JOURNAL OF INFORMATION AND COMMUNICATION TECHNOLOGY
https://e-journal.uum.edu.my/index.php/jict

How to cite this article:
Ouadad, R., & Mouncif, H. (2025). Exploring deep learning techniques for sentiment analysis in online education platforms: A case study of Coursera reviews. *Journal of Information and Communication Technology*, *24*(4), 1-35. https://doi.org/10.32890/jict2025.24.4.1

Exploring Deep Learning Techniques for Sentiment Analysis in Online Education Platforms: A Case Study of Coursera Reviews

[1]Raja Ouadad & [2]Hicham Mouncif
LIMATI Laboratory, Mathematics and Informatics Department,
Polydisciplinary Faculty, Sultan Moulay Slimane University, Morocco

[*1]raja.ouadad@usms.ac.ma
[2]h.mouncif@usms.ac.ma
*Corresponding author

Received: 27/1/2025 Revised: 10/5/2025 Accepted: 3/8/2025 Published: 31/10/2025

ABSTRACT

The rise of online courses, accelerated by the COVID-19 pandemic, has underscored the need for effective educational models capable of addressing the challenges posed by remote learning. This study focuses on the development of sentiment classifiers using the Coursera reviews dataset to evaluate the polarity of student feedback. This research improved student engagement and support in online education by applying sophisticated sentiment analysis techniques. We explored a comprehensive methodology encompassing various pre-processing techniques, advanced tokenisation methods, and a range of deep learning architectures, including Feedforward Neural Networks (FNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRUs), and Bidirectional Encoder Representations from Transformers (BERT)-based models. Each model's performance is optimised through meticulous hyperparameter tuning using the Optuna framework. Results indicated that BERT is the best model, achieving a recall of 97.50% and an accuracy of 96.83%, while Bidirectional LSTM (BiLSTM) closely followed with a recall of 96.55% and an accuracy of 96.71%. In contrast, simpler models like FNN and RNN exhibited lower accuracy (92.83% and 87.83%, respectively). These findings underscore the importance of advanced models in capturing contextual meanings and highlight the effectiveness of leveraging embeddings, attention mechanisms, and tailored pre-processing strategies, which significantly improve sentiment classification performance.

Keywords: Sentiment analysis, online education, deep learning, hyperparameter tuning.

INTRODUCTION

Digital platforms and social media have become essential tools for modern communication, enabling billions of users to express their views and share information on diverse subjects (Abuhammad & Ahmed, 2024). Among the many other applications, these platforms have played a significant role in the rise of online education. Online courses have emerged as a widely adopted educational model, allowing students to balance their studies alongside other commitments (Osmanoğlu et al., 2020). The COVID-19 pandemic further underscored the significance of online education, as institutions worldwide transitioned from traditional face-to-face teaching to remote learning (Washington et al., 2021). However, this rapid shift presented substantial challenges, risking the quality and effectiveness of the educational experience (Washington et al., 2021).

Massive Open Online Courses (MOOCs) offer accessible educational materials to students globally, utilising various resources such as videos, slides, texts, quizzes, and discussion forums to foster interaction between learners and instructors (Onan, 2021). Despite their potential, research indicated that MOOCs experience significantly lower completion rates compared to traditional in-person courses (Onan, 2021). To mitigate this issue, studies emphasise the importance of tutor interventions based on feedback and sentiments expressed by students in forums. Sentiment analysis, powered by machine learning and natural language processing, enables the detection of emotions like confusion or boredom, allowing for timely support (Altrabsheh et al., 2014). This information can also inform adaptive learning systems, offering tailored interventions, personalised content, and alerts based on individual student behaviour (Madani et al., 2020).

Sentiment analysis is the process of identifying and interpreting emotions from text, yielding insights into the feelings and opinions of individuals (Chumwatana, 2018). However, this task is particularly challenging, as emotions are often embedded in context and subtle linguistic cues rather than explicit terms like “confused” or “happy” (Baqach & Battou, 2021). Studies have shown that machine learning and deep learning models outperform traditional lexicon-based methods, which rely on labour-intensive manual feature engineering to extract emotions from text effectively. Sentiment analysis algorithms encompass a broad spectrum, ranging from traditional machine learning techniques such as Naïve Bayes, Decision Trees, and Support Vector Machines (SVMs) to advanced deep learning models like Recurrent Neural Networks (RNNs) (Birjali et al., 2021). Deep learning algorithms excel in automatically extracting meaningful features from textual data, allowing them to deliver strong performance and often surpass traditional machine learning models in various studies (Cen et al., 2020; Sultana et al., 2018). Before delving into RNNs, it is essential to understand Feedforward Neural Networks (FNNs). FNNs transmit information in one direction—from the input layer to the output layer—without retaining the memory of past inputs. While effective for tasks such as text classification, FNNs struggle with sequential data, making capturing relationships between distant words within a sentence challenging.

In contrast, RNNs are specifically designed to manage sequential data by maintaining hidden states that store information from previous inputs. However, they face the vanishing gradient problem, where gradients can diminish during backpropagation, impeding effective learning and the ability to capture long-term dependencies, especially when semantically related words are separated in a text. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models were developed to address these challenges. Both models utilised gating mechanisms that enable selective retention or discarding of information from previous inputs, enhancing their capacity to capture long-range dependencies. The LSTM model features a memory cell designed for prolonged information storage, while the GRU model employs a more streamlined gating mechanism, enhancing computational efficiency.

Recently, the transformer model, which leverages an attention mechanism, has shown remarkable success in natural language processing. The attention mechanism allows the model to assign varying levels of importance to different segments of the input sequence, facilitating the creation of informative embeddings. It computes a weighted sum of the input embeddings, with the weights derived from a learned compatibility function that evaluates the relationship between query and key embeddings. This approach effectively captures long-range dependencies within the input sequence, resulting in richer and more informative representations (Bahdanau et al., 2014). Considering this, a structured methodology was implemented to optimise the performance of sentiment classifiers utilising the Coursera reviews dataset. The process began with various pre-processing techniques, including text normalisation, stop word removal, and applying of stemming or lemmatisation to refine the dataset.

Advanced tokenisation methods were employed to convert the text into suitable formats for analysis while effectively handling diverse vocabulary. Each architecture's top-performing models underwent a thorough analysis of pre-processing techniques and hyperparameter tuning using the Optuna framework. The FNNs utilised Stanford's Global Vectors for Word Representation (GloVe) embedding (Pennington et al., 2014) for enhanced semantic representation, while the RNN models incorporated multiheaded attention mechanisms inspired by Vaswani et al. (2023). Additionally, LSTM and GRU cells were integrated with attention mechanisms to capture essential patterns in the input sequences. Lastly, a Bidirectional Encoder Representations from Transformers (BERT)-based model sourced from Hugging Face was explored as a powerful approach for sentiment classification, demonstrating the effectiveness of various methodologies in improving model performance.

The remainder of this paper is structured as follows: The following section reviews related work in the field, then details the materials and methods used in this study, including the dataset, pre-processing techniques, and the deep learning models employed for sentiment analysis. The results and analysis section presents a comparative analysis and discussion of the findings, highlighting the performance of each model and providing insights into their strengths and weaknesses. The paper concludes by summarising the study's key insights and implications and outlining directions for future research in the field.

RELATED WORKS

This section reviews state-of-the-art approaches to sentiment analysis, particularly in educational settings like MOOCs and e-learning platforms. Recent research highlights the growing reliance on deep learning techniques to assess student satisfaction, identify challenges, and enhance learning experiences. Key advancements include the integration of recurrent, convolutional, and transformer-based architectures, which excel at capturing complex patterns and emotions in student feedback. Efforts to improve pre-processing, hyperparameter tuning, and attention mechanisms further contribute to the development of robust and generalised sentiment classifiers. A summary of the related works is presented in Table 1.

Kandhro et al. (2019) employed LSTMs with word embeddings for analysing student feedback, achieving superior results over traditional models like SVM and Naïve Bayes by addressing contextual limitations. Zhang et al. (2022) proposed a BERT+BiGRU model for polarity classification, demonstrating its superiority over baseline machine learning and deep learning models utilising Word2Vec embeddings. Similarly, Kastrati et al. (2020) developed an aspect-based sentiment analysis framework using the convolutional neural network (CNN)+FastText, achieving high F1 scores for MOOC-specific categories.

Onan (2021) conducted a comprehensive analysis of 66,000 MOOC reviews, revealing the efficacy of ensemble learning methods and LSTMs with GloVe embeddings for sentiment prediction. Pu et al. (2021) introduced an ensemble model combining CNNs and BiLSTMs with Word2Vec and GloVe, optimised via multi-objective gray wolf optimisation, achieving over 91% F1 scores on Chinese course evaluations. Mrhar et al. (2021) employed a Bayesian CNN-LSTM model to examine 140,320 Coursera course reviews, reaching an accuracy of 91%. Their approach established a connection between the sentiment expressed in forum posts and both dropout likelihood and course performance. Additionally, Chen et al. (2022) and Tzeng et al. (2022) leveraged deep learning for sentiment classification and clustering, providing actionable insights for improving MOOCs based on feedback analysis. Peng et al. (2022) extended sentiment analysis to emotion classification, using CNNs, BiLSTMs, and attention mechanisms to categorise emotions accurately. Koufakou (2024) explored advanced NLP models like BERT, RoBERTa, and XLNet, with RoBERTa excelling in sentiment polarity detection and SVM in topic classification.

Jebbari et al. (2025) proposed a hybrid neural network for sentiment analysis in MOOC discussion forums, combining BiLSTM for word-level processing and CNN for capturing character-level patterns. The model achieved a high accuracy of 93.11%, highlighting the effectiveness of hybrid deep learning architectures in analysing sentiment in educational contexts. In alignment with these advances, Ananthi Claral Mary (2025) proposed a CNN-LSTM hybrid model that demonstrated high accuracy in analysing student feedback from online learning platforms. This work highlights how deep learning can effectively interpret unstructured student input to support improvements in teaching practices. Their findings underscore the importance of hyperparameter optimisation and advanced NLP techniques in refining sentiment analysis for educational settings. Despite significant advancements, challenges remained in sentiment analysis, including limited annotated datasets, a lack of context-aware methods, and poor generalisability across platforms. Additionally, many models lack interpretability, hindering actionable insights. This study addresses these gaps by leveraging advanced deep-learning architectures and pre-processing techniques to enhance sentiment analysis and extract meaningful insights from student feedback in MOOCs and e-learning platforms.

Table 1

Summary of the Related Works

Authors	Dataset	Models	Focus	Key Findings
Kandhro et al. (2019)	3000 student feedback responses from 30 courses (2017-2018), collected via Google Forms	LSTM	Teacher evaluation through LSTM-based sentiment analysis using student feedback.	Achieved state-of-the-art accuracy by identifying optimal hyperparameters, outperforming traditional methods.
Zheng et al. (2020)	25,000 user reviews from Chinese online education platforms	BERT, BiGRU, Word2Vec	Classifying sentence polarity	BERT+BiGRU outperformed all other models, highlighting BERT's superiority over Word2Vec.

(continued)

Authors	Dataset	Models	Focus	Key Findings
Kastrati et al. (2020)	21,940 reviews from 15 Coursera courses.	Deep learning network, CNN, FastText	Aspect-based sentiment analysis	Framework achieved F1 scores of 86.13% for aspect classification and 82.10% for sentiment classification.
Onan et al. (2021)	66,000 reviews from MOOCs	Machine learning, ensemble learning, LSTM	Analysis of MOOC reviews	Ensemble methods outperformed traditional models; LSTM achieved the best predictive performance among deep learning models.
Pu et al. (2021)	20,240 student evaluations with emotion labels from a Chinese online course platform	Word2Vec, GloVe, Bidirectional LSTM, CNN	Sentiment analysis of online evaluations	A novel ensemble model achieved an F1 score of over 91%, outperforming seven comparison models.
Mrhar et al. (2021)	140,320 Coursera course reviews	Bayesian CNN-LSTM	Sentiment analysis in MOOCs	Outperforms traditional CNN, LSTM, and CNN-LSTM models in accuracy, precision, recall, and F1-score; strong correlation found between forum sentiment and dropout rates.
Chen et al. (2022)	186,738 reviews from Coursera and edX	Deep learning model	Learner perceptions of MOOCs	Found generally positive views of MOOCs, but negative feedback focused on technical issues and course design.
Tzeng et al. (2022)	1,786 learner feedback responses from four MOOCs	A five-layer Deep Neural Network (DNN) model with a rectified linear unit (ReLU) activation function	Analysing and categorising student feedback	Deep learning system provides insights into student perceptions, aiding course improvement.
Peng et al. (2022)	10,000 comments from online teaching evaluations	CNN, BiLSTM, BiLSTM attention mechanism	Emotion classification in comments	CNN-BLSTM Model significantly outperformed baseline methods in classifying emotions into eight categories.
Zyout et al. (2024)	Hybrid dataset (3,820 student comments: 2,773 formal + 1,047 ChatGPT-generated)	BERT + Bi-LSTM + Attention	Evaluation of embedding techniques with attention mechanisms for sentiment analysis	BERT-based model with Bi-LSTM and attention achieved top F-scores (89%, 88%), outperforming other methods (67%–69%).

