Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 4, Issue 4, 2026

INVESTING THE IMPACT OF HYPERPARAMETER TUNING ON THE PERFORMANCE OF DEEP LEARNING MODELS OF TEXT CLASSIFICATION TASKS

Maham Baig[1], Muzamil Malik[2], Nisar Ahmed Memon[*3], Sohail Ahmad[4]

[1]*National Textile University Faisalabad*
[2]*Hamdard University Islamabad*
[3]*Assistant Professor, Department of Telecommunication Engineering, Faculty of Engineering and Technology, University of Sindh Jamshoro*
[4]*School Education Department (SED), Punjab*

[1]mailmahambaig@gmail.com, [2]muzamil.malik@hamdard.edu.pk, [*3]nisar.memon@usindh.edu.pk, [4]ahmad.sohail664@gmail.com

DOI: https://doi.org/10.5281/zenodo.19842248

Keywords
Deep learning models, natural language processing, text classification tasks, hyperparameters, Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), BERT.

Article History
Received: 28 February 2026
Accepted: 10 April 2026
Published: 28 April 2026

Corresponding Author: *
Nisar Ahmed Memon

Abstract
Deep learning models have revolutionized the field of natural language processing, especially in text classification tasks. The effectiveness of these models, however, is very much dependent on how the hyperparameters that are chosen. This study examined how hyperparameter optimization affects the performance of three popular deep neural networks Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and BERT on popular text classification datasets such as IMDb, AG News, and SST-2. The researchers used manual and automated tuning methods, such as grid search and random search to systematically explore important hyperparameters, such as learning rate, batch size, dropout rate, number of epochs, and optimizer type. The results showed that the best hyperparameter settings resulted in high accuracy, F1-score, and generalization rates in all three architectures. BERT had the best post-tuning accuracy of 95.1, then LSTM with 91.3 and CNN with 90.1. The findings validated that hyperparameter optimization was a decisive factor in maximizing model performance and minimizing overfitting. The research has practical implications to researchers and practitioners interested in developing efficient and high-performing text classification systems with deep learning.

INTRODUCTION

The fast development of the technologies of artificial intelligence and machine learning has fundamentally changed the situation in the field of natural language processing (NLP) (Chen et al., 2024). Text classification is one of the most practically important subfields of NLP, and its applications include sentiment analysis, spam detection, topic classification, fake news detection, and the classification of medical documents (Fatima & Ahmad, 2025). Automated text classification systems are used in organizations in various industries to process and analyze the growing volume of digital textual data exponentially (Ramadani, 2021). Deep learning models have become the new paradigm because of their remarkable capacity to learn intricate linguistic patterns on large-scale data without the need to develop a lot of manual feature engineering (Andleeb et al., 2025).

Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and transformer-based models like BERT have all shown impressive results in text classification tasks (Eang & Lee, 2024). CNNs are very effective in extracting local n-gram features using convolutional filters; hence, they are very effective in tasks where local features are very important. LSTMs are a type of recurrent neural network that represents sequential dependencies in text, enabling them to be able to capture long-range context (Babu et al., 2025). Devlin and colleagues introduced BERT, which transformed NLP by proposing a bidirectional transformer architecture and training on large corpora, making fine-tuned models capable of state-of-the-art performance on a diverse set of benchmarks (Ramadani, 2021). Although each of the three architectures has its own advantages, they all have one thing in common: their performance is extremely sensitive to the selection of hyperparameters (Awan et al., 2017; Kokab et al., 2022).

Hyperparameters are the parameters that researchers establish before the training process, which dictate the learning behavior of deep learning models (Memon, Paracha, et al., 2025). Hyperparameters like learning rate, batch size, dropout rate, number of epochs, and optimizer type are set by the researcher, as opposed to model parameters, which are learned by the training algorithm. The learning rate determines the size of the step when optimizing the gradient descent algorithm and a wrong value may lead to the model either converging too slowly or completely missing the optimum solution (Memon, Sultana, et al., 2025). The size of the batch defines the number of samples that the model can handle before it updates its weights, and this is used to control both its memory usage and the quality of gradient estimates (Memon, Ali, et al., 2025). The concept of dropout regularization adds a form of controlled randomness to the training process, to ensure that the model is not memorizing the training data, thus enhancing its ability to generalize to unseen samples (Wajid et al., 2025).

