Multi-class Sentiment Analysis on Online Drug Reviews Using Different Deep Learning Techniques

|  |  |  |
| --- | --- | --- |
| Rezaul Haque1,  Department of Computer Science and Engineering,  East-West University,  Dhaka 1212, Bangladesh,  [rezaulh603@gmail.com](mailto:rezaulh603@gmail.com) | Katura Gania Khushbu2,  Department of Computer Science and Engineering,  East-West University,  Dhaka 1212, Bangladesh,  [katurakhushbu@gmail.com](mailto:katurakhushbu@gmail.com) | Sazzad Hossain Rony3  Department of Computer Science and Engineering,  East West University,  Dhaka 1212, Bangladesh,  [sazzad.cse050@gmail.com](mailto:sazzad.cse050@gmail.com) |
| Saddam Hossain Laskar4  Department of Computer Science and Engineering,  East West University,  Dhaka 1212, Bangladesh,  [sazzad.cse050@gmail.com](mailto:sazzad.cse050@gmail.com) |  |  |

# **Abstract**

With the rise of the internet, user-generated content from social networking sites has emerged as a key driving factor, including a wealth of information about medications, diagnoses, treatments, and, disorders. Customers' opinions regarding previously used medicines are contained in the data in the form of comments, which can be used to detect crucial adverse drug reactions. This information can be used to get significant insights using machine learning approaches like sentiment analysis. It is frequently impossible for a potential buyer to review all of the comments before making a purchase choice. Drug assessments are beneficial to both health-care professionals and the general public since they give important medical information. Assessing opinions toward various aspects of drug evaluations can provide important insights, support decision-making, and improve public monitoring systems by revealing collective experience. Another key issue is the unstructured and linguistic nature of the evaluations, which finds it challenging for users to categorize comments into useful insights. Previous research has used both machine and deep learning algorithms to perform categorization on drug reviews, with deep learning classifiers outperforming machine learning classifiers. So, using appropriate NLP and ML algorithms, our key objective in this study is to acquire a higher categorization score than earlier research studies. We achieved our aim by applying SA on medicine reviews to detect positive, negative, and neutral user comments training five machine learning algorithms on two distinct feature extraction and four deep learning classifiers on two different word embedding approaches. Our results show that out of all ML algorithms, random forest trained on count vectorizer surpass previous study results, with accuracy and F1 score of 96.65% and 96.42%, respectively. Obtained results of ML algorithms were further improved using BiLSTM model trained on GloVe embedding obtaining an accuracy and F1 score of 97.40% and 97.42%, respectively.

# **Introduction**

The emergence of Web 2.0 has made it possible for the internet to become a more participatory platform for its users, there are now large volumes of user-generated content in  social networking and online shopping websites[1]. The data growth of these platforms has been phenomenal, and the significant influence that it has on users and their families is being investigated by an increasing number of sectors in order to acquire knowledge into their user community and, as a corollary, to drive change. Sentiment Analysis (SA) have been used extensively to help mine user-generated content across several domains such as products[2], restaurants[3], health and movies[4].

Massive amounts of online data created from user comments are assessed autonomously in the pharmaceutical and healthcare industries in order to obtain helpful insights on the efficacy and side effects of medicines. These insights are obtained from the evaluation of large volumes of online data. The industry manufacturers use these information to enhance the performance of their pharmacovigilance systems by identifying issues faster, compare the online reputation of brands, post marketing drug surveillance and provide safe drugs without any side effects. Moreover, the detection of consumers suffering from adversarial drug reaction can result in saving many lives which can be achieved by mining online user reviews towards a particular brands or drugs. The early indication of adverse drug reactions is also necessary for risk management purposes as financial and reputational losses can be faced upon issuing lawsuit. People post their thoughts about the effectiveness or side effects of drugs on different online forums and due to the immense growth of reviews it has become challenging task to extract accurate sentiments using sentiment analysis. Moreover, on the medical/healthcare domain the predicted results of sentiment analyzer need to be more accurate than other domains. As a consequence of this, over the course of the years, a significant amount of research on sentiment analysis on drug reviews has been carried out in to gain an understanding of the patient's level of satisfaction regarding factors such as contentment, surroundings, accessibility, the cordiality of the staff, and the effectiveness of the proceeding. By automatic classification of positive and negative reviews of consumers new or unknown adverse drug effects can also be identified.

Subjectivity analysis is one type of analysis, and one type of subjectivity analysis is known as sentiment analysis[5]. Sentiment analysis examines the feelings included in a text with the intention of understanding the sentiment polarities of opinions towards various aspects of a subject [6]. It is still regarded as a highly difficult challenge due to the fact that user-generated material is expressed in a variety of different and intricate ways utilizing natural language. Unfortunately, due to privacy and ethical concerns, this method is not frequently used in the medical field [7]. Drug related posts in medical forums, are mostly conversational in nature, and are thus representative of prevailing public opinion. Sentiment analysis on medication reviews have seen substantial development over the course of the years, progressing from fundamental principles to more complex machine learning approaches like as deep learning. DL is one of the most renowned technologies utilized in a variety of SA tasks. In the work of [8] authors, present a drug recommender system that predicts sentiments into binary class using machine learning models trained on different feature extractors. In [9] authors, performed binary-class (positive and negative) and multi-class (positive, neutral, and negative) sentiment classification using a number of different machine learning techniques to assess the amount of efficiency possessed by a certain medication. In addition, fuzzy-rough feature selection technique was utilized by authors of [10] to train ML classifier to predict multi-class sentiments on drug reviews. Most of the research works have been done to perform binary or multiclass classification on a dataset collected from UCI ML repository (drugs.com) [11]. Previous research works have commonly used the dataset to train different feature extractors such TF-IDF, Word2Vec, and BoW on traditional ML algorithms to predict sentiments into positive-negative or negative-neutral-positive. However, their models suffered from low accuracy score due to the imbalance distribution of class, high dimensionality feature vector, and lack of preprocessing techniques. Authors of [12] solved the low classification score problem achieving 93% accuracy score on multi-class; training Artificial Neural Network (ANN) model on Count Vectorizer (CV) feature extractor. The study suggests that adaptation of DL classifiers can result in a significant classification score along with performance for the task of drug SA. Similar works can be seen done by the authors of [13] where several combinations of DL models namely CNN, LSTM, BiLSTM, and BERT was trained to predict on drug reviews into 3 classes.

However, DL classifiers has some drawbacks such as requirement of large amount of data and human intervention, high computational cost, highly parameter sensitivity, and hard to debug [14], [15]. On the other hand, ML models requires less computational cost and human intervention than DL classifiers[16]. Previous studies on drug SA with high accuracy score was achieved by DL models and performance of the classification score of their model still requires significant improvement. The performance of the ML classifiers highly depends on text preprocessing and feature extraction techniques. So, it is also important to select adequate text cleansing operation and feature extractor out of the many for the task of drug SA. In addition, there is no real-world web application that can automatically classify sentiments of drug reviews which can be very useful for pharmaceuticals and health care experts.

Therefore, the primary objective of this research is to enhance the accuracy scores of earlier research works by employing both ML and DL algorithms that have been trained on different feature extraction and word embedding strategies. Our contributions toward the accomplishment of the objective are as follows:

* Collected 215063 drug reviews from drugs.com website and labeled the reviews into three classes: positive, negative, and neutral.
* Performed text preprocessing using Python’s NLTK package to reduce noisiness from the text and enhance the text quality.
* Trained five different ML algorithms on two feature extractor such as TF-IDF and CV. The ML algorithms trained in this study are: Random Forrest (RF), Support Vector Machine( SVM), Passive Aggressive (PAG), Logistic Regression (LR), and Stochastic Gradient Descent (SGD).
* Trained four DL algorithms on two word embedding techniques such as keras and GloVe. The DL algorithms trained in this study are: LSTM, Bidirectional LSTM (BiLSTM), GRU, and Bidirectional GRU (BiGRU).
* Presented a comparative analysis of performance of the classifiers on the basis of accuracy, error analysis, and models performance.
* Built a real-world web application based on the highest performing classifier that can predict drug reviews into three classes using Flask framework.