(continued)

Authors	Dataset	Models	Focus	Key Findings
Koufakou (2024)	10,610 publicly available course reviews from various bootcamp-type courses on CourseReport	BERT, RoBERTa, XLNet, SVM	Sentiment analysis and topic detection	RoBERTa excelled in sentiment detection (95.5% accuracy); SVM was best for topic classification (79.8% accuracy).
Jebbari et al. (2025)	24684 anonymised MOOC forum posts	BiLSTM + CNN	Sentiment analysis using a hybrid deep learning model at the word and character levels	Achieved 93.11% accuracy; effectively captures linguistic nuances; demonstrates strong potential of hybrid models in online education sentiment analysis
Ananthi Claral Mary (2025)	2,650 learner reviews from VLEs in India (1415 positive, 1010 neutral, 225 negative)	CNN, LSTM, CNN-LSTM	Sentiment prediction to support the optimisation of teaching methods	Hybrid CNN-LSTM outperformed CNN and LSTM in accuracy, precision, recall, and F1-score; achieved 97% accuracy.

MATERIALS AND METHODS

Material

This section provides an overview of the resources used in this study, focusing on the dataset and tools that form the foundation of the sentiment analysis process. We begin by describing the Coursera reviews dataset, including its structure, characteristics, and any pre-processing steps applied to enhance data quality. Additionally, the tools, libraries, and computational resources employed in the study are outlined, as they play a critical role in building, training, and evaluating the deep learning models used for sentiment classification.

Tools and Development Environment

Our experiments were implemented using PyTorch along with the TorchText library. PyTorch is a flexible deep-learning framework that offers dynamic computation graphs, making it particularly suitable for complex model building and experimentation. TorchText provides essential tools for text processing, enabling efficient handling of natural language data. For hyperparameter optimisation, we employed Optuna, an efficient and flexible framework for automated hyperparameter tuning. Optuna's adaptive search algorithms allow it to find optimal parameters more quickly than traditional methods, which is essential for improving model performance without exhaustive manual tuning. For our development environment, we utilised Google Colaboratory (Colab), which supports running code on both GPU and CPU, providing a robust setup for deep learning tasks.

Dataset Description

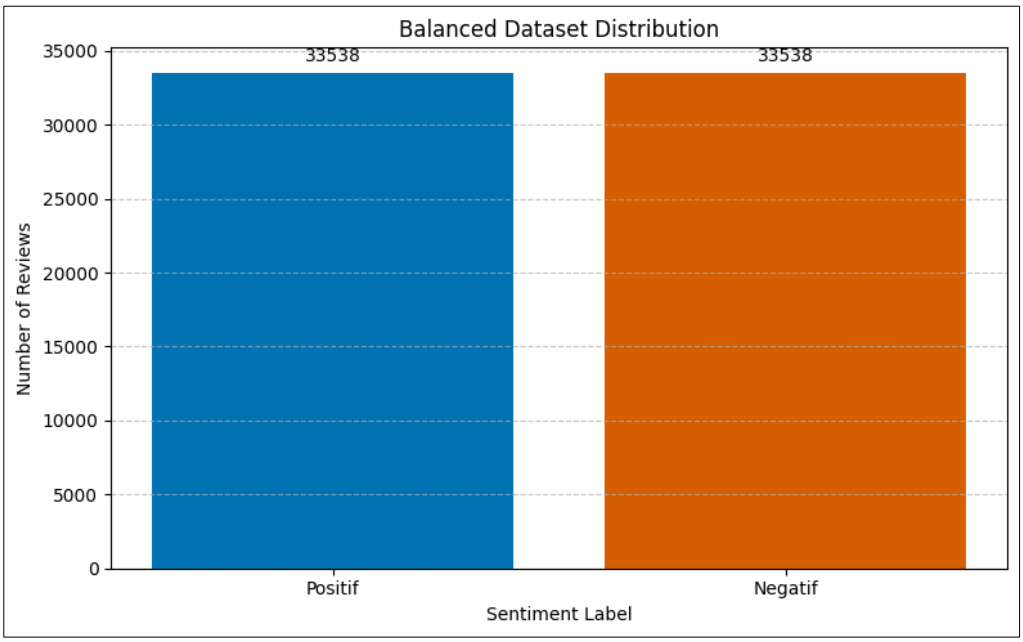
We built a text corpus for the sentiment analysis by scraping course reviews from Coursera.org, a well-known MOOC platform, using the Python library BeautifulSoup. The initial dataset consisted of 85,270 reviews from various domains. Each review on Coursera includes a five-star rating, contributing to the

overall course rating. To generate labelled data, we assigned a label of 1 (positive) to reviews with more than four stars and 0 (negative) to those with fewer than two stars. Reviews with three stars, non-alphabetic content, or written in languages other than English were removed. Language detection was performed using Google's "langdetect"_library in Python. After these pre-processing steps, the final dataset contained 75,542 reviews. The exclusion of three-star reviews was made to reduce ambiguity and ensure a clearer distinction between positive and negative sentiment, which is beneficial for binary classification. However, we acknowledge that this approach may introduce certain limitations. Neutral reviews often contain nuanced opinions, and their exclusion may reduce the representativeness of the dataset. Additionally, using star ratings as sentiment labels assumes a direct alignment between the numeric rating and the textual sentiment, which may not always hold true in practice. These considerations highlight potential biases introduced during pre-processing, which are important to keep in mind when interpreting the results.

After collecting the data, we observed that the corpus was imbalanced. To address this, we applied under-sampling to achieve an approximately equal number of positive and negative reviews. As a result, the final dataset consisted of 67,076 MOOC reviews, with 33,538 labelled as negative and 33,538 as positive. Figure 1 shows the distribution of the balanced dataset, illustrating the split between positive and negative labels.

Figure 1

Balanced Dataset Distribution



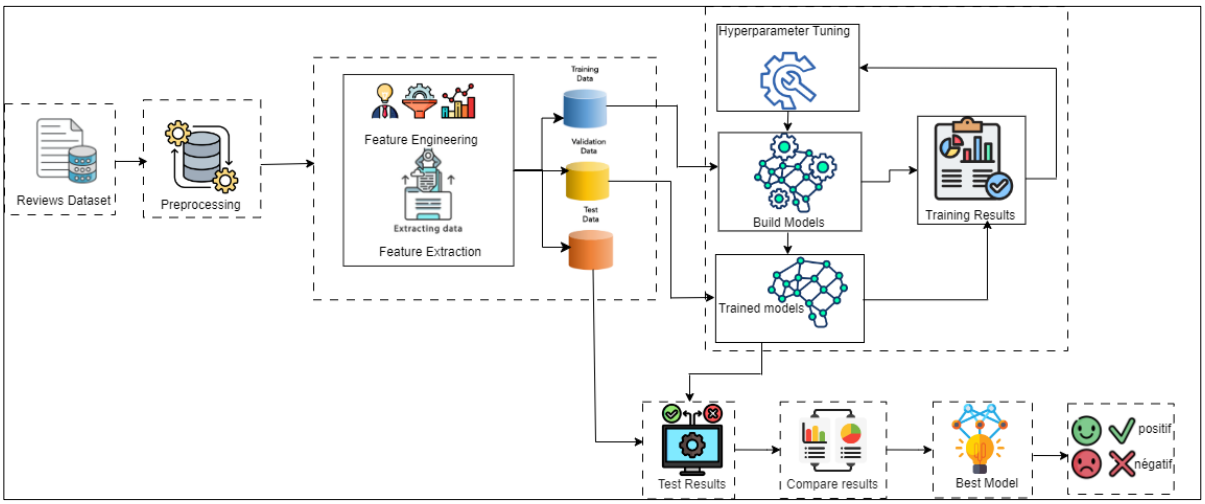
Method

This section outlines the development process of deep learning models for sentiment analysis, leveraging architectures such as FNN, RNN, LSTM, BiLSTM, GRU with attention mechanisms, and BERT. The Coursera reviews dataset is used, and pre-processing, tokenisation, and feature extraction techniques are applied to prepare the data. As illustrated in Figure 2, the workflow begins with data pre-processing, followed by feature engineering and splitting the data into training, validation, and test sets.

The models are built and optimised through hyperparameter tuning to enhance performance. Finally, the results are compared to identify the best model capable of accurately predicting the sentiment polarity, either positive or negative.

Figure 2

Workflow for Developing Deep Learning Models for Sentiment Analysis

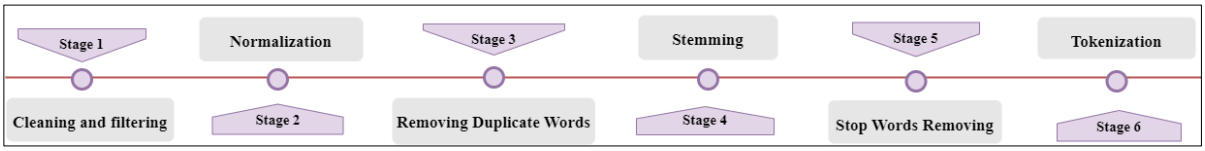


Data Pre-processing

Pre-processing is a crucial step in data preparation, involving removing noisy and non-informative elements (e.g., hashtags) to enhance classification accuracy (Ouadad & Mouncif, 2024). Data from online education platforms and courses is often unstructured, so thorough cleansing is necessary to gain meaningful insights and achieve better results. Key steps include tokenisation, which splits text into tokens using separators like spaces or commas; stemming, which reduces words to their root forms by removing suffixes; and normalisation, which converts informal text into standardised formats to identify word variations. Stop word removal eliminates low-value words, such as conjunctions and prepositions, that do not affect sentiment or emotion classification. This streamlined process enhances outcomes and ensures reliable text analysis, as illustrated in Figure 3.

Figure 3

Diagram of the Pre-processing Phase



Data Splitting

We split our dataset into three distinct subsets: training, validation, and testing. Eighty-two percent of the samples were allocated for training to ensure a robust evaluation of model performance, providing the models with substantial data to learn from. This larger training set enhances the models' learning

capacity and improves their generalisation capabilities. Subsequently, 10% of the samples were reserved for validation, allowing for effective hyperparameter tuning and model optimisation. Finally, the remaining 8% of the samples were designated for testing, ensuring an unbiased assessment of the models' ability to generalise to unseen data. Table 2 presents the specific number of samples allocated to each subset. This stratified approach helps mitigate overfitting and enhances the reliability of the results. By using an 82-10-8 split, the study benefits from increased training data while still maintaining sufficient samples for validation and testing, thereby striking a balance that supports effective model development and evaluation.

Table 2

Dataset Splitting for Training, Validation, and Testing

Subset	Percentage	Number of Reviews
Training	82%	55,000
Validation	10%	6,708
Testing	8%	5,366

Feature Extraction

Data must be integrated since deep learning frameworks cannot directly process raw inputs. Various embedding techniques are available to achieve this. The embedding layer assigns a vector representation to each word in the input text, capturing its semantic relationship with other words. Word embedding is a powerful word-level representation capable of capturing the semantic meaning of words. Each word is mapped to a fixed-size vector containing values representing various semantic features, initialised randomly and adjusted during training. Several established techniques have been developed for this purpose. One common approach is one-hot encoding, where each word is represented as a 2D vector of zeros with a value of 1 at the position corresponding to its index (Baqach & Battou, 2021).

Deep learning techniques have shown superior performance in this task. Models like Word2Vec train neural networks to capture word relationships by encoding each word as a vector. It allows mathematical measures, such as cosine similarity, to quantify the semantic similarity between words effectively. In our work, we utilised GloVe embeddings, specifically with 100 dimensions. GloVe is a popular word embedding technique that captures the global statistical information of words in a corpus by representing them as dense vectors. We employed two different techniques for embedding usage:

i. Approach 1: This technique involves embedding each word's vectors.
ii. Approach 2: This method utilises the sum or average of word vectors to represent the overall meaning of a review.

Deep Learning Models

This section outlines the results of our experiments, highlighting both the initial and optimised versions of the algorithms implemented for sentiment classification. We utilised the following models: FNN, RNN, LSTM, Bidirectional LSTMs (BiLSTM), Gated Recurrent Units (GRU) with attention mechanisms, and BERT-based architectures. The choice of these algorithms was driven by their proven effectiveness in handling sequential data, capturing contextual information, and understanding nuanced

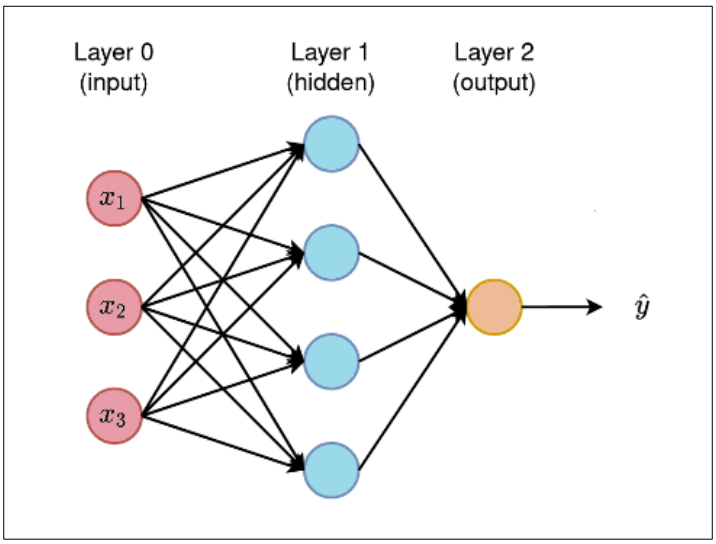
sentiments expressed in text. For each model, we provide detailed descriptions of the parameters used, with a focus on how attention mechanisms were incorporated to enhance the models' abilities to identify key patterns in the input sequences.

