SENTIMENT ANALYSIS OF TRADING 212 PLATFROM BASED ON TRUSTPILOT REVIEWS

Student Name: Itunuoluwa Abidoye
Student ID: 202136265
Course Code: 771767_A23_T2
Course: Applied Artificial Intelligence

Abstract

Sentiment analysis has surfaced as a decisive tool concerning the understanding of users' behaviour and the improvement of service delivery, which is especially useful in the finance sector. The scope of this report is towards the utilization of sentiment analysis of Trading 212 platform with the oversampling of over 22,000 reviews from Trustpilot, which had data imbalances, that were corrected by the increase of the sample up to 41,000 for the achievement of satisfying results. Through the employment of both traditional machine learning models (Naive Bayes, Logistic Regression, Support Vector Machine classifier) and advanced deep learning frameworks (RNN, LSTM, GRU) this analysis evaluates their ability to produce meaningful Insights which would give a general review of the user experience and can assist in redesigning and tailoring services to the users.

Introduction

In the fast-improving world of financial services, the grasping of customer feelings is of great importance for a company to be in the race against its rivals. Trading 212 is a popular trading platform which interacts with a lot of users daily thereby, getting the feedback of the users by the means of online reviews. This feedback gives the necessary information to the users' reaction to the platform's services and features which is an essential part of the research.

Sentiment analysis, which is a subdivision of the natural language processing (NLP), is a scientific method that is utilized to extract, measure, and study the affective states and the subjective information from these reviews (Ahmad & Umar, 2023). The importance of this analysis is in its capacity to affect business strategies by improving customer service and making possible marketing that is tailored to a target audience. Sentiment analysis helps in the comprehension of the customer requests and reviews, which is a need for the survival of a customer-oriented business in a customer focus-oriented market.

Studies and report of today indicate that companies that have the capacity to analyse customers' feedback with the help of sentiment analysis are the ones who are capable to improve their services and increase market share. Although NLP has been well-developed, this study will be a unique opportunity to make discoveries about the customers sentiment on the company's product. Taking into consideration the huge amount of unstructured data on Trustpilot, the sentiment analysis tools are the key to the successful utilization of user feedback into the actionable insights.

Background Review

Sentiment analysis, also known as opinion mining, is a natural language processing (NLP) technique used to determine the sentiment expressed in a piece of text (Ahmad & Umar, 2023). It has been widely applied in various domains, including product and company reviews, to gauge public opinion. In the implementation of sentiment analysis, various traditional and deep learning models have been explored to improve the accuracy and efficiency of sentiment classification tasks.

One of the earliest and simplest models used is the Naive Bayes classifier, which operates on Bayes' theorem and assumes independence between features (Pang et al., 2002). While computationally efficient and easy to implement, its independence assumption often leads to suboptimal performance in complex datasets. Despite its simplicity, (Pang et al., 2002)found that Naive Bayes performed reasonably well but was outperformed by Support Vector Machines (SVM).

Support Vector Machines (SVM) are popular for sentiment analysis due to their ability to handle high-dimensional spaces and text data effectively (Joachims, 1998). SVMs are robust to overfitting and provide high accuracy, though they can be computationally expensive and complex to tune. Comparative studies found SVMs to outperform Naive Bayes and logistic regression in sentiment classification tasks (Pang et al., 2002).

Decision Trees and Random Forests split data based on feature values to build tree-like models (Breiman, 2001). Decision trees are easy to interpret and handle both numerical and categorical data but can be prone to overfitting. Random Forests, an ensemble method, mitigate this issue and provide robust performance with improved accuracy over single decision trees (Breiman, 2001).

In recent years, deep learning models have significantly enhanced sentiment analysis performance. Convolutional Neural Networks (CNNs), originally designed for image processing, have been adapted for text data by applying convolutional filters to capture local dependencies (Kim, 2014). CNNs effectively capture local patterns and are less prone to overfitting but may struggle with long-range dependencies. Kim (2014) demonstrated that CNNs outperformed traditional models, especially with pre-trained word embeddings.

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) Networks are designed to handle sequential data and capture long-range dependencies by maintaining a memory of previous inputs (Hochreiter & Schmidhuber, 1997) LSTMs are powerful for capturing context over long sequences and mitigating the vanishing gradient problem but can be computationally intensive and require large datasets. Tang et al., (2015) demonstrated that LSTM-based models significantly outperformed traditional methods and CNNs, particularly for longer reviews.

The most recent advancement in NLP for sentiment analysis is the transformer architecture, particularly BERT (Bidirectional Encoder Representations from Transformers). BERT uses a bidirectional approach to pre-train a transformer model on a large corpus, capturing context from both directions (Devlin et al., 2019). BERT achieves state-of-the-art performance across various NLP tasks due to its ability to capture contextual information but requires substantial computational resources. Devlin et al. (2019) reported that BERT achieved unprecedented accuracy in sentiment analysis tasks, surpassing previous models, including LSTMs and CNNs.

The progression from traditional models like Naive Bayes and SVMs to deep learning models such as CNNs, LSTMs, and transformers like BERT has brought significant improvements to sentiment analysis. Each method has strengths and limitations, with deep learning models generally providing superior accuracy and context understanding at the cost of higher computational requirements. Future research in sentiment analysis may focus on improving the efficiency and scalability of these advanced models.

SMART Objectives:

- **Specific**: To apply sentiment analysis to over 22,000 Trustpilot reviews of Trading 212, identifying overall sentiment trends related to user satisfaction and concerns.
- **Measurable**: Utilize NLP metrics such as polarity scores to quantify sentiment, aiming for a model accuracy of over 80%.
- **Achievable**: Given the availability of DAIM computational resources, and the proven efficacy of the chosen models (Gaussian Naive Bayes, Logistic Regression, SVM, LSTM, RNN and GRU) in similar datasets, achieving high accuracy in sentiment classification is feasible.
- **Relevant**: The project aligns with Trading 212's strategic goals of enhancing customer satisfaction and improving service quality.
- **Time-bound**: The analysis will be conducted over 6 weeks, allowing sufficient time for model training, evaluation, and refinement.

Dataset Description

The dataset comprises 22,640 reviews sourced from Trustpilot which is suitable for the NLP application, specifically targeted towards the financial trading platform, Trading 212. Each entry in the dataset includes the full text of a customer review, along with a numeric rating ranging from 1 to 5. The dataset is a structured tabular-textual data with the following main fields: 'review_text' and 'review_rating,' which are crucial for our analysis. The Dataset was gotten using the web scrapping Beautiful Soup method and saved to a CSV file for analysis.

