**Sentiment Analysis of Drug Reviews using Supervised**

**Machine Learning and Deep Learning**

# Introduction

Pharmacovigilance (PV) ensures the safety and efficacy of medications throughout the drug lifecycle, informing healthcare professionals, patients, pharmaceutical companies, and regulatory agencies. Post-market surveillance is key to understanding drug performance in the real-world, identifying adverse effects, ensuring benefits continue to outweigh any risks. The global PV software market size is growing, predicted as $207.7 million by 2024 (Tepsivo, 2021). Natural language processing offers cost-saving PV solutions automating sentiment classification of large amounts of publicly available patient drug reviews.

Sentiment analysis of drug reviews is underexplored in the literature with most machine (ML) and deep learning (DL) performed on the Drugs.com UCI repository (Vijayaraghavan & Basu, 2020). The dataset covers many diseases but also presents model performance challenges due to the uneven distribution of review ratings, discussed in detail in section 1.1.

This research will explore the effects of balanced class sentiment analysis comparing ML and DL techniques in combination with vectorization and word embedding transfer learning, to improve performance, over currently published solutions.

## Background review

The most recent research compared term frequency-inverse document frequency (TF-IDF) and Count Vectoriser (CV) and supervised machine learning algorithms linear Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), multinomial, Gaussian, Bernoulli and complement Naïve Bayes (NB), and XGB for positive, negative, and neutral sentiment classification. TF-IDF-SVM achieved the highest accuracy (95%) compared to CV-SVM (91%), TF-IDF combined with LR, RF and multinomial-NB also achieved 90%, 89% and 89% accuracy, respectively. No improvement was seen with cross-validation. Improved performance could be achieved by balancing the classes, although over-sampling increases computational time. Improved accuracy was also observed for SVM on Druglib.com and WebMD.com datasets. The work does not extend to word embeddings, DL techniques such as RNNs or transformers (Alaie et al., 2024).

Garg (2021) compared bag of words (BoW), TF-IDF, Word2Vec and several ML classifiers on the unbalanced dataset with TF-IDF-SVM outperforming all other models with 93% accuracy, recommending oversampling and algorithm optimization for increased performance (Garg, 2021). TF-IDF with subsequent FastText embedding did not improve SVM accuracy (Yadav & Vishwakarma, 2020). LR achieved 92% with n-gram and feature space reduction of terms higher than a defined document frequency threshold. Model hyperparameters were tuned for best Cohens Kappa score using 5-fold cross validation and unbalanced classes were accounted for with weight inversely proportional to the class frequency. Majority of classification errors occurred on neutral reviews, potentially improved by converting to binary classification (Gräßer et al., 2018). RF outperformed SVM, multiple-layer perceptron (MLP) and NB, in binary classification with an accuracy of 94%, However, SVM still achieved best multi-class performance (Uddin et al., 2022). NB outperformed DT, RF and KNN achieving accuracy of 87% for multi-class classification (Basiri et al., 2020).

The importance of effective preprocessing is emphasised by Gurdin et al. (2020) comparing unigrams and word embeddings for NB, RF, SVM and CNN across disease areas. SVM achieved the highest F1-scores but error analysis revealed negation and acronym handling, and spell checking could have improved performance (Gurdin et al., 2020).

Comparison of DL models, convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM), hybrid approaches (CNN-BiLSTM, BiLSTM-CNN) and pre-trained embeddings (Word2Vec supplemented with additional training data, and Bidirectional Encoder Representations from Transformers (BERT)), identified BERT-BiLSTM as the best performing model marginally over CNN, determined by micro- and macro-F1-score, compared to TF-IDF-SVM. The CNN was the optimal model due to decreased computational time with negligible trade-off in performance compared to BERT-BiLSTM. The dataset was again unbalanced, with class performance dictated by the number of instances (Colón-Ruiz & Segura-Bedmar, 2020).

Vijayaraghavan et al. (2020) focus on conditions with the most reviews (birth control, depression, and pain) and classify each separately using SVM, RF, LR and unidirectional LSTM, Gated Recurrent Units (GRU) and an artificial neural network (ANN). CV outperformed TF-IDF for all models. Low accuracy (67%) was achieved with Word2Vec embeddings. SVM was the most accurate ML algorithm (82-97% accuracy across diseases) with comparable performance for LSTM and GRU (87-92%). The classes are unbalanced and the work does not consider bidirectional models (Vijayaraghavan & Basu, 2020).

LSTM performed below (50%) RF and SVM for four disease areas. RF achieved an accuracy of 87.6% and 87.5% for BoW and TF-IDF, respectively. The classes again were unbalanced and LSTM performance could be improved by training the model with multiple epochs, not explored due to computational time (Helae et al., 2022). Ahmed et al. (2023) combined patient reviews and ratings with similar illnesses and medications as LSTM classification features. LSTM outperformed SVM, KNN, RF and ANN, achieving training accuracy of 91%, however overfitting produced a validation accuracy of 80% (Ahmed et al., 2023).

Haque et al. (2023) compared TF-IDF and CV for five ML algorithms (RF, SVM, passive aggressive (PAG) LR, and stochastic gradient descent) with word embedding techniques (Word2Seq and GloVe) for LSTM, BiLSTM, GRU, and bidirectional GRU (BiGRU). CV outperformed TF-IDF, with RF achieving the highest accuracy (96.7%) followed by LR (96%). PAG and SVM also performed well, with accuracy above 95.3%. BiLSTM outperformed all models but required more training data and time with negligible performance gain. The dataset was unbalanced reflected in poorer classification of neutral reviews (Haque et al., 2023). Review length standardisation and custom stop words could be explored as others have done (Basiri et al., 2020).

GRU achieved 85.8% and 82.6% for positive and negative reviews, respectively, in binary classification compared to LSTM, BiLSTM, BiGRU, simple RNN with and without embedding. GRU achieved good performance versus computational time. The dataset is unbalanced for binary classification and this wasn’t accounted for using over- or down-sampling (Kumar et al., 2023).

Classification of drug reviews using transformers has seen comparison of BERT models with raw text input with TF-IDF-RF and NB. BioClinicalBERT model outperformed all models achieving 87% accuracy (Shiju & He, 2022). A transformer (topicT-AttNN) with LSTM and attention mechanism outperformed baselines achieving an F1-score of 96% (Job et al., 2023).