Feedforward Neural Network (FNN)

FNNs have garnered significant interest in recent years due to their adaptable architecture and strong capacity to represent complex patterns (Zhang et al., 2022). As a type of artificial neural network, FNNs facilitate a unidirectional flow of information, progressing from the input layer to the output layer without cycles or feedback loops. The architecture consists of an input layer, one or more hidden layers, and an output layer, with neurons in each layer fully connected to the next. This design is appreciated for its simplicity, flexibility, and ability to learn intricate input-output relationships. Figure 4 illustrates the architecture of an FNN as applied to sentiment analysis. It depicts the flow of information from the input layer, which represents textual data, through hidden layers that capture complex patterns, ultimately reaching the output layer that predicts sentiment polarity. The interconnections among the neurons enable the model to effectively learn and classify sentiments as positive, negative, or neutral. This structural design is fundamental to the effectiveness of FNNs in processing and analysing sentiment in textual data.

Figure 4

The Structure of a FNN



Building upon this architecture, we implemented an FNN for sentiment classification to evaluate its performance on our dataset. We explored two techniques for representing input data: the padding-based approach, which ensures consistent sequence lengths, and the mean-based approach, which averages word embeddings across sequences. Both methods employed 100-dimensional GloVe embeddings to capture the semantic relationships within the text. To optimise the model's performance, we leveraged Automated Optimal Model Search and Learning Curves, ensuring the selection of the best configuration based on these input strategies. The following sections detail the optimised hyperparameters and performance metrics for each method.

Padding-based Approach

In this method, each review is represented by the first 100 words (after stopwords and irrelevant words are removed by the tokeniser). If the review contains fewer words, zero-padding is applied to ensure all input vectors have the same length. This allows the model to account for word order and positional relevance. Table 3 presents the key hyperparameters used in the padding-based approach. These parameters provide essential insights into the model's design and training configuration, ensuring consistency compared to other models.

Table 3

Hyperparameters for the Padding-based Approach

Parameter	Value
Optimiser	Stochastic Gradient Descent (SGD)
Learning Rate	0.02
Loss Function	Cross-Entropy Loss
Batch Size	512
Epochs	20
Hidden Layers	2
Hidden Layer Sizes	512, 256 neurons
Activation Function	Sigmoid
Embedding Dimension	100

The FNN demonstrated commendable performance, achieving 85.61% accuracy and a macro-average F1_score as detailed in Table 4. The confusion matrix in Figure 5 shows that the model accurately identified 87% of negative sentiments and 84% of positive sentiments, with misclassifications of 13% and 16%, respectively. Additionally, the accompanying Receiver Operating Characteristic (ROC) curve demonstrates the model's effectiveness, achieving an Area Under the Curve (AUC) of 0.856, highlighting its strong ability to distinguish between positive and negative sentiments.

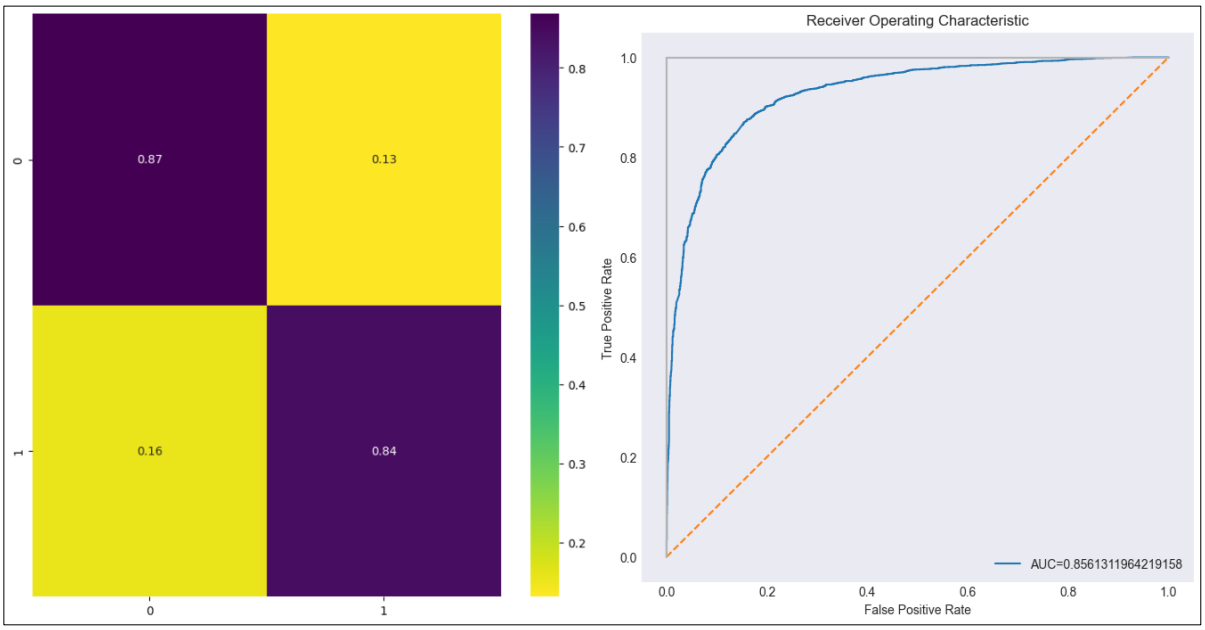
Table 4

Initial FNN Model Evaluation Metrics

Metric	Value
Total Training Time	5.21 minutes
Correct Predictions	4594 / 5366
Mean Loss	0.46
Recall	85.61%
Precision	86.48%
Accuracy	85.61%
Macro Averaged F1-score	85.61%

Figure 5

Confusion Matrix and ROC Curve of the Initial FNN Model



Mean-based Approach

Each review is represented in this method by averaging the word vectors into a single "review-level" vector. This representation ensures that longer reviews do not exert a disproportionate influence compared to shorter ones, making the input vectors comparable across all reviews. This approach aligns with Wieting et al. (2015), which demonstrated that averaging vectors yields superior performance over summing them. The mean-based approach also simplifies input data, enhancing computational efficiency while effectively capturing overall sentiment. However, this method has its limitations; it overlooks contextual nuances by not retaining information about word order or specific word combinations. This trade-off highlights the need to balance between computational efficiency and the ability to convey sentiment effectively. The observation justified the choice to average vectors, as longer reviews often produced larger vector magnitudes that did not necessarily correlate with the sentiment expressed. Shorter reviews can convey similar sentiments, making it reasonable to average the vectors for a uniform dimensional representation.

To implement this approach, we developed a function to calculate the average vector for each review, generating a single 300-dimensional embedding. We subsequently created datasets and data loaders with a batch size of 512, maintaining consistency with prior methods. The model was then trained using the same parameters, with the only variation being the input to the sequential layer. The results obtained from the Mean-Based Approach are summarised in Table 5.

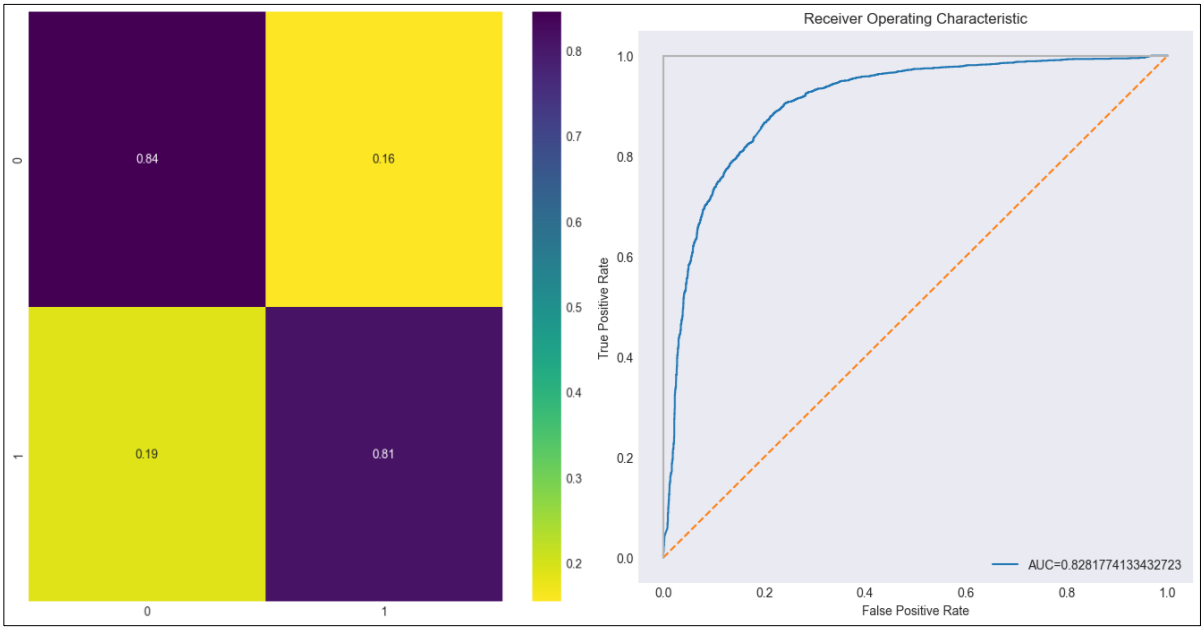
Table 5

FNN Mean Approach Evaluation Metrics

Metric	Value
Training Time	39.61 seconds
Predicted Correctly	4444 / 5366
Mean Loss	0.49
Recall	82.82%
Precision	83.96%
Accuracy	82.82%
Macro Averaged F1-score	82.81%

Figure 6

Mean-based Confusion Matrix and ROC Curve



The mean-based method demonstrated impressive computational efficiency, completing training in just 39.61 seconds, as shown in Table 5. The model accurately predicted 4444 out of 5366 instances, yielding an accuracy of 82.82% and a mean loss of 0.49, with recall and precision recorded at 82.82% and 83.96%, respectively. While these metrics are closely aligned with the padding-based approach, the mean-based method offers a significant advantage in execution time, being three times faster due to the reduced data size. It is suggested that increasing the permissible word length or dimensionality of embeddings in the padding approach could further extend execution time. Given its comparable performance in effectively capturing sentiment, we adopted the mean-based methodology for subsequent experiments.

Automated Optimal Model Search and Learning Curves

To streamline identifying the optimal model during training, we developed a function that trains the model over several epochs and saves the version with the highest macro-averaged F1 score for subsequent evaluation. This function also generates learning curves illustrating each epoch's training and validation mean loss. Before hyperparameter tuning, we assessed seven activation functions tailored to this neural network architecture, modifying the configuration based on previous models. Each activation function was trained and evaluated on the test set under consistent conditions, ensured by establishing a random seed before testing. Metrics and learning curves were recorded and compiled into a Pandas DataFrame for comprehensive model performance analysis. Table 6 summarises the performance of various activation functions used during neural network training.

Table 6

Performance Comparison of Activation Functions in Neural Network Training

Activation Function	Training Loss - Validation Loss Average Difference	Best F1 Score Achieved After (Epochs)	Predicted Correctly From 5366	Mean Loss	Recall (%)	Precision (%)	Accuracy (%)	Macro Averaged F1-score (%)	Time secondes
ReLU	0.000364967	19	4043	0.68	75.34	76.36	75.34	75.34	50.98
Leaky ReLU	0.000366970	19	4046	0.68	75.40	76.45	75.40	75.39	54.28
Exponential Linear Unit (ELU)	0.001800489	29	4166	0.58	77.64	81.35	77.64	77.56	57.65
Sigmoid Linear Units (SiLU)	0.000155436	29	4060	0.69	75.66	76.86	75.66	75.65	68.77
Hardswish	0.000153633	29	4067	0.69	75.79	76.82	75.79	75.78	66.71
ReLU6	0.000364967	19	4043	0.68	75.34	76.36	75.34	75.34	57.53
PReLU	0.000452119	29	4069	0.67	75.83	80.13	75.83	75.71	64.28

ELU achieved the highest F1-score and accuracy but exhibited instability, with fluctuating F1-scores across epochs and inconsistent performance between training and validation sets, raising concerns about its generalisation. Additionally, ELU's inability to produce zero outputs may limit its effectiveness for sentiment analysis. In contrast, ReLU demonstrated more stable behaviour, with minimal differences between training and validation losses, and maintained a consistent recall of 80%. Its stability without overfitting or underfitting makes it a more reliable choice. Consequently, we selected ReLU for subsequent hyperparameter tuning and model optimisation. To refine our model further, we utilised Optuna for hyperparameter optimisation, testing various configurations to identify the optimal architecture and training parameters. Table 7 summarises the best hyperparameters identified through this process, along with their search space and relevant results.