Data Suitability: The large volume of data ensures that the analysis can be statistically significant and highly representative of the customer base. Additionally, the presence of both text and numeric ratings allows for a dual approach to sentiment analysis both quantitative and qualitative.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 22640 entries, 0 to 22639
Data columns (total 5 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   review_title          22640 non-null  object
 1   review_date_original  22640 non-null  object
 2   review_rating         22640 non-null  object
 3   review_text           16841 non-null  object
 4   page_number           22640 non-null  int64
dtypes: int64(1), object(4)
memory usage: 884.5+ KB
```

Figure1: Dataset structure of Trading 212 from Trustpilot

Exploratory Data Analysis

Pre-processing: The scraped dataset from Trustpilot was inspected and cleaned by checking for null and empty entries. During this process, approximately 5799 null entries were found in the 'review_text' column. These null entries were addressed by using the rating column to categorize reviews as positive or negative. Several pre-processing steps were then applied to further prepare the data for analysis:

- **Handling Missing Data**: Identified and cleaned 5799 null entries in the 'review_text' column.
- **Sentiment Categorization**: Used the rating (1 to 5) column to categorize reviews as positive or negative.
- **Text Normalization**: Conversion of all text to lowercase to ensure uniformity.
- **Noise Removal**: Stripping out unnecessary punctuation and non-alphanumeric characters.
- **Tokenization**: Breaking down possible paragraphs into sentences and sentences into words.
- **Stop Words Removal**: Filtering out common words (such as 'and', 'the', etc.) that do not contribute to sentiment analysis.
- **Lemmatization**: Reducing words to their base or root form.
- **Feature Engineering:** Created additional features from the text data, such as trigrams, bigrams, word cloud, sentiment scores, or TF-IDF vectors.
- **Data Splitting:** Split the dataset into training and testing sets to facilitate model building and evaluation.

These steps ensured that the dataset was properly prepared for subsequent analysis and modelling tasks.

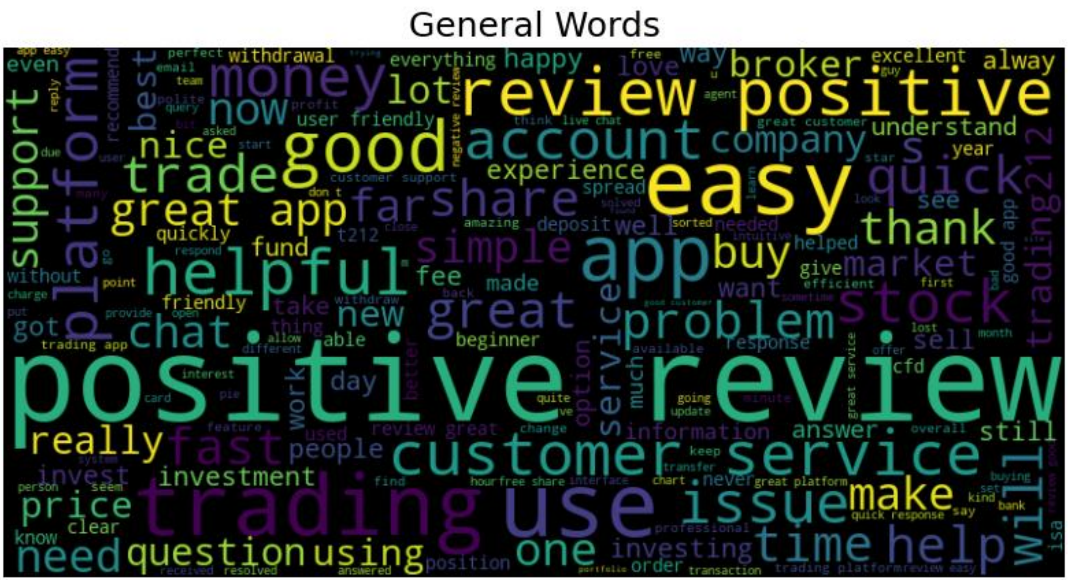

Figure 2: General Word Cloud words

Figure 3: (Left) Positive Word Cloud (Right) Negative Word Cloud Words

Handling Imbalances: The initial analysis revealed a disparity in the distribution of ratings as seen in Figure 4, with a majority leaning towards higher ratings (Rating 5). To address this imbalance, oversampling and under-sampling balancing technique was considered to ensure that the model is not biased towards the majority class.

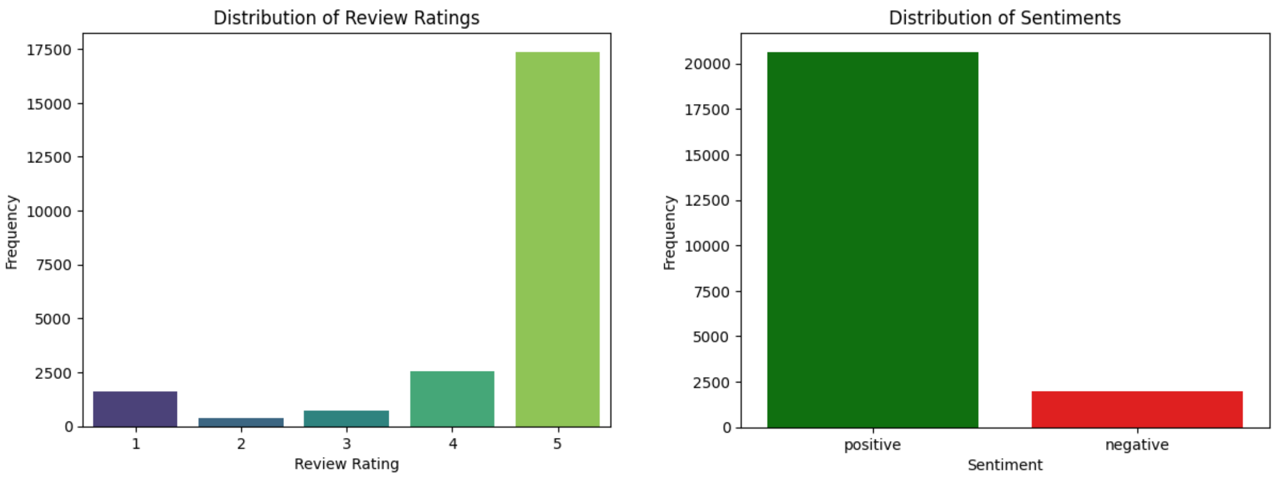

Figure 4: (Left) Unbalanced Rating Distribution Plot (Right) Sentiment Distribution

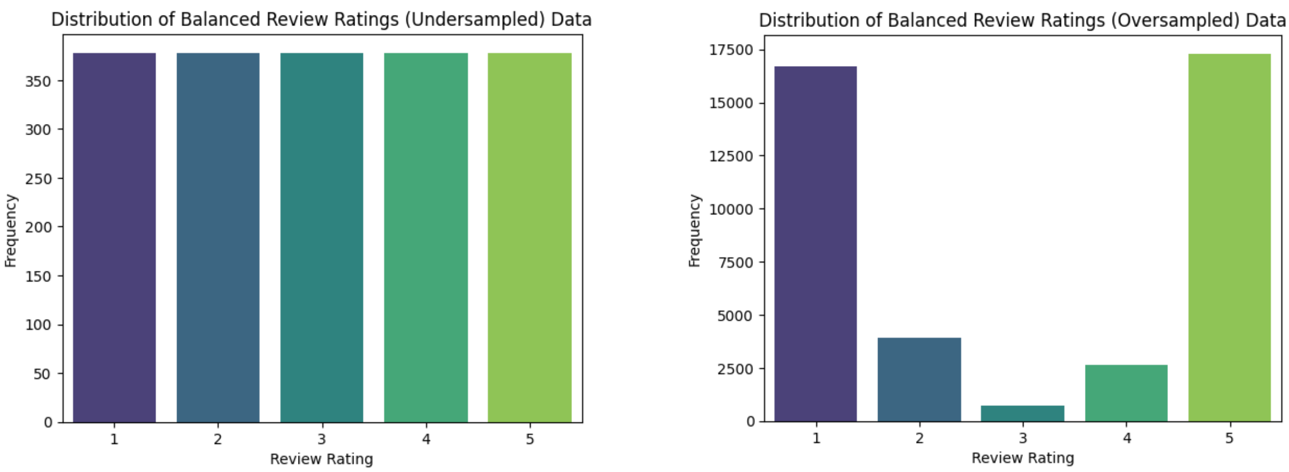

Figure 5: (Left) Rating Under-sampled Plot (Right) Rating Over-sampled Plot.

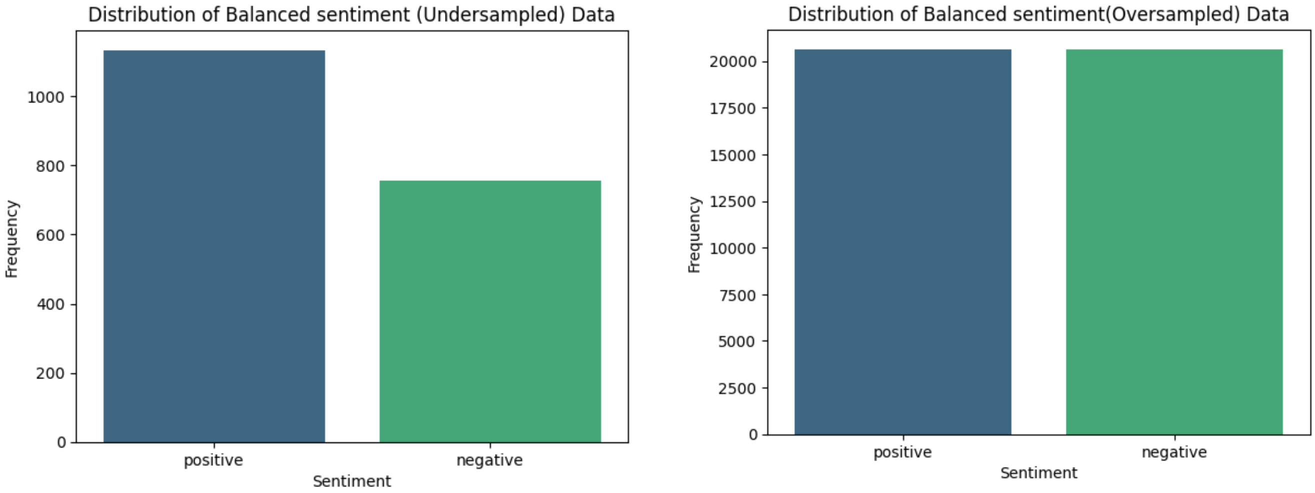

Figure 6: (Left) Sentiment Under-sampled Plot (Right) Sentiment Over-sampled Plot

Baseline Performance: A majority class classifier (Sentiment Oversampled Data) was established as the baseline, which prevents bias and predicts the most frequent class. This provided a benchmark for subsequent, more sophisticated models.

