents are going to use different forms of a word, such as schedule, scheduled, and scheduling. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set. Stemming and Lemmatization are methods used as part of text-preparation process before it analyzes the meaning behind a word in Sentiment Analysis. Stemming algorithms works by chopping off the ends of words in the hope of achieving the goal of reducing inflectional forms correctly most of the time, and often includes the removal of derivational affixes. Lemmatization usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word [3]. However, Lemmatization is used in this work as the comprehension of the meaning of the reviews is crucial in the classification process and it is more accurate than Stemming. A Corpus-based approach is adopted and helps to solve the problem of finding opinion words with context specific orientations. It depends on patterns that occur together along with a seed list of opinion words to find other opinion words in a large corpus.

While deep learning algorithms deal with numbers, the data we have in this work is text. In order to be able to classify these textual data using deep learning classifiers, these textual data need to be transformed into numbers. This process or transformation of text into numbers is called text vectorization. Text vectorization is an important step enabling deep learning classifiers for analyzing the textual data. Count Vectorization (Bag of Words model) and TF-IDF vectorization are one of the two ways to vectorize the textual data of movie reviews. Bag of Words model (Count Vectorization) converts the data to numeric values like 1 and 2 based on the number of times they appear in the text. Whereas, TF-IDF is chosen for vectorizing the textual data because it takes into consideration the importance of a word across the complete list of documents. Within each document, each word is measured for its relevance in that document and is given weight according to how relevant it is to that document. Therefore, if a word exists in many documents, the weight given to that word is diminished, as it is not useful for discerning the documents. TF-IDF creates a matrix in which rows represent the documents, columns represent the words, and values represent the relevance of the words in the documents. TF, term frequency, measures how many times a word exists in a given document and IDF, inverse document frequency, measures how many times that word exists across a set of documents [4].

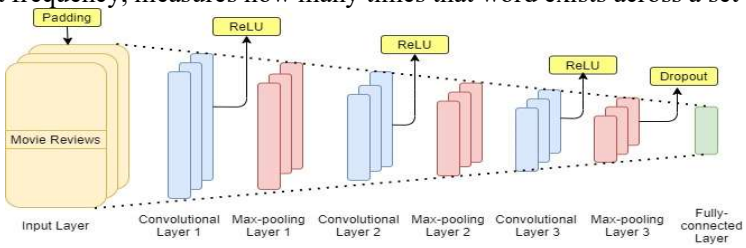

Figure 2: Architecture of CNN for IMDb movie reviews sentiment classification

One vital data mining function is classification, which builds a model for labelling testing data based on previous training data. Assigning classes to reviews can be done by such model which predicts the label of new data. Some of the classification algorithms that has proven their efficiency in this work are Linear Support Vector Classifier (LinearSVC) and Convolutional Neural Network (CNN). This paper addresses sentiment analysis of movie reviews as a classification task and different classification algorithms are considered and compared to assess their performances for the task at hand. SVC algorithm classifies by analyzing data and recognizing patterns, so-called supervised learning methods and is able to recognize separate hyperplanes that maximizes margins between two different classes. However, SVC lacks the problem of selecting appropriate parameters or features as these will greatly influence the classification accuracy results [4]. As a result, CNN is also adopted and has been widely used in NLP recently, as it is effective for text classification by extracting significant features of the text after local information of texts is stored using convolutional layers, padding, dropout and activation function.

III. EXPERIMENTS

This paper uses the IMDB Dataset of 50K Movie Reviews from Kaggle as the dataset for text sentiment classification [5]. This dataset is divided into two sets for training and testing, with 37,500 reviews of movies downloaded from IMDb allocated for the training data and 12,500 reviews for the testing data. In each dataset, the number of reviews labelled “positive” and “negative” is equal. The reason behind this split percentage is that it is commonly used split in research [6]. LinearSVC and CNN are the two models used for sentiment classification after the data pre-processing techniques are once completed.