Table 7

Optimised Hyperparameters for FNN Architecture

Hyperparameter	Search Space	Best Value
Number of Layers	1 to 5	2
Number of Nodes	2 to 512	512
Activation Function	ReLU, ELU, Leaky ReLU, etc.	ReLU
Dropout Rate	0.01 to 0.3	0.38
Learning Rate	Wide range	0.001
Optimiser	Adam, SGD, RMSProp, Adagrad	Adam
Training Epochs	Fixed	30

The evaluation of the optimised model, as detailed in Table 8, revealed a training loss and validation loss average difference of only 0.0166, indicating strong generalisation. The model achieved an impressive accuracy of 92.83%, with a recall of 92.83% and a precision of 92.46%. Figure 7 presents the confusion matrix, showcasing a high true positive rate alongside the ROC curve with an impressive AUC of 0.93, confirming the model's strong ability to discriminate between sentiment classes. These results demonstrate that the optimised architecture enhances performance and ensures reliability in sentiment classification tasks.

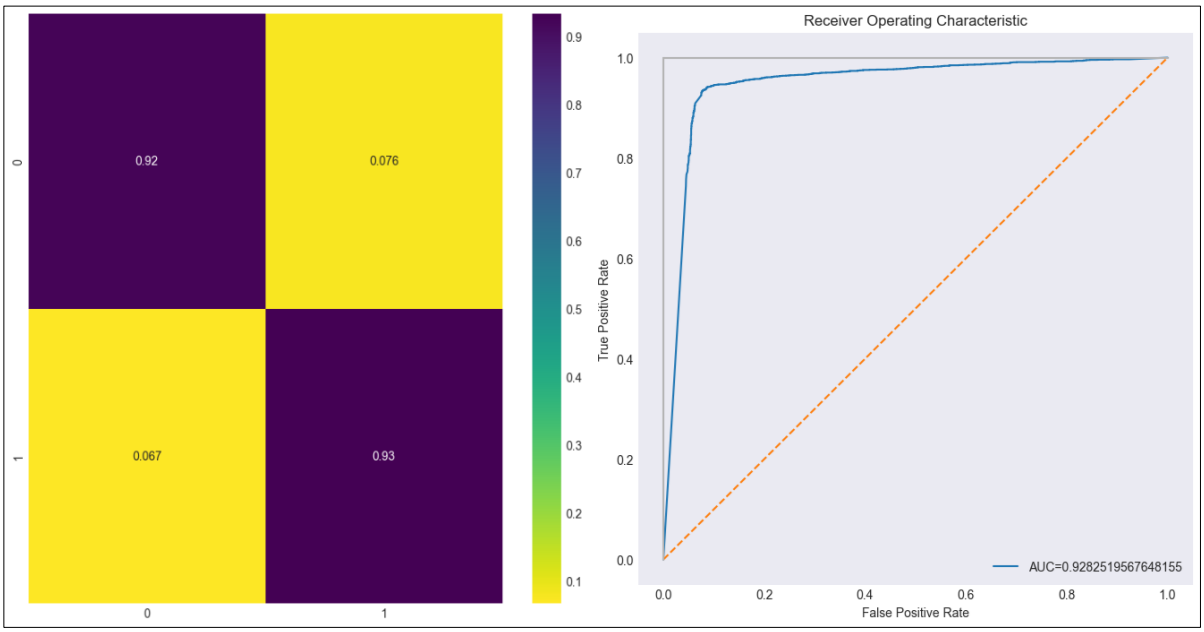
Table 8

Evaluation Results of the Optimised FNN

Metric	Value
Training Time	8.94 minutes
Training Loss - Validation Loss Avg	0.0166
Predicted Correctly	4981 / 5366
Mean Loss	0.39
Recall	92.83%
Precision	92.46%
Accuracy	92.83%
Macro Averaged F1-Score	92.83%

Figure 7

Confusion Matrix and ROC Curve of the Optimised FNN

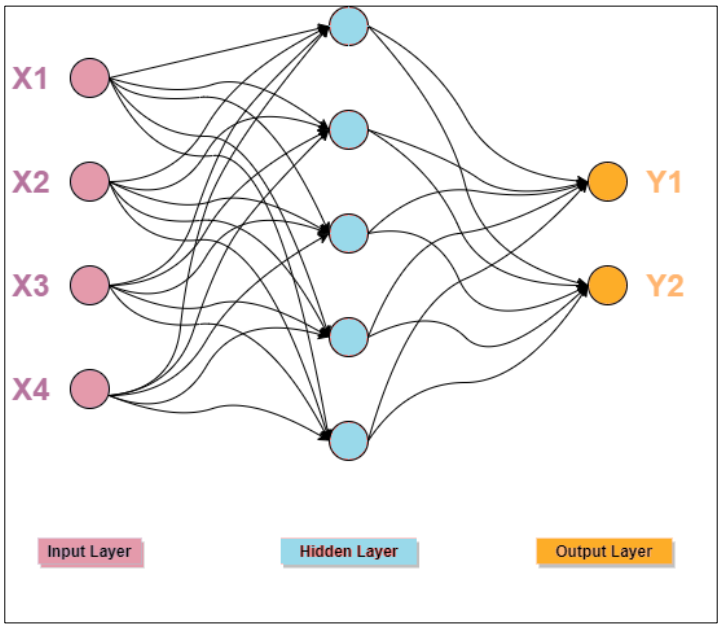


Recurrent Neural Network (RNN)

RNNs are specialised to handle sequential data by retaining a hidden state that reflects information from prior inputs, enabling the network to capture temporal dependencies (Mienye et al., 2024). Figure 8 illustrates the structure of an RNN as applied to sentiment analysis. In this architecture, the input layer (X1 to X4) represents word embeddings derived from tokens in a text sequence, such as individual words in a review. The hidden layer captures context by taking input not only from the current step but also from previous time steps, maintaining "memory" across the sequence—essential for processing phrases with nuanced meaning (e.g., "not good"). Finally, the output layer (Y1, Y2) generates predictions after processing the sequence, providing probabilities or class predictions, such as positive or negative sentiment. This structure allows RNNs to handle sequential dependencies, making them well-suited for tasks like sentiment analysis.

Figure 8

The Structure of an RNN



In the context of our study, we implemented an RNN architecture for sentiment classification to assess its efficacy on our dataset. Given the sequential characteristics inherent to textual data, we selected a padding approach for generating word embedding instead of employing the mean-based technique, which had shown superior performance with Feedforward Neural Networks (FNNs). This choice is particularly crucial as RNNs leverage vector representations for each word, which are instrumental in preserving the context and sequence of the input data. The word embedding is trained on extensive datasets and effectively captures the semantic relationships among words, thus serving as a fundamental component for contextual analysis within the RNN framework (Mikolov et al., 2013; Hochreiter & Schmidhuber, 1997). We experimented with a straightforward RNN utilising PyTorch's RNN class, which implements a multi-layer Elman RNN with tanh non-linearity, as shown in Equation 1.

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \quad (1)$$

This formulation allows the network to maintain a hidden state that encapsulates information from previous time steps, enabling the model to capture the temporal dependencies inherent in sequential data. Our initial tests employed two RNN layers with a hidden size of 128 over twenty epochs, providing a foundation for further refinement and exploration of advanced architectures. Table 9 outlines the performance metrics achieved during the training process, clearly comparing the RNN's effectiveness under the specified configuration.

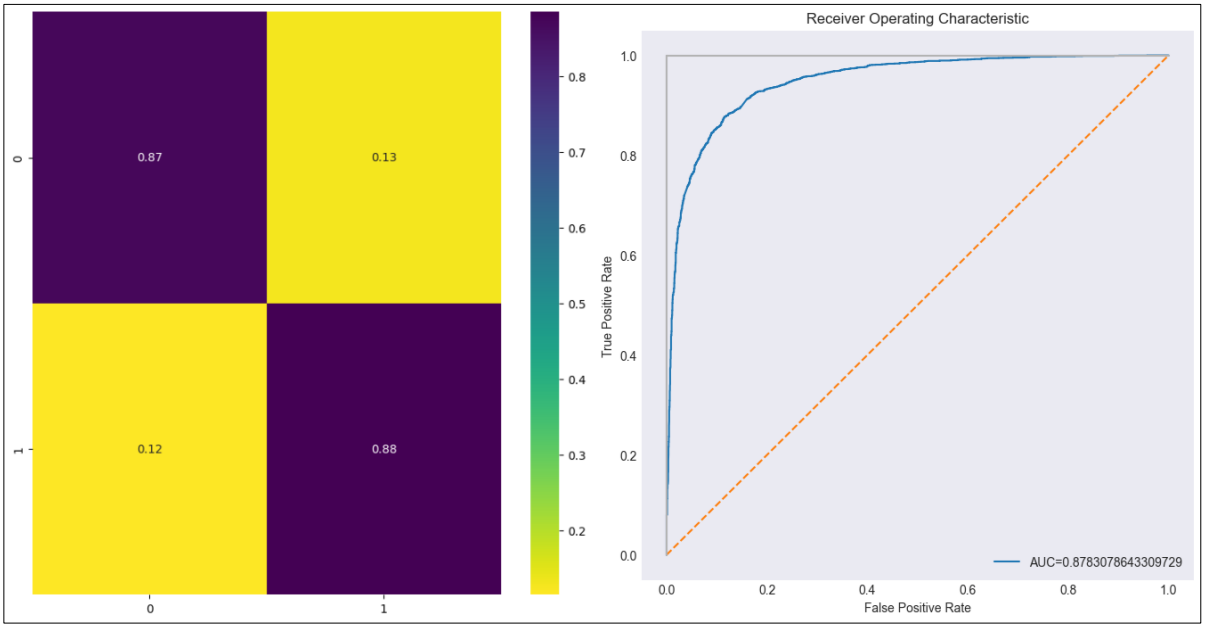
Table 9

Evaluation Results of RNN

Metrics	Value
Training Time	05.07 minutes
Predicted Correctly	4713 / 5366
Mean Loss	0.29
Recall	88.48%
Precision	87.34%
Accuracy	87.83%
Macro Averaged F1-score	87.83%

Figure 9

Confusion Matrix and ROC Curve of the RNN Model



The evaluation of the RNN model, as summarised in the results, demonstrated a commendable accuracy of 87.83%, a recall of 88.48% and a precision of 87.34%. These metrics indicate that the model is proficient in identifying both positive and negative sentiments within the dataset. Figure 9 highlights the confusion matrix, indicating a high true positive rate and effective classification, while the ROC curve with an AUC of 0.878 confirms the model's ability to distinguish between sentiment classes. These results demonstrate that the basic RNN architecture delivers reliable performance even without extensive fine-tuning, providing a strong foundation for future enhancements and optimisations.

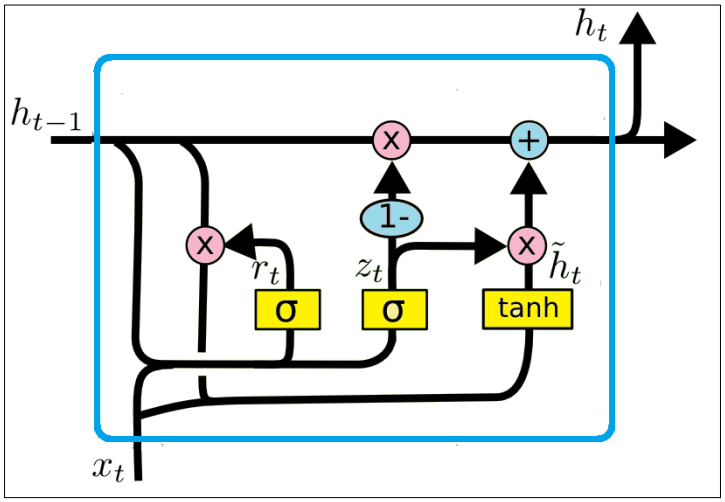
Gated Recurrent Unit (GRU)

GRUs are a type of RNN architecture specifically designed to handle sequential data by maintaining a hidden state that reflects information from prior inputs, thus capturing temporal dependencies essential for various applications, including sentiment analysis (Dey & Salem, 2017). Figure 10 illustrates the

internal architecture of a GRU. Two key gates govern the flow of information: the update gate (z_t) determines how much of the previous hidden state h_{t-1} is carried forward, allowing the model to retain long-term dependencies. The reset gate (r_t) controls how much past information is forgotten by modulating the influence of h_{t-1} when computing the candidate hidden state $\tilde{h}_t$. These dynamics are formalised in Equation 2, where the GRU ensures efficient temporal dependency capture while mitigating the vanishing gradient problem, making it highly effective for sequential tasks like sentiment analysis.