Traditional Machine Learning Methods

Gaussian Naive Bayes (NB): Gaussian Naive Bayes is a variant of the Naive Bayes classifier that assumes features follow a Gaussian (normal) distribution. It is effective for datasets where features can be assumed to be normally distributed, especially with a large feature set (Rish, 2001).

Multinomial Naive Bayes: Multinomial Naive Bayes is tailored for multinomially distributed data and is commonly used for text classification, where features (words or n-grams) are represented as frequency counts (Mccallum & Nigam, 1998).

Logistic Regression: Logistic Regression is used for binary classification problems, modelling the probability of class membership. It is robust to noise and less prone to overfitting, though it requires a good amount of data for stable results (Hosmer & Lemeshow, 2000).

Support Vector Machine (SVM): SVMs are powerful classifiers for feature-rich data, finding the hyperplane that best separates different classes by maximizing the margin between them. They are effective in high-dimensional spaces but can be computationally intensive (Cortes, 1995).

Random Forest: Random Forest is an ensemble learning method that constructs multiple decision trees during training. It improves classification accuracy and robustness by reducing overfitting and aggregating the predictions of individual trees (Breiman, 2001 ; Aslam et al., 2024).

Table 2: Traditional Model Comparison

Model	Strength	Weaknesses
Gaussian Naive Bayes (NB)	• Fast training and prediction times. • Works well with high-dimensional data.	• Assumes feature independence and may perform poorly if this assumption does not hold. • Less effective with skewed data distributions, which are common in sentiment analysis where lexical features often have uneven distributions.
Multinomial Naive Bayes	• Particularly effective for text classification with discrete frequency counts. • Simple and easy to understand and implement.	• Like Gaussian NB, it assumes independence between features, which is rarely the case in text data. • It cannot learn interactions between words.
Logistic Regression	• Provides probabilities for predictions and robust performance on linearly separable classes. • Works well with smaller datasets compared to complex models.	• Struggles with high-dimensional data without regularization. • Limited to linear decision boundaries unless polynomial features are used.
Support Vector Machine (SVM)	• Effective in high-dimensional spaces. • Works well with a clear margin of separation and is effective in complex domains where there is a clear degree of separation.	• Requires full data loading into memory, thus not suitable for large datasets. • Performance depends heavily on the choice of the kernel function.
Random Forest	• Effective for a wide variety of data types and tasks.	• Can be slow to train, especially with many trees. • Not as easy to visually interpret as other

	• Can handle large datasets with higher dimensionality.	models like decision trees.