The paper is organized as Sect. 2, related works, Sect. 3 contain the methodology of the research which consists of data collection and labeling, utilized text preprocessing, feature extraction techniques and training parameters of ML models, and opted evaluation metrics, Sect. 4 contain result analysis and discussion , In the final Sect. 5, we conclude our research work providing the limitations and potential future works.

# **Related Works**

There has been significant effort put into utilizing ML/DL algorithms to discern the feelings of user evaluations, which coincides with a dramatic increase in the improvement of AI. Restaurant, ecommerce, and other industries are frequently utilizing SA to understand opinions of consumers to grow their business. In spite of the widespread use of sentiment analysis across a wide variety of application areas, the pharmaceutical domain has gotten a significantly smaller amount of attention. On the other hand, a many developments have been recorded in the more recent literature. This is due to the relevance of mining medication reviews, which might contribute to a variety of different healthcare stakeholders.

Many research studies implemented drug SA using different preprocessing and feature extraction techniques. Unfortunately, there is a lack of research work on drug SA with great accuracy score due to the inappropriate preprocessing and feature extraction techniques, and unsuited training parameters. Due to the lack of ground truth dataset most of them used dataset of drugs.com and trained ML/DL algorithms to perform either binary (positive and negative) or multiclass (positive, neutral, and negative) classification. The study in [8] presents a drug recommendation system based on ML algorithms such as LR, Perceptron, Ridge Classifier, Multinomial Naïve Bayes, SGD, and SVM where binary classification is performed to identify sentiments in positive or negative. To train ML models they used four feature extractors and their LR model resulted in highest accuracy score of 91%. On the similar dataset authors of [10] proposed fuzzy-rough feature-selection based ML model to classify sentiments into three classes. They used BOW and TF-IDF to train naïve bayes, random forest, decision tree, and riper models where random forest with TF-IDF obtained the highest accuracy of 67%. In [17] authors proposed a linguistic approach for drug SA on multiclass dataset which was collected from WebMD. Their approach outperforms two types of SVM model with an accuracy of 69%, exceeding the score to 7%. Authors of [18] investigated the effect of sentiment analysis features in detecting adversarial drug reaction from online posts. They created dataset from twitter and DailyStrength and performed binary classification to achieve 80% accuracy score.

In recent years DL algorithms have emerged as the most popular technique to work with drug SA. In [12] authors trained both ML and DL models with different feature extractors such as TF-IDF, Count Vectorizer, and Word2Vec to classify drug sentiments into multi-class. On the testing data their ANN model obtained the highest accuracy score of 93.85% with count vectorizer. Authors of [13] did a similar work where comparison of several DL classifiers performance on multi-class drug SA is presented. They trained CNN, LSTM, BiLSTM, BERT, and combination of these models on Word2Vec word embedding, achieve the highest F1 score of 0.90 with combined model of BERT and LSTM. These two papers have obtained significant classification on drug SA task with DL algorithms but their ML models resulted in poor performance on testing data.

It is abundantly obvious from the published works that machine learning models of earlier works do not have a substantial accuracy score on ML algorithms. Therefore, it is essential to devise appropriate methods for enhancing the performance of the models in order to meet the demands of the field. And we believe with the proper selection of preprocessing, feature extraction, and ML model the results of previous results can be improved. According to our best knowledge, there is no research on multi-class SA on drug reviews obtained from drugs.com that have a great accuracy score using ML algorithms.

# **Methodology**

## **Data collection**

This section contains a description of the experimental dataset for the drug reviews SA problem. It is referred to as the Drug Review Dataset, and it was obtained from the UCI repository. The dataset includes user evaluations on various medications, as well as ailments that are associated with certain medications, and a range of ratings from 0 to 10 stars that indicate the level of pleasure felt by users. The dataset also contains drug’s name, condition of patient, useful count, and number of people found the review helpful. As was noted before, each drug review is given a rating on a scale that ranges from 0 to 9, with 0 representing the least satisfied patients to 9 representing the most satisfied patients. We categorized the dataset with three classes according to the three levels of polarity based on the review's rating: negative (class 0; rating less than 4), neutral (class 1;  rating  greater than 4 and less than 7) and positive (class 2; rating greater than 7). Figure 1 showcases the distribution of labels for each class. The final dataset contains 215063 drug reviews where there are 105433, 100071, and 9559 number of reviews categorized into positive, negative , and neutral class respectively. Since ML models cannot comprehend labels in text form, we required to convert the class labels into numerical format and Sklearn’s Label Encoder package was utilized to numeric encode the classes.

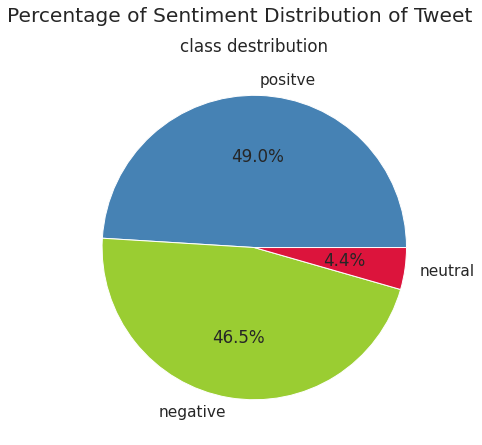


Figure 1. Number of reviews for each class.

|  |  |  |  |
| --- | --- | --- | --- |
| Class | Total Reviews | Total Words | Unique Words |
| Positive | 105433 | 4145737 | 30752 |
| Negative | 100071 | 4319347 | 31418 |
| Neutral | 9559 | 230376 | 8561 |

## **Text preprocessing**

Text preprocessing is a highly important phase in the process of sentiment analysis since the text that is gathered from the internet typically contains a great deal of noise in the form of adverts, HTML elements, scripts, punctuation, and white space. Data preprocessing is done to eliminate the incomplete, noise and inconsistent data[19]. By removing all of them, we were able to cut down on the amount of noise in the text, which eventually led to an improvement in the effectiveness and precision of the classification models that were applied to the text. We also carried out some special data cleaning steps like stop words removal and digit removal. Text normalization was also performed in this study, where English contractions were expended, lemmatization and spell correction were utilized to reduce dimensionality of the data. The techniques were performed with the help of Python’s NLTK and RegEx library. The noise removal and text normalization processes require a great deal of caution since they might occasionally result in the loss of a small number of rows from the dataset, which in turn leads to a reduction in accuracy. Figure 2 illustrates the word cloud of most frequently occurring tokens of each classes after performing all the preprocessing steps discussed above.

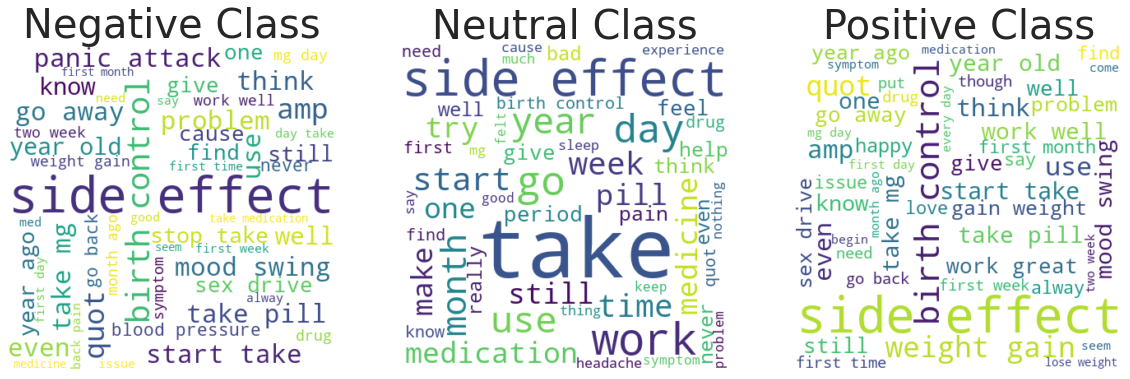


Figure 2. Word cloud of most frequently occurring terms for each class of reviews.

