A comparison between Machine Learning and Deep Learning techniques for Textual Feedback Analysis

Urooj Abdul Haleem and Syed Zaffar Qasim⋆

⋆ Department of Computer & Information Systems Engineering, NED University, Karachi, Pakistan

Abstract- The higher educational institutions gather student feedback after the end of each semester to improve the quality of education. The feedback consists of a grading scale to answer the questions followed by a textual response conveying the sentiments regarding the student's experience. Since there is a considerable amount of response influx, going through every single textual feedback is time consuming; hence the need arises to extract sentiments from individual comments and classify them as positive, negative or neutral. The aim of our research is the comparison of various machine learning and deep learning approaches for developing an effective sentiment classification system for instructors. In this study, we analyzed student feedback consisting of 19000 comments and trained various machine learning and deep learning algorithms using several feature extraction techniques. Among the different algorithms employed, a cascading neural network consisting of CNN combined with LSTM using Glove word embedding outperformed all the other architectures giving an accuracy of 91.27%.

Index Terms- Deep learning, feedback analysis, opinion mining, sentiment analysis, text mining

I INTRODUCTION

Sentiment analysis involves analyzing the textual appraisal and opinions of people toward objects and their properties [1, 2]. The objects include products, services, persons, organizations, topics etc. It is also called opinion mining and has gained significant attention in recent years as a powerful tool for understanding and analyzing human emotions, opinions, and attitudes from various types of texts.

One of the most critical areas where sentiment analysis can be applied is in the field of education, specifically in analysing student feedback on instructors. A fundamental step in this analysis is to find the sentiment orientation, also called polarity, of some textual unit e.g. a single comment of a student. This polarity can be categorized as fixed labels which are positive, negative and neutral.

This research paper aims to apply sentiment analysis techniques to student feedback data collected from the Rate-MyProfessors.com website, a popular platform where students can rate and provide feedback on their instructors and academic environment. The site contains a wealth of information, including student evaluations of instructors, textual comments, and overall ratings. The data from this website can provide valuable insights into student perceptions of instructors, which can be used to enhance the education quality and improve the learning experience. A distinctive feature of textual feedback is that it allows the students to identify various issues and problems that is otherwise not possible with scale-based scores. Additionally, the students can give valuable suggestions for improvement in course management and syllabus modification.

In this work, two different approaches have been used for sentimental analysis of student reviews. The first approach is a deep learning approach, which is like the one proposed by Aytu˘g Onan [3]. Onan's approach uses a deep learning model to analyze student feedback and extract opinions and sentiments from the text. The second approach that we have employed is a classification with feature extraction technique, like the one discussed by Avinash and Sivasankar in [4]. This approach uses various feature extraction techniques to extract relevant information from the commercial product feedback data and classify it into different sentiments.

The goal of this research is to compare the performance of these two approaches and determine which one is more effective for sentiment analysis of student feedback on instructors. We hope that the outcomes of this study will give valuable understanding into student perceptions of instructors and help the education community to improve the quality of education.

The organization of the rest of the paper is mentioned next here. In sec II, we cover a brief account of the past work related to sentiment analysis using machine learning (ML) and deep learning (DL) techniques. In the next section III, the detailed methodology for carrying out the textual feedback analysis is presented. It includes the description of the dataset, proposed framework, feature extraction and the various ML and DL methodologies. Then the section IV covers the details of the experiments conducted, their results and a discussion on the analysis. The conclusion of this study will be covered in sec V.

II RELATED WORK

A Machine Learning Techniques for Sentimental Analysis

Jagdale et al. [5] applied the Naïve Bayes and Support Vector Machine (SVM) approaches of ML on a dataset acquired from Amazon which contained reviews about computers and other electronic products. The reviews were grouped into positive and negative. For camera reviews, the accuracy obtained using Naïve Bayes was 98.17% and with SVM, it was 93.54%.

In the work carried out by Sudhanshu Kumar et al. [6], machine learning techniques were used to analyze the sentiments on book preferences on a segmented data. The book reviews were collected using questionnaire along with the gender and age related data. The segmentation was based obviously on gender and age to examine the influence of these factors on user reviews. The sentiment analysis is performed with different ML techniques that include Naïve Bayes (NB), SVM, Maximum Entropy etc. The reviews were preprocessed by removing the punctuations and stop words and feature extraction was carried out using Bag-of-words model.

The authors in [7] used a lexicon-based approach to classify student comments as positive, negative, or neutral. They used a dataset of student comments and applied a lexicon-based approach, which utilizes a lexicon of words along with the corresponding sentiment scores. The scores were calculated by counting the number of positive and negative words in a comment. The authors compared the functioning of lexicon-based method with alternative machine learning approaches like Naïve Bayes and Decision Tree. They found that their lexicon-based approach performed better with respect to accuracy and F1-score.

Naresh and Krishna [8] performed sentiment analysis to classify twitter dataset of an airline into positive, negative and neutral using machine learning approaches. The analysis was carried out in three stages. The first stage involved data collection and preprocessing for noise removal. In the second stage, feature extraction was performed on the preprocessed data using an optimization technique in which larger dataset was split into smaller subproblems and the last stage classified the training dataset into the various classes. The combination of sequential minimal optimization and decision tree algorithms gave the best accuracy of 89.47% in comparison to other approaches.