Justification: Multinomial NB, Logistic Regression, and SVM are chosen for their effectiveness in binary classification tasks common in sentiment analysis (Ahmad & Umar, 2023).Logistic Regression provides a good baseline for understanding feature importance, SVM is robust against overfitting, and Multinomial NB can be very effective given the proper feature normalization and if the assumption of feature independence roughly holds. These models have been shown to perform well in financial sentiment analysis scenarios, where the features are often well-defined and benefit from linear classifiers (Ahmad & Umar, 2023).

Deep Learning Methods

Long Short-Term Memory (LSTM): LSTM networks are a type of RNN designed to capture long-term dependencies by maintaining a memory cell that preserves information over prolonged periods, addressing the vanishing gradient problem (Hochreiter & Schmidhuber, 1997).

Recurrent Neural Network (RNN): RNNs handle sequential data by maintaining a hidden state that captures information about previous inputs, making them suitable for tasks where data order is important (Mikolov et al., 2010).

Gated Recurrent Unit (GRU): GRUs are a simpler variant of RNNs that combine the forget and input gates into a single update gate, reducing complexity and computational cost while effectively capturing dependencies in sequential data (Cho et al., 2014).

Bidirectional Encoder Representations from Transformers (BERT): BERT is a transformer-based model for NLP tasks that uses a bidirectional approach to pre-train on a large corpus, capturing context from both directions to achieve state-of-the-art performance (Devlin et al., 2019).

Convolutional Neural Network (CNN): CNNs, adapted for text data, apply convolutional filters to input data to capture local patterns and extract meaningful features, making them effective for tasks like sentence classification (Kim, 2014).

Table 3: Deep Learning Model Comparison

Model	Strength	Weaknesses
Long Short-Term Memory (LSTM)	• LSTMs are particularly effective in capturing long-range dependencies in text data, crucial for understanding the context in customer reviews that may affect sentiment (Hochreiter & Schmidhuber, 1997).	• They are computationally complex and hence require a lot of training time and resources which can be a constraint for the quick

	• Avoids the vanishing gradient problem typical in standard RNNs.	deployment of models or on a large scale. • They are Prone to overfitting on smaller datasets
Recurrent Neural Networks (RNN)	• RNNs are straightforward to implement and can model time-series data effectively, capturing temporal dynamics which is beneficial for text-based data (Hochreiter & Schmidhuber, 1997).	• Suffers from the vanishing gradient problem, making it difficult to learn long-term dependencies(Li et al., 2024). • Not as powerful as LSTMs or GRUs for handling exceptionally long sequences.
Gated Recurrent Unit (GRU)	• GRUs reduce the complexity of LSTMs and often provide similar performance with fewer parameters, which can lead to faster training times (Chung et al., 2014).	• Computationally intensive compared to traditional methods. • They underperform behind LSTMs in tasks requiring complex pattern extraction from long sequences.
Bidirectional Encoder Representations from Transformers (BERT)	• BERT are pre-trained on large text and fine-tuned for specific tasks, providing significant improvements over older models. • BERT can generate a rich understanding of language variation, which is extremely beneficial in sentiment analysis. It has been shown to achieve state-of-the-art results on a variety of NLP tasks (Devlin et al., 2019).	• Computationally resource intensive model which makes its usage in resource scarce places difficult.