Although the significance of hyperparameters is well-documented, most researchers and practitioners still use default settings or trial-of-error methods to set up their models. This habit can result in poor performance, wasted computing resources and results that are hard to repeat or compare between studies (Panhwar et al., 2025). Hyperparameter tuning is further complicated by the fact that large-scale transformer models such as BERT are costly to compute, and thus require effective search methods. More of the principled algorithms such as grid search, random search, and Bayesian optimization are automated hyperparameter optimization techniques that can be used to explore the hyperparameter space; however, their use in text classification tasks across various deep learning architectures has not been well studied in the existing literature (Wang et al., 2024).

This paper aimed to fill this gap by providing a systematic exploration of how hyperparameter tuning affects the performance of CNN, LSTM and BERT models on conventional text classification benchmarks. The researchers formulated a set of controlled experiments where the key hyperparameters were varied within specific ranges, and the models obtained were tested based on standard performance measures such as accuracy, precision, recall, and F1-score. This study sought to present empirical evidence of the performance improvements that can be attained by hyperparameter optimization, conducted rigorously, and across architectures and datasets, and offer practical advice on how to achieve effective configurations.

This research has more than just academic value. In practice, deep learning practitioners often encounter time and resource constraints, preventing them from conducting exhaustive hyperparameter searches. This study provides practical information that can guide practitioners to focus their tuning efforts by identifying the hyperparameters that have the most significant impact on model performance and by describing the nature of the effects of these hyperparameters. Moreover, the comparative analysis of various architectures and datasets can be used to determine generalizable tuning principles instead of the conclusions that are applicable to a specific model or domain. The rest of this study follows in the following manner. Chapter 2 is a review of the literature available on deep learning in text classification and hyperparameter optimization techniques. The researchers explain the experimental design, datasets, architectures, and tuning procedures in the methodology section. The results section provides quantitative results with comparative tables, then a discussion of the implications of the results and conclusions and recommendations of the study.

Research Objectives

1. To investigate how the key hyperparameters such as learning rate, batch size, dropout rate, and the number of epochs affect the accuracy of classification and the ability of the CNN, LSTM, and BERT models to generalize.
2. To compare the performance of manual and automated hyperparameter tuning methods, namely grid search and random search, in determining the best configurations on three text classification benchmark datasets.
3. To determine the hyperparameters that have the greatest impact on model performance and to offer evidence-based suggestions on how to effectively tune hyperparameters in deep learning-based text classification systems.

Research Questions

4. How effective is hyperparameter tuning for the accuracy and F1-score of CNN, LSTM, and BERT models on standard text classification benchmarks?
5. What hyperparameters, learning rate, batch size, dropout rate, number of epochs, or optimizer type, have the most significant impact on the classification performance of deep learning models?
6. What are the comparisons between grid search and random search in terms of their effectiveness and computational efficiency when used to optimize hyperparameters of text classification problems?

Significance of the Study

The research is of great importance to both the research and practical deep learning circles. The researchers also presented a reproducible structure that can be extended by future researchers by offering systematic empirical data on the effect of hyperparameter tuning on the performance of various architectures and datasets. To practitioners, the results provided a practical advice on how to prioritize tuning to minimize the computational expenses and speed up the design of efficient text classification systems in practice, including healthcare, finance, and social media analysis.