## **Feature extraction**

ML/DL algorithms cannot process textual data and it requires the strings in the forms numerical input to perform classification[20]. In order to transform reviews into feature vector for ML classifiers, we implemented two feature extraction techniques: TF-IDF vectorizer and count vectorizer using Sklearn’s feature extraction library. On the other hand, DL classifiers were trained on two distinct word embedding techniques such as keras and GloVe embedding.

### **TF-IDF Vectorizer**

The TF-IDF algorithm is a well-known technique for determining the significance of a given word in relation to a given document. When determining the term frequency of a certain term, the ratio of the number of times that term appears in a text to the total number of words in a document is used as the calculation [21]. IDF, which stands for "Inverse Document Frequency," is a method that determines the significance of a phrase. There are several words and phrases, such as "the," "a," "but," etc., that are used frequently yet do not have any significant meaning. The formula for calculating IDF is as follows: IDF (t) = log(N/DF), where N is the total number of documents and DF is the total number of documents that include the phrase t. The transformation of information from its narrative structure into a vector space can be done more effectively using TF-IDF. This enables us to locate phrases inside a text that are essential and carry a lot of weight. The following equation 1-3 are the formulae for TF-IDF, where i refers to the word and j refers to the document.

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |
|  | (3) |

The characteristics of the text that will be used by supervised machine learning algorithms will be derived from N-Gram. These are n tokens taken sequentially from the text that was provided. It is possible for the value of n to be 1, 2, 3, 4 and so on. It is referred to as a unigram if the value of n is considered to be one, a bigram if n is equal to two, a trigram if n is equal to three, and so on. In this study, TF-IDF was applied with the help of Sklearns feature extraction library where the input was cleaned reviews, maximum feature was set to 12,000, and N-gram range was set to unigram and bigram.

### **Count Vectorizer**

The top-frequency terms are chosen for the vocabulary via the Count Vectorizer method, which then builds a sparse representation of the texts across the vocabulary. The documents are transformed into a matrix that contains token counts by using the Count Vectorizer. To begin, the documents are segmented, and then a dense matrix is constructed based on the frequency with which each token appears in the texts. In order to construct the matrix, we first filter out all of the document collection's stop words. The next step is to clean up the vocabulary by removing any terms that are found in less than four different texts. In this method, any phrases that are used insufficiently will be eliminated. In this particular research study, CV was applied to the cleaned reviews utilizing a unigram and bigram N-gram range with a default maximum feature parameter set.

### **Word Embedding**

The innovation of word embedding has highly improved the performance of DL classifiers for the task of text classification[22]. Word embedding is an improved version on bag-of-words that maps tokens of textual data into a dense vector representation of words. It can extract the relative and semantic information from the text a collection of statistical language modeling techniques and provide a continuous vector space for DL classifiers to work on [23]. DL classifiers take the data into input, hidden, and output layer to extract meaningful features from the text. The embedding layer is fed onto the input layer of the model to learn from the vectorized words. For the task of text classification there are different type of word embedding techniques and among them Word2sequence and GloVe was utilized in this experiment.

Initially, we employed word2sequence embedding technique which enables a text corpus to be vectorized by turning each text into a sequence of integers. This is achieved by using Keras’s Tokenizer and padding function where each integer is the index of a token in a dictionary. In this experiment, the dataset was tokenized with a vocabulary size of 43,000, embedding dimension of 200, and padding size of 100. We set the vocabulary size to be 43,000 tokens cause the dataset contained 42565 unique tokens and embedding dimension was set to a low number of 200 cause higher dimension provides computational difficulties for DL classifiers. We maintain the length of the text sequence's equal length by padding zeros at the end because the length of the comments varies (post-padding).

We also employed pretrained GloVe embedding, an unsupervised learning approach that can generate word representations for global vectors. GloVe is simply a log-bilinear approach with a scaled least-squares purpose that was developed on a 6 billion token corpora. The corpus was built with Wikipedia2014 and Gigaword5, with a vocabulary of the top 400,000 most frequently occurring terms and a context window size of 10. In this experiment, we used word vectors with dimensions of 300, vocabulary sizes of 43,000 tokens, and padding sizes of 100 to initialize GloVe embeddings.

## **Training Baseline Models**

In ML studies baseline model functions to contextualize the results of trained models that lacks complexity and may have little predictive power. Selecting a baseline model is necessary to make a strong comparison between the performance of the classifiers. In our study, highest performing ML classifier was selected as the baseline model which was RF trained on CV. Before training the ML model, we need to separate the dataset into training and testing set. After performing feature extraction, we break the whole dataset into fractions of 80% and 20% for the training sample and testing set procedure. We then trained our model with training set that corresponds of labeled 172044 drug reviews and tested the performance on testing set that consists of unlabeled 43011 drug reviews. Description of the opted ML models are as follows:

### Random Forrest

The classification technique that consists of many decision trees is referred to as the Random Forest Classifier. Randomness is used by the algorithm to construct each individual tree in order to encourage the growth of statistically independent forests[24]. These forests are then utilized to draw correct conclusions based on the predicting abilities of the forest. In order to put the RF into action, there are two main parameters that need to be configured: the number of trees to be used as estimators (n estimators), and the kind of criteria to be applied. According to the findings of a number of research [25], it is possible to attain good outcomes by sticking with the system's default settings. However, the enormous number of trees will produce a consistent outcome of varying relevance, as stated in [26]. In addition, [27] claimed that utilizing a greater number of trees than what is required may be superfluous, but that this does not make the model less accurate. Both the n estimator parameter (which was set to 100), as well as the criteria parameter (which was set to entropy), went through a series of tests and evaluations in this study in order to identify the best possible RF model for classification.

### Support Vector Machine

This is an effective approach that can serve either the objective of regression and classification. A hyperplane is drawn to demarcate the separate categories. This approach performs exceptionally well with regression, and the influence of the SVM is more pronounced as the dimension space becomes more complex. Additionally, SVMs function admirably even when the number of observations is far higher than the dimension size. One disadvantage is that it does not work very well with large datasets. According to authors [28], the radial basis function (RBF) kernel of the SVM classifier is widely implemented and demonstrates good result. In order to put the SVM method into practice, we relied on the RBF kernel. When implementing the SVM classifier with an RBF kernel, there are two criteria that need to be specified. These are the optimal parameters of cost (C), and the kernel width parameter. Adjusting the degree to which non-separable training data are stiff can be accomplished by manipulation of the C parameter, which determines the maximum magnitude of misclassification that can be tolerated. The refining of the form of the class-dividing hyperplane may be affected by the kernel width parameter, which is denoted by. Raising the value of C may result in the model being overfit, but higher the amount of will change the geometry of the hyperplane that divides classes which could influence the classification's accuracy. When we trained the SVM model, we set the kernel type to be RBF and provided the cost parameter a value of 1.

### Passive Aggressive (PAG)

PAG algorithms are a class of ML algorithms that are frequently utilized in software designed for working with large amounts of data [29]. The algorithm is frequently used for numerous kinds of large-scale and online learning. In online ML methods, the input data arrives in the order that it was requested, and the machine learning model is updated in the same order. This is in contrast to standard batch learning, which uses the full training dataset simultaneously [30]. PAG model was trained with the default parameters, and the maximum number of iterations was set to 200.

### Logistic Regression (LR)

A logistic function is used to describe a binary dependent variable in the most fundamental version of the statistical model known as logistic regression [31]. However, many more complicated extensions of this model do exist. This widely used approach for classifying data is a member of the Generalized Linear Models subclass. Logistic regression is a statistical technique that may be used to predict the probabilities that describe the results of an experiment. This technique is also known as the Ensemble Learning algorithm. We trained the LR model using the cost parameter set to 1, the maximum iteration set to 200, the tolerance set to 0.001, and the liblinear solver settings.