Convolutional Neural Networks (CNN)	• CNNs are faster to train than RNNs and can capture hierarchical patterns in data,	• They cannot handle sequential data as effectively as RNNs,

	which can be useful in picking up patterns in sentence structures (Kim, 2014).	missing capturing temporal dependencies.

Justification: LSTMs and GRUs are chosen for this analysis due to their capability of understanding and keep information for longer text sequences which is necessary to accurately gauge sentiments from those regular customer reviews. RNN's, which was also used even though are simpler and less resource intensive, are the basis of performance comparison and are effective for shorter or less complex text data.

Implementation and Refinement

Table 4: Model Implementation and Refinement

Model	**Libraries Used**	**Procedures Followed:** Data Pre-processing (DP), Model Building (MB), Model Evaluation (ME), Model Training (MT)	**Finetuning Strategies:** Hyperparameter Tuning (HT), Regularization(R)	**Justification**
Logistic Regression	pandas, scikit-learn (LogisticRegression, train_test_split, classification_report, confusion_matrix), matplotlib.pyplot, seaborn	**DP:** Loaded, cleaned, and applied TF-IDF vectorization. **MB:** Trained Logistic Regression model**.** **ME:** Accuracy, F1 score, confusion matrix, and classification report.	**HT:** GridSearchCV **R:** Tuned regularization strength using (C), and L2	Regularization controls overfitting, ensuring model generalizes well to new data.
Multinomial Naive Bayes	pandas, scikit-learn (MultinomialNB, train_test_split, GridSearchCV, classification_report, confusion matrix), matplotlib.pyplot, seaborn	**DP:** Applied TF-IDF vectorization**.** **MB:** Trained MultinomialNB model. **ME:** Accuracy, F1 score, confusion matrix, and classification report.	**HT:** GridSearchCV Hyperparameter **R:** Tuned smoothing parameter using (alpha).	Proper tuning of alpha helps manages the influence of rare features, improving prediction reliability.
Support Vector Machine (SVM)	pandas, scikit-learn (SVC, train_test_split, classification_report,	**DP:** Applied TF-IDF vectorization.	**HT:** GridSearchCV. **R:** Adjusted kernel type and	Selecting the right kernel and C value enhances the SVM's ability to handle non-linear data and avoid overfitting.

	confusion_matrix), matplotlib.pyplot, seaborn	**MB:** Trained SVC model. **ME:** Accuracy, F1 score, confusion matrix, and classification report.	regularization parameter using (C)	
Recurrent Neural Network (RNN)		**DP:** Tokenized and padded sequences. **MB:** Constructed RNN using Sequential, Embedding, and SimpleRNN layers. **MT:** Compiled and trained model. **ME:** Used accuracy, F1 score, confusion matrix. ROC-AUC, PR-AUC.		Regularization techniques are essential to avoid overfitting and ensure reliable performance on unseen data.