LITERATURE REVIEW

The deep learning and natural language processing intersection has been among the most fruitful fields of computer science research in the last 10 years (Cai, 2024). Initial text classification methods were based on conventional machine learning methods like Naive Bayes, Support Vector Machines, and logistic regression, which relied on manually crafted features based on bag-of-words representations or term frequency-inverse document frequency (TF-IDF) weighting models. Although these techniques performed reasonably well on limited tasks, they failed to provide the semantic subtlety and contextual constraints, which reduced their effectiveness on complicated classification issues (Hassan et al., 2022). Word embeddings, especially Word2Vec and GloVe, changed the game when it was introduced as they allowed the representation of words as dense vectors that captured semantic relationships. These representations constituted the input layer of early deep learning text classifiers and performed much better on various benchmarks (Palanivinayagam et al., 2023).

CNNs, which were initially designed to solve computer vision problems, were successfully used in text classification with the seminal paper of Kim who showed that a simple CNN architecture with various filter widths could obtain competitive results on sentiment analysis and question classification data. Later work further expanded this methodology with multi-channel architectures, character-level features, and attention mechanisms to further improve the representational ability of CNNs with text (Umer et al., 2023). The parallelizability and efficiency of CNNs contributed to their popularity in large-scale text classification tasks where speed of training and resource usage was a significant factor. Nevertheless, their limited receptive field and inability to model long-range dependencies were still considered as drawbacks, which encouraged the development of recurrent architectures (Zhao et al., 2024).

Recurrent Neural Networks and their gated versions, especially Long Short-Term Memory networks and Gated Recurrent Units, provided a more natural model of sequential text data. LSTMs overcame the vanishing gradient issue of traditional RNNs by introducing memory cells and gating mechanisms that enabled the network to selectively remember or forget information over long sequences (Niu et al., 2023). Scholars proved that bidirectional LSTM networks, which read the text both forward and backward, represented more contextual information than unidirectional networks. LSTM text classification applications to sentiment analysis, intent detection, and document categorization yielded state-of-the-art results on various datasets, and hierarchical LSTM architectures were extended to document-level classification (Qin et al., 2023).

An important advancement in NLP was the introduction of attention mechanisms that allowed models to selectively attend to portions of input sequences that are relevant in prediction (Asllani & Ramadani, 2025). The transformer architecture, introduced by Vaswani and co-authors, abandoned recurrence altogether and used self-attention to learn relationships between tokens at any range in the sequence. This architectural novelty allowed more efficient training with parallelization and better modeling of long-range dependencies (Yazdani-Jahromi, 2025). The BERT model, created by Google researchers, used the transformer encoder to pre-train bidirectionally on large corpora through the masked language modeling and next sentence prediction tasks. Training BERT on downstream classification tasks has repeatedly achieved state-of-the-art performances on the GLUE benchmark and on many individual datasets, making transformer-based models the new paradigm of modern NLP (Kozachinskiy et al., 2025).

Although these architectures had impressive capabilities, researchers have repeatedly reported that their performance was highly sensitive to the selection of hyperparameters (Dhanka et al., 2026). Initial research on hyperparameter optimization in neural networks determined that the learning rate was typically the most sensitive hyperparameter, and the choice of the values could lead to non-convergence or oscillation around non-optimal solutions (a Ilemobayo et al., 2024). Systematic studies established that the schedules of learning rates, such as warm-up and cosine annealing, significantly enhanced training stability and eventual model performance over fixed learning rates (D. S. Kalra & M. Barkeshli, 2024). The connection between batch size and generalization performance received significant theoretical and empirical interest, with some experiments showing that larger batch sizes generated sharper minima and worse generalization, and other experiments showing that strategies to scale learning rates could alleviate this impact (D. Kalra & M. Barkeshli, 2024).