### Stochastic Gradient Descent (SGD)

Commonly used in neural networks, SVMs, and LR, the following algorithm is well adapted to permit discriminative training of linear models under convex loss functions. [32]. The SGD is one of the most common ML methods for model optimization. The SGD algorithm is a more advanced version of the gradient descent method. This algorithm is referred to as a stochastic estimation of the gradient descent optimization method because it uses an interpolation of gradient rather than an actual gradient by arbitrarily subsampling the entire training set of data. The approach is widely used due to the fact that it has a high efficiency and is simple to construct for datasets that contain redundant observations[33]. In this work, LR is utilized as a loss function in modeling, and the maximum number of iterations that are used is 200.

## **Training DL Models**

We splited our dataset into 75% training, 5% validation, and 20% testing set where the training and validation set will be used to train LSTM, BiLSTM, GRU, and BiGRU models. The testing set will be used to make predictions on the model after training in order to compare the effectiveness of the classifiers. However, all of these classifiers have their own advantages and disadvantages. The DNN network is highly sensitive to parameters and it is necessary to find out proper parameters to train them.

There are numerous RNN-based model variations that work well with sequential input data, including audio, music, text, name entity recognition, etc. However, due to the vanishing and exploiting gradient problem, it often does not perform well in long-term dependence. [34]. Since the network's half that is closest to the output is updated and the other half that is farther away from the output is improperly updated. That's why LSTM, a particular kind of RNN that can learn long-term dependencies, was developed. They may develop a future model based on past and present data by learning from past data, which is accomplished by incorporating a number of gates into their network design to recall past data. As a result, the input values are only traversed once ( i.e. from left to right, input to output). Additionally, GRU have been utilized recently in order to alleviate the shortcomings of standard RNNs with large text[35]. The advantage of the GRU model is that it may be used to decide how much information should be remembered from previous steps, how much should be handed back, and how much should be received in the current synthesis steps. Through these gate controls, GRU is able to learn long texts effectively. Consequently, the RNN networks with memory operation are obviously more appropriate for the task of text classification. But LSTM is hard to train fast and accurately as it takes a lot of resources [36]. In Bi-LSTM model the given input data is utilized twice for training (i.e., first from left to right, and then from right to left) which makes up for the shortcomings of LSTM by learning from the previous information of the current word, fully considering the semantic features between contexts, and acquiring more comprehensive features and feature information. The Bi-GRU, which merges the "forget" and "input" gates into a single update gate, is a minor modification of the Bi-LSTM. Along with making various adjustments, it blends the hidden state and cell state.

In our research, we trained these models on word2sequence and glove embedding separately. Therefore, Figure 7 represents our used deep neural networks which was implemented using Keras. With the training and validation data the models are individually trained with batches of size 128. The model is configured to train, at most, with 100 epochs. However, to avoid overfitting an early stopping method which monitors the validation accuracy per epoch with factor of 0.1, patience of 2, minimum delta of 0.0001 and minimum learning rate of 1e-06 was incorporated that monitors the accuracy of the model are embedded within the training stage. As it’s a multiclass problem the loss function used for training the models is the “sparse categorical cross entropy” and the used optimizer is “Adam” with the learning rate 1e-02 and epsilon of 1e-08 provided. Then the model is evaluated on the 20% test data to make prediction on three classes of drug reviews.

## **Evaluation**

The performance of our proposed models was evaluated based on their accuracy, precision, and recall scores, as well as their F1 score. Equation 4-7 is used to calculate the traditional evaluation metrics.

|  |  |
| --- | --- |
|  | (4) |
|  | (5) |
|  | (6) |
|  | (7) |

In Equations (4), (5), and (6), the number of data identified as positive among the data labeled as positive is referred to as the true positive (TP), and the number of data classed as negative among the data labeled as negative is referred to as the true negative (TN). A false negative (FN) is the number of data that were supposed to be negative but were really labeled as positive in the dataset, while a false positive (FP) is the number of data that were supposed to be positive but were actually marked as negative in the dataset. Accuracy is defined as the number of properly predicted cases divided by the total number of predictions provided by the model; however, any approach might perform differently in relation to the number of instances that are correctly classified, as seen in Equation (4). According to Equation, precision is measured as the percentage of documents that were correctly identified as positive by the model (5). As indicated in Equation, recall is the proportion of documents that were categorized as positive by the model out of the total number of documents that really had positive tags (6). In addition, the F1-score is the overall average of the recall and accuracy scores, as seen in Equation (7). In addition, an error analysis was performed with the help of a confusion matrix and a ROC-AUC curve.

# **Results Analysis and Discussion**

## **Performance of ML model**

Table 1 displays the findings of the baseline models performed on the drug reviews. Both the TF-IDF and Count vectorizer feature extractors demonstrated that the RF model had superior performance to the others in terms of evaluation metrics. We also could observe that the models trained with count vectorizer outperformed models trained TF-IDF. Models trained with count vectorizer resulted in an excellent accuracy score where each model can be seen performing at the same level of evaluation scores. Compared to the count vectorizer, models trained with TF-IDF showcases that only RF can reach a significant classification score, whereas rest of the algorithm sufferers from low accuracy score. In addition, SGD model can be seen performing worst for both the feature extractors.

TABLE 1. Results of all the classifiers trained on TF-IDF and count vectorizer.

| **Feature** | **Models** | **Evaluation Metrics** | | | |
| --- | --- | --- | --- | --- | --- |
| **Accuracy** | **Precision** | **Recall** | **F1** |
|  | RF | 96.65% | 96.60% | 96.13% | 96.42% |
| CV | LR | 96.04% | 96.94% | 95.54% | 96.24% |
| PAG | 95.57% | 94.56% | 95.51% | 95.48% |
| SVM | 95.33% | 94.46% | 95.61% | 95.52% |
|  | SGD | 95.27% | 94.86% | 95.41% | 95.37% |
|  | RF | 93.29% | 92.39% | 94.59% | 93.69% |
| TF-IDF | SVM | 90.65% | 89.99% | 90.11% | 89.77% |
| PAG | 90.39% | 90.00% | 90.90% | 90.41% |
|  | LR | 89.62% | 89.81% | 86.99% | 89.63% |
|  | SGD | 86.01% | 87.02% | 85.82% | 86.54% |

Based on the Figure 3, we were able to deduce that the CV approaches did, in fact, have an effect on the performance of any classifier. The accuracy of models that were trained using the Count vectorizer was greater than that of models trained with other techniques on a variety of datasets. In addition, the RF model is capable of achieving the highest classification scores for any of the feature extractors.

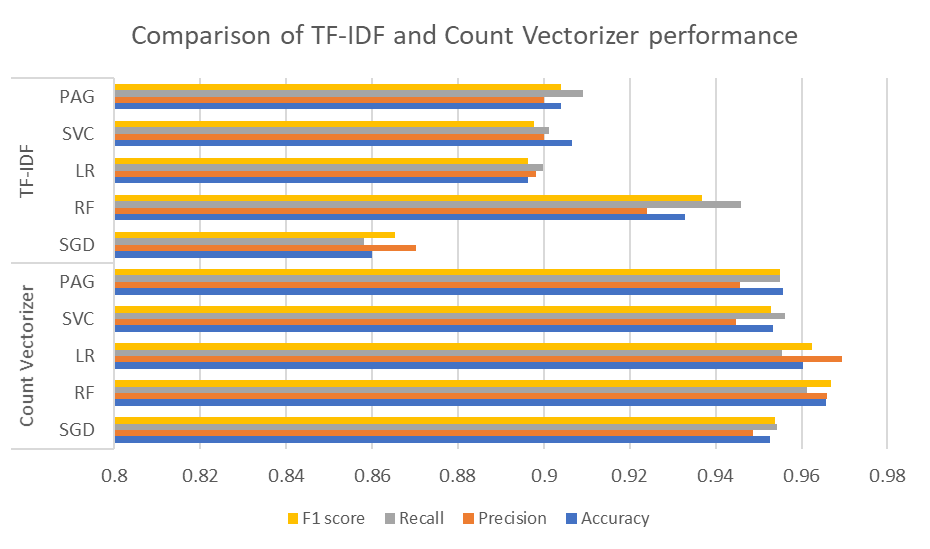


Figure 3. Results of count vectorizer and TF-IDF for each algorithm

We notice that out of the classifiers trained on TF-IDF, RF outperforms others with an accuracy and F1 score of above 93%. SVM, PAG, and LR performs equally well on the testing data with an accuracy score between 89% to 91%. And lastly, SGD classifier performed the worst with the lowest accuracy and F1 score of 86% and 86.5%, respectively. For models trained on CV, RF model beat other models in terms of all experimental assessment parameters, with an accuracy score of 96.65% and an F1 score of 96.42%, respectively. LR can be seen performing equally well to RF with an accuracy score of 96%. PAG and SVM also performed well on the testing data obtaining accuracy and F1 score above 95.3%. Finally, SGD model achieved the lowest score among the models with an accuracy score of 95.27%. However, all the classification score of the models indicates that each algorithms have performed equally well on CV features.