The Linear Support Vector Classifier (SVC) investigates information, characterizes choice limits and uses the components for the calculation, which are performed in the input space. The vital information is presented in two arrangements of vectors, each of size m. At this point, each datum (expressed as a vector) is ordered into a class. Next, the machine identifies the boundary between the two classes that is far from any place in the training samples. The boundary characterizes the classification edge, expanding the edge lessens ambivalent choices. Unigrams, bigrams and trigrams are the different n-grams that are used to extract sets of words from the reviews in TFIDF. The data is fitted onto this model using the TF-IDF vectorization technique as it gives a higher accuracy than Count Vectorization (Bag of Words model) and a classification report is achieved as shown in Table 1. The proposed CNN consists of three convolutional layers with appropriate padding in the input layer

and a max-pooling layer because the small size of our input dimension does not require additional layers to extract features/patterns, and ends with a fully-connected layer. One may argue that stacking more convolutional layers might be better, as the deeper network is known to capture higher level patterns. But in this paper, only three layers are used to avoid overfitting. As text is one-dimensional, the words are converted into word embeddings to visualize words in two-dimensions, each word along one axis and other axis for the elements of vectors. In order to optimize the CNN model, the following settings were used: a batch size of 64; number of filters = 100; embedding dimension = 100; size of filters to either 3,4 or 5 (n-gram); input dimension is dependent on the length of the review and the output dimension is 1. A filter size of [n*width] is used where 'n' is the number of sequential words and width is the depth of the filter or dimensional embeddings. The output of this filter is the weighted sum of all elements covered by the filter (single real number). The main idea is that each filter will learn a different feature to extract. For example, each of the [n*width] filters look for the occurrence of different bi-grams that are relevant for analysing sentiment of movie reviews. And the same goes for different sizes of filters (n-grams) with heights of 3,4 and 5. This model has 100 filters of 3 different sizes (n-grams), i.e., 300 different n-grams. Later, these are concatenated into a single vector and passed through a linear layer to predict the sentiment. The strides for the kernels is set to 1. Every layer took ReLU as an activation function. The number of epochs used to train this model is five and was enough to view valid and satisfactory results of values of accuracy and loss. The number of the trainable parameters is 2,620,801 and ADAM optimizer is adopted with a learning rate of 0.0001. Since the dropout is a regularization technique to avoid over-fitting in neural networks, the dropout for each layer is set to 0.5, which is related to the fraction of the input units to drop and it will be turned ON during training and OFF during testing. Through backpropagation, the weights of the filters are updated so that whenever certain n-grams that are highly indicative of the sentiment are seen, the output of the filter is the highest value amongst all. This high value is then passed through the max pooling layer if it is the maximum value in the output. Binary Cross Entropy with Logits Loss is used here for better numerical stability. To evaluate the overall performance of the classifiers, we consider several performance metrics such as precision, recall, f1-score, and accuracy and a classification report is achieved for CNN as shown in Table 2. A model is built which prompts users to type in their reviews real-time which will display results immediately with the predicted label along with how accurate the prediction is and could be implemented on social media platforms as a future work.

Table 1: Classification Report of LinearSVC Model

	Precision	**Recall**	**f1-score**	**support**
Negative Reviews	0.91	0.89	0.90	6157
Positive Reviews	0.89	0.92	0.91	6343

Table 2: Classification Report of CNN

	Precision	**Recall**	**f1-score**	**support**
Negative Reviews	0.91	0.89	0.90	6157
Positive Reviews	0.89	0.92	0.91	6343

Table 3: Comparison between LinearSVC and CNN

Classification Model	**LinearSVC**	**CNN**
Training Accuracy	97.34%	96.52%
Test Accuracy	93.23%	88.56%
Validation Accuracy	90.29%	86.97%
Training Loss	0.228	0.105
Test Loss	0.459	0.298
Validation Loss	0.520	0.348

Based on the Table 3, we can observe that this work is able to classify movie reviews very well using both LinearSVC and CNN models with competitive and satisfactory results. We can observe that LinearSVC performs slightly better than CNN, however, LinearSVC does not consider domain-independent sentiment classification and the loss values are larger than that of CNN although it has slightly higher accuracy values than CNN. CNN can overcome many challenges of sentiment analysis. For example, words in a specific region are more likely to be related than words far away. Thus, CNN can automatically and adaptively extract spatial hierarchies of features out of written reviews that may capture different writing styles of users.