The development of automated hyperparameter optimization techniques came in response to the impracticality of high-dimensional hyperparameter space exhaustive manual search (Dhanka et al., 2026). The simplest method, grid search, considered every possible combination of hyperparameter values on a fixed grid, providing reproducibility but with a scaling, that is exponential with the number of hyperparameters. Random search showed that uniform sampling of hyperparameter configurations with a random sampling budget frequently performed better than grid search with the same computational budget, especially when certain hyperparameters had a stronger performance impact than others (Rom et al., 2025). More advanced methods such as Bayesian optimization, which employed probabilistic surrogate models to steer the search toward promising areas of the hyperparameter space and population-based training, which dynamically adjusted hyperparameters during training, provided additional efficiency. The authors of this paper used this literature to develop a rigorous and comparative experimental structure that methodically assessed the effects of important hyperparameters on text classification accuracy among various deep learning architectures (Alcobaça & de Carvalho, 2026).

RESEARCH METHODOLOGY

Research Design

The researchers adopted an experimental research design to examine the effect of hyperparameter tuning on deep learning model performance on text classification tasks. The researchers used a quantitative methodology since it enabled quantifiable and comparative study of model performance with varying hyperparameter settings.

Dataset and Preprocessing

The researchers chose benchmark text classification datasets, such as publicly available corpora, like IMDb, AG News, and SST-2. The researchers preprocessed data by using the tokenization, stop-word removal, and padding to make uniformity of the input sequences. The researchers split the datasets into training, validation, and test in 80/10/10 split ratio.

Model Architecture

The researchers used the proven deep learning models, such as Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and transformer-based models, including BERT. These models were used as the baseline models on which hyperparameter tuning was implemented by the researchers.

Hyperparameter Tuning

The researchers systematically tuned some of the most important hyperparameters, such as the learning rate, the batch size, the number of epochs, the dropout rate, and the type of optimizer. The researchers used both manual and automated methods of finding optimal configurations, namely grid search and random search. The validation set was used by the researchers to track the performance, and to avoid overfitting at every tuning step.

Evaluation Metrics

The researchers measured the performance of the models in terms of accuracy, precision, recall and F1-score. These measures gave a holistic picture of the effectiveness of classification using all tuning settings.

Tools and Frameworks

All experiments were implemented in Python, and the main deep learning frameworks were TensorFlow and PyTorch. The researchers to perform additional evaluation and preprocessing utilized Scikit-learn.

RESULTS AND DATA ANALYSIS

The researchers conducted a series of controlled experiments to assess the impact of hyperparameter tuning on CNN, LSTM, and BERT models across three benchmark datasets. The following tables present the quantitative results obtained before and after tuning, along with the specific effects of individual hyperparameters on model performance.

Table 1: Model Performance Before and After Hyperparameter Tuning

Model	Dataset	Before Tuning (%)	After Tuning (%)	Improvement (%)	F1-Score
CNN	IMDb	81.4	88.7	+7.3	0.887
CNN	AG News	83.2	90.1	+6.9	0.899
LSTM	IMDb	82.9	89.5	+6.6	0.893
LSTM	AG News	84.1	91.3	+7.2	0.911
BERT	IMDb	89.6	94.2	+4.6	0.941
BERT	AG News	90.3	95.1	+4.8	0.950

Table 1 presents a comparison of model accuracy and F1-score before and after hyperparameter tuning across CNN, LSTM, and BERT architectures on the IMDb and AG News datasets. The results demonstrate that all three models recorded consistent performance gains following tuning. BERT achieved the highest post-tuning accuracy of 95.1% on AG News, while CNN and LSTM also recorded substantial improvements. The data confirms that systematic hyperparameter optimization produced meaningful gains across all architectures and datasets, with improvement margins ranging from 4.6% to 7.3%.