A hybrid algorithm was recommended by Nasim et. al. [9] for sentiment analysis of students reviews using machine learning and lexicon-based methods. The two ML algorithms employed were SVM and Random Forest. The sentiment analysis model was trained using a combination of TF-IDF and lexicon-based features. A comparative analysis was carried out between the recommended model and alternative sentiment analysis models. The experimental outcomes reveal that the developed model gives better performance than the other methods, with respect to accuracy and F-measure, for sentiment analysis of student feedback.

B Deep Learning Techniques for sentimental analysis

Deep learning has been used regularly in the area of natural language processing (NLP) for various purposes that include sentiment analysis. Recently many people have suggested different deep learning architectures for these tasks, particularly for the analysis of student feedback.

One popular architecture uses convolutional neural networks (CNNs) for sentiment analysis. A study by Santos and Gatti [10] suggested a CNN-based technique for sentiment analysis of movie reviews and Twitter messages, showing that the model attained remarkable performance. W. Souma et. al. [11] have presented a very useful dimension in sentiment analysis in which the average stock price of a share is compared just before and after a news article related to the stock is released. The objective is to examine whether agents can forecast the variation in stock prices based on the sentiments of financial news articles. The DL model is trained using RNN alongwith long short-term memory (LSTM). It combines the Natural Language processing method with the DL hierarchical models for performing financial classification and forecasting. The NLP approach extracts news with positive and negative polarity. It uses two data sets, the first is TRNA and the other is the TRTH for DJIA 30 Index having duration from 2003 to 2013. On the average, it predicts negative news as negative and positive news as positive.

The sentiment analysis deep learning models that combine CNN and LSTM networks and support vector machines (SVM) are developed and validated in [12] on 8 review datasets and text-based tweets of various fields. A comparison is done between the hybrid models and three singular models, SVM, LSTM, and CNN. For the evaluation of each technique, both the reliability and computation time were taken into account.

In [13], an ensemble DL model has been developed by the authors for improving the sentiment analysis on Arabic tweets. It involved integrating the CNN and LSTM approaches for the necessary prediction. The model makes use of a word vector representation which is already trained and no other feature engineering is used here. It is shown to outperform the latest DL model on the Arabic Sentiment Tweets Dataset (ASTD). The approaches were compared on the basis of the F1-score performance metric.

III PROPOSED METHODOLOGY

A About the dataset

For this research, we have taken the dataset from the website RateMyProfessors.com, an online platform that helps educational organizations take feedback from students regarding their experience with their course teachers. It allows students to express opinions regarding the course content or any suggestion they would like to give to the administration for improving the quality of education. On the other hand, it allows teachers to look at the statistics generated from student ratings. It enables them to learn about the point of view of the students which helps them improve their teaching methodology.

The dataset contains 19000 comments along with the individual student rating and the average rating (See table 1).

Table 1: Student Ratings

Student Rating	Average Rating	Comments
5	4.7	Excellent professor. Hilarious, fun and good
1.6	1.5	Warning: By far, the worst online teacher. No communication skills, no desire to help
3.5	3.5	One word: DRY

The comments were categorized into three categories namely positive, neutral and negative based on student ratings ranging between 0 to 5. If the rating was higher than 3.5, it was rated as positive. If it was below 2.5, it was rated as negative and if it was between 2.5 and 3.5, it was rated neutral. In this manner, 11200 positive comments, 4400 negative comments and 3400 neutral comments were gathered. The positive comments were down-sampled and the rest of the categories were up-sampled. Models were trained and tested on this dataset and then were further tested on another dataset containing 100 comments collected within the classes at our institution.

B Proposed Framework

Figure 1 displays the framework that we adopted in this research. The sentiment analysis process, after data collection, was carried out in several stages.

The first stage involved preprocessing the text data by removing numbers and punctuation, then converting to lowercase, tokenizing, removing stop words and lemmatizing to make the data more consistent and noise-free.

In the next stage, we used Feature Extraction Techniques (FETs) such as Bag of words, global vector etc to obtain meaningful features from the student dataset. The tensor flow library was then applied to split the data into two groups for training and testing.

The experiment was further divided into three stages. Firstly, the machine learning techniques were employed for training and testing the model using the BoW, TF-IDF and word2vec features and their performance was recorded. In the second stage, the model was trained with a hybrid of machine learning algorithms and ensemble techniques and got evaluated again using the same three features. Finally deep learning architecture was used to train and test the model using word2vec and GloVe features. Their performance was evaluated against various measures like accuracy and F1-Score.

C Preprocessing

The student feedback is obtained in the English language as natural language. The machine learning model cannot process text input, so we must convert it into a format it can understand. Preparing the data is the first step in this process, as text data is challenging due to noise such as incorrect punctuation, slang, emoticons, and spelling mistakes. Words like ”BOO,” ”prof,” and ”tooooo much” can confuse the model, and common words like ”they,” ”you,” and ”he” are known as stop words, which don't convey any emotion. To improve the accuracy of the model, these stop words should be removed. The preprocessing steps applied to the dataset include:

- removing numbers and punctuation,
- converting characters to lowercase,
- tokenization,
- removing stop words,
- removing no comments,
- replacing contraction

For example, the following table 2 shows an example of original comment and the corresponding preprocessed comment:-

Table 2: Original and preprocessed comment

Original Comment	I overall enjoyed this class because the assignments were straightforward and interesting. I just did not enjoy the video project because I felt like no one in my group cared enough to help
Preprocessed Comment	overall enjoyed class assignments straightforward interesting not enjoy video project felt like no one group cared enough help

D Feature Extraction

In this research study, the preprocessed data was transformed into different feature sets using unigram and weighting schemes (BOW, TF-IDF, word2vec) as discussed below:-

D.1 Bag of Words (BoW)

Bag-of-words (BoW) [14] is a method of feature extraction for NLP that represents text as a set of words, where each word is a feature and the frequency of the word in the text is the feature value. In the case of unique words, they will be represented by a vector with a value of 1 in the corresponding entry in the vocabulary and 0 in the rest of the entries. For example, if the vocabulary is {'He', 'explains', 'the', 'concepts', 'very', 'well', 'but', 'not', 'algorithm'} and the document is ”He explains the concepts very well but not explains the algorithm”, the BoW for this document is {1, 2, 2, 1, 1, 1, 1, 2, 2, 1}.