Long Short-Term Memory (LSTM)	tensorflow.keras (layers, models, preprocessing), NumPy, matplotlib.pyplot	**DP:** Tokenized and padded sequences. **MB:** Built LSTM model using Sequential, Embedding, and LSTM layers. **MT:** Compiled model, trained with suitable loss function and optimizer. **ME:** Used accuracy, F1 score, confusion matrix, ROC-AUC, PR-AUC.	**HT:** Tuned batch size (64, 128), Additional Model Layer **R:** Additional Dropout rate 20%, L2 Regularization	Proper tuning of hyperparameters is crucial for achieving optimal performance and efficiency.
Gated Recurrent Unit (GRU)		**Data Pre-processing:** Tokenized and padded sequences. **Model Building:** Designed GRU model using Sequential, Embedding, and GRU layers.		Regularization techniques help control the model's complexity, ensuring good generalization to new data.

		MT: Compiled model, trained with suitable configurations **ME:** Used accuracy, F1 score, confusion matrix, ROC-AUC, PR-AUC.		

Model Evaluation

Each model was evaluated using accuracy and F1 scores. Among traditional machine learning models, the Support Vector Machines (SVM) achieved the highest performance with an accuracy of 1.00 and an F1 score of 1.00. In the deep learning category, the Long Short-term Memory (LSTM) model outperformed others with an accuracy of 0.989 and an F1 score of 0.99. Fine-tuning techniques such as hyperparameter tuning and regularization were used to observe model performance. The results were visualized using confusion matrices to show classification performance and training history plots to illustrate the learning process of the deep learning models.

Table 5: Naive Bayes Multinomial Model Result

Models	Accuracy	F1-Score	Precision	Recall
Baseline Model	95%	95%	95%	95%
Regularization using (Alpha - 0.01)	96%	96%	96%	96%
Hyperparameter (Grid Search CV)	96%	96%	96%	96%

The regularized Naive Bayes model, after hyperparameter tuning with GridSearchCV, achieved an accuracy, F1-score, precision, and recall of 96%, indicating a slight improvement over the baseline model and regularization. The confusion matrix (Figure 7, Right) confirms balanced performance across all classes.

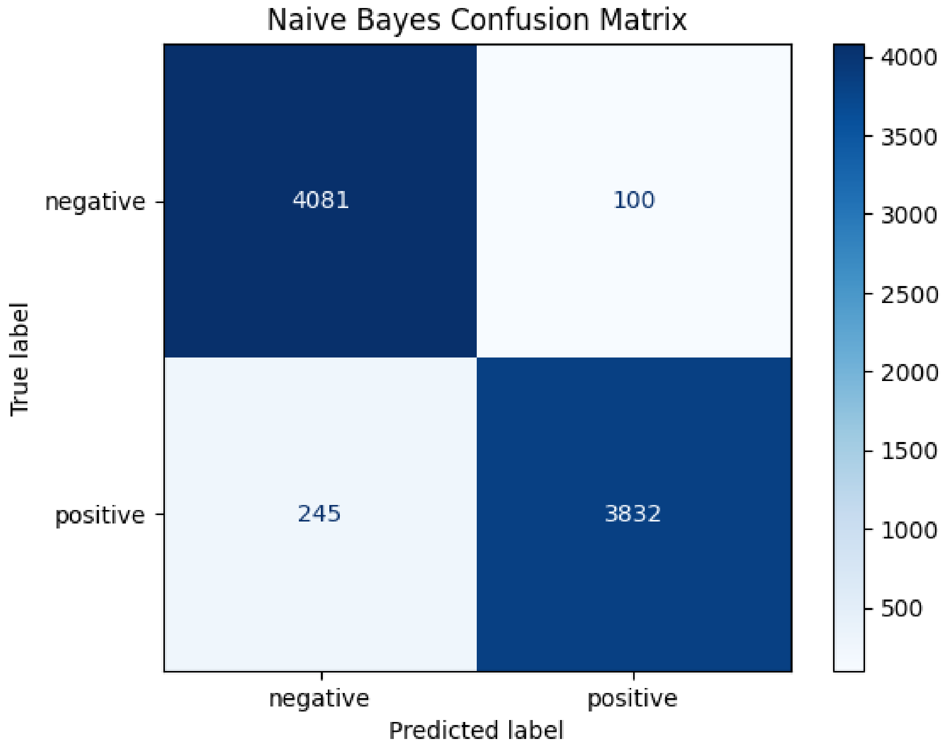

Figure 7: Best Confusion Matrix for Naive Bayes

Table 6: Logistics Regression Classifier Model Result

Models	Accuracy	F1-Score	Precision	Recall
Baseline Model	97%	97%	97%	97%
Regularization using (C-100, L2, Liblinear)	97%	97%	97%	97%
Hyperparameter (GridSearchCV)	99%	99%	99%	99%

The regularized Logistic Regression model, optimized with GridSearchCV, outperformed the baseline, and regularized models with an accuracy, F1-score, precision, and recall of 99%. The best confusion matrix (Figure 8) shows near-perfect classification performance.

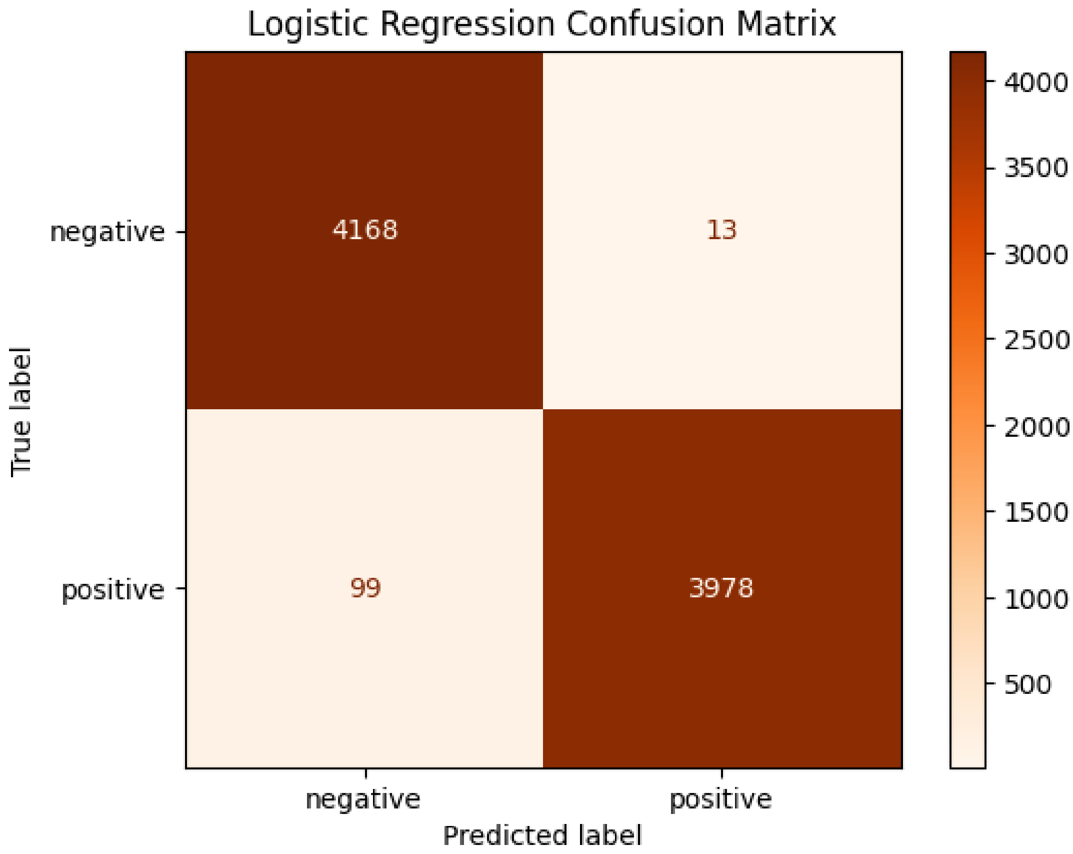

Figure 8: Best Logistics Regression Confusion Metrics (Hyperparameter)

Table 7: Support Vector Machines Model Result

Models	Accuracy	F1-Score	Precision	Recall
Baseline Model	98%	98%	98%	98%
Regularization using (C -10, scale, rbf)	99%	99%	99%	99%
Hyperparameter (GridSearchCV)	100%	100%	100%	100%

The regularized SVM model, fine-tuned with GridSearchCV, achieved perfect scores across all metrics (accuracy, F1-score, precision, and recall of 100%), making it the best-performing traditional model. The confusion matrix (Figure 9) demonstrates flawless classification accuracy.

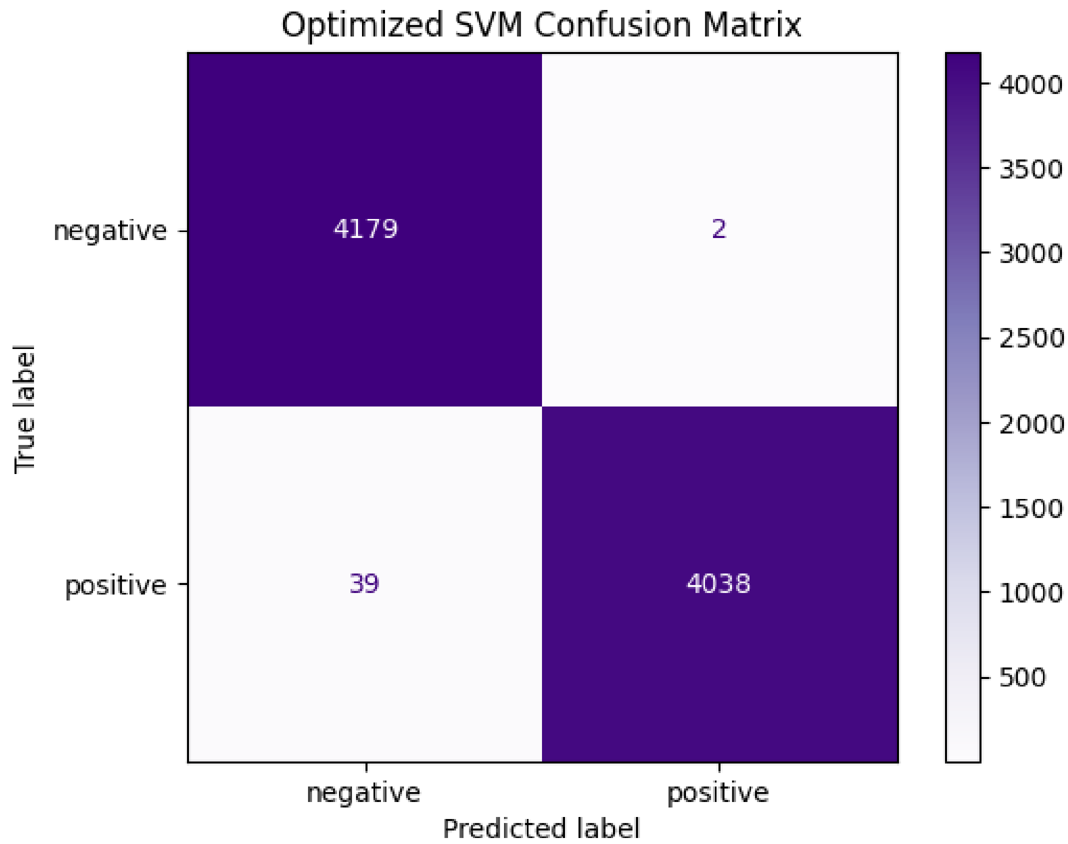

Figure 9: Best Confusion SVM Metrics using Hyperparameter.

Table 8: Deep learning Model Long Short-term Memory (LSTM) Result

Models	Accuracy	F1-Score	Precision	Recall	ROC-AUC	PR-AUC