REFERENCES

1. Hannah Kim and Young-Seob Jeong, “Sentiment Analysis Using Convolutional Neural Networks”, Applied Sciences, vol. 1, no. 2, pp. 1-3, June 2019.
2. Walaa Medhat, Ahmed Hassan, Hoda Korashy, “Sentiment analysis algorithms and applications: A survey”, Ain Shams University, vol. 5, no. 1093-1113, pp. 2-3, April 2014.
3. Abdullah Alsaeedi, Mohammad Zubair Khan, “A Study on Sentiment Analysis Techniques of Twitter Data”, International Journal of Advanced Computer Science and Applications (IJACSA), vol. 10, no. 2, pp. 7-8, 2019.
4. Munir Ahmad, Shabib Aftab, “Sentiment Analysis using SVM: A Systematic Literature Review”, International Journal of Advanced Computer Science and Applications (IJACSA), vol. 9, no. 2, pp. 3-4, 2018.
5. “IMDB Dataset of 50K Movie Reviews”, Kaggle.com, 2019. [Online]. Available: https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews
6. Mais Yasen, Sara Tedmori, “Movie Reviews Sentiment Analysis and Classification”, ResearchGate, vol. 1, no. 2, pp. 1-5, April 2019.
7. “Sentiment Analysis - Cleaning, EDA & BERT (88% Acc)”, Kaggle.com, 2019. [Online]. Available: https://www.kaggle.com/madz2000/sentiment-analysis-cleaning-eda-bert-88-acc