The main benefit of using the BoW model is that it's simple to implement and computationally efficient. However, it does not consider the context or order of the words in the text. To overcome this, techniques such as n-grams and TF-IDF can be used.

D.2 Term frequency-Inverse document frequency (TFIDF)

Term Frequency-Inverse Document Frequency (TF-IDF) [14] is a feature extraction method for NLP that assigns a weight to each word in a document based on its frequency in the document and in the entire corpus of documents. The term frequency (TF) measures how often a word appears in a document whereas the inverse document frequency (IDF) measures how rare a word is across the entire corpus.

The TF-IDF weight for a word in a document is calculated as the product of its TF and IDF:

$$\mathrm{TF} - \mathrm{IDF} = \mathrm{TF}(\mathrm{word}, \mathrm{document}) \times \mathrm{IDF}(\mathrm{word}) \quad (1)$$

$$TF(word, document) = \frac{NOWD}{TWD} \quad (2)$$

Where

TWD = Total no. of words in the document

NOWD = No. of occurrences of word in the document

$$IDF(word) = \log\left(\frac{TND}{ND}\right) \quad (3)$$

where

ND = No. of documents containing the word

TND = Total no. of documents

One of the main benefits of using TF-IDF is that it can help to reduce the dimensionality of the feature space by giving less importance to words, such as stop words, and more importance to rare and informative words.

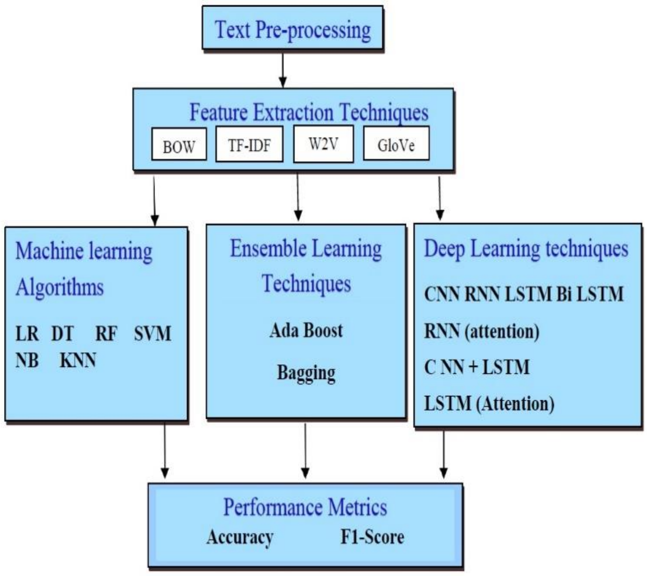

Figure 1: Proposed Framework

D.3 n-gram

In natural language processing, an n-gram [15] is a contiguous sequence of n words. When using n-grams as features, the goal is to capture the meaning of the text in a way that reflects the context in which the words appear. For example, the word ”play” can have different meanings depending on the context, and by including the preceding and following words in the form of bigrams or trigrams, it can be possible to capture the meaning more accurately.

n-grams are also useful in handling cases where the context of a word is essential, such as idioms, collocations, and phrasal verbs. For example, the bigrams ”kick the” and ”the bucket” would not have much meaning separately, but together they form the idiomatic expression ”kick the bucket” which means dying.

It's important to note that while n-grams can be useful in capturing context, they also increase the dimensionality of the data and the computational cost of processing it, so a trade-off needs to be made between capturing more context and the complexity of the model. For our research, we have used unigram as it proved to be more resourceful than the other combinations.

D.4 Global vectors

Global Vectors (GloVe) is a word embedding method developed by researchers at Stanford University that represents words as high-dimensional vectors in a continuous vector space. These vectors capture the meaning and context of words in a way that can be applied in NLP applications such as sentiment analysis, text classification, and machine translation.

The GloVe model learns word embeddings by training on a large corpus of text data using a technique called matrix factorization. The model takes a word-word co-occurrence matrix and counts the number of times each word appears in the same context as every other word in the corpus as input.

The model then factorizes this matrix into two lower-dimensional matrices, one representing the words and the other representing the context. The dot product of these two matrices results in the word vectors, which are used as the word embeddings.

D.5 Word2Vec

Word2Vec [16] is a word embedding technique that represents words as dense vectors in a continuous space, capturing the semantic meaning and context of words. It uses a neural network architecture to predict the context words given a target word, and the network weights are used as the word vectors.

There are two variants of word2vec: Continuous Bag-of-Words (CBOW) and Skip-Gram. CBOW predicts target words from their context words, while Skip-Gram predicts context words from a target word. Both models are trained on a large corpus of text, and the learned word vectors are then used in various NLP tasks such as text classification, sentiment analysis, and machine translation.

E Machine learning Algorithms

After feature extraction, the six conventional machine learning algorithms [17, 18], discussed below, were applied to build classification models based on these features.