When classification is performed, the ROC Curve, which also goes by the name ROC-AUC, is utilized. This curve illustrates the balance that must be struck between sensitivity and specificity. Before we can generate the ROC curve, we need to first determine the True Positive Rate (TPR) and the False Positive Rate (FPR) for each of the different thresholds. For every threshold, the values of the FPR and the TPR are shown along the x and y axis, respectively. A classifier is meant to offer places along the diagonal where FPR and TPR are identical in order to serve as a starting point for further analysis. The test's precision increases proportionately with the distance the curve travels toward the 45-degree diagonal of the ROC space. This field will take values between 0 and 1, with 0 indicating a classification of the class that is entirely erroneous and 1 indicating a classification that is completely accurate. Figure 4(a) and 4(b) shows the AUC-ROC curve for RF models trained with TF-IDF and count vectorizer, respectively. The figure 4(a) indicates that the RF model trained with TF-IDF has a mean AUC score of 97% where label 0 (negative), and label 1 (neutral), and label 2 (positive) obtains AUC score of 99%, 92%, and 99%, respectively. On the other hand, figure 4(b) indicates that Count vectorized RF achieves a mean AUC score of 98% where negative and positive class corresponds of 99% AUC score and the neutral class outperforms the performance of TF-IDF vectorized RF’s neutral class with an AUC score of 96%. Both the AUC ROC curve of figure suggests RF model trained with count vectorizer can perform with great AUC score for all the classes compared to the TF-IDF vectorized model.

|  |  |
| --- | --- |
|  |  |

Figure 4. ROC-AUC curve of highest performing classifier trained on TF-IDF and count vectorizer.

## **Performance of DL classifiers**

Table 2 displays performance of DL classifiers trained on GloVe and Keras embedding based on opted evaluation metrics. From the table we can see, BiLSTM model trained on GloVe embedding was the best performing algorithm, outperforming our baseline and all other DL classifiers. The results of Table 2 showcases that DL classifiers trained on GloVe embedding can achieve a significantly higher score than the models trained on keras embedding. On the other hand, all the models trained on keras embedding failed to beat the classification scores of baseline model where, LSTM performed the worst among all the DL classifiers.

TABLE 2. Results of DL classifiers trained on GloVe and Keras embedding.

| **Feature** | **Models** | **Evaluation Metrics** | | | |
| --- | --- | --- | --- | --- | --- |
| **Accuracy** | **Precision** | **Recall** | **F1** |
|  | BiLSTM | 97.40% | 97.01% | 97.68% | 97.42% |
| GloVe | LSTM | 97.20% | 97.08% | 95.66% | 97.34% |
| GRU | 95.28% | 95.46% | 95.51% | 95.48% |
| BiGRU | 95.21% | 94.82% | 95.32% | 95.26% |
|  | BiGRU | 95.09% | 95.09% | 94.09% | 94.59% |
| Keras | BiLSTM | 92.30% | 92.30% | 82.94% | 87.37% |
| GRU | 92.21% | 92.21% | 82.95% | 87.34% |
|  | LSTM | 91.20% | 91.20% | 80.18% | 85.34% |

The Figure 5 compares the performance of DL classifiers trained on keras and GloVe embedding based on the traditional evaluation metrics. Overall, it is clear that BiLSTM and BiGRU have the highest percentage of classification score in all evaluation measurements for GloVe and Keras embedding, respectively. Meanwhile, BiGRU (GloVe) and LSTM (Keras) have the overall lowest percentage in this DL classifier result. As per the table, the Glove feature has the highest percentage in the evaluation metrics than the Keras feature, which has 97.40% Accuracy, 97.01% Precision, 97.68% Recall, and 97.42% F1 respectively, but BilSTM placed second in Keras embedding. In terms of the percentage of BiGRU of Keras, it covers 95.09% Accuracy, 95.09% Precision, 94.09% Recall, and 94.09% F1 individually, which are less than BiLSTM from GloVe. Interestingly, BiGRU encountered the lowest percentage evaluation in the GloVe. LSTM comes in second place which is inaugurated with more than 97% Accuracy, Precision, and F1, and obtained 95.66% in Recall at the GloVe embedding. However, LSTM from Keras positioned last on its evaluation. Singly, GRU stands third in both Glove and Keras but the fact tends GRU(Keras) failed to achieve even 90% when GRU(GloVe) got more than 95% in each evaluation segment.

Figure 5. Comparison of performance of DL classifiers trained on keras and GloVe embedding

Figure 6 showcases the results of all the DL models trained on keras embedding. For the models trained on keras embedding, the best accuracy, precision, recall, and F1 measure values are achieved with the use of BiGRU algorithm. The following highest F1 score is 94.59%, and the recall and precision scores are 95.09% and 94.09%, respectively. BiLSTM and GRU models obtained almost similar accuracy score of 92% but they encountered a low recall and F1 score of less than 83% and 88%, respectively. Lastly, LSTM model performed the worst with the lowest classification score where the accuracy and precision score were 91.2% and F1 score of 85.34%. The following discussion indicates that the highest performing DL model BiGRU trained on keras embedding could not achieve a better result than our baseline model.

Figure 6. Performance of DL classifiers on Keras embedding

The results of each DL classifiers trained on GloVe Embedding are displayed in the Figure 7. The BiLSTM and LSTM classifier beat others classifier in terms of all evaluation criteria with accuracy score of 97.40% and an F1 score of 97.42% respectively. LSTM can be seen performing equally well to BiLSM with an accuracy score of 97.20% and f1 score 97.34%. GRU also performed good enough by obtaining accuracy 95.28% and F1 score above 95.48%. Finally, BiGRU classifier achieved the lowest score among the models with an accuracy score of 95.217%. Our classification scores of BiLSTM model have outperformed the performance of baseline classifier, increasing the accuracy, recall, precision, and F1 scores to 0.75%, 0.41%, 1.55%, 1%, respectively.

Figure 7. Performance of DL classifiers on GloVe embedding

## **Error Analysis**

The confusion matrix is one of the approaches that is both insightful and easy method for measuring the accuracy and completeness of a machine learning system. Its primary application is in classification jobs, particularly those in which the results may include two or more types of classes. The technique was utilized to perform error analysis on the highest performing model trained on each feature extraction techniques. With the help of confusion matrix, we tried to find meaningful insights about the ML/DL model results.