## Objective

Many ML and DL approaches have been compared on the Drugs.com review dataset but none have properly addressed class imbalance for sentiment analysis. Higher accuracy has been achieved with binary classification, but these models fail to capture the nuances of less polarised reviews.

SVM, RF and NB perform well with TF-IDF or CV. GRU and LSTM with word embedding also achieve good performance with lower computational time compared to BiLSTM and BiGRU.

Balancing the entire dataset by data augmentation or oversampling will increase the size of an already large dataset, compounding DL running times. Subsampling the diseases with the most reviews and balancing the classes using data augmentation or oversampling, along with downsampling the entire dataset has not been explored.

This research, achievable in a four-week timeframe, will contribute to the field comparing performance of TF-IDF and CV for RF and NB, and GloVe word embeddings with GRU and LSTM on class balanced Drugs.com reviews. Performance will be benchmarked against published model accuracy for unbalanced classes.

# 2 Methodology

## 2.1 Data

The Drugs.com dataset, sourced from UCI machine learning repository dataset, totals 215,063 patient drug reviews split into training and test sets (75%/25%) (UC Irvine Machie Learning Repository, 2018). The dataset is the largest and most used for drug review sentiment classification in the literature, providing a range of sentiments (Gräßer et al., 2018). The features are detailed in Appendix 7.1, Figure 1. The data is provided in tsv format, with numerical and categorical data and unstructured free text reviews. Ratings are unbalanced, ranging from 1-10, 10 indicating excellent (Appendix 7.1, Figure 2).

## 2.2 Cleaning and preprocessing

The training and test sets were combined, and ratings segmented into sentiment classes: positive (rating 7-10), neutral (5,6), and negative (1-4), aligning with previous literature (Appendix 7.1, Figure 3) (Haque et al., 2023). The resulting classes are unbalanced, with 66% positive, 9% neutral and 25% negative reviews.

Review length standardisation was performed including all reviews with length <1400 (Appendix 7.1, Figure 4).

Preprocessing steps performed were lowercasing, contraction expansion, spell checking, HTML, special character, URL, non-ASCII, punctuation, numerical data and stopword removal, whitespace standardisation, tokenisation, and lemmatisation. Technique details and libraries are detailed in Appendix 7.2, Table 1.

The data was subject to 75%/25% split into training and test sets, with 20% training data for DL model validation. 10% of the training dataset was used for optimisation (Scikit-learn, n. d.-i).

Time did not allow for custom abbreviation, acronym and stopword dictionaries and negation handling, which may improve preprocessing quality. TextBlob is an off-the-shelf library for handling contractions and spell-checking, and missed instances were observed. Contraction removal could result in loss of sentiment meaning.

N-grams (bi-/tri-grams) or part-of-speech (POS) tagging was not incorporated due to increased computational time, and increased dimensionality and sparsity (Liu & Sun, 2023).

### 2.2.1 Class balancing

Random oversampling, SMOTE oversampling, and downsampling were applied for class balancing. The five diseases with the highest review counts were subsampled to perform oversampling (Appendix 7.1, Figure 5). The entire dataset was downsampled.

Random oversampling duplicates samples from the minority classes until they are balanced with the majority class (Imbalanced learn, n. d.-a). Downsampling decreases the majority classes by randomly removing instances, reducing the dataset size, potentially affecting model performance. Both techniques are applicable to text data, computationally efficient but can cause over-fitting (Lee & Seo, 2022).

SMOTE can only be applied after vectorisation converting text to numerical values. Applied to a multi-class dataset, it treats each class independently and oversamples the minority class to match the number in the majority class. This process is repeated for each minority class, resulting in a balanced distribution (Imbalanced learn, n. d.-b). Advantages to SMOTE are increase representation of minority classes with synthetic samples, bias mitigation, improving model discriminative power (Wongvorachan et al., 2023).

Word clouds of the downsampled and random oversampled datasets for the positive, neutral, and negative classes indicate side effects as a prominent theme across all sentiment classes, along with frequency (date, week, time) (Appendix 7.1, Figures 6-11). Pain appears frequently in both positive and negative reviews. Customised stopwords like medication and pill could be removed, not considered in this analysis.

## 2.3 Models

### 2.3.1 ML

#### 2.3.1.1 Model comparison

Challenges to overcome in text classification are the large feature space and non-linear multi-class classification. Interpretability is also difficult, with classification more explainable by DT, LR and NB. SVM classification finds the optimal hyperplane separating data classes. DTs recursively partition feature space based on criteria like information gain. RF is an ensemble of DTs, robust to overfitting and effective in capturing complex relationships. LR models probability of binary outcomes linearly. NB is probabilistic, assuming feature independence for efficient classification, particularly in text tasks (Kumar et al., 2020). The strengths and weaknesses of each text classification method are detailed in Appendix 7.1, Table 2. MultinomialNB was chosen because it provides probabilistic predictions providing an interpretable, simpler and efficient analysis, performing well on the Drugs.com dataset previously (Basiri et al., 2020). RF (ensemble technique) was chosen for its robustness to overfitting, although more computationally intensive, often outperforming other algorithms on the dataset (Vijayaraghavan & Basu, 2020; Helae et al., 2022).

#### 2.3.1.2 Vectorisation

TF-IDF determines document word importance relative to a corpus, representing them as numerical vectors. Term frequency (TF) measures document word frequency. Inverse document frequency (IDF) determines word importance across a collection of documents, reducing weight for frequently occurring words. Discriminative words with high TF-IDF scores occur frequently in a specific document but are rare across the dataset (Scikit-learn, n. d.-d). CV counts the frequency terms in each document (Scikit-learn, n. d.-c).

#### 2.3.1.3 MultinomialNB optimisation

Sklearn was used for multinomialNB (Scikit-learn, n. d.-a). Gridsearch was used for alpha and fit\_prior optimisation, scoring using accuracy, with 5-fold cross-validation (Scikit-learn, n. d.-g). Alpha is a smoothing parameter used to prevent zero probabilities by adding a small positive value to feature counts. Fit prior is Boolean parameter determining whether to learn class prior probabilities from the data (Scikit-learn, n. d.-j).