E.1 Naïve *Bayes (NB)*

NB is infact an application of Bayes' theorem with the assumption that the features of a data point are independent of each other, hence the name Naïve. It is efficient in terms of computation time, making it a good choice for large datasets. For each data point, the algorithm calculates its probability of belonging to each class, and the point is finally assigned to the class with the highest probability. The different applications in which Naïve Bayes is successfully applied are sentiment analysis, text classification and spam filtering [19].

E.2 Decision Tree (DT)

It uses a tree-like structure where each node represents a feature and each leaf represents a prediction. The algorithm starts by choosing the feature that best separates the data into classes. It then recursively applies the same process to the subsets of data until a stopping criterion is met. Decision trees are easy to interpret, handle missing values and are suitable for multi-class problems [20].

E.3 Random Forest (RF)

It is a classification and regression method based on ensemble learning. It employs multiple decision trees that collectively make predictions. A different random subset of the data and a random subset of features is used for training each tree. The predictions of all trees is averaged to make final prediction. RF is famous for its good performance and capability to deal with data having multiple dimensions.

E.4 Logistic Regression (LR)

In this statistical method, a logistic function is used to model the association between the dependent and one or more independent variables. The predicted values are mapped by the logistic function to a probability $p \in [0,1]$. The data points are then classified into one of two classes using p. Logistic Regression is easy to interpret, fast to train and can handle non-linear relationships by transforming the independent variables [21].

E.5 Support Vector Machines (SVM)

The main function of this method is to find the hyperplane that maximally separates the data into classes. SVM is merited for handling non-linear relationships and high-dimensional data by using kernel functions. It is also effective in handling small datasets, as it is less prone to overfitting than other algorithms. SVM has been successfully used in various domains like bioinformatics, computer vision, and NLP [22, 23].

E.6 K-Nearest Neighbors (KNN)

The basic approach is to first identify the nearest K data points to a particular test point and then performs prediction by finding the majority class or average value of those K nearest points. KNN is effective in handling small datasets and is useful for problems with high-dimensional data, as it does not require explicit computation of feature weights. It is a good choice for problems where the relationships between features are not well understood or are non-linear [18].

F Ensemble Learning Models

Ensemble learning [24, 25] is a method where multiple supervised learning models work together to make a prediction. The idea is that the combination of these models will result in a more robust and accurate prediction. In the past, research has shown that ensemble learning can be effective in sentiment analysis. The current study uses two ensemble learning methods (AdaBoost and bagging) [18] with six supervised learning algorithms.

AdaBoost is an ensemble learning method that trains the base models sequentially and builds a new model at each round. Bagging (Bootstrapped Aggregating), an ensemble machine learning technique used to reduce the variance in the prediction of models by combining the output of multiple models. It works by training multiple models on different randomly selected subsets (with replacement) of the training data. The final prediction is typically made by taking the average or majority vote of the individual models' predictions.

G Deep learning Models

Deep learning [26] is a subfield of machine learning that focuses on using artificial neural networks with multiple layers (hence the term ”deep”) to learn complex representations of data. The key difference between deep learning and other forms of machine learning lies in the depth of the model and the way it learns from data. Unlike traditional machine learning algorithms that rely on manually crafted features, deep learning algorithms learn a hierarchical representation of the data through multiple non-linear transformations, allowing them to learn complex features and make highly accurate predictions automatically. Deep learning has proven to be highly effective for tasks such as natural language processing, speech and image recognition among others.

G.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) [27, 28] are a type of deep learning algorithms especially appropriate for image and video recognition applications.

A CNN comprises different layers including convolutional layers, pooling layers and fully connected layers. The convolutional layers involve learning the convolutional filters and applying them to the input data for feature extraction. The function of pooling layers is to reduce the dimensionality of the feature maps and to increase the robustness of the features to small translations and deformations. This allows the model to focus on the most important features and to ignore irrelevant or redundant information. The purpose of fully connected layers is to classify the input data using the learned features.

One of the unique characteristics of CNNs is that they use a technique called weight sharing, which allows them to learn translation-invariant features, meaning that they are robust to small translations and deformations of the input data.

G.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [29] are a type of deep learning algorithm that are particularly well-suited for sequence-to-sequence tasks, such as natural language processing, speech recognition, and time series forecasting. The basic idea behind RNNs is to introduce feedback connections in the neural network architecture, allowing the network to maintain information from previous time steps and use it to inform its predictions at the current time step.

This is particularly useful for sequence-to-sequence tasks, where the output at the current time step depends on the input and the output at previous time steps.

An RNN consists of a sequence of recurrent layers, each of which takes as input the current input and the hidden state from the previous time step. The hidden state is a vector that encodes the information that the network has learned so far and it is updated at each time step. The output of the RNN is generated by the final recurrent layer, which maps the hidden state to the final output. In RNN, the weights are updated in backpropagation through a large number of timesteps causing the problem of vanishing and exploding gradient.

G.3 Long Short-Term Memory

The basic idea behind LSTMs [29] is to introduce a set of memory cells and gates into the RNN architecture, allowing the network to selectively store and retrieve data from earlier time steps hence overcoming the problem of vanishing or exploding gradients that can occur in traditional RNNs. This allows them to better handle long-term dependencies and to make more accurate predictions. Another unique characteristic of LSTMs is that they can handle variable length sequences, unlike traditional feedforward neural networks which are designed to handle fixed-length inputs. This allows LSTMs to process sequences of dissimilar sizes without requiring the padding or truncation.