Baseline Model	98.7%	99%	99%	99%	99.6%	99.5%
Regularization using (Dropout rate = 0.2, L2 Regularization)	91.9%	92%	92%	92%	97.1%	96.9%
Hyperparameter (Extra Layer, Batch size 128)	98.9%	99%	99%	99%	99.7%	99.6%

The LSTM model, after hyperparameter tuning with a batch size of 124, reached an accuracy of 98.9%, with F1-score, precision, and recall at 99%, and ROC-AUC and PR-AUC scores of 99.7% and 99.6%, respectively. This indicates superior performance compared to the baseline and regularized models.

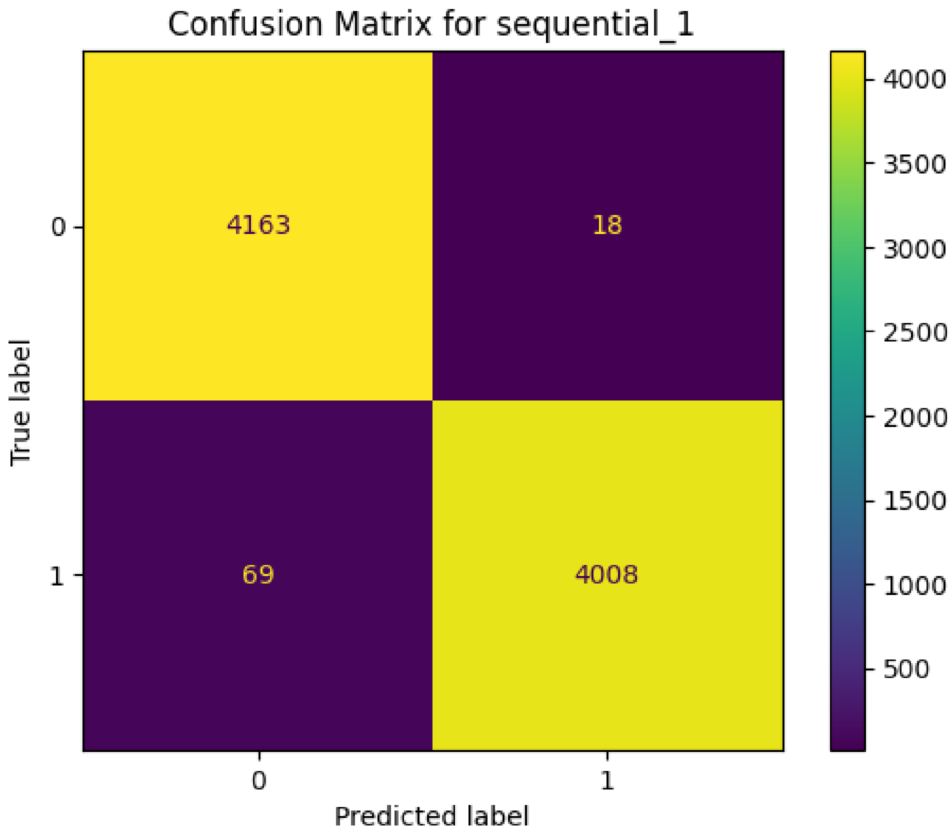

Figure 10: Best Confusion LSTM Metrics using Hyperparameter.

Table 9: Deep learning Model Gated Recurrent Unit (GRU) Result

Models	Accuracy	F1-Score	Precision	Recall	ROC-AUC	PR-AUC
Baseline Model	98.9%	99%	99%	99%	99.5%	99.5%
Regularization using (Dropout rate = 0.2, L2 Regularization)	96.3%	96%	96%	96%	99.0%	99.0%
Hyperparameter (Extra Layer, Batch size 128)	98.8%	99%	99%	99%	99.6%	99.5%

The GRU model, fine-tuned similarly to the LSTM with a batch size of 124, achieved an accuracy of 98.8%, with F1-score, precision, and recall at 99%, and ROC-AUC and PR-AUC scores of 99.6% and 99.5%, respectively, demonstrating robust performance.

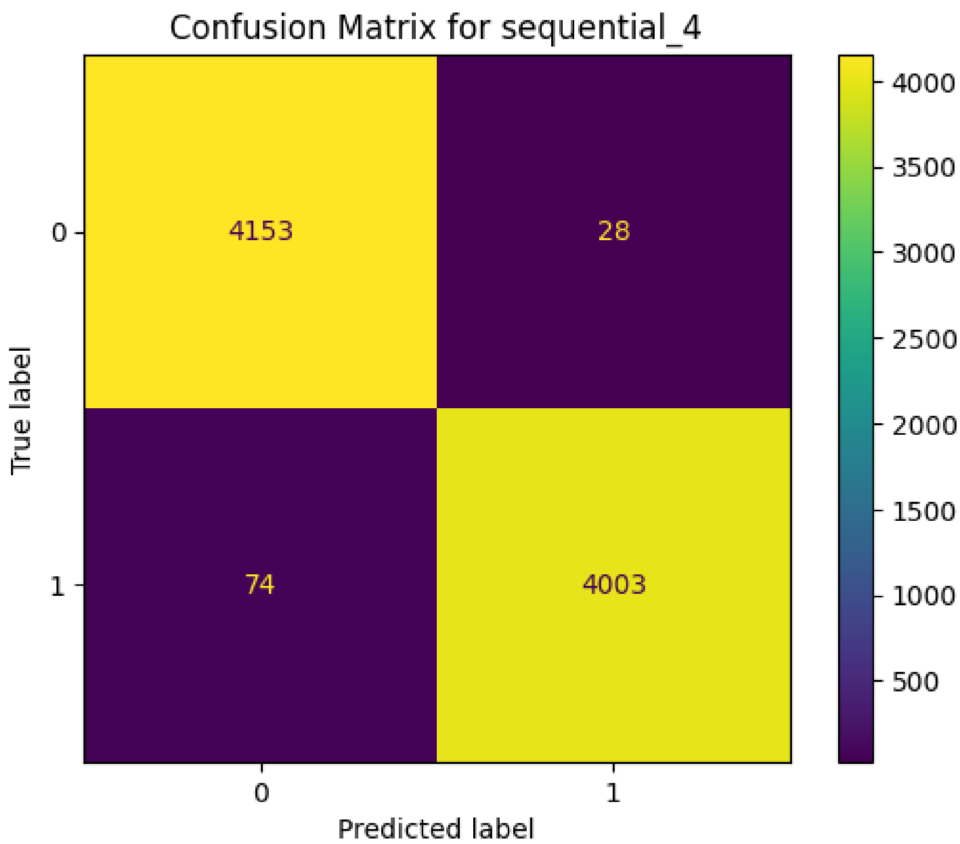

Figure 11: Best Confusion GRU Metrics using Hyperparameter.

Table 10: Deep learning Model Recurrent Neural Networks (RNN) Result

Models	Accuracy	F1-Score	Precision	Recall	ROC-AUC	PR-AUC
Baseline Model	97.3%	97%	97%	97%	99.3%	99.2%
Regularization using (Dropout rate = 0.2, L2 Regularization)	97.1%	97%	97%	97%	99.2%	99.1%
Hyperparameter (Extra Layer, Batch size 128)	98%	98%	98%	98%	99.3%	99.3%

The RNN model, with hyperparameter tuning, showed an accuracy of 98%, F1-score, precision, and recall at 98%, and ROC-AUC and PR-AUC scores of 99.3%, confirming its effectiveness, although slightly lower than LSTM and GRU.

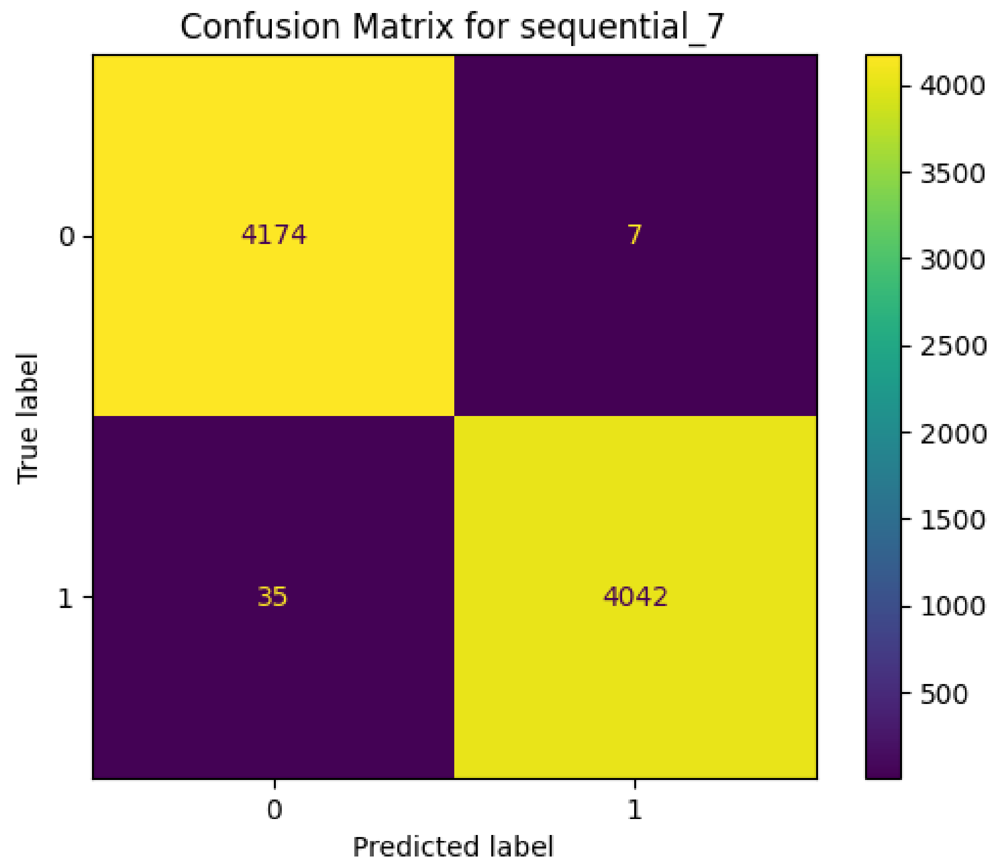

Figure 12: Best Confusion RNN Metrics using Hyperparameter.

In academic writing, the MLA style, used in the humanities, features an author-page citation format and a Works Cited list, focusing less on publication dates (Modern Language Association, 2016). APA style, prevalent in psychology and social sciences, emphasizes recent research using an author-date citation format but has complex rules (American Psychological Association, 2020). Chicago style, common in history and business, allows flexibility with its Notes and Bibliography or Author-Date systems, but requires detailed formatting (University of Chicago Press, 2017). Harvard style, popular in the UK and Australia, combines simplicity and detail in its author-date format, though it varies slightly (Pears & Shields, 2019)). The Vancouver style, used in medicine and science, utilizes a numerical citation system that is compact but less direct for sourcing (International Committee of Medical Journal Editors, 2020). For this report, Harvard style is adopted due to its simplicity, balanced information, and versatility.

References

Ahmad, H.O. & Umar, S.U. (2023) Sentiment Analysis of Financial Textual data Using Machine Learning and Deep Learning Models. *Informatica (Slovenia)*, 47(5), 153–158. Available online: https://doi.org/10.31449/inf.v47i5.4673.