Figure 10

The GRU Architecture



$$
\begin{aligned}
z_t &= \sigma(W_z \cdot [h_{t-1}, x_t]) \\
r_t &= \sigma(W_r \cdot [h_{t-1}, x_t]) \\
\tilde{h}_t &= \tanh(W \cdot [r_t * h_{t-1}, x_t]) \\
h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t
\end{aligned}
\tag{2}
$$

We implemented a simple Gated Recurrent Unit (GRU) network utilising PyTorch's built-in GRU class, which provides an efficient and flexible framework for constructing recurrent architecture. This network comprised a single GRU layer that employed the following computations to process sequential data. Where z_t is the update gate and h_t is the candidate activation. This formulation enabled the GRU to effectively capture temporal dependencies in sequential data while maintaining computational efficiency, a characteristic that is particularly beneficial for tasks like sentiment analysis and time series forecasting (Dey & Salem, 2017; Chung et al., 2014). The design of GRUs allows them to mitigate the vanishing gradient problem commonly associated with traditional RNNs, making them suitable for processing longer sequences. Additionally, the GRU architecture balances between performance and computational requirements, which is advantageous in practical applications (Bahdanau et al., 2014). Table 10 presents the performance metrics obtained, providing insight into the initial GRU model's effectiveness within the given configuration.

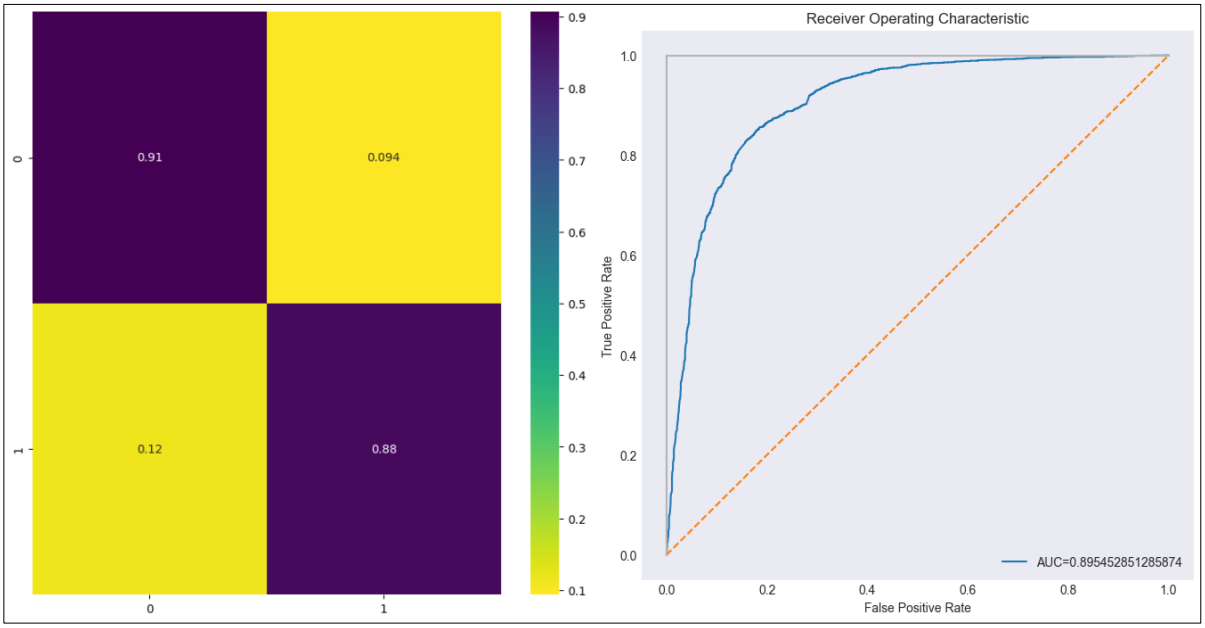
Table 10

Performance Results of Simple GRU

Metrics	Value
Training Time	23.14 minutes
Predicted Correctly	4805 / 5366
Mean Loss	0.25
Recall	88.45%
Precision	90.43%
Accuracy	89.55%
Macro Averaged F1-score	89.54%

Figure 11

Confusion Matrix and ROC Curve of the Initial GRU Model



The initial GRU model delivered promising results, achieving 89.55% accuracy, 88.45% recall, and 90.43% precision, resulting in a balanced F1 score of 89.54%. The confusion matrix (Figure 11) confirms effective classification with minimal errors, while the ROC curve (Figure 11) demonstrates strong discriminatory power with an AUC of 0.895. These metrics highlight the GRU's efficiency in capturing sequential patterns, providing a robust baseline for future optimisation. Before implementing network enhancements, we conducted a final optimisation using Optuna to identify improved hyperparameters for the learning rate and hidden state size in the current model configuration. The selected GRU model employs a bidirectional structure and stacks multiple GRU layers, enhancing its learning capacity. This architecture processes input sequences in both forward and reverse directions, allowing the network to capture contextual information from both past and future elements, making it effective in understanding nuanced patterns in sequential data, such as sentiment expressed in reviews.

In addition, the model incorporates multiheaded attention mechanisms, enabling it to focus on different parts of the input sequence simultaneously, which improves its ability to capture important features and dependencies. This combination of bidirectional GRUs and attention mechanisms significantly enhances the model's capability to understand complex relationships within the data, leading to more accurate predictions. Such architecture strikes a balance between performance and efficiency, making it particularly well-suited for tasks like sentiment analysis, where both temporal dependencies and specific word associations play a critical role in classification. Table 11 summarises the optimal hyperparameters used for the optimal GRU model.

Table 11

Optimised Hyperparameters for Optimal GRU Architecture

Hyperparameter	Search Space	Best Value
Learning Rate (LR)	0.001, 0.005, 0.01, 0.1	0.0045
Epochs	1,3, 5, 10, 20	3
Input Size	50, 100, 200	100
Embedding Size	50, 100, 200, 300	100
Hidden Size	16, 32, 48, 64	48
Layers	1, 2, 3, 4	2
Dropout Rate	0.1, 0.2, 0.3	0.15
Clip Value	0.5, 0.7, 1.0	0.7
Heads	1, 2, 3, 4	3
Classes	binary classification	2

The evaluation of the best GRU model, as presented in Table 12, demonstrates a significant performance improvement. The model achieved an accuracy of 95.50%, with a recall of 95.50% and a precision of 94.64%, indicating a strong capability to identify both positive and negative sentiments correctly. The macro-averaged F1_score of 95.50% confirms that the model effectively balances precision and recall across both classes. With a mean loss of 0.36, the model shows minimal error during predictions. Additionally, the training process was completed in 33.74 minutes, highlighting computational efficiency. The confusion matrix (Figure 12) confirms effective classification with minimal errors, while the ROC curve (Figure 12) highlights strong discriminatory power with an AUC of 0.954. These results suggest that the optimised GRU network performs exceptionally well with the chosen parameters and sets a robust benchmark for future improvements.

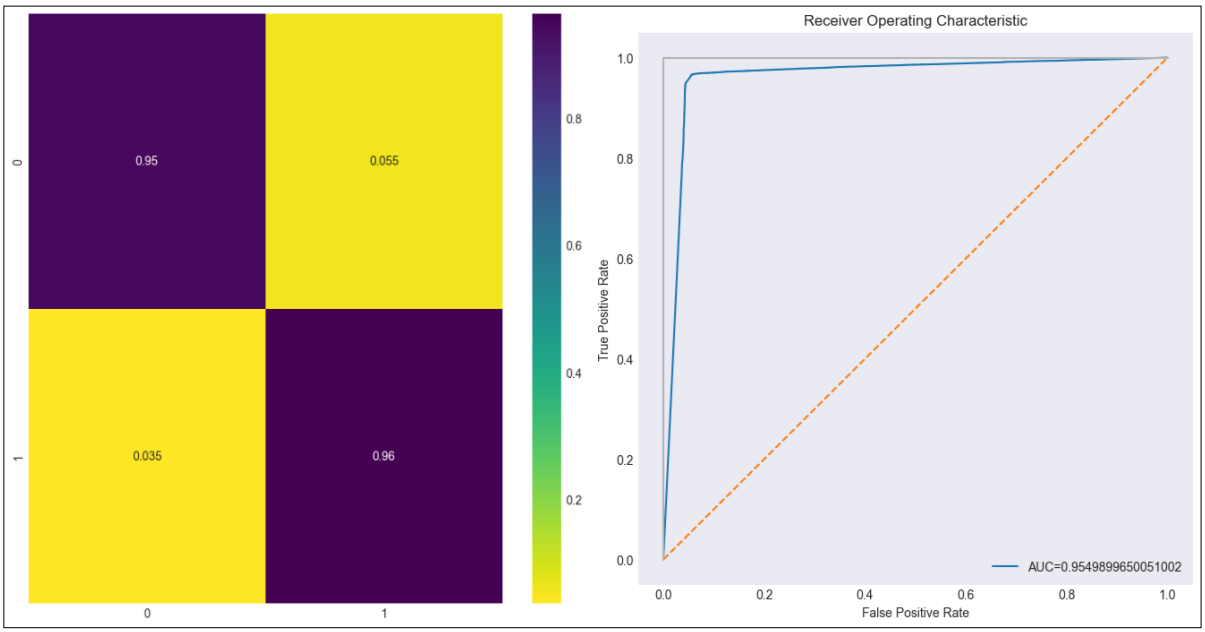
Table 12

Evaluation Results for the Best GRU Model

Metric	Value
Training Time	33.74 minutes
Mean Loss	0.36
Recall	95.50%
Precision	94.64%
Accuracy	95.50%
Macro Averaged F1-score	95.50%

Figure 12

Confusion Matrix and ROC Curve of the Best GRU Model

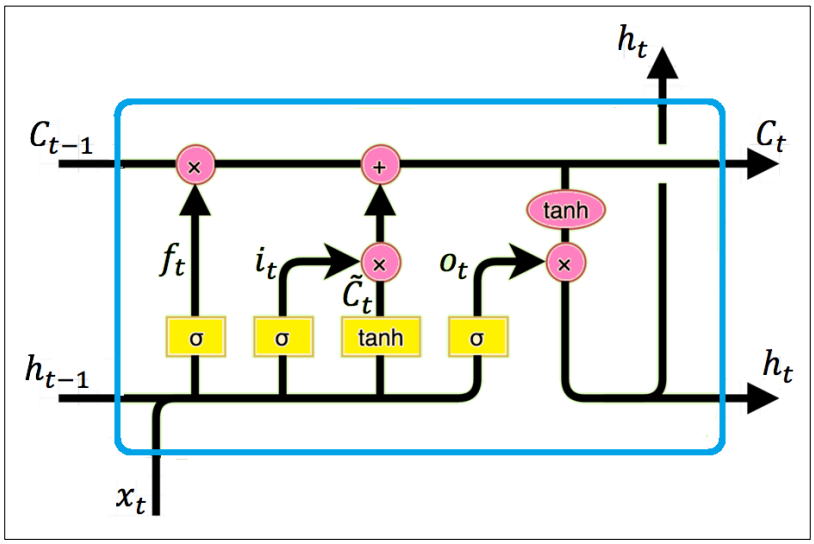


Long Short-term Memory (LSTM)

LSTM networks, developed by Pennington et al. (2014), address the vanishing gradient problem common in standard RNNs by introducing gating mechanisms that control the flow of information. These gates allow LSTMs to maintain and update their internal state over extended sequences, making them effective for modelling long-term dependencies. As shown in Figure 13 and explained in Equation 3, the LSTM cell consists of key components: the forget gate (f_t) determines which parts of the previous cell state (c_{t-1}) to discard, while the input gate (i_t) decides what new information, represented by the candidate cell state ($\tilde{c}_t$), should be added. These gates collaboratively update the cell state (c_t), which serves as long-term memory. The output gate (o_t) filters the current output and produces the hidden state (h_t), which is passed to the next step and used for predictions. This process also incorporates the Hadamard product ($\odot$), which regulates the flow and retention of information. This structure enables LSTMs to effectively capture both short- and long-term dependencies, making them ideal for tasks like sentiment analysis, where understanding sequential patterns and context is critical.

Figure 13

The Architecture of LSTM Network



$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}x_{t-1} + W_{ci}K \odot c_{t-1} + b_i) \\
f_t &= \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}K \odot c_{t-1} + b_f\right) \\
c_t &= f_t \odot K\, c_{t-1} + i_t \odot K\, tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}Kc_t + b_o) \\
h_t &= o_t \odot Ktanh(c_t)
\end{aligned}
\tag{3}
$$

After careful experimentation, Table 13 presents the optimised hyperparameters identified for the LSTM architecture. These values were selected to enhance the model's performance by balancing accuracy and efficiency within the defined search space. Each hyperparameter was tuned to ensure the network can effectively capture sequential dependencies while maintaining stability during training.