An LSTM network comprises a sequence of layers, containing a set of memory cells and gates. The gates monitor the information flow across the memory cells, enabling the LSTM to save and pick relevant information from previous time steps selectively. The LSTM network output is generated by the final LSTM layer, which maps the hidden state to the final output.

G.4 LSTM with attention mechanism

LSTM is a form of RNN architecture developed to tackle the problem of vanishing gradients in RNNs. An LSTM network comprises of memory cells, gates (input, forget, and output), and fully connected layers. The attention mechanism [30] is a method for selectively focusing on particular sections of the input sequence when performing predictions. It enables the model to measure the significance of each part of the sequence differently and to focus on the most relevant information.

In the context of Natural Language Processing (NLP), the attention mechanism can be used to weigh the importance of different words in a sentence and to focus on the most relevant words when making predictions. When combined with an LSTM network, the attention mechanism provides a way for the model to dynamically focus on the most relevant parts of the input sequence for each prediction. This allows the LSTM network to make predictions based on a weighted combination of information from different parts of the input sequence. It has been shown to significantly improve the performance of NLP models on various tasks such as machine translation and text classification. Combining LSTM with an attention mechanism provides a powerful tool for learning long-term dependencies in data sequences.

G.5 CNN combined with LSTM

Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks are two popular deep learning architectures used for text classification tasks. A combination of these two architectures, known as CNN-LSTM, has been shown to improve the accuracy of text classification models. The CNN-LSTM model applies convolutional filters to extract high-level features from the input text data and then passes the output through a sequence of LSTM cells to capture the temporal dependencies in the data. It allows the model to learn local and global features from the input text, leading to better classification performance.
Several studies have demonstrated the effectiveness of the CNN-LSTM model on various text classification tasks, including sentiment analysis, topic classification, and spam detection. For example, [31] used a CNN-LSTM model to classify sentiment in movie reviews, achieving state-of-the-art performance on the dataset.

G.6 RNN with an attention mechanism

Recurrent Neural Networks (RNNs) are a type of neural network commonly used for processing sequential data. One of the significant challenges in processing sequential data is the ability to capture dependencies between different parts of the sequence. One solution to this problem is the use of attention mechanisms [32].

Attention mechanisms are a type of mechanism that allows the model to focus on specific sections of the input sequence while making predictions. It is done by associating a weight to each input element which is a function of its importance to the prediction. The weights are learned during training and are used to calculate a weighted sum of the input elements, which is then used as the input to the next layer of the model.

In the case of RNNs with attention, the attention mechanism is used to compute a weighted sum of the hidden states of the RNN. It allows the model to selectively attend to specific parts of the input sequence, rather than simply relying on the final hidden state of the RNN.

It has been demonstrated that the adoption of attention mechanisms in RNNs has improved performance on a broad range of applications, including speech recognition, machine translation and image captioning. The ability to focus on specific sections of the input sequence allows the model to record complex dependencies between different parts of the sequence, leading to improved performance.

H Evaluation Metrics

In evaluating machine learning and deep learning algorithms, two common metrics used are classification accuracy (ACC) and F1-score. These measures have been widely employed to assess the performance of predictive models.

H.1 Accuracy

Accuracy [33] is a widely used evaluation measure in ML and DL. It is used to quantify the model performance in terms of the number of correct predictions it makes. It is defined as the ratio of no. of correct predictions (NCP) to the total no. of predictions (TNP). It can be calculated as follows:

$$Accuracy = \frac{NCP}{TNP} \qquad (4)$$

Accuracy is a simple and intuitive measure and is particularly useful when the classes in the target variable are balanced. However, it can be incorrect in situations where the classes are imbalanced, as a model can show high accuracy by simply predicting the majority class. In such cases, other measures such as F1-score, recall and precision should be used to better understand the performance of the model.

H.2 F1-Score

The F1-score [33] measures a model's accuracy that balances precision (Pr) and recall (Re). It is a harmonic mean of these two quantities which is calculated as follows:

$$F1 - Score = \frac{\Pr * Re}{\Pr + Re} * 2 \qquad (5)$$

where Pr is the ratio of true positive predictions to the total of true positive and false positive predictions with the model, and Re is the ratio of true positive predictions to the total of true positive and false negative predictions.

This score gives a balance among Pr and Re and is especially helpful when the classes in the target variable are imbalanced, or when there is an unequal cost of false negatives and false positives. In these cases, simply optimizing for accuracy can result in a model that prioritizes one type of error over the other, which may not be desirable.

IV Experiment And Result

After preprocessing the data and feature extraction, the experiment was conducted in three batches. Firstly, models were trained on six different supervised machine learning algorithms: three feature extraction techniques, namely BoW, TF-IDF, and word2vec, in combination with the unigram model. Labeling techniques like encoding were used to encode the sentiments of each comment. In the second batch, ensemble techniques (AdaBoost and Bagging) were used with a machine-learning algorithm to boost their performance. Five deep-learning architectures were deployed in the last batch with two word embedding techniques.

This section presents the results of classification accuracy and F1-Score, as produced by traditional supervised learning techniques and ensemble learning algorithms. Tables 3 and 4 represent the accuracy and F1-Score values obtained from machine learning algorithms for the given dataset.