Aslam, R.W., Shu, H., Naz, I., Quddoos, A., Yaseen, A., Gulshad, K. & Alarifi, S.S. (2024) Machine Learning-Based Wetland Vulnerability Assessment in the Sindh Province Ramsar Site Using Remote Sensing Data. *Remote Sensing*, 16(5). Available online: https://doi.org/10.3390/rs16050928.

Breiman, L. (2001) *Random Forests*. Machine Learning, 45, pp.5-32.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. (2014) Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Available online: http://arxiv.org/abs/1406.1078.

Chung, J., Gulcehre, C., Cho, K. & Bengio, Y. (2014) Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Available online: http://arxiv.org/abs/1412.3555.

Cortes, C. (1995) *Support-Vector Networks*. Machine Learning, 20(3), pp.273-297.

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Available online: http://arxiv.org/abs/1810.04805.

Hochreiter, S. & Schmidhuber, J. (1997) LONG SHORT-TERM MEMORY. *Long Short-Term Memory. Neural Computational,* [Preprint].

Hosmer, D.W., & Lemeshow, S., 2000. Applied Logistic Regression. 2nd ed. New York: Wiley.

Joachims, T. (1998) *Text Categorization with Support Vector Machines: Learning with Many Relevant Features*.

Kim, Y. (2014) Convolutional Neural Networks for Sentence Classification. Available online: http://arxiv.org/abs/1408.5882.

Li, L., Zhu, L., Huang, F., Wang, D., Li, X., Wu, T. & He, Y. (2024) Post-Disaster Emergency Communications Enhanced by Drones and Non-Orthogonal Multiple Access: Three-Dimensional Deployment Optimization and Spectrum Allocation. *Drones*, 8(2). Available online: https://doi.org/10.3390/drones8020063.

Mccallum, A. & Nigam, K. (1998) *A Comparison of Event Models for Naive Bayes Text Classification*.

Mikolov, T., Karafiát, M., Burget, L., Honza, J.'', Cernock ́ycernock ́y, '' ˇ & Khudanpur, S. (2010) *Recurrent neural network based language model*.
Pang, B., Lee, L. & Vaithyanathan, S. (2002) *Thumbs up? Sentiment Classification using Machine Learning Techniques*. EMNLP. Available online: http://www.cs.cornell.edu/people/pabo/movie-review-data/.

Pears, R. & Shields, G. (2019) *Cite them right: the essential guide to referencing and plagiarism*.

Rish, I. (2001) *An Empirical Study of the Naïve Bayes Classifier An empirical study of the naive Bayes classifier*. Available online: https://www.researchgate.net/publication/228845263.

Tang, D., Qin, B. & Liu, T. (2015) *Document Modeling with Gated Recurrent Neural Network for Sentiment Classification*. Association for Computational Linguistics. Available online: http://ir.hit.edu.cn/.

Modern Language Association. (2016). MLA Handbook (8th ed.). *Modern Language Association of America.* Available online: https://owl.purdue.edu/owl/research_and_citation/mla_style/mla_style_introduction.html

American Psychological Association. (2020). Publication Manual of the American Psychological Association (7th ed.). *American Psychological Association.* Available online: https://www.scribbr.com/apa-style/apa-seventh-edition-changes/

University of Chicago Press. (2017). The Chicago Manual of Style (17th ed.). University of Chicago Press. Available online: https://www.chicagomanualofstyle.org/tools_citationguide.html

International Committee of Medical Journal Editors. (2020). Recommendations for the Conduct, Reporting, Editing, and Publication of Scholarly Work in Medical Journals. Accessed online: http://www.icmje.org/icmje-recommendations.pdf