### **Error Analysis of ML Models**

Figure 8(a) and 8(b) shows the confusion matrix of highest performing model trained with TF-IDF and count vectorizer, respectively. In Figure 8(a), we can see that the confusion matrix of RF reveals that the model is able to forecast positive and negative classes of drug reviews better than neutral class reviews. This is based on the testing data of 43011 drug reviews. Out of the 1922 neutral class reviews the model can accurately predict 1396 as neutral and incorrectly predict 255 and 271 reviews as negative and positive, respectively. From figure 8(b) we can see, Count vectorized RF model performed significantly well on positive and negative class compared to the TF-IDF vectorized RF model, as 20425 positive and 19457 negative classes are correctly predicted out of the 20972 positive and 20117 negative reviews, respectively. In addition, neutral class can also be seen achieving a significant true positive rate as 1428 classes are correctly classified out of 1922 reviews. Both the models are encountering problem of inaccurate prediction of neutral class. However, RF model trained with count vectorizer performed better at classifying all the three classes with great accuracy and less wrong predictions compared to the TF-IDF vectorized RF model. On the 20% test data the model makes wrong predictions of 3.2%, 25.7%, and 2.6% for negative, neutral, and positive class.

|  |  |
| --- | --- |
|  |  |

Figure 8. Confusion matrix of highest and lowest performing classifier trained on TF-IDF vectorizer.

### **Error Analysis of DL Models**

Confusion matrix of highest performing model trained on keras (BiGRU) and GloVe (BiLSTM) embedding is illustrated in figure 9(a) and 9(b), respectively. We considered 20% testing data which corresponds of 43013 reviews to generate the confusion matrix. In figure 9(a), we can see that like baseline line models, BiGRU trained on keras embedding can predict the negative and positive class with great accuracy but makes high number of wrong predictions for neutral class. Out of the 20,042 negative, 1899 neutral, and 21072 positive labels the BiGRU models accurately predicts 18945, 1744, and 20213 labels, respectively. Furthermore, the evaluation metrics indicates that the model could not beat the results of baseline model. Confusion matrix of GloVe+BiLSTM model shown in figure 9(b) indicates that the negative and positive class is more accurately predicted than Keras+BiGRU model, but performs comparatively poor on neutral class. The model makes wrong predictions of 2%, 13.3%, and 2.1% for negative, neutral, and positive class, respectively. Compared to the Keras+BiGRU model, the following model increases the accurate predictions to 3.47% and 1.97% with 19627 negative and 20620 positive labels. In addition, the model could not achieve a better performance on neutral class with 6% increase of wrong predictions. However, due to the ability of accurately making predictions on negative and positive reviews than others, the model outperforms baseline models performance.

|  |  |
| --- | --- |
|  |  |

Figure 9. Confusion matrix of highest performing classifier trained on Keras and GloVe embedding

Figure 10(a) and 10(b) illustrates the validation accuracy and validation loss for each model trained with GloVe embedding for 100 epochs. From the figure 10(a), we can see the validation accuracy of BiLSTM and LSTM model is increasing with the increment of epochs. Though, BiLSTM performed slightly well with better validation accuracy scores for each epoch. Furthermore, BiGRU performed the worst throughout the epochs with lowest validation accuracy scores. The validation loss of each model for 100 epochs is shown in figure 10(b), which shows that the loss of each model is decreasing until 45th epoch and after that it starts to overfit with an increase validation loss. BiLSTM and LSTM model continued the whole process with lower validation loss compared to the other algorithms. GRU and BiGRU did not perform well as its validation loss decreases until 45th epoch but the loss is comparatively higher and the loss kept going high for the rest of the epochs.

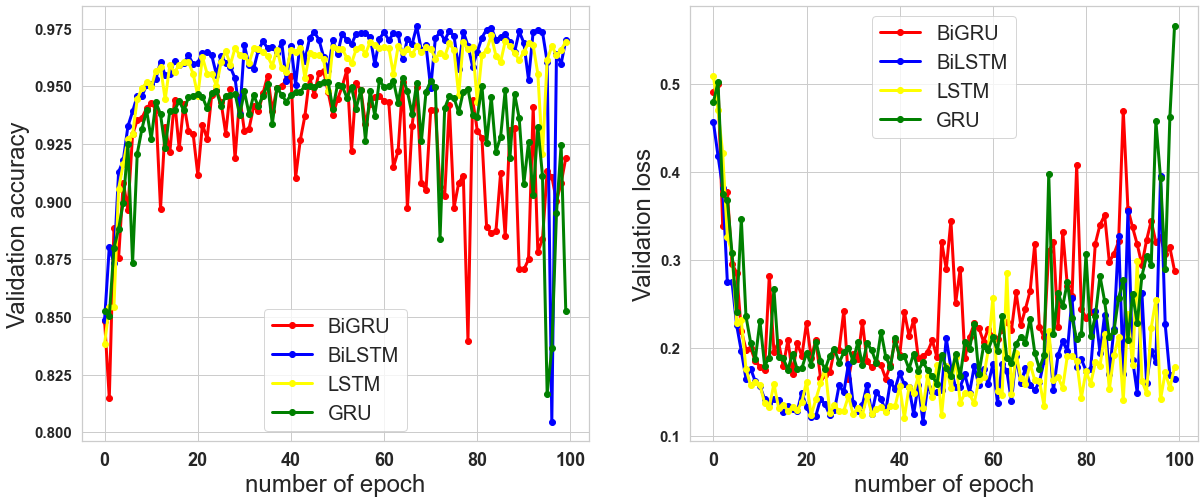


Figure 10. Comparison of validation accuracy and loss for each epochs of all the classifiers trained on GloVe embedding

## **State of the art comparison**

Finally, we present a state-of-the-art comparison between the results of us to the prior research works conducted on drug SA.As discussed in the related works section, most of the research works have utilized DL classifiers to obtain a great accuracy score. And their studies lack significant classification score using ML classifiers which is computationally fast compared to DL models. In this study, we outperformed the results of previous research works [8], [12], [13] on multi-class drug SA conducted on drugs.com dataset. Table 3 indicates that with the use of ML algorithms we have obtained a better accuracy score than prior works on DL algorithms.

TABLE 3. State of art comparison

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference | Dataset | Approach | Number of class | Accuracy |
| **Our approach** | **Drugs.com** | **ML** | **3** | **96.6%** |
| **DL** | **97.4%.** |
| [12] | Drugs.com | DL | 3 | 93.8% |
| [8] | Drugs.com | ML | 2 | 93% |
| [13] | Drugs.com | DL | 3 | 90.4% |

## **Web Application Development**

We also built a web application based automatic drug review categorization using the highest performing ML model. Flask framework was utilized to make the application due to its scalability, lightweight features, and availability of python libraries for the task of SA. Figure 11 shows the result page of the web application where a user provides review of a drug and the count vectorized RF model predicts the text as a neutral class.

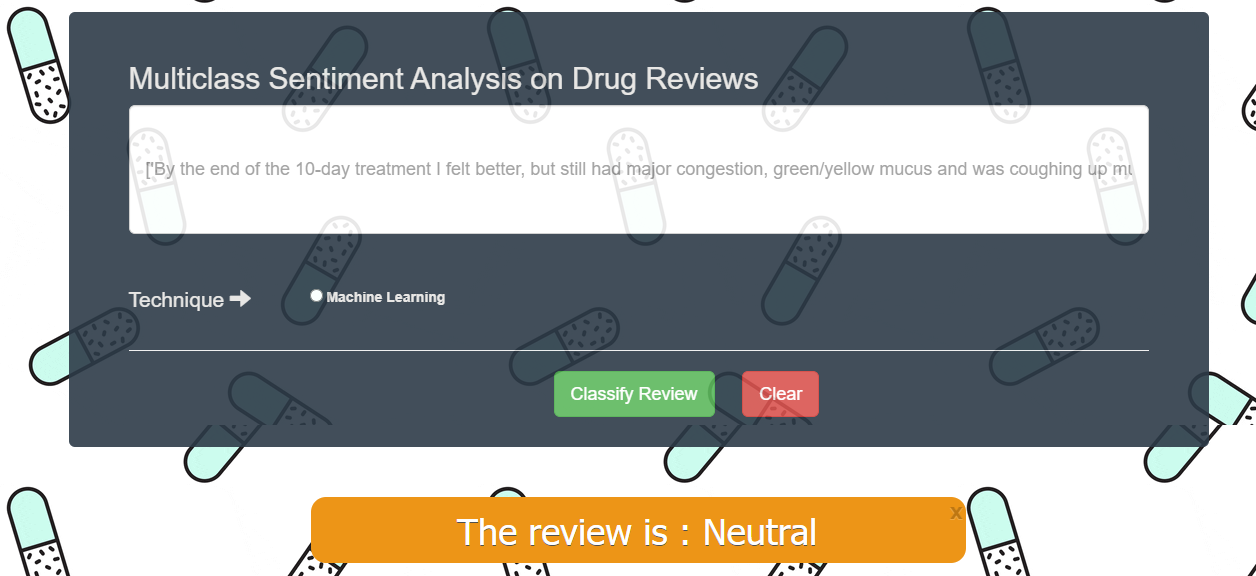


Figure 11. Web application on drug SA

# **Discussion**

This paper presents a comparative performance analysis ML/DL algorithms trained on different feature extraction and word embedding techniques to detect drug sentiments into three classes: positive, neutral, and negative. Since the dataset used in this study had lower number of neutral reviews, ML models encountered difficulties to predict the class accurately. On the other hand, DL algorithms trained on GloVe predicted the neutral class more accurately, exceeding the accuracy score of ML models. Though we obtained better results than previous research works but the score can still be improved by increasing the neutral reviews in the dataset.