Table 3: Accuracy for machine learning model

Algorithms	BOW	TF-IDF(unigram)	W2V
DT	75.65	74.70	76.26
RF	82.10	83.31	85.53
NB	71.00	76.55	
KNN	65.26	69.90	80.24
LR	74.37	76.00	66.66
DT (Ada boost)	78.00	77.72	87.33
RF (Ada boost)	89.93	84.23	90.12
SVM (Ada boost)	73.45	72.25	77.89
LR (Ada boost)	79.56	73.33	65.99
DT (bagging)	82.22	80.89	83.35
RF (bagging)	82.34	82.17	83.67
SVM (bagging)	81.11	82.12	82.21
KNN (bagging)	67.79	80.01	80.90
LR (bagging)	76.69	78.99	82.11

Table 4: F1-Score for machine learning model

Algorithms	BOW	TF-IDF(unigram)	W2V
DT	0.76	0.72	0.76
RF	0.82	0.83	0.85
NB	0.71	0.76	
KNN	0.63	0.70	0.80
LR	0.80	0.75	0.66
DT (Ada boost)	0.78	0.77	0.86
RF (Ada boost)	0.90	0.83	0.90
SVM (Ada boost)	0.75	0.72	0.77
LR (Ada boost)	0.79	0.73	0.66
DT (bagging)	0.78	0.83	0.83
RF (bagging)	0.83	0.83	0.83
SVM (bagging)	0.81	0.82	0.82
KNN (bagging)	0.67	0.80	0.80
LR (bagging)	0.75	0.79	0.82

A Machine Learning

This study analyzed the classification accuracy of various supervised learning algorithms and ensemble methods when applied to three different configurations obtained from three text representation schemes of the dataset. The algorithms studied were Random Forest (RF), Naïve Bayes (NB), Support Vector Machines (SVMs), Logistic Regression (LR), K-Nearest Neighbours (KNN), and Decision Tree (DT). Results showed that RF achieved the highest accuracy, followed by DT and SVMs. The word2vec with the vector size of 100 and 200 had the best accuracy, followed by BoWs in second place and unigram features with TF-IDF weighting coming in third. The study found that word embedding models outperformed classic feature extraction techniques. However, RF applied with BoW almost gave the same accuracy as with word2vec.

The study also looked at the impact of ensemble methods, including AdaBoost, and Bagging, on the predictive performance of the supervised learning methods. First, a base ML algorithm like decision tree, logistic regression, or support vector machine is chosen. Next, multiple bootstrap samples of the training set are created by randomly selecting training examples with replacements. Each of these bootstrap samples is given equal weight, and a base model is trained on the weighted training set. The error rate of the base model on the training set is then computed. In AdaBoost, the weights of the misclassified examples are increased, while the weights of the correctly classified examples are decreased. It puts more emphasis on the examples that are difficult to classify. Training a base model, computing the error rate, and updating the weights of the training examples are repeated for a fixed number of iterations. Finally, the outputs of the base models are combined by weighing them according to their error rates, and the resulting ensemble model is evaluated on a test set. The results showed using ensemble methods improved the accuracy of the supervised learning algorithms. The highest accuracy among the configurations studied was obtained by the AdaBoost ensemble of RF, with a classification accuracy of 90.12%. In summary, RF performs better than other machine learning algorithms when combined with AdaBoost for text classification because it reduces the variance of the model, while AdaBoost reduces the bias. This combination allows the resulting model to make accurate predictions on high-dimensional and noisy datasets.

Overall, the results of this study provide valuable insights into the accuracy of various supervised learning algorithms and ensemble methods when applied to text corpus configurations. The findings can be helpful for practitioners in choosing the most appropriate machine learning algorithm for their text classification task, as well as for researchers in developing new approaches for text classification.

B Deep Learning

This research used deep learning-based sentiment analysis on a text-based dataset. As mentioned in the previous section, two word embedding schemes, word2vec and GloVe, were used to represent the textual comments. Six deep learning architectures were considered, including CNN, RNN, bidirectional RNN-AM, LSTM, CNN combined with LSTM, and bidirectional LSTM for processing the text data.

These deep learning architectures were implemented and trained using different libraries such as TensorFlow and Keras. The optimal performance for each model was obtained through hyperparameter optimization based on Bayesian optimization with a Gaussian process. Almost 80% of the data was used as the training set and validation set, while the rest was used as the testing set. In the case of word2vec, both continuous skip-gram and CBOW methods were considered, with varying dimensions and sizes of vectors of projection layers. The general structure of deep learning-based sentiment analysis is summarized in Figure 1.

Table 5 present the classification accuracy obtained by the seven deep learning approaches.

The results from the empirical analysis showed that, for the text corpus, the GloVe word embedding scheme outperformed the alternate word embedding schemes. The word2vec skip-gram model obtained the worst performance for prediction. The results indicated that the best predictive performance was achieved with vector size equal to 300 and a value of 300 for dimension projection layer. The highest predictive performance among the deep learning architectures was achieved by CNN combined with cascaded layers of LSTM followed by the bidirectional RNN-AM. These findings can contribute to the advancement of deep learning-based sentiment analysis in text analytics.

Table 5: Accuracy for deep learning techniques

Algorithms	Vector-Size	W2Vec	Glove
CNN	200	80.13	82.13
RNN	200	84.02	84.22
LSTM	200	73.63	75.63
Bidirectional-LSTM	200	80.22	80.22
Bidirectional-LSTM-with-Attention-mechanism	200	87.52	87.52
CNN-with-LSTM	200	89.45	90.22
CNN	300	79.87	81.99
RNN	300	85.06	85.06
MLP	300	78.09	78.09
Bidirectional-LSTM	300	80.03	81.99
Bidirectional-RNN	with		
Attention-mechanism	300	-	90.76
Bidirectional-LSTM-with-Attention-mechanism	300	83.31	87.35
CNN-with-LSTM	300	88.07	91.29

The number of epochs is a hyperparameter that determines the number of times the model is to be trained on the dataset. Generally, the more epochs a model is trained for, the better it will perform on the training data. However, training for too many epochs may cause overfitting, which is the situation in which the results from the model are very good on the training data but poor on unseen, new data.