APPENDIX

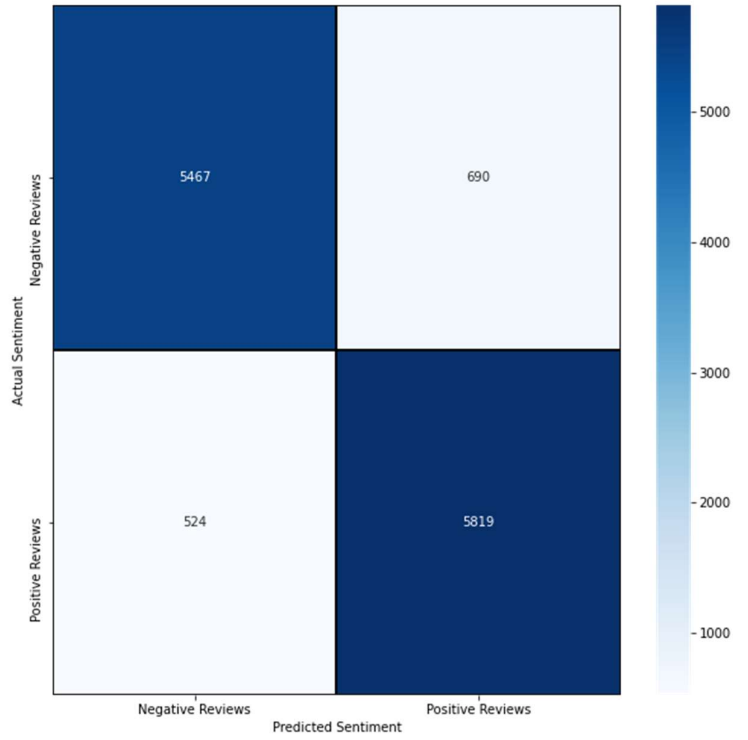

Figure 3: Confusion Matrix of LinearSVC model with TF-IDF

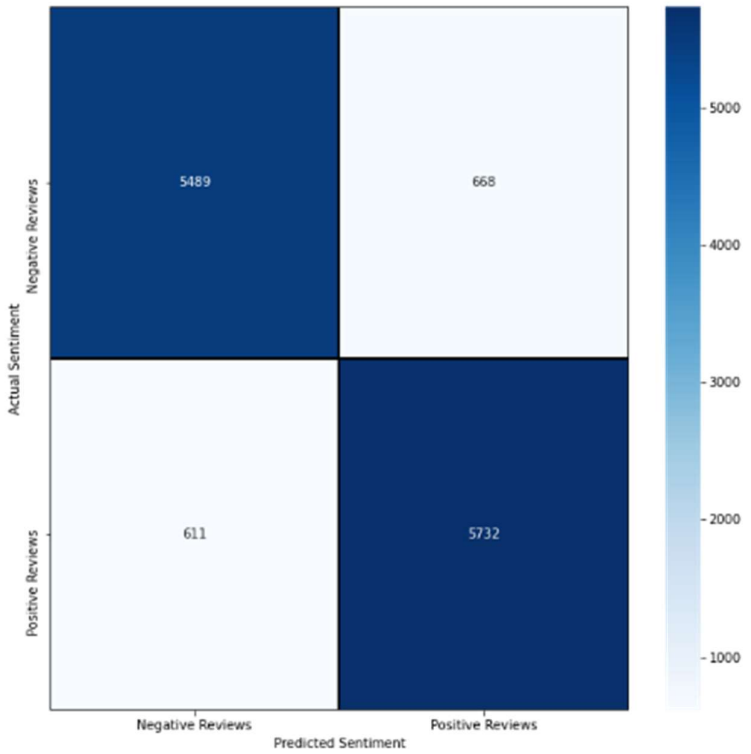

Figure 4: Confusion Matrix of CNN model

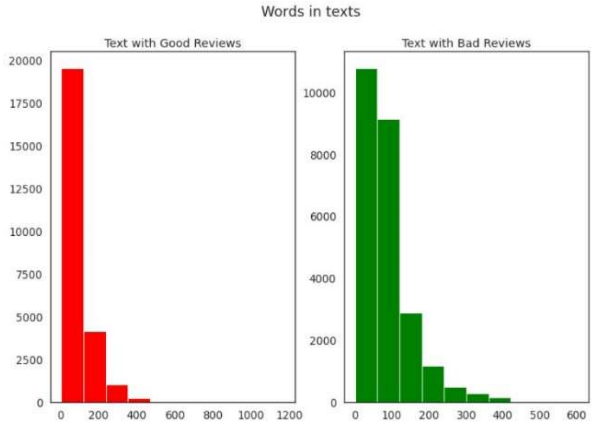

Figure 5: Plot of number of words in positive and negative reviews

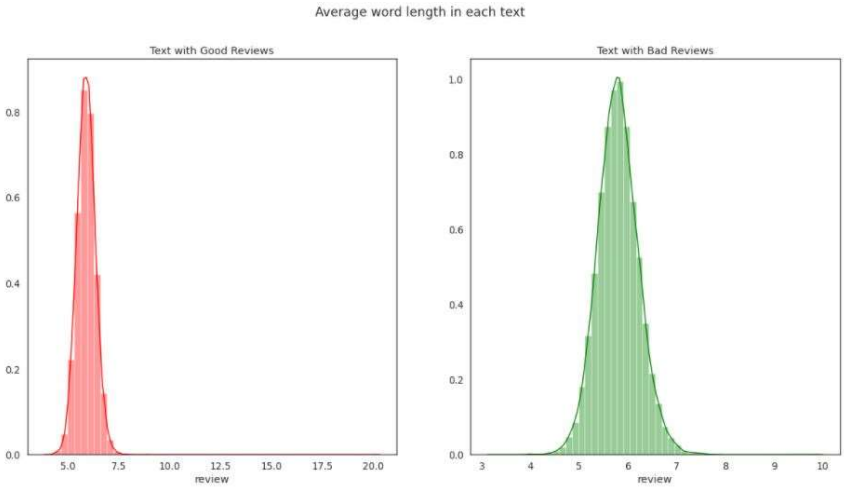

Figure 6: Plot of average word length in each review

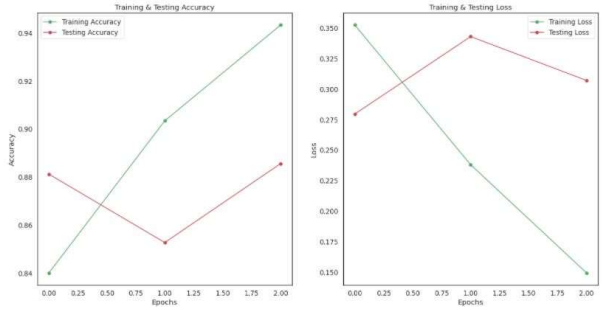