#### 2.3.1.4 RF optimisation

Sklearn was used for RF, an ensemble method (Scikit-learn, n. d.-b). Gridsearch was used for n\_estimators, max\_depth, max\_samples\_split and min\_samples\_leaf optimisation, scoring using accuracy, with 5-fold cross-validation (Scikit-learn, n. d.-g).

N\_estimators determines the number of DT, where a higher number generally improves performance but increases computational complexity. max\_depth controls the maximum depth of each tree, preventing overfitting by limiting tree complexity. min\_samples\_split specifies the minimum number of samples required to split an internal node, reducing overfitting by ensuring nodes have a minimum size for splitting. Lastly, min\_samples\_leaf sets the minimum number of samples allowed at a leaf node, preventing overfitting by enforcing a minimum sample requirement at leaf nodes (Scikit-learn, n. d.-b).

### 2.3.2 DL

#### 2.3.2.1 Model comparison

LSTM and GRU are types of recurrent neural network (RNN) architectures, with LSTM utilizing memory cells and gates to control information flow and GRU being a simplified version with fewer parameters. BiLSTM and BiGRU extend these by processing text bidirectionally, capturing context from both past and future words. Transformers employ self-attention mechanisms to weigh the importance of each word in the text, achieving state-of-the-art performance by efficiently capturing global dependencies (Xiao & Zhu, 2023). The strengths and weaknesses of each method are detailed in Appendix 7.1, Table 3. LSTM and GRU were chosen because of their reduced complexity and computational time compared to BiLSTM, BiGRU and transformers, whilst still providing excellent accuracy that might be improved by balancing the dataset (Haque et al., 2023).

#### 2.3.2.2 GloVe word embedding

GloVe is an unsupervised learning algorithm for obtaining vector representations for words, capturing semantic relationships between words based on co-occurrence statistics (Pennington et al., 2014b).

The common crawl pre-trained word vectors file (42B tokens, 1.9M vocab, uncased, 300d vectors, 1.75 GB) was chosen for its larger size compared to the Wikipedia set, more suitable than the Twitter set for social media applications (Pennington et al., 2014a). Sequence padding ensured uniform length (Tensorflow, n. d.-i). The embedding layers created for downsampled and random oversampled datasets (the latter using top\_five\_diseases tokens, representative of the vocabulary in the oversampled dataset) were set to trainable, allowing updates to improve performance (Tensorflow, n. d.-b). Vectorised SMOTE balanced data cannot be used for word embedding, requiring tokenised data as input.

The data was not one-hot encoded to reduced memory requirements by reducing dimensionality, with model loss function set as sparse\_categorical\_crossentropy (ref).

#### 2.3.2.3 Optimisation

Tensorflow Keras was used for model architecture (Tensorflow, n. d.-h; n. d.-c; n. d.-d; n. d.-a; n. d.-b). Randomisedsearch and KerasClassifier were used for hyperparameter optimisation, scoring using accuracy, with 5-fold cross-validation (Scikeras, n. d.; Scikit-learn, n. d.-h).

Hyperparameter tuning optimised dropout, learning rate, recurrent dropout (specific to the recurrent connections between the hidden states of consecutive time steps) and others (Hanifi et al., 2024) (Tensorflow, n. d.-e; n. d.-f). Need to explain what the different hyperparameters do. Link to hyperparameter table? Adam optimiser and ReLU activation function set, according to others in the literature.

L1 or L2 regularisation was explored, adding a penalty term to the loss function during training, which encourages minimised weights (Keras, n. d.). Dropout regularisation (dropout layers optimised). Dense layers. Effect of one versus two layers and batch size on model performance was also investigated. Flatten layer.

## 2.4 Evaluation

Model performance was evaluated using accuracy (correctly classified samples), precision (accuracy of positive predictions), recall (proportion of true positive predictions of actual positives), F1-score (balancing precision and recall), and confusion matrix (Scikit-learn, n. d.-e; n. d.-f). Micro-average receiver operating characteristic-area under the curve (ROC-AUC) considers each prediction as a binary classification problem, averaging the results for multi-classification. The greater the area, the better the model is able to distinguish between classes (Scikit-learn, n. d.-f).

For DL models, training and validation accuracy and loss were monitored, determining overfitting and for training epoch number optimisation.

### 2.4.1 Baselines

Architecture and performance of literature baseline models, for unbalanced datasets, is recorded in Appendix 7.2, Table 4.

# 3 Results and discussion

## 3.1 ML

TF-IDF vectorised data outperformed CV for the downsampled, random oversampled and SMOTE oversampled dataset for both NB and RF (Appendix 7.2, Table 6,7), in contrast to previous work where RF achieved improved accuracy on the entire unbalanced dataset using the BoW CV approach (Haque et al., 2023). Optimal hyperparameters are detailed in Appendix 7.2, Table 5.

Best performance for both models was achieved on the SMOTE oversampled dataset (Appendix 7.1, Figures 12, 13, Appendix 7.2, Table 6,7). This is expected as SMOTE creates new synthetic samples for class balancing as opposed to random duplication for random oversampling.

MultinomialNB achieved a micro average AUC-ROC of 87.5% on the SMOTE oversampled dataset, but accuracy was below the baseline model**.** However, neutral class classification was much improved, with precision slightly below the positive and neutral classes (Appendix 7.2, Table 4, 6).

There was no difference in accuracy between random oversampled and downsampled data for both NB and RF. For these datasets, RF neutral class precision was comparable to the other classes, however, recall and F1-score were slightly reduced, indicating a lower proportion of actual positives that were correctly identified by the model, observable in the confusion matrix (Appendix 7.1, Figures 14, 15).

RF outperformed NB, achieving an impressive 98% accuracy, 99.8% micro average AUC-ROC, with precision, recall and F1-score all greater than 95% for all classes (Appendix 7.1, Figures 13). TF-IDF-RF also outperformed the baseline model (97% accuracy), with improved neutral review classification, achieving greater than 96% precision, recall and F1-score, comparatively to 25% error (Haque et al., 2023).