Table 13

Optimised Hyperparameters for LSTM Architecture

Hyperparameter	Search Space	Best Value
Input Size (INPUT)	Length of input sequences	300
Hidden Units (HIDDEN)	64, 128, 256, 264	64
Learning Rate (LR)	0.001, 0.005, 0.01, 0.03	0.005
Epochs (EPOCHS)	3, 5, 10	3
Number of Layers	1, 2, 3, 4	4
Embedding Size	50, 100, 300	100
Dropout Rate (DROPOUT)	0.1, 0.15, 0.2, 0.3	0.15
Clip Value	0.5, 0.7, 0.9	0.9
Heads (Attention)	1, 2, 3	3
Optimiser	Adam, SGD, RMSprop	Adam
Loss Function	CrossEntropy, MSELoss	CrossEntropyLoss

Table 14 presents the performance metrics of the optimised LSTM model. The model correctly predicted 4973 out of 5366 samples, achieving a mean loss of 0.20. It demonstrated robust performance with a recall of 93.37%, precision of 92.10%, and an accuracy of 92.68%, indicating its ability to balance between identifying true positives and minimising false positives effectively. The macro-average F1-score of 92.68% reflects strong performance across both classes. The confusion matrix (Figure 14) reveals that 92% of the negative class (label 0) and 93% of the positive class (label 1) were correctly identified, further reinforcing the model's reliability in sentiment classification. Following this, the ROC curve (Figure 14) will further confirm the optimised LSTM's discrimination ability between the two classes, complementing the detailed evaluation.

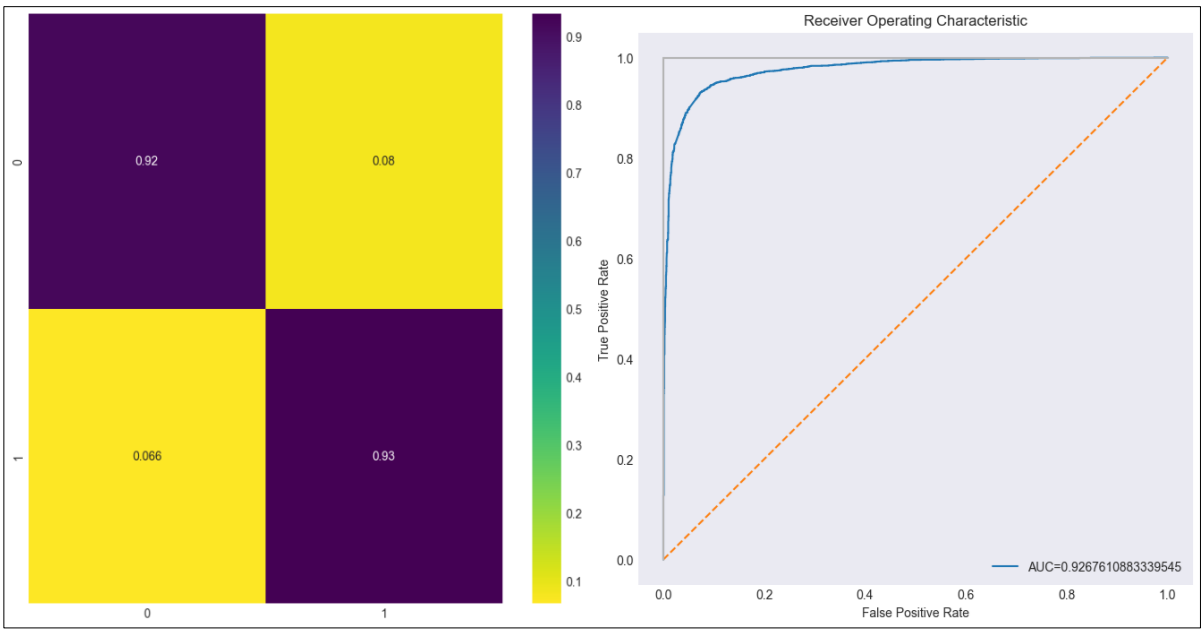
Table 14

Evaluation Metrics for the Best LSTM Model

Metrics	Values
Training Time	10.25 minutes
Predicted Correctly	4973 / 5366
Mean Loss	0.20
Recall	93.37%
Precision	92.10%
Accuracy	92.68%
Macro Averaged F1-score	92.68%

Figure 14

Confusion Matrix and ROC Curve of the Optimised *LSTM*

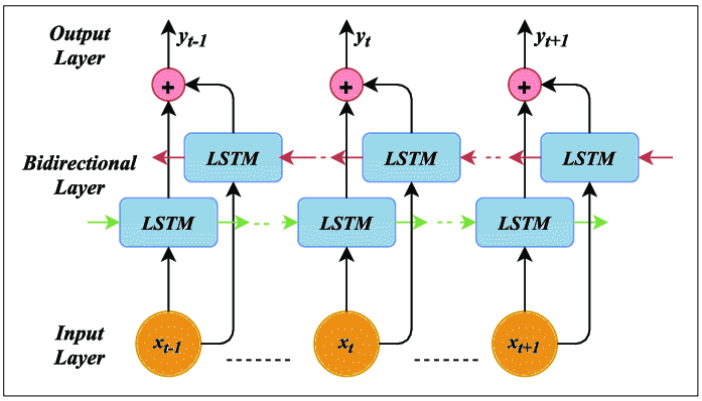


Bidirectional LSTM (BiLSTM)

The BiLSTM, illustrated in Figure 15, enhances the traditional LSTM architecture by handling the sequence in both forward and backwards directions (Kiperwasser & Goldberg, 2016). This method enables the network to gather context from past and future elements, improving its capacity to understand dependencies within the sequence more comprehensively. In BiLSTM, the input layer processes text sequences typically represented as embeddings that capture the semantic meaning of words. Each word in the input sequence is transformed into a fixed-dimensional vector, allowing the model to retain information about word relationships. The bidirectional layer consists of two LSTM units: one processes the sequence from the start to the end (forward), while the other processes it from the end to the start (backwards). Combining the outputs from both directions at each step gives the model a richer representation of the context, which is particularly beneficial for tasks like sentiment analysis. Finally, the output layer produces sentiment predictions by applying an activation function like softmax to the combined hidden states. This structure enhances the model's ability to accurately classify sentiments, as it can effectively consider the influence of words in preceding and following contexts.

Figure 15

The Architecture of BiLSTM Network



By using optimal hyperparameters—such as the number of layers, hidden units, learning rate, and dropout rate—the BiLSTM is trained to minimise loss and improve performance metrics. After training, the model's effectiveness is assessed on a test set to measure accuracy, precision, recall, and F1-score, ensuring it performs well in real-world applications. As shown in Table 15, the best model achieved an accuracy of 96.71%, with a recall of 96.55%, a precision of 96.87%, and a macro-average F1-score of 96.71%. The mean loss was 0.10, and the model correctly predicted 12,973 out of 13,415 instances, with a total training time of 115.35 minutes. The model's performance is further visualised through its confusion matrix (Figure 16) and ROC curve (Figure 16), demonstrating its reliability for real-world applications.

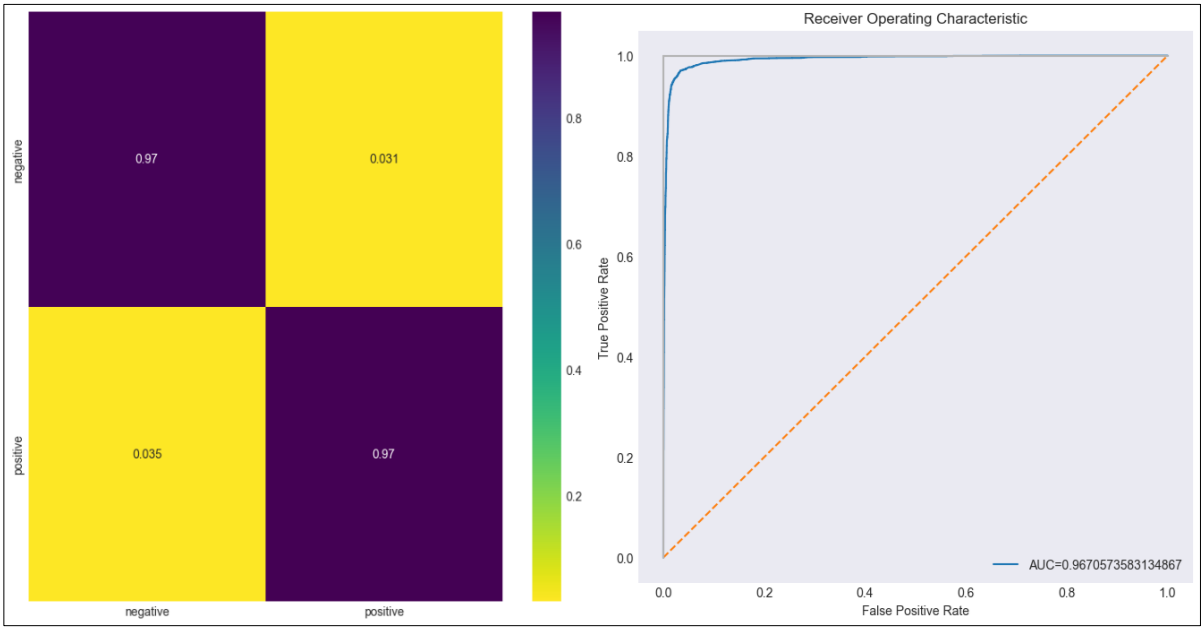
Table 15

BiLSTM Best Model Evaluation Results

Metric	Value
Training Time	15.35 minutes
Predicted Correctly	5180/5366
Mean Loss	0.10
Recall	96.55%
Precision	96.87%
Accuracy	96.71%
Macro Averaged F1-score	96.71%

Figure 16

Confusion Matrix and ROC Curve of the BiLSTM Model



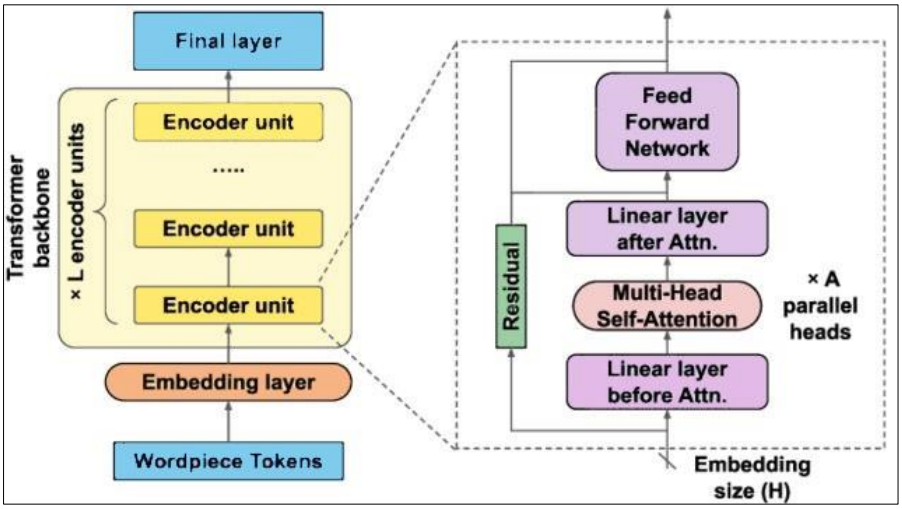
Bidirectional Encoder Representations from Transformers (BERT)

The BERT model introduced by Google in 2018 excels at understanding contextual information bidirectionally, making it highly effective for natural language processing tasks. By capturing the nuances of language through the analysis of context from both preceding and succeeding words, BERT enhanced performance in various applications, including sentiment analysis, question answering, and language translation (Devlin et al., 2019). As illustrated in Figure 17, the Transformer architecture comprises multiple stacked encoder units, each containing two main components: a self-attention sub-unit and a feedforward network (FFN) sub-unit, both connected via residual links. The self-attention sub-unit features a multi-head self-attention layer flanked by fully connected layers, while the FFN sub-unit consists solely of fully connected layers.

BERT's architecture is defined by three hyperparameters: the number of encoder units (L), the embedding vector size (H), and the number of attention heads in each self-attention layer (A). The values of L and H determine the model's depth and width, whereas A influences the number of contextual relationships each encoder can focus on (Kandhro et al., 2019). The authors of BERT released two pre-trained models: BERTBASE (L = 12; H = 768; A = 12) and BERTLARGE (L = 24; H = 1024; A = 16). The cased variant of BERT is particularly noteworthy because it retains the capitalisation of words, which can significantly affect meaning in many languages. This feature enhances its effectiveness in sentiment analysis, allowing for a more nuanced understanding of text where distinctions, such as between "Apple" (the brand) and "apple" (the fruit), are crucial. Consequently, the BERT cased model improves the accuracy of sentiment predictions in applications like product and service reviews.