Previously many research has been done on drug SA using both ML and DL algorithms where DL classifiers resulted in significant accuracy score but ML models suffered from poor accuracy score. So, in this article, we firstly focused on exceeding the classification score of previous research works on drug SA using only the ML algorithms combined with several NLP techniques. Out of the two feature extractors, RF model trained on count vectorizer obtained the highest accuracy and F1 score of 96.65% and 96.42%, respectively. Our RF model outperforms the highest performing DL algorithms of previous research on drug SA [12]. Text classification presents challenges that are well suited for RF classifiers due to the large dimensionality and noise of the data. Since the dataset we used in this experiment was imbalanced, a greater classification score can be achieved with RF than with other methods thanks to the majority vote on the predictions made by all of the decision trees in the forest. It is also worth to mention that all the classifiers trained with Count vectorizer have performed equally well and outperformed the results of prior research works. Utilization of several preprocessing techniques cleaned the reviews and reduced the dimensionality of the data. As a result, with the reduced dimension of data all the ML algorithms reached a significant accuracy score. In addition, throughout the research we found out that models trained with count vectorizer can obtain greater accuracy than TF-IDF vectorized models for the task of multi-class sentiment classification on drug reviews. The more straightforward and honest manner in which CV represents the words of the evaluations is the primary contributor to its superior performance. While CV mostly indicates the total number of words that are included in the reviews, TFIDF primarily represents the importance of the words that are present in the reviews. The TFIDF encoding performs worse than the CV encoding because it conceals the complete contextual etymology of the text inside the review.

Both BiLSTM and LSTM models trained on GloVe embedding outperformed the performance of count vectorized RF with an accuracy and F1 score of over 97%. The BiLSTM model can process inputs both forward and backward in time due to the BiLSTM processing chain, which replicates the LSTM processing chain. By including a second hidden layer, BiLSTM improves upon the unidirectional LSTM by enabling hidden-to-hidden interconnections to transmit in the opposing temporal sequence. As a result, the model can make use of data from both the present and the future. For sentiment classification issues, it is beneficial for a model to be aware of both the past and the future contexts. The method enables BiLSTM to take the future context into account. Additionally, without keeping duplicate context information, its layer learns bidirectional long-term dependency between time steps in time series or sequence data. When we want the network to learn from the full time series at each time step while simultaneously having access to contextual data, these dependencies are essential. As a result, it proved to be an excellent performance for our study. However, compared to other classifiers, the BiLSTM model has the drawback of requiring more training data and time.

# **Conclusion**

It is highly significant to examine the feelings of reviews via the use of AI technology in our day and age, which is characterized by the fast growth of Internet technology and social networks. When we go out to do anything, be it go shopping, make an online buy, or go to a restaurant, we first look at the reviews to ensure that we are making the best choice. Reviews are becoming an increasingly important part of our everyday lives. The utilization of drug reviews is possible to offer light on the knowledge of users' preferences and drug experiences, which can be exploited to assist with making decisions by healthcare experts and promote health.

However, previous research works on drug SA using ML suffered from poor accuracy score and motivated by this, in this research SA on drug reviews was studied using several ML (RF, SVM, LR, PAG, SGD) and DL (LSTM, BiLSTM, GRU, BiGRU) classifiers. Firstly, we trained the ML models with two different feature extraction techniques such as TF-IDF and Count vectorizer to identify which algorithm can result in better classification score. Secondly, two distinct word embedding techniques namely keras and GloVe embedding was used to train DL classifiers to further improve the results and overcome the limitations of ML models. To evaluate the result of the algorithms, traditional evaluation metrics like accuracy, precision, recall, and f1 score was used and error analysis was carried out using confusion matrix and ROC-AUC curve. Our results show that out of the ML algorithms, RF trained on count vectorizer can obtain the highest accuracy score of 96% outperforming the results of previous research works. Furthermore, models trained with count vectorizer outperformed the results of TF-IDF vectorized model due to blatant openness in representing the words of the reviews. For the DL classifiers, BiLSTM model trained on GloVe can be seen outperforming the results of ML classifiers with an accuracy and F1 score of 97.40% and 97.42%, respectively. Because it can deal with forward-backward dependencies from feature sequences, resolving gradient disappearance, and long-term dependence, BiLSTM performs better because it can extract pertinent information from lengthy reviews more effectively. We also built a web application based on the count vectorized RF model that can automatically categorize drug reviews into three classes.

Because language representation models are becoming increasingly popular for use in classification problems, we intend to include them into our future work alongside variational autoencoders and adversarial networks as part of our semi-supervised analysis strategies. In this manner, we intend to investigate other approaches in the hope of decreasing our reliance on annotated corpora. In addition, in order to increase the level of fine-tuning that can be applied to our strategies, we will investigate the use of semantic characteristics in addition to contextual ones.

# **Reference**

[1] R. K. Atkinson, K. Sabo, and Q. Conley, “The participatory web,” *Handbook of Technology in Psychology, Psychiatry and Neurology: Theory, Research, and Practice*, pp. 91–120, Aug. 2012, doi: 10.1080/1369118X.2012.665935.

[2] X. Fang and J. Zhan, “Sentiment analysis using product review data,” *Journal of Big Data*, vol. 2, no. 1, pp. 1–14, Dec. 2015, doi: 10.1186/S40537-015-0015-2/FIGURES/9.

[3] L. Li, L. Yang, and Y. Zeng, “Improving Sentiment Classification of Restaurant Reviews with Attention-Based Bi-GRU Neural Network,” *Symmetry 2021, Vol. 13, Page 1517*, vol. 13, no. 8, p. 1517, Aug. 2021, doi: 10.3390/SYM13081517.

[4] R. Parveen, N. Shrivastava, and P. Tripathi, “Sentiment classification of movie reviews by supervised machine learning approaches using ensemble learning voted algorithm,” *2nd International Conference on Data, Engineering and Applications, IDEA 2020*, Feb. 2020, doi: 10.1109/IDEA49133.2020.9170684.

[5] A. Montoyo, P. Martínez-Barco, and A. Balahur, “Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments,” *Decision Support Systems*, vol. 53, no. 4, pp. 675–679, Nov. 2012, doi: 10.1016/J.DSS.2012.05.022.

[6] M. Wankhade, A. C. S. Rao, and C. Kulkarni, “A survey on sentiment analysis methods, applications, and challenges,” *Artificial Intelligence Review*, pp. 1–50, Feb. 2022, doi: 10.1007/S10462-022-10144-1/TABLES/7.

[7] K. Denecke and Y. Deng, “Sentiment analysis in medical settings: New opportunities and challenges,” *Artificial Intelligence in Medicine*, vol. 64, no. 1, pp. 17–27, May 2015, doi: 10.1016/J.ARTMED.2015.03.006.

[8] S. Garg, “Drug recommendation system based on sentiment analysis of drug reviews using machine learning,” *Proceedings of the Confluence 2021: 11th International Conference on Cloud Computing, Data Science and Engineering*, pp. 175–181, Jan. 2021, doi: 10.1109/CONFLUENCE51648.2021.9377188.