## 3.2 DL

GRU vs LSTM with word embeddings, how did they compare to baseline (literature). Classification performance for each class (pos/neg/neut). What model performed best?

Only performed on random oversampling five disease subset.

Optimal hyperparameters (Appendix 7.2, Table 8, 9). Did regularisation help? Dropout layers prevent overfitting? Optimal number of layers, effect of batch size (32 vs 64). Mention flatten layer.

Could have included optmiser and activation function optimisation, didn’t do in this work to reduce parameter space. Could be explored in future work.

Did word embedding transfer learning with GRU/LSTM perform the task better than ML models?

# 4 Conclusion

SMOTE oversampling provided best dataset for classification (ML and DL?)

Best ML model (TF-IDF-RF on SMOTE oversampled data)

Best DL model

Best performing model overall, effect of balancing classes, performance for different classes (pos/neg/neut). Negative classification is improved compared to baseline.

Future work-

Transformer

Word embeddings with ML.

# 5 Referencing styles

MLA, popular in humanities disciplines, employs parenthetical in-text citations including the author's last name and page number, alongside a page of cited works, listing sources. APA, predominant in social sciences, features parenthetical citations with author's last name and publication year, coupled with a references page with bibliographic details. Chicago style, adopted in arts and humanities, offers two citation systems, either footnotes/endnotes or author-date citations, accompanied by a bibliography. Vancouver style for scientific fields, utilises numerical in-text citations along with a sequentially numbered reference list. Harvard style, employed across disciplines, displays author-date citations in the text and a reference list (Neville, 2012). Bibliographic formatting differences are detailed in Appendix 7.2, Table 11. Harvard Hull referencing was chosen for this report.

*Word count 3000*

# 6 References

Alaie, A. I., Farooq, U., Bhat, W. A., Khurana, S. S. & Singh, P. (2024) An empirical study on sentimental drug review analysis using lexicon and machine learning-based techniques. *SN Computer Science*, 5(1), 63.

Ahmed, I., Ahmad, M., Chehri, A. & Jeon, G. (2023) A heterogeneous network embedded medicine recommendation system based on LSTM. *Future Generation Computer Systems*, 149, 1-11.

Alaie, A. I., Farooq, U., Bhat, W. A., Khurana, S. S. & Singh, P. (2024) An empirical study on sentimental drug review analysis using lexicon and machine learning-based techniques. *SN Computer Science*, 5(1), 63.

Basiri, M. E., Abdar, M., Cifci, M. A., Nemati, S. & Acharya, U. R. (2020) A novel method for sentiment classification of drug reviews using fusion of deep and machine learning techniques. *Knowledge-Based Systems*, 198, 105949.

Colón-Ruiz, C. & Segura-Bedmar, I. (2020) Comparing deep learning architectures for sentiment analysis on drug reviews. *Journal of Biomedical Informatics*, 110, 103539.

Garg, S. (2021) Drug recommendation system based on sentiment analysis of drug reviews using machine learning, *11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. Noida, India, 28-29 January. IEEE.

Gräßer, F., Kallumadi, S., Malberg, H. & Zaunseder, S. (2018) Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning, *Proceedings of the 2018 International Conference on Digital Health*. Lyon, France, 23-26 April. Association for Computing Machinery.

Gurdin, G., Vargas, J. A., Maffey, L. G., Olex, A. L., Lewinski, N. A. & McInnes, B. T. (2020) Analysis of inter-domain and cross-domain drug review polarity classification. *AMIA Joint Summits on Translational Science Proceedings*, 2020, 201-210.

Hanifi, S., Cammarono, A. & Zare-Behtash, H. (2024) Advanced hyperparameter optimization of deep learning models for wind power prediction. *Renewable Energy*, 221, 119700.

Haque, R., Laskar, S. H., Khushbu, K. G., Hasan, M. J. & Uddin, J. (2023) Data-driven solution to identify sentiments from online drug reviews. *Computers*, 12(4), 87.

Helae, M. Q. Y., Ebrahimi, D. & Alzhouri, F. (2022) Data analytics in the pharmacology domain, *International Journal of Big Data and Analytics in Healthcare (IJBDAH)*. Hershey, PA, USA: IGI Global.

Imbalanced learn (n. d.-a) *RandomOverSampler.* Available online: <https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.RandomOverSampler.html> [Accessed 06/04/2024].

Imbalanced learn (n. d.-b) *SMOTE.* Available online: <https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html> [Accessed 13/03/2024].

Job, S., Tao, X. H., Li, Y. F., Li, L. & Yong, J. M. (2023) Topic integrated opinion-based drug recommendation with transformers. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7(6), 1676-1686.

Keras (n. d.) *Layer weight regularizers.* Available online: <https://keras.io/api/layers/regularizers/> [Accessed 06/04/2024].

Kumar, A., Dabas, V. & Hooda, P. (2020) Text classification algorithms for mining unstructured data: a SWOT analysis. *International Journal of Information Technology*, 12(4), 1159-1169.

Kumar, A., Kumar, N., Kuriakose, J. & Kumar, Y. (2023) A review of deep learning-based approaches for detection and diagnosis of diverse classes of drugs. *Archives of Computational Methods in Engineering*, 30(6), 3867-3889.

Lee, W. & Seo, K. (2022) Downsampling for binary classification with a highly imbalanced dataset using active learning. *Big Data Research*, 28, 100314.

Liu, Z. & Sun, M. (2023) Representation learning and NLP, in Liu, Z., Lin, Y. & Sun, M. (eds), *Representation Learning for Natural Language Processing*. Singapore: Springer Nature Singapore, 1-27.

Maslej-Krešňáková, V., Sarnovský, M., Butka, P. & Machová, K. (2020) Comparison of deep learning models and various text pre-processing techniques for the toxic comments classification. *Applied Sciences*, 10(23), 8631.

Neville, C. (2012) Referencing: principles, practice and problems. *RGUHS Journal of Pharmaceutical Sciences*, 2(2), 1-8.

NLTK (2023) *Natural language toolkit documentation.* Available online: <https://www.nltk.org/> [Accessed 06/04/2024].