Figure 17

Flowchart of the BERT Model



In our study, we tested both the BERT-base cased and uncased models for sentiment classification and concluded that the cased model performed better by effectively capturing emotional nuances. Our final architecture employs the BERT-base cased model. To optimise performance. We utilised the AdamW optimiser to optimise performance, stabilised the training process with gradient clipping, and implemented a learning rate scheduler to enhance convergence. The model was trained over four epochs with a batch size of 32, leveraging an expanded dataset and longer sequences to address the complexities of online commentary. Key hyperparameters are summarised in Table 16, ensuring the reproducibility of our setup.

Table 16

Optimal Hyperparameters for BERT-base Cased Model

Hyperparameter	Search Space	Best Value
model type	base-uncased, base-cased	BERT-base cased
max_length	80, 128, 256	256
batch_size	16, 32	32
epochs	1, 2, 3, 4, 5	4
hidden_size	64, 128	64
classes	2 (binary classification)	2
dropout	0.1, 0.2, 0.3	0.1
learning_rate	{1e-5, 2e-5, 5e-5}	5e-5
epsilon	1e-8	1e-8
clip_value	0.6, 0.8	0.6
optimiser	Adam, AdamW, SGD	AdamW
Scheduler	Linear Scheduler, Cosine Annealing, Exponential	Linear Scheduler
Loss function	CrossEntropyLoss, FocalLoss	CrossEntropyLoss

It is evident that the BERT-based cased classifier, when utilising increased sequence length and implementing early stopping, has significantly enhanced its performance. The model has shown a highly promising performance. As summarised in Table 17, it achieved an accuracy of 97.41%, correctly predicting 5,227 out of 5,366 samples. The macro-average F1 score of 97.41% further emphasises its balanced ability to manage both positive and negative sentimental classes. The model demonstrates robust predictive power with a precision of 97.43% and a recall of 97.39%. The ROC-AUC curve (Figure 18) achieves an impressive AUC score of 0.967, reflecting the model's strong ability to differentiate between positive and negative sentiments. Additionally, the low mean loss of 0.08 highlights minimal deviation between predicted and actual labels. The confusion matrix (Figure 18) demonstrates high classification accuracy for both sentiment classes (97%). These findings established the BERT-based model as a highly effective tool for sentiment analysis, even in its initial implementation, making it well-suited for applications like review classification.

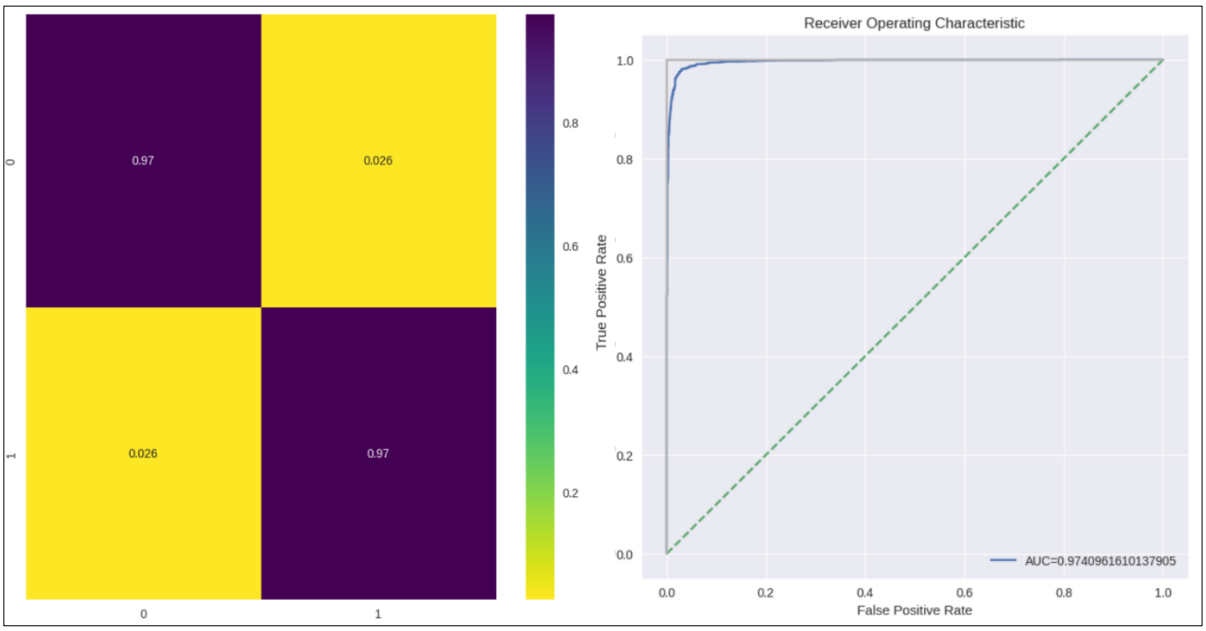
Table 17

Evaluation Results for BERT Model

Metric	Value
Training Time	75.86 minutes
Predicted Correctly	5196/5366
Mean Loss	0.34
Recall	97.50%
Precision	96.21%
Accuracy	96.83%
Macro Averaged F1-score	96.83%

Figure 18

Confusion Matrix and ROC Curve of the BERT Model



RESULTS AND DISCUSSIONS

This section compares the results obtained after optimising the hyperparameters of each model to achieve their best performance. The metrics reported, including training time, correct predictions, mean loss, accuracy, recall, precision, and macro-averaged F1-score, reflect the models' performance under their optimal configurations. Comparative analysis is conducted to highlight the strengths and weaknesses of each model, providing a comprehensive view of their capabilities in sentiment classification of student's reviews. The comparison Table 18 shows that BERT achieved the highest recall (97.50%), demonstrating its effectiveness in identifying positive reviews. BiLSTM closely followed, showing substantial precision (96.87%) and attaining the highest macro-average F1_score (96.71%).

In contrast, FNN and RNN exhibited lower accuracy scores (92.83% and 87.83%, respectively), highlighting their limitations in capturing the complexity of student reviews. Despite these limitations, efforts were made for hyperparameter optimisation, testing various configurations to identify the optimal architecture and training parameters. These results emphasised the advantages of using more advanced architectures, such as GRU, LSTM, BiLSTM, and BERT, which outperform simpler models. It is suggested that capturing temporal dependencies and contextual meanings is crucial for accurate sentiment analysis in educational data.

Table 18

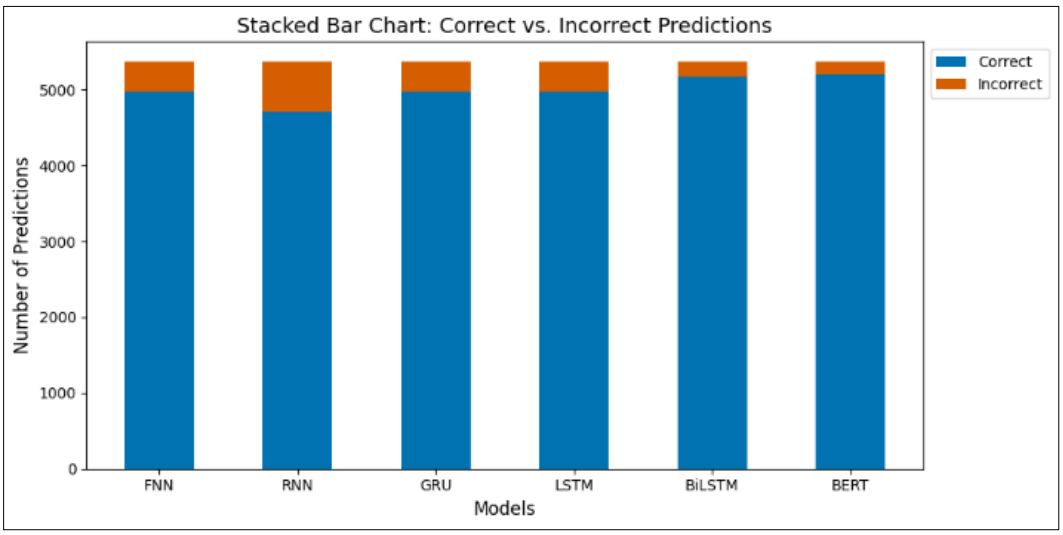
Comparison Table of Optimised Models' Performance

Model	Mean Loss	Recall %	Precision %	Accuracy %	Macro Averaged F1-Score%
FNN	0.39	92.83	92.46	92.83	92.83
RNN	0.29	88.48	87.34	87.83	87.83
GRU	0.36	95.50	94.64	95.50	95.50
LSTM	0.20	93.37	92.10	92.68	92.68
BiLSTM	0.10	96.55	96.87	96.71	96.71
BERT	0.34	97.50	96.21	96.83	96.83

The stacked bar chart (Figure 19) shows the proportion of correct vs. incorrect predictions across all models. Both BiLSTM and BERT demonstrated excellent performance, with most predictions being correct. GRU and LSTM also performed well, showing fewer incorrect predictions than simpler architectures like FNN and RNN. It indicated that these advanced models are better suited for capturing patterns in sentiment data, especially those with sequential dependencies.

Figure 19

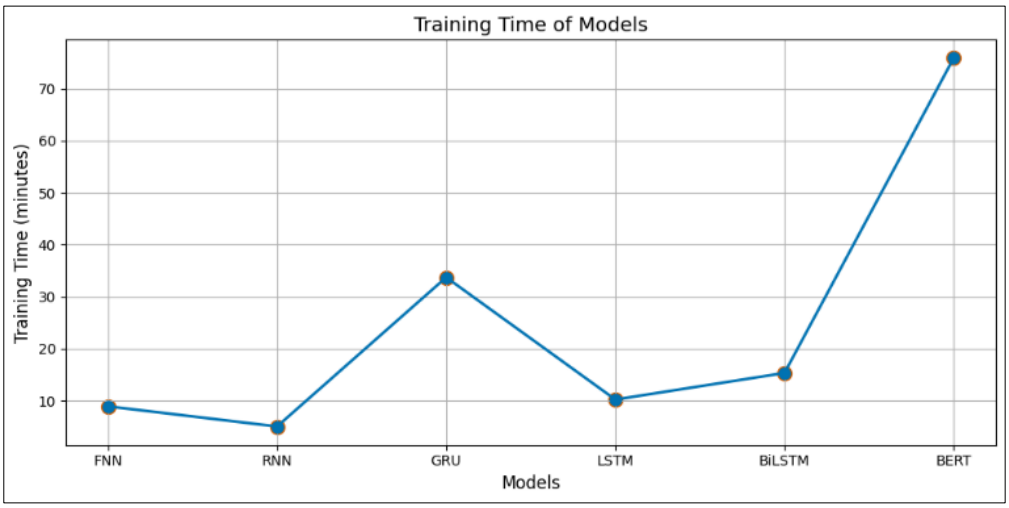
Proportion of Correct vs. Incorrect Predictions



The line graph (Figure 20) illustrates the training time for each model. As expected, BERT requires the longest training time—exceeding 70 minutes—due to the complexity of its architecture and large parameter space. In comparison, BiLSTM strikes a reasonable balance, delivering high accuracy with moderate training time. While faster than LSTM and BiLSTM, GRU still takes longer than FNN and RNN. It is attributed to the gated mechanisms that enable it to retain long-term dependencies, enhancing the model's effectiveness in sentiment classification. FNN and RNN, the simplest architecture, complete their training in approximately 10 minutes, making them viable for applications where computational efficiency precedes performance.

Figure 20

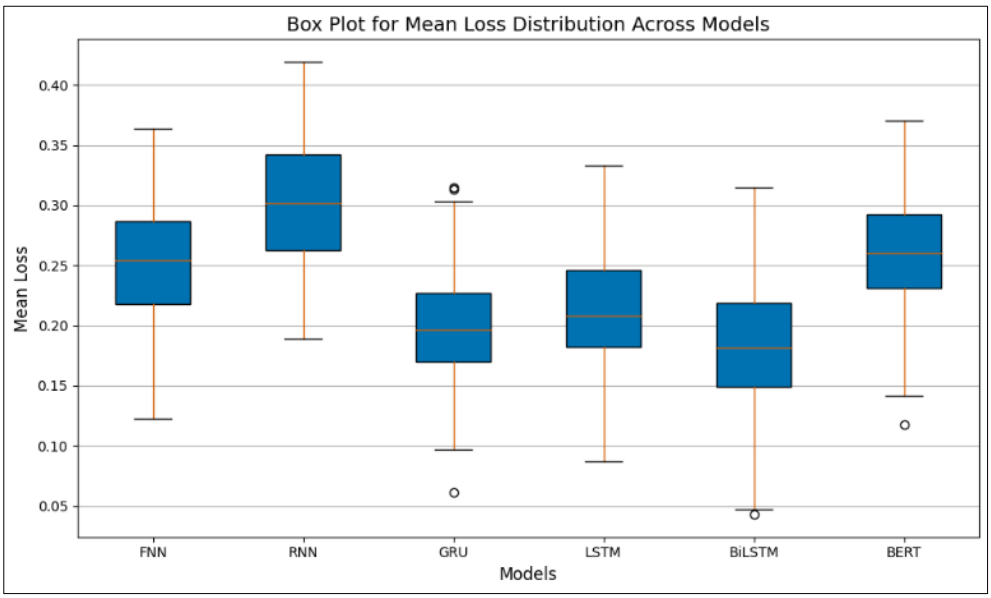
Training Time Comparison across Models



The box plot (Figure 21) illustrates the distribution of mean loss across models, revealing that BiLSTM and LSTM exhibited the lowest mean loss with smaller variance, indicating superior performance and stability during training. In contrast, RNN and FNN show higher mean loss and variability, highlighting their limitations in handling complex data. Interestingly, despite its high performance, BERT has a relatively higher mean loss, likely due to its complexity and fine-tuning challenges on the specific dataset. GRU presents a balanced loss, making it a practical alternative to LSTM. Overall, BiLSTM and LSTM are reliable for sentiment analysis, while RNN's poor performance underscores the need for advanced gating mechanisms, and BERT's loss reflected the trade-offs of using large, pre-trained models.