Figure 6: Plot of values of accuracy and loss of LinearSVC model

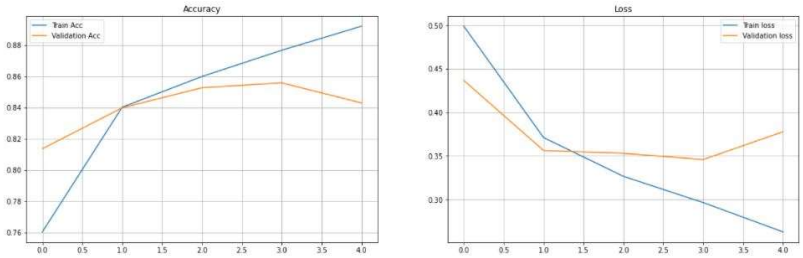

Figure 7: Plot of values of accuracy and loss of CNN model

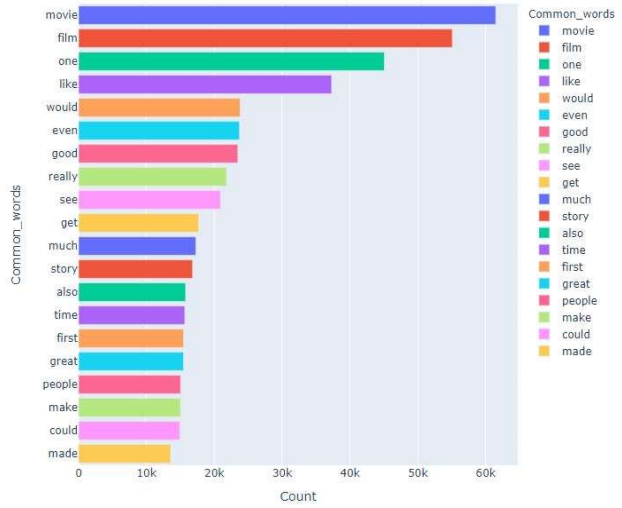

Figure 8: Plot of common unigrams in movie reviews dataset [7]

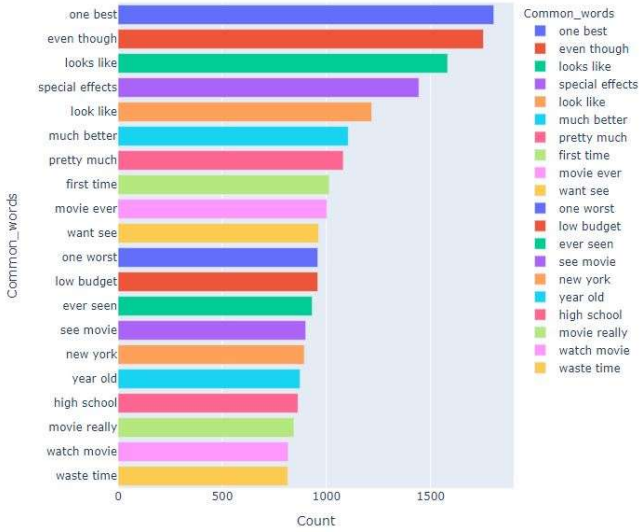

Figure 9: Plot of common bigrams in movie reviews dataset [7]

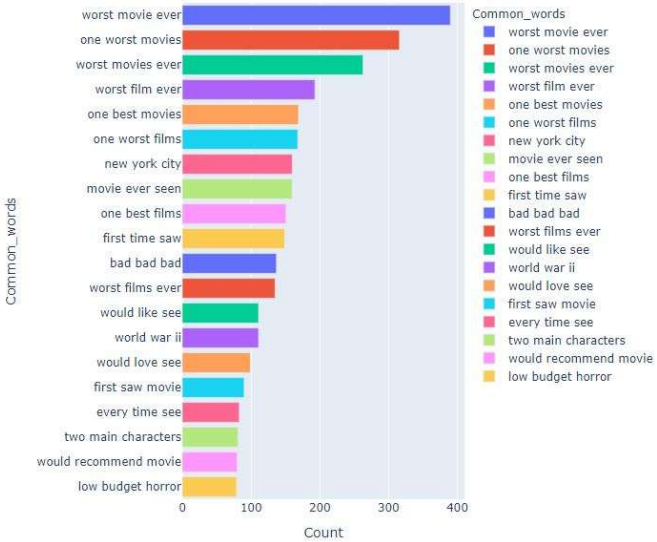

Figure 10: Plot of common trigrams in movie reviews dataset [7]