Pennington, J., Socher, R. & Manning, C. D. (2014a) *GloVe: global vectors for word representation.* Available online: <https://nlp.stanford.edu/projects/glove/> [Accessed 13/03/2024].

Pennington, J., Socher, R. & Manning, C. D. (2014b) GloVe: global vectors for word representation, *Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, October 25–29.

Python (2024) *re - Regular expression operations.* Available online: <https://docs.python.org/3/library/re.html> [Accessed 06/04/2024].

Scikeras (n. d.) *scikeras.wrappers.KerasClassifier.* Available online: <https://adriangb.com/scikeras/stable/generated/scikeras.wrappers.KerasClassifier.html> [Accessed 04/04/2024].

Scikit-learn (n. d.-a) *Naive Bayes.* Available online: <https://scikit-learn.org/stable/modules/naive_bayes.html> [Accessed 04/04/2024].

Scikit-learn (n. d.-b) *sklearn.ensemble.RandomForestClassifier.* Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html> [Accessed 04/04/2024].

Scikit-learn (n. d.-c) *sklearn.feature\_extraction.text.CountVectorizer.* Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html> [Accessed 04/04/2024].

Scikit-learn (n. d.-d) *sklearn.feature\_extraction.text.TfidfVectorizer.* Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html> [Accessed 04/04/2024].

Scikit-learn (n. d.-e) *sklearn.metrics.classification\_report.* Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html> [Accessed 04/04/2024].

Scikit-learn (n. d.-f) *sklearn.metrics.confusion\_matrix.* Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html> [Accessed 04/04/2024].

Scikit-learn (n. d.-g) *sklearn.model\_selection.GridSearchCV.* Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html> [Accessed 04/04/2024].

Scikit-learn (n. d.-h) *sklearn.model\_selection.RandomizedSearchCV.* Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html> [Accessed 04/04/2024].

Scikit-learn (n. d.-i) *sklearn.model\_selection.train\_test\_split.* Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html> [Accessed 04/04/2024].

Scikit-learn (n. d.-j) *sklearn.naive\_bayes.MultinomialNB.* Available online: <https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html> [Accessed 04/04/2024].

Shiju, A. & He, Z. (2022) Classifying drug ratings using user reviews with transformer-based language models, *10th International Conference on Healthcare Informatics (ICHI)*. Rochester, USA, 11-14 June. IEEE.

Tensorflow (n. d.-a) *tf.keras.layers.Dense.* Available online: <https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense> [Accessed 09/04/2024].

Tensorflow (n. d.-b) *tf.keras.layers.Embedding.* Available online: <https://www.tensorflow.org/api_docs/python/tf/keras/layers/Embedding> [Accessed 09/04/2024].

Tensorflow (n. d.-c) *tf.keras.layers.GRU* Available online: <https://www.tensorflow.org/api_docs/python/tf/keras/layers/GRU> [Accessed 09/04/2024].

Tensorflow (n. d.-d) *tf.keras.layers.LSTM.* Available online: <https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM> [Accessed 09/04/2024].

Tensorflow (n. d.-e) *tf.keras.optimizers.Adam.* Available online: <https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Adam> [Accessed 09/04/2024].

Tensorflow (n. d.-f) *tf.keras.optimizers.RMSprop.* Available online: <https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/RMSprop> [Accessed 09/04/2024].

Tensorflow (n. d.-g) *tf.keras.preprocessing.text.Tokenizer.* Available online: <https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer> [Accessed 09/04/2024].

Tensorflow (n. d.-h) *tf.keras.Sequential.* Available online: <https://www.tensorflow.org/api_docs/python/tf/keras/Sequential> [Accessed 09/04/2024].

Tensorflow (n. d.-i) *tf.keras.utils.pad\_sequences.* Available online: <https://www.tensorflow.org/api_docs/python/tf/keras/utils/pad_sequences> [Accessed 09/04/2024].

Tepsivo (2021) *The cost of pharmacovigilance.* Available online: <https://www.tepsivo.com/blog/the-cost-of-pharmacovigilance/> [Accessed 04/04/2024].

TextBlob (n. d.) *Installation.* Available online: <https://textblob.readthedocs.io/en/dev/install.html> [Accessed 06/04/2024].

UC Irvine Machie Learning Repository (2018) *Drug reviews (Drugs.com).* Available online: <https://archive.ics.uci.edu/dataset/462/drug+review+dataset+drugs+com> [Accessed 04/04/2024].

Uddin, M. N., Bin Hafiz, M. F., Hossain, S. & Islam, S. M. M. (2022) Drug sentiment analysis using machine learning classifiers. *International Journal of Advanced Computer Science and Applications*, 13(1), 92-100.

Vijayaraghavan, S. & Basu, D. (2020) Sentiment analysis in drug reviews using supervised machine learning algorithms. *arXiv preprint arXiv:2003.11643*, abs/2003.11643.

Wongvorachan, T., He, S. & Bulut, O. (2023) A comparison of undersampling, oversampling, and SMOTE methods for dealing with imbalanced classification in educational data mining. *Information*, 14(1), 54.

Xiao, T. & Zhu, J. (2023) Introduction to transformers: an NLP perspective. *arXiv preprint arXiv:2311.17633*, abs/2311.17633.

Yadav, A. & Vishwakarma, D. K. (2020) A weighted text representation framework for sentiment analysis of medical drug reviews, *6th International Conference on Multimedia Big Data (BigMM)*. New Delhi, India, 24-26 September. IEEE.