Figure 21

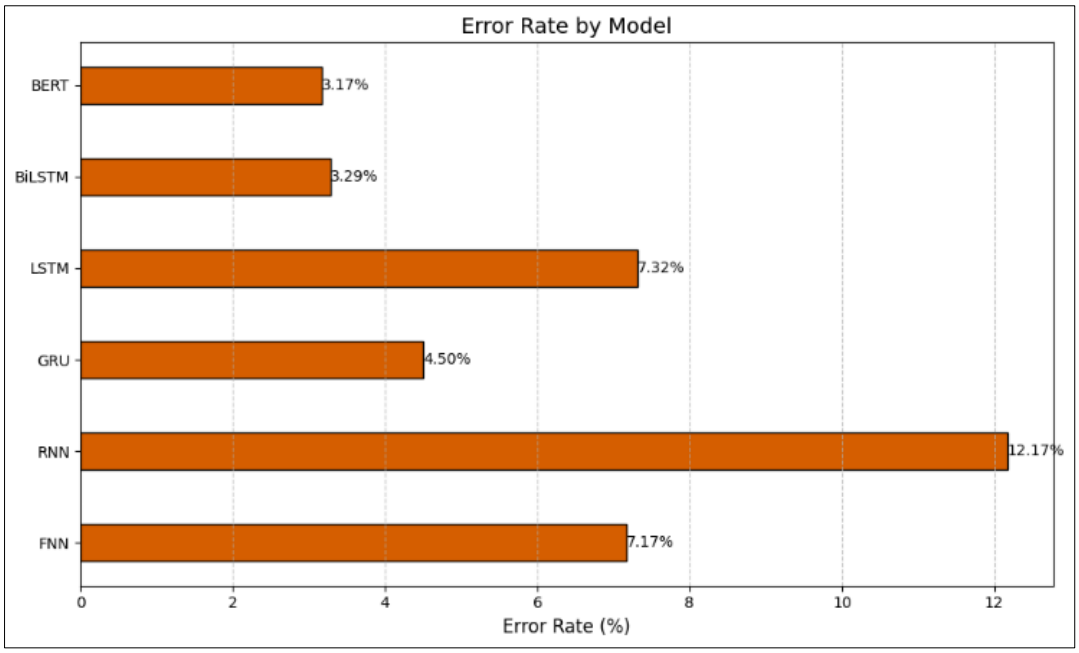
Distribution of Mean Loss across Sentiment Analysis Models



The bar chart (Figure 22) presents the error rates for each evaluated model. BERT achieves the lowest error rate at 3.17%, followed closely by BiLSTM with 3.29% and LSTM with 3.32%. These models demonstrate their capability to effectively handle complex text patterns, reducing the chances of misclassification. Despite being faster, the GRU model shows a slightly higher error rate of 4.50%. FNN and RNN exhibited the highest error rates, with 7.17% and 12.17%, respectively, reflecting their limitations in managing sequential dependencies and contextual information within the data. This chart reinforces the finding that BERT and BiLSTM are among the top-performing models, both in terms of accuracy and error reduction, making them ideal choices for sentiment classification tasks. However, it also highlights the practical performance of GRU as a lightweight option that balances between efficiency and error management.

Figure 22

Error Rate by Models



The results indicated that advanced architectures like BiLSTM and BERT are highly effective for sentiment analysis tasks, particularly when nuanced contextual understanding and sequential patterns are critical. Their ability to handle complex dependencies suggested their suitability for applications involving sentiment classification in educational settings, where accurate analysis of student feedback can lead to actionable insights for course improvement. Moreover, GRU emerged as a practical alternative for scenarios that balance computational efficiency and accuracy, making it an attractive choice for real-time applications or when deploying models on resource-constrained devices. The study also highlights the limitations of simpler models like FNN and RNN, which may be unsuitable for complex tasks without significant architectural enhancements. It underscores the importance of leveraging gating mechanisms, bidirectional processing, and pre-trained architectures for high-stakes applications.

In conclusion, while BERT and BiLSTM offered state-of-the-art performance, their computational demands should be considered in the context of practical deployment. GRU provided a promising middle ground, balancing accuracy and efficiency, whereas FNN and RNN may only be viable for

simpler tasks with less complex data. These findings provide valuable guidance for selecting appropriate models for sentiment analysis tasks, particularly in educational domains. These results confirmed the effectiveness of our proposed approach in binary sentiment classification for learner reviews. In future work, sentiment analysis insights could be practically integrated into MOOC platforms to personalise course recommendations, identify content needing improvement, or provide instructors with automated feedback summaries. This would enhance the overall learner experience and support data-driven decision-making in online education.

CONCLUSION

In conclusion, this study highlighted the effectiveness of advanced deep learning models like BiLSTM and BERT for sentiment analysis of student reviews, particularly in capturing complex feedback patterns essential for improving course quality. However, BERT's high computational requirements may restrict its use in real-time applications. GRU and LSTM offer more efficient alternatives, balancing performance with computational efficiency. While FNN and RNN struggle with complex patterns, they remain viable baseline models or lightweight options for simpler, less resource-demanding tasks.

Future research should focus on developing efficient models that leverage data augmentation and lightweight architectures to improve accessibility and scalability. Additionally, employing ensemble methods could further enhance performance. It is also crucial to consider the ethical and privacy implications of applying sentiment analysis to educational data. Ensuring student reviews are anonymised and processed in compliance with data protection standards will help maintain trust and uphold ethical research practices. Finally, integrating explainable AI techniques will make these models interpretable and actionable, facilitating informed decision-making in online education and ensuring that student feedback leads to meaningful improvements in course offerings.

ACKNOWLEDGMENT

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

REFERENCES

Abuhammad, A. S., & Ahmed, M. A. (2024). Automatic negation detection for semantic analysis in Arabic hotel reviews through lexical and structural features: A supervised classification. *Journal of Information and Communication Technology*, *23*(4), 709-744. https://doi.org/10.32890/jict2024.23.4.5

Altrabsheh, N., Cocea, M., & Fallahkhair, S. (2014). Learning sentiment from students' feedback for real-time interventions in classrooms. *Adaptive and Intelligent Systems*, 40–49. https://doi.org/10.1007/978-3-319-11298-5_5

Ananthi Claral Mary, T. (2025). Hybrid deep learning model to predict students' sentiments in higher educational institutions. *International Journal of Interactive Mobile Technologies (iJIM), 19*, 46–61. https://doi.org/10.3991/ijim.v19i01.50883

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv*, *1409*.

Baqach, A., & Battou, A. (2021). *Towards a user-oriented adaptive system based on sentiment analysis from text.* E3S Web of Conferences, 297, 01010. https://doi.org/10.1051/e3sconf/202129701010

Birjali, M., Kasri, M., & Beni Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, *226*, 107134. https://doi.org/10.1016/j.knosys.2021.107134

Cen, P., Zhang, K., & Zheng, D. (2020). Sentiment analysis using deep learning approach. *Journal on Artificial Intelligence*, *2*(1), 17–27. https://doi.org/10.32604/jai.2020.010132

Chumwatana, T. (2018). Comment analysis for product and service satisfaction from Thai customers' review in social network. *Journal of Information and Communication Technology*, *17*(2), 271-289. https://doi.org/10.32890/jict2018.17.2.8254

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS 2014 Workshop on Deep Learning, December 2014*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding (arXiv:1810.04805). *arXiv*. https://doi.org/10.48550/arXiv.1810.04805

Dey, R., & Salem, F. M. (2017). *Gate-variants of Gated Recurrent Unit (GRU) neural networks*. *2017* IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), 1597–1600. https://doi.org/10.1109/MWSCAS.2017.8053243

Jebbari et al. (2025). A novel hybrid model for sentiment analysis in MOOC forums with hybrid word and character-level neural networks. *Indonesian Journal of Electrical Engineering and Computer Science*, *37*(3), Article 3. https://doi.org/10.11591/ijeecs.v37.i3.pp1758-1771

Kandhro, I. A., Wasi, S., Kumar, K., Rind, M., & Ameen, M. (2019). Sentiment analysis of students' comment using long-short term model. *Indian Journal of Science and Technology*, *12*(8), 1–16. https://doi.org/10.17485/ijst/2019/v12i8/141741

Kastrati, Z., Imran, A. S., & Kurti, A. (2020). Weakly supervised framework for aspect-based sentiment analysis on students' reviews of MOOCs. *IEEE Access*, 8, 106799–106810. https://doi.org/10.1109/ACCESS.2020.3000739

Kiperwasser, E., & Goldberg, Y. (2016). Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, *4*, 313–327. https://doi.org/10.1162/tacl_a_00101

Koufakou, A. (2024). Deep learning for opinion mining and topic classification of course reviews. *Education and Information Technologies*, 29(3), 2973–2997. https://doi.org/10.1007/s10639-023-11736-2

Madani, Y., Ezzikouri, H., Erritali, M., & Hssina, B. (2020). Finding optimal pedagogical content in an adaptive e-learning platform using a new recommendation approach and reinforcement learning. *Journal of Ambient Intelligence and Humanized Computing*, *11*(10), 3921–3936. https://doi.org/10.1007/s12652-019-01627-1

Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent neural networks: A comprehensive review of architectures, variants, and applications. *Information*, *15*(9), Article 9. https://doi.org/10.3390/info15090517

Mrhar, K., Benhiba, L., Bourekkache, S., & Abik, M. (2021). A Bayesian CNN-LSTM model for sentiment analysis in Massive Open Online Courses MOOCs. *International Journal of Emerging Technologies in Learning (iJET)*, *16*(23), 216-232. https://doi.org/10.3991/ijet.v16i23.24457

Onan, A. (2021). Sentiment analysis on massive open online course evaluations: A text mining and deep learning approach. *Computer Applications in Engineering Education*, *29*(3), 572–589. https://doi.org/10.1002/cae.22253

Osmanoğlu, U. Ö., Atak, O. N., Çağlar, K., Kayhan, H., & Can, T. (2020). Sentiment analysis for distance education course materials: A machine learning approach. *Journal of Educational Technology and Online Learning*, *3*(1), 31–48. https://doi.org/10.31681/jetol.663733

Ouadad, R., & Mouncif, H. (2024). *Sentiment analysis of students feedback in online courses using supervised, ensemble, and transfer learning methods.* Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security, 1–7. https://doi.org/10.1145/3659677.3659733

Peng, H., Zhang, Z., & Liu, H. (2022). A Sentiment analysis method for teaching evaluation texts using attention mechanism combined with CNN-BLSTM model. *Scientific Programming*, 2022, 1–9. https://doi.org/10.1155/2022/8496151

Pennington, J., Socher, R., & Manning, C. (2014). *GloVe: Global Vectors for Word Representation. In* A. Moschitti, B. Pang, & W. Daelemans (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

Sultana, J., Sultana, N., Yadav, K., & AlFayez, F. (2018). *Prediction of sentiment analysis on educational data based on deep learning approach*. 2018 21st Saudi Computer Society National Computer Conference (NCC), 1–5. https://doi.org/10.1109/NCG.2018.8593108

Tzeng, J.-W., Lee, C.-A., Huang, N.-F., Huang, H.-H., & Lai, C.-F. (2022). MOOC Evaluation system based on deep learning. *The International Review of Research in Open and Distributed Learning*, *23*(1), 21–40. https://doi.org/10.19173/irrodl.v22i4.5417

Vaswani et al. (2023). Attention is all you need (arXiv:1706.03762). *arXiv*. http://arxiv.org/abs/1706.03762

Washington et al. (2021). *Sentiment analysis and topic modeling on tweets about online education during COVID-19*. https://radar.auctr.edu/islandora/object/mc.ir.fac.pub%3A2022_washington_patrick_2/

Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2015). Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*

Zhang, Z., Feng, F., & Huang, T. (2022). FNNS: An effective feedforward neural network scheme with random weights for processing large-scale datasets. *Applied Sciences*, *12*(23), Article 23. https://doi.org/10.3390/app122312478

Zyout, I., & Zyout, M. (2024). Sentiment analysis of student feedback using attention-based RNN and transformer embedding. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *13*(2), 2173. https://doi.org/10.11591/ijai.v13.i2.pp2173-2184