Figure 2 represents training and validation accuracy; we see a steep rise in training accuracy, whereas testing data accuracy improved slowly but surely. As seen in Figure 2, training accuracy reaches 1, but the validation accuracy reaches up to 92%. The reason for the halted progress can be seen in Figure 3 as training loss reduces, validation loss becomes stagnant, indicating that the model is overfitted to the training data.

After experimenting with three batches, the model that provided the best accuracy was chosen to predict sentiments from one hundred comments collected within university premises. In this case, CNN with LSTM cascaded architecture was chosen. It assigned probabilities to all three classes, and the class with the highest probability was selected as the predicted sentiment. When a sentence was passed to the model, it went through preprocessing steps first and then outputted the predicted sentiment (positive, negative, or neutral) based on the highest probability in the final output vector.

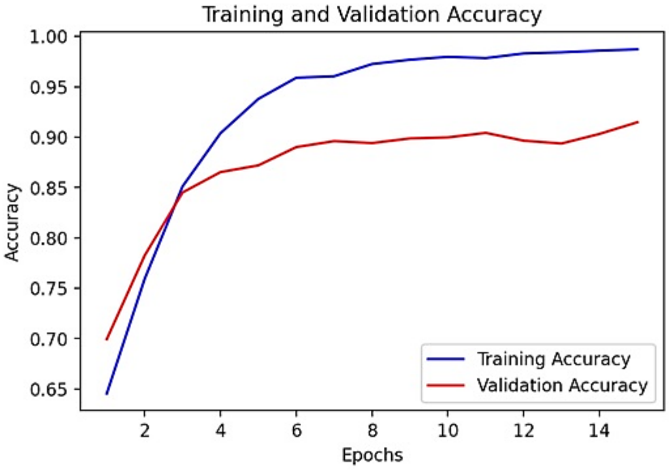

Figure 2: Training and Validation Accuracy

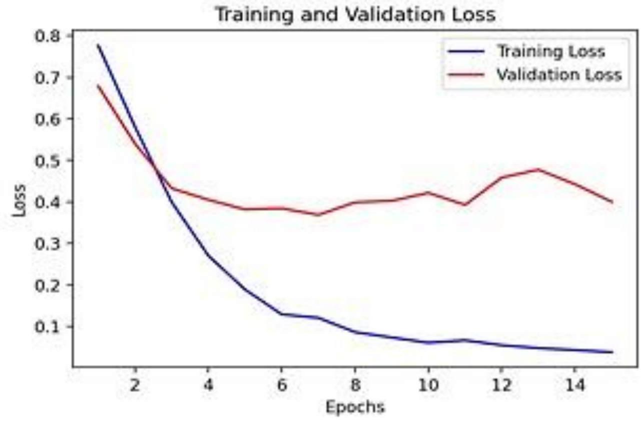

Figure 3: Training and Validation Loss

C Discussion

The study evaluated various machine learning techniques including standard classifiers, ensemble methods, and deep learning models. Results showed that ensemble methods generally have better performance over traditional classifiers, while deep learning models outperformed ensemble techniques. The highest classification accuracy of 91.29% was achieved by using a cascaded Neural Network of CNN combined with LSTM, followed closely by Bi-directional Recurrent Neural Network Mechanism (RNN-AM) with a Global Vectors (GloVe) word embedding representation.

The experimental outcomes demonstrate that deep learning frameworks have the ability to produce noteworthy results in education for tasks related to machine learning and data mining. It is worth noting that conventional machine learning techniques produced similar results and consumed less computational resources and time than deep learning architecture. Using other embedding techniques may produce better results, and the possibility of outperforming other deep learning techniques presides.

The dataset used in the empirical analysis was gathered from Ratemyprofessors.com, where most contributors are from instructors and schools in the USA. This specific dataset used in the study may impact the results of machine learning and data science. Most of the data set consisted of positive comments due to which we had to down sample the comments as it affected our accuracy.

This is considered a constraint of the research. However, the method used in the study applies a machine learning-based methodology for sentiment classification and can be applied in other languages for sentiment analysis with proper preprocessing.

V Conclusion

This paper presents a text-mining method for analysing instructor evaluation reviews. A dataset of 20,000 reviews was collected for the study. Out of which, 1000 were discarded for not providing helpful information. The analysis consisted of evaluating various machine learning algorithms, including conventional supervised methods (such as Decision tree, Naïve Bayes, Logistic Regression, Support Vector Machines, Random Forest and k-Nearest Neighbors), ensemble learning techniques (AdaBoost, and Bagging), and DL models (CNN, RNN, Bidirectional RNN with Attention Mechanism, Bidirectional LSTM and CNN combined with LSTM). The study utilized two conventional text representation techniques (BOW and TF-IDF) with conventional machine learning algorithms and two word embedding approaches (word2vec and GloVe) with deep learning models. According to the results, the methods based on deep learning outperformed ensemble and supervised learning methods for sentiment classification.