# 7 Appendix

## 7.1 Figures

A screenshot of a computer

Description automatically generated

**Figure 1.** Drugs.com dataset

A graph of a bar chart

Description automatically generated with medium confidence

**Figure 2.** Drugs.com rating distribution

A graph with different colored squares

Description automatically generated

**Figure 3.** Drugs.com sentiment class distribution

A green line with black dots

Description automatically generated

**Figure 4.** Review length boxplot, indicating outliers outside 1.5x IQR whiskers

A graph of positive and negative

Description automatically generated

**Figure 5.** Top five disease sentiment class distribution

A close up of words

Description automatically generated

**Figure 6.** Negative sentiment wordcloud for downsampled dataset

A close up of words

Description automatically generated

**Figure 7.** Neutral sentiment wordcloud for downsampled dataset

A close up of words

Description automatically generated

**Figure 8.** Positive sentiment wordcloud for downsampled dataset

A close up of words

Description automatically generated

**Figure 9.** Negative sentiment wordcloud for random oversampled dataset

A close up of words

Description automatically generated

**Figure 10.** Neutral sentiment wordcloud for random oversampled dataset

A close up of words

Description automatically generated

**Figure 11.** Positive sentiment wordcloud for random oversampled dataset

A yellow and purple squares with numbers

Description automatically generated

**Figure 12.** Confusion matrix for NB sentiment classification of SMOTE oversampled TF-IDF dataset

A yellow and purple squares with black numbers

Description automatically generated

**Figure 13.** Confusion matrix for RF sentiment classification of SMOTE oversampled TF-IDF dataset

A yellow and purple squares with numbers

Description automatically generated

**Figure 14.** Confusion matrix for RF sentiment classification of downsampled TF-IDF dataset

A yellow and purple squares with numbers

Description automatically generated

**Figure 15.** Confusion matrix for RF sentiment classification of random oversampled TF-IDF dataset

**Figure 16.** Training and validation accuracy and loss for 1-layer GRU sentiment classification of random oversampled dataset

**Figure 17.** Confusion matrix for 1-layer GRU sentiment classification of random oversampled dataset

**Figure 18.** Training and validation accuracy and loss for 2-layer GRU sentiment classification of random oversampled dataset

**Figure 19.** Confusion matrix for 2-layer GRU sentiment classification of random oversampled dataset

**Figure 20.** Training and validation accuracy and loss for 1-layer LSTM sentiment classification of random oversampled dataset

**Figure 21.** Confusion matrix for 1-layer LSTM sentiment classification of random oversampled dataset

**Figure 22.** Training and validation accuracy and loss for 2-layer LSTM sentiment classification of random oversampled dataset

**Figure 23.** Confusion matrix for 2-layer LSTM sentiment classification of random oversampled dataset

**Figure 24.** Confusion matrix for 1-layer GRU sentiment classification of random oversampled dataset

**Figure 25.** Training and validation accuracy and loss for 1-layer GRU/LSTM sentiment classification, batch size 32, of random oversampled dataset

## 7.2 Tables

**Table 1.** Preprocessing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Steps | Technique | Library | Code | Description |
| 1 | Lowercasing |  | .lower() |  |
| 2 | Spell-checking and contraction expansion | TextBlob\* | .correct() |  |
| 3 | URL, email address, HTML tag, non-ASCII characters, punctuation, special characters, numerical data, whitespace removal | re\*\* | .sub() |  |
| 4 | Tokenisation | tensorflow.keras.preprocessing.text\*\*\* | Tokenizer() | Divide text in tokens (words) |
| 5 | Stopword removal | nltk.corpus\*\*\*\* | stopwords | Removing common words without significant meaning |
| 6 | Lemmatisation | nltk.stem | WordNetLemmatizer() | Reducing words to their base or dictionary form, applying linguistic rules |
| \*(TextBlob, n. d.)  \*\*(Python, 2024)  \*\*\*(Tensorflow, n. d.-g)  \*\*\*\*(NLTK, 2023) | | | | |

**Table 2.** Comparison of ML text classification models (Kumar et al., 2020).

|  |  |  |
| --- | --- | --- |
| Model | Strengths | Weaknesses |
| DT | - Interpretable  - Handles non-linear relationships  - Robust to outliers with feature space partitioning | - Prone to overfitting, especially with deep trees  - Unstable with small variations in training data  - Limited expressiveness |
| Multinomial-NB | - Simple and fast, good for large scale text classification  - Handles irrelevant features and high dimensionality  - Interpretable probabilistic predictions | - Strong independence assumption, might not hold true in practice  - Sensitive to feature distribution differences between training and testing data  - Limited expressiveness |
| LR | - Interpretable predictions by probability estimations  - Simpler  - Handles linear relationships between features and target variable | - Limited to linear relationships  - Sensitive to outliers  - Optical performance dependant on optimised feature selection |
| SVM | - Effective in high-dimensional spaces, suitable for text classification with many features  - Handles linear and non-linear relationships due to different kernel functions  - Robust against overfitting due to regularisation | - Computationally expensive  - Hyperparameter tuning crucial to performance (choice of kernel/regularisation)  - Low interpretability as not probabilistic |
| RF | - Ensemble learning combines multiple trees  - Averages predictions so more robust against overfitting  - Handles high-dimensionality without feature selection | - Lack of interpretability  - Computational complexity and resource/time  - Memory consumption |

**Table 3.** Comparison of DL text classification models (Maslej-Krešňáková et al., 2020)

|  |  |  |
| --- | --- | --- |
| Model | Strengths | Weaknesses |
| GRU | - Simpler architecture for faster training and inference  - Fewer parameters, more memory-efficient.  - Effective in capturing short-term dependencies in sequential data | - May struggle with capturing long-term dependencies compared to LSTM  - Limited expressiveness compared to more complex models like transformers |
| LSTM | - Captures both short-term and long-term dependencies in sequential data  - Mitigates vanishing gradient problem  - More expressive than GRU due to additional memory cell | - More complex architecture compared to GRU, with slower training and higher memory requirements  - More parameters, making it computationally more expensive |
| BiGRU | - Captures both past and future context of each word, improving understanding of context  - Simpler architecture compared to BiLSTM, faster training and inference  - Effective in capturing dependencies in both directions of the input sequence | - Doubles the number of parameters compared to unidirectional GRU, increasing computational complexity  - May overfit, especially with large datasets and complex architectures |
| BiLSTM | - Captures both past and future context of each word, improving understanding of context  - Effective in capturing dependencies in both directions of the input sequence | - Doubles the number of parameters compared to unidirectional LSTM, increasing computational complexity  - May overfit, especially with large datasets and complex architectures |
| Transformer | - Self-attention mechanism allows capturing global dependencies in the input sequence  - Parallelisation of computation leads to faster training compared to sequential models like LSTMs and GRUs  - State-of-the-art performance | - Higher memory requirements due to attention mechanisms, limiting scalability to very large datasets  - Complex architecture and training process requiring significant computational resources  - Less interpretable compared to other models due to lack of sequential processing |