[9] M. N. Uddin, M. F. bin Hafi, S. Hossain, and S. M. M. Islam, “Drug Sentiment Analysis using Machine Learning Classifiers,” *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 1, pp. 92–100, 2022, doi: 10.14569/IJACSA.2022.0130112.

[10] T. Chen, P. Su, C. Shang, R. Hill, H. Zhang, and Q. Shen, “Sentiment Classification of Drug Reviews Using Fuzzy-rough Feature Selection,” *IEEE International Conference on Fuzzy Systems*, vol. 2019-June, Jun. 2019, doi: 10.1109/FUZZ-IEEE.2019.8858916.

[11] F. Gräßer, H. Malberg, S. Kallumadi, and S. Zaunseder, “Aspect-Based sentiment analysis of drug reviews applying cross-Domain and cross-Data learning,” *ACM International Conference Proceeding Series*, vol. 2018-April, pp. 121–125, Apr. 2018, doi: 10.1145/3194658.3194677.

[12] S. Vijayaraghavan and D. Basu, “Sentiment Analysis in Drug Reviews using Supervised Machine Learning Algorithms,” Mar. 2020, Accessed: Jun. 04, 2022. [Online]. Available: https://arxiv.org/abs/2003.11643v1

[13] C. Colón-Ruiz and I. Segura-Bedmar, “Comparing deep learning architectures for sentiment analysis on drug reviews,” *Journal of Biomedical Informatics*, vol. 110, p. 103539, Oct. 2020, doi: 10.1016/J.JBI.2020.103539.

[14] M. M. Khan, S. Hossain, P. Mozumdar, S. Akter, and R. H. Ashique, “A review on machine learning and deep learning for various antenna design applications,” *Heliyon*, vol. 8, no. 4, p. e09317, Apr. 2022, doi: 10.1016/J.HELIYON.2022.E09317.

[15] A. Hernández-Blanco, B. Herrera-Flores, D. Tomás, and B. Navarro-Colorado, “A Systematic Review of Deep Learning Approaches to Educational Data Mining,” *Complexity*, vol. 2019, 2019, doi: 10.1155/2019/1306039.

[16] I. H. Sarker, “Machine Learning: Algorithms, Real-World Applications and Research Directions,” *SN Computer Science 2021 2:3*, vol. 2, no. 3, pp. 1–21, Mar. 2021, doi: 10.1007/S42979-021-00592-X.

[17] J.-C. Na and W. Y. M. Kyaing, “Sentiment Analysis of User-Generated Content on Drug Review Websites,” *Journal of Information Science Theory and Practice*, vol. 3, no. 1, pp. 6–23, Mar. 2015, doi: 10.1633/JISTAP.2015.3.1.1.

[18] I. Korkontzelos, A. Nikfarjam, M. Shardlow, A. Sarker, S. Ananiadou, and G. H. Gonzalez, “Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts,” *Journal of Biomedical Informatics*, vol. 62, pp. 148–158, Aug. 2016, doi: 10.1016/J.JBI.2016.06.007.

[19] P. Chandrasekar and K. Qian, “The Impact of Data Preprocessing on the Performance of a Naïve Bayes Classifier,” *Proceedings - International Computer Software and Applications Conference*, vol. 2, pp. 618–619, Aug. 2016, doi: 10.1109/COMPSAC.2016.205.

[20] S. M. and A. R. Pais, “Classification of Phishing Email Using Word Embedding and Machine Learning Techniques,” *Journal of Cyber Security and Mobility*, vol. 11, no. 3, pp. 279–320–279–320, May 2022, doi: 10.13052/JCSM2245-1439.1131.

[21] Y. R. Saputri and H. Februariyanti, “SENTIMENT ANALYSIS ON SHOPEE E-COMMERCE USING THE NAÏVE BAYES CLASSIFIER ALGORITHM,” *Jurnal Mantik*, vol. 6, no. 2, pp. 1349–1357, Jun. 2022, doi: 10.35335/MANTIK.V6I2.2397.

[22] K. N. Singh, S. D. Devi, H. M. Devi, and A. K. Mahanta, “A novel approach for dimension reduction using word embedding: An enhanced text classification approach,” *International Journal of Information Management Data Insights*, vol. 2, no. 1, p. 100061, Apr. 2022, doi: 10.1016/J.JJIMEI.2022.100061.

[23] Y. Shi, Y. Yang, and Y. Liu, “Word embedding representation with synthetic position and context information for relation extraction,” *Proceedings - 9th IEEE International Conference on Big Knowledge, ICBK 2018*, pp. 106–112, Dec. 2018, doi: 10.1109/ICBK.2018.00022.

[24] Y. Mansour and M. Schain, “Learning with Maximum-Entropy Distributions,” *Machine Learning 2001 45:2*, vol. 45, no. 2, pp. 123–145, Nov. 2001, doi: 10.1023/A:1010950718922.

[25] M. Z. Islam, J. Liu, J. Li, L. Liu, and W. Kang, “A semantics aware random forest for text classification,” *International Conference on Information and Knowledge Management, Proceedings*, pp. 1061–1070, Nov. 2019, doi: 10.1145/3357384.3357891.

[26] A. Liaw, M. W.-R. news, and undefined 2002, “Classification and regression by randomForest,” *cogns.northwestern.edu*, vol. 2, no. 3, 2002, Accessed: Jun. 05, 2022. [Online]. Available: https://cogns.northwestern.edu/cbmg/LiawAndWiener2002.pdf

[27] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[28] D. Shi and X. Yang, “Support Vector Machines for Land Cover Mapping from Remote Sensor Imagery,” pp. 265–279, 2015, doi: 10.1007/978-94-017-9813-6\_13.

[29] U. Sharma, S. Saran, and S. M. Patil, “Fake News Detection using Machine Learning Algorithms,” *International Journal of Engineering Research & Technology*, vol. 9, no. 3, Feb. 2021, doi: 10.17577/IJERTCONV9IS03104.

[30] W. Antoun, F. Baly, and H. Hajj, “AraBERT: Transformer-based Model for Arabic Language Understanding,” Feb. 2020, Accessed: Jun. 05, 2022. [Online]. Available: http://arxiv.org/abs/2003.00104

[31] J. Huang, “Detecting Fake News With Machine Learning,” *Journal of Physics: Conference Series*, vol. 1693, no. 1, p. 012158, Dec. 2020, doi: 10.1088/1742-6596/1693/1/012158.

[32] A. el Bilali, A. Taleb, A. Nafii, B. Alabjah, and N. Mazigh, “Prediction of sodium adsorption ratio and chloride concentration in a coastal aquifer under seawater intrusion using machine learning models,” *Environmental Technology and Innovation*, vol. 23, Aug. 2021, doi: 10.1016/J.ETI.2021.101641.

[33] X. Wang *et al.*, “Deep Generative Mixture Model for Robust Imbalance Classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022, doi: 10.1109/TPAMI.2022.3178914.

[34] J. Zhang, Y. Zeng, and B. Starly, “Recurrent neural networks with long term temporal dependencies in machine tool wear diagnosis and prognosis,” *SN Applied Sciences*, vol. 3, no. 4, pp. 1–13, Apr. 2021, doi: 10.1007/S42452-021-04427-5/FIGURES/7.

[35] M. Zulqarnain, R. Ghazali, Y. M. M. Hassim, and M. Rehan, “Text classification based on gated recurrent unit combines with support vector machine,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 4, pp. 3734–3742, Aug. 2020, doi: 10.11591/IJECE.V10I4.PP3734-3742.

[36] J. P. T. Mo *et al.*, “Design and Implementation of an Explainable Bidirectional LSTM Model Based on Transition System Approach for Cooperative AI-Workers,” *Applied Sciences 2022, Vol. 12, Page 6390*, vol. 12, no. 13, p. 6390, Jun. 2022, doi: 10.3390/APP12136390.