Amongst the different configurations considered for comparison, the highest accuracy of 91.29% was obtained using a cascaded CNN combined with an LSTM in combination with a GloVe based word embedding representation.

References

[1] E. Cambria, D. Das, S. Bandyopadhyay, A. Feraco, et al., "A practical guide to sentiment analysis," 2017.

[2] J. Zhao, K. Liu, and L. Xu, "Sentiment analysis: mining opinions, sentiments, and emotions," 2016.

[3] A. Onan, "Mining opinions from instructor evaluation reviews: a deep learning approach," Computer Applications in Engineering Education, vol. 28, no. 1, pp. 117–138, 2020.

[4] M. Avinash and E. Sivasankar, "A study of feature extraction techniques for sentiment analysis," in Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS 2018, Volume 3, pp. 475–486, Springer, 2019.

[5] R. S. Jagdale, V. S. Shirsat, and S. N. Deshmukh, "Sentiment analysis on product reviews using machine learning techniques," in Cognitive Informatics and Soft Computing: Proceeding of CISC 2017, pp. 639–647, Springer, 2019.

[6] S. Kumar, M. Gahalawat, P. P. Roy, D. P. Dogra, and B.-G. Kim, "Exploring impact of age and gender on sentiment analysis using machine learning," Electronics, vol. 9, no. 2, p. 374, 2020.

[7] K. Z. Aung and N. N. Myo, "Sentiment analysis of students' comment using lexicon based approach," in 2017 IEEE/ACIS 16th international conference on computer and information science (ICIS), pp. 149–154, IEEE, 2017.

[8] A. Naresh and P. Venkata Krishna, "An efficient approach for sentiment analysis using machine learning algorithm," Evolutionary intelligence, vol. 14, pp. 725–731, 2021.

[9] Z. Nasim, Q. Rajput, and S. Haider, "Sentiment analysis of student feedback using machine learning and lexicon based approaches," in 2017 international conference on research and innovation in information systems (ICRIIS), pp. 1–6, IEEE, 2017.

[10] C. Dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers, pp. 69–78, 2014.

[11] W. Souma, I. Vodenska, and H. Aoyama, "Enhanced news sentiment analysis using deep learning methods," Journal of Computational Social Science, vol. 2, no. 1, pp. 33–46, 2019.

[12] C. N. Dang, M. N. Moreno-Garc´ıa, and F. De la Prieta, "Hybrid deep learning models for sentiment analysis," Complexity, vol. 2021, pp. 1–16, 2021.

[13] M. Heikal, M. Torki, and N. El-Makky, "Sentiment analysis of arabic tweets using deep learning," Procedia Computer Science, vol. 142, pp. 114–122, 2018.

[14] G. Hackeling, Mastering Machine Learning with scikit-learn. Packt Publishing Ltd, 2017.

[15] Y. HaCohen-Kerner, D. Miller, and Y. Yigal, "The influence of preprocessing on text classification using a bag-of-words representation," PloS one, vol. 15, no. 5, p. e0232525, 2020.

[16] K. Kalaivani, S. Uma, and C. Kanimozhiselvi, "A review on feature extraction techniques for sentiment classification," in 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), pp. 679–683, IEEE, 2020.

[17] O. Theobald, Machine learning for absolute beginners: a plain English introduction, vol. 157. Scatterplot press London, UK, 2017.

[18] M. Kubat and J. Kubat, An introduction to machine learning, vol. 2. Springer, 2017.

[19] K. M. Leung et al., "Naive bayesian classifier," Polytechnic University Department of Computer Science/Finance and Risk Engineering, vol. 2007,pp. 123–156, 2007.

[20] S. B. Kotsiantis, "Decision trees: a recent overview," Artificial Intelligence Review, vol. 39, pp. 261–283, 2013.

[21] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, Applied logistic regression, vol. 398. John Wiley & Sons, 2013.

[22] I. Steinwart and A. Christmann, Support vector machines. Springer Science & Business Media, 2008.

[23] D. Meyer and F. Wien, "Support vector machines," The Interface to libsvm in package e1071, vol. 28, p. 20, 2015.

[24] X. Dong, Z. Yu, W. Cao, Y. Shi, and Q. Ma, "A survey on ensemble learning," Frontiers of Computer Science, vol. 14, pp. 241–258, 2020.

[25] O. Sagi and L. Rokach, "Ensemble learning: A survey," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, p. e1249, 2018.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, pp. 436–444, 2015.

[27] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," IEEE transactions on neural networks and learning systems, 2021.

[28] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al.,

"Recent advances in convolutional neural networks," Pattern recognition, vol. 77, pp. 354–377, 2018.

[29] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," Physica D: Nonlinear Phenomena, vol. 404, p. 132306, 2020.

[30] W. Zheng, P. Zhao, K. Huang, and G. Chen, "Understanding the property of long term memory for the lstm with attention mechanism," in Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2708–2717, 2021.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in neural information processing systems, vol. 30, 2017.

[32] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," Neurocomputing, vol. 452, pp. 48–62, 2021.

[33] H. Dalianis and H. Dalianis, "Evaluation metrics and evaluation," Clinical text mining: secondary use of electronic patient records, pp. 45–53, 2018.

AUTHORS

First Author – Urooj Abdul Haleem, BE (Computer Systems), Master of Engineering (Computer Systems) student at NED university,

Second Author – Syed Zaffar Qasim, PhD (Computer Systems Engineering), Department of Computer & Information Systems Engineering, NED University,

Correspondence Author – Syed Zaffar Qasim, PhD (Computer Systems Engineering),