**Table 4.** Model baselines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Weighted Average Precision | Weighted Average Recall | Weighted Average F1-score | Neutral class performance |
| MultinomialNB\* | 0.8722 | 0.8775 | 0.8722 | 0.8626 | 0.3730 recall, 0.5371 F1-score |
| CV-RF\*\* | 0.9665 | 0.9660 | 0.9613 | 0.9642 | 25.7% errors |
| TF-IDF-RF\*\* | 0.9329 | 0.9239 | 0.9459 | 0.9369 |  |
| GRU\*\*\* | 0.9528 | 0.9546 | 0.9551 | 0.9548 |  |
| LSTM\*\*\* | 0.9720 | 0.9708 | 0.9566 | 0.9734 |  |
| \*(Basiri et al., 2020)  \*\*(Haque et al., 2023)  \*\*\*(Haque et al., 2023) Adam optimiser, learning rate=0.01, Embedding layer, Dropout layer, LSTM/GRU layer, Dense/Dropout (units=64), Dense/Dropout (units=32), Dense/Dropout (units=16), Flatten layer, Dense (units=3, Softmax) | | | | | |

**Table 5.** ML optimal hyperparameters

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Dataset | Vectorisation | Best Parameters |
| MultinomialNB | Downsampled | TF-IDF | 'alpha': 2.0, 'fit\_prior': True |
| CV | 'alpha': 2.0, 'fit\_prior': False |
| Random oversampled | TF-IDF | 'alpha': 1.0, 'fit\_prior': True |
| CV | 'alpha': 2.0, 'fit\_prior': True |
| SMOTE oversampled | TF-IDF | 'alpha': 0.1, 'fit\_prior': False |
| CV | 'alpha': 0.5, 'fit\_prior': True |
| RF | Downsampled | TF-IDF | 'max\_depth': None, 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 300 |
| CV | 'max\_depth': None, 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 300 |
| Random oversampled | TF-IDF | 'max\_depth': None, 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 300 |
| CV | 'max\_depth': None, 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 300 |
| SMOTE oversampled | TF-IDF | 'max\_depth': None, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 300 |
| CV | 'max\_depth': None, 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 300 |

**Table 6.** MultinomialNB performance

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | Vectorisation | Class | Accuracy | Precision | Recall | F1-score | Micro Average ROC AUC |
| Downsampled | TF-IDF |  | 0.60 |  |  |  | 0.783 |
| Negative |  | 0.66 | 0.58 | 0.62 |  |
| Neutral |  | 0.52 | 0.57 | 0.54 |  |
| Positive |  | 0.62 | 0.65 | 0.64 |  |
| CV |  | 0.60 |  |  |  | 0.773 |
| Negative |  | 0.66 | 0.59 | 0.63 |  |
| Neutral |  | 0.53 | 0.53 | 0.53 |  |
| Positive |  | 0.60 | 0.67 | 0.63 |  |
| Random oversampled | TF-IDF |  | 0.60 |  |  |  | 0.785 |
| Negative |  | 0.66 | 0.59 | 0.63 |  |
| Neutral |  | 0.53 | 0.55 | 0.54 |  |
| Positive |  | 0.62 | 0.65 | 0.64 |  |
| CV |  | 0.60 |  |  |  | 0.773 |
| Negative |  | 0.66 | 0.59 | 0.63 |  |
| Neutral |  | 0.53 | 0.53 | 0.53 |  |
| Positive |  | 0.60 | 0.67 | 0.63 |  |
| SMOTE oversampled | **TF-IDF** |  | **0.71** |  |  |  | **0.875** |
| **Negative** |  | **0.79** | **0.69** | **0.74** |  |
| **Neutral** |  | **0.65** | **0.75** | **0.70** |  |
| **Positive** |  | **0.71** | **0.70** | **0.70** |  |
| CV |  | 0.64 |  |  |  | 0.803 |
| Negative |  | 0.73 | 0.50 | 0.59 |  |
| Neutral |  | 0.55 | 0.70 | 0.62 |  |
| Positive |  | 0.68 | 0.71 | 0.69 |  |

**Table 7.** RF performance

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | Vectorisation | Class | Accuracy | Precision | Recall | F1-score | Micro Average ROC AUC |
| Downsampled | TF-IDF |  | 0.73 |  |  |  | 0.881 |
| Negative |  | 0.73 | 0.75 | 0.74 |  |
| Neutral |  | 0.77 | 0.67 | 0.71 |  |
| Positive |  | 0.70 | 0.78 | 0.74 |  |
| CV |  | 0.71 |  |  |  | 0.869 |
| Negative |  | 0.71 | 0.75 | 0.73 |  |
| Neutral |  | 0.75 | 0.62 | 0.68 |  |
| Positive |  | 0.68 | 0.77 | 0.72 |  |
| Random oversampled | TF-IDF |  | 0.73 |  |  |  | 0.881 |
| Negative |  | 0.73 | 0.75 | 0.74 |  |
| Neutral |  | 0.77 | 0.67 | 0.71 |  |
| Positive |  | 0.70 | 0.78 | 0.74 |  |
| CV |  | 0.71 |  |  |  | 0.870 |
| Negative |  | 0.71 | 0.75 | 0.73 |  |
| Neutral |  | 0.76 | 0.61 | 0.68 |  |
| Positive |  | 0.68 | 0.77 | 0.72 |  |
| SMOTE oversampled | **TF-IDF** |  | **0.98** |  |  |  | **0.998** |
| **Negative** |  | **0.98** | **0.96** | **0.97** |  |
| **Neutral** |  | **1.00** | **0.98** | **0.99** |  |
| **Positive** |  | **0.95** | **0.98** | **0.97** |  |
| CV |  | 0.93 |  |  |  | 0.989 |
| Negative |  | 0.94 | 0.89 | 0.92 |  |
| Neutral |  | 0.94 | 0.95 | 0.94 |  |
| Positive |  | 0.92 | 0.96 | 0.94 |  |

**Table X.** DL model optimisation

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| No. | Model | Architecture\* | units | learning\_rate | dropout | recurrent\_dropout | dense\_dropout | L1/L2 regularisation | Training accuracy |  |
| 1 | GRU | Embedding, dropout, GRU, 3 dense layers (64,32,16 units), flatten, classification\*\* | 128 | 0.001 | 0.2 | 0.1 | 0.2 | None | 0.501 |  |
| 2 | GRU | Embedding, dropout, GRU, 3 dense layers (64,32,16 units), classification | 128 | 0.1 | 0.4 | 0.3 | 0.2 | None | 0.473 |  |
| 3 | GRU | Embedding, dropout, GRU, 2 dense layers (32,16 units), classification | 128 | 0.01 | 0.3 | 0.1 | 0.2 | None | 0.529 |  |
| 4 | GRU | Embedding, dropout, GRU, 1 dense layer (32 units), classification | 128 | 0.001 | 0.3 | 0.3 | 0.2 | None | 0.554 |  |
| 5 | GRU | Embedding, GRU, 1 dense layer (16 units), classification | 128 | 0.001 | 0.4 | 0.3 | 0.2 | None | 0.582 |  |
| 6 | GRU | Embedding, GRU, 1 dense layer (64 units), flatten, classification | 128 | 0.001 | 0.3 | 0.2 | 0.2 | None | 0.612 |  |
| 7 | GRU | Embedding, GRU, classification | 128 | 0.01 | 0.3 | 0.1 | - | None | 0.637 |  |
| 8 | LSTM | Embedding, dropout, LSTM, 3 dense layers (64,32,16 units), flatten, classification\*\* | 128 | 0.001 | 0.2 | 0.2 | 0.2 | L2 (lambda 0.001) | 0.610 |  |
| 9 | BiLSTM | Embedding, dropout, BiLSTM, 3 dense layers (64,32,16 units), flatten, classification\*\* | 128 | 0.01 | 0.3 | 0.3 | 0.2 | None | 0.515 |  |
| 10 | BiLSTM | Embedding, dropout, BiLSTM, 1 dense layer (32 units), flatten, classification | 128 | 0.01 | 0.3 | 0.3 | 0.2 | None |  |  |
| 11 | BiLSTM | Embedding, BiLSTM, classification | 128 | 0.01 | 0.3 | 0.3 | - | None |  |  |
| \*Adam optimiser, ReLU activation, 5-fold cross-validation, n\_iterations = 50  \*\*Baseline architecture (Haque et al., 2023) | | | | | | | | | |  |

**Table X.** DL model performance on random oversampled dataset

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model\* | Batch size | Class | Accuracy | Precision | Recall | F1-score |
| 1 | 64 |  | 0.64 |  |  |  |
| Negative |  | 0.71 | 0.57 | 0.63 |
| Neutral |  | 0.53 | 0.65 | 0.59 |
| Positive |  | 0.70 | 0.71 | 0.70 |
| 1 | 32 |  | 0.61 |  |  |  |
| Negative |  | 0.60 | 0.71 | 0.65 |
| Neutral |  | 0.50 | 0.49 | 0.49 |
| Positive |  | 0.75 | 0.63 | 0.68 |
| 6 | 64 |  | 0.63 |  |  |  |
| Negative |  | 0.63 | 0.70 | 0.66 |
| Neutral |  | 0.57 | 0.43 | 0.49 |
| Positive |  | 0.67 | 0.75 | 0.70 |
| 7 | 64 |  | 0.62 |  |  |  |
| Negative |  | 0.63 | 0.68 | 0.65 |
| Neutral |  | 0.55 | 0.47 | 0.51 |
| Positive |  | 0.68 | 0.72 | 0.70 |
| 8 | 64 |  | 0.64 |  |  |  |
| Negative |  | 0.63 | 0.71 | 0.67 |
| Neutral |  | 0.57 | 0.54 | 0.56 |
| Positive |  | 0.73 | 0.67 | 0.70 |
| 9 | 32 |  | 0.65 |  |  |  |
| Negative |  | 0.71 | 0.60 | 0.65 |
| Neutral |  | 0.56 | 0.61 | 0.58 |
| Positive |  | 0.70 | 0.75 | 0.72 |
| 10 | 32 |  | 0.66 |  |  |  |
| Negative |  | 0.67 | 0.68 | 0.68 |
| Neutral |  | 0.56 | 0.66 | 0.61 |
| Positive |  | 0.78 | 0.63 | 0.70 |
| 11 | 32 |  | 0.65 |  |  |  |
| Negative |  | 0.70 | 0.59 | 0.64 |
| Neutral |  | 0.55 | 0.65 | 0.59 |
| Positive |  | 0.72 | 0.71 | 0.71 |
| \*Run for 30 epochs, with early stopping (patience = 3) | | | | | | |

**Table 11.** Referencing style bibliographic formatting

|  |  |
| --- | --- |
| Style | Format |
| MLA | Haque, Rezaul, et al. "Data-Driven Solution to Identify Sentiments from Online Drug Reviews." *Computers*, vol. 12, no. 4, 2023, p. 87. |
| APA | Haque, R., Laskar, S. H., Khushbu, K. G., Hasan, M. J., & Uddin, J. (2023). Data-driven solution to identify sentiments from online drug reviews. *Computers*, 12(4), 87. |
| Chicago | Haque, Rezaul, Saddam Hossain Laskar, Katura Gania Khushbu, Md Junayed Hasan, and Jia Uddin. "Data-Driven Solution to Identify Sentiments from Online Drug Reviews." *Computers* 12, no. 4 (2023): 87. |
| Vancouver | Haque R, Laskar SH, Khushbu KG, Hasan MJ, Uddin J. Data-Driven Solution to Identify Sentiments from Online Drug Reviews. *Computers*. 2023;12(4):87. |
| Harvard | Haque, R., Laskar, S. H., Khushbu, K. G., Hasan, M. J., & Uddin, J. (2023) 'Data-Driven Solution to Identify Sentiments from Online Drug Reviews', *Computers*, 12(4), p. 87. |
| Harvard Hull | Haque, R., Laskar, S. H., Khushbu, K. G., Hasan, M. J. & Uddin, J. (2023) Data-driven solution to identify sentiments from online drug reviews. *Computers*, 12(4), 87. |