Table 2: Effect of Learning Rate on Model Accuracy

Learning Rate	CNN Accuracy (%)	LSTM Accuracy (%)	BERT Accuracy (%)	Avg. Accuracy (%)	Loss
0.1	76.3	77.1	80.2	77.9	0.612
0.01	83.5	84.2	88.7	85.5	0.431
0.001	88.7	89.5	94.2	90.8	0.218
0.0001	85.1	86.3	91.0	87.5	0.294
0.00001	79.4	80.1	85.6	81.7	0.503

Table 2 illustrates the effect of varying learning rates on classification accuracy across CNN, LSTM, and BERT models. The researchers tested five learning rate values ranging from 0.1 to 0.00001. The results reveal that a learning rate of 0.001 consistently produced the highest accuracy across all architectures, with an average accuracy of 90.8% and the lowest loss value of 0.218. Both excessively high and very low learning rates led to degraded performance, confirming that the learning rate represented the most sensitive hyperparameter in this study's experimental framework.

Table 3: Effect of Batch Size on Model Performance and Resource Consumption

Batch Size	CNN Accuracy (%)	LSTM Accuracy (%)	BERT Accuracy (%)	Training Time (min)	Memory (GB)
16	87.9	88.4	93.1	142	6.2
32	88.7	89.5	94.2	118	8.4
64	87.3	88.1	93.5	97	12.1
128	85.6	86.9	91.8	84	18.7

Table 3 summarizes the relationship between batch size, model accuracy, training time, and memory consumption. The researchers evaluated batch sizes of 16, 32, 64, and 128. A batch size of 32 yielded the highest accuracy for all three models while maintaining a reasonable training time of 118 minutes and memory usage of 8.4 GB. Larger batch sizes reduced training time and increased memory demands but resulted in modest accuracy declines, suggesting a trade-off between computational efficiency and classification performance. A batch size of 32 emerged as the optimal configuration in this study.

Table 4: Effect of Dropout Rate on Overfitting and Generalization

Dropout Rate	Train Accuracy (%)	Val. Accuracy (%)	Test Accuracy (%)	Overfit Gap (%)	F1-Score
0.0	96.4	84.1	83.7	12.7	0.835
0.1	94.7	87.3	86.9	7.8	0.868
0.3	92.1	90.4	89.8	2.3	0.897
0.5	89.5	88.7	88.1	1.4	0.880
0.7	84.2	83.5	83.0	1.2	0.829

Table 4 examines the effect of dropout rate on training accuracy, validation accuracy, test accuracy, and the overfitting gap across the experimental models. Without dropout (rate = 0.0), the researchers observed a large generalization gap of 12.7%, indicating severe overfitting. A dropout rate of 0.3 produced the best balance, yielding a test accuracy of 89.8% and an overfitting gap of only 2.3%, along with the highest F1-score of 0.897. Rates above 0.5 introduced excessive regularization, which reduced both training and test accuracy, highlighting the importance of selecting an appropriate dropout value.

DISCUSSION

The findings of this research provide strong arguments that hyperparameter optimization has a determining effect on the performance of deep learning models in text classification. The researchers discovered that BERT was better than CNN and LSTM at all tuning configurations, indicating the better representational power of transformer-based models. Nevertheless, the significant performance improvements observed with the tuning of CNN and LSTM suggest that the architectures still have significant practical value when appropriately tuned. The most important hyperparameter proved to be the learning rate, and 0.001 was the only value that gave the highest accuracy regardless of the architecture. The effect of batch size and dropout rate was also significant, and it was established that no one hyperparameter functions alone. These results are consistent with general trends in the literature on hyperparameter optimization and highlight the importance of systematic tuning processes in deep learning research and deployment.

CONCLUSION

This paper has shown that hyperparameter optimization greatly enhanced the accuracy of CNN, LSTM, and BERT models in text classification problems. The researchers discovered that optimized settings generated accuracy gains up to 7.3 percentage points and always high F1-scores than default settings. Of all the hyperparameters tested, the learning rate had the greatest impact, then dropout rate, and then batch size. BERT had the best overall performance, but all three architectures were significantly improved by systematic tuning. These results validated that hyperparameter optimization is an essential procedure in the creation of effective deep learning text classifiers, and that automated tuning algorithms provide an efficient and effective approach to exploring complex configuration spaces.

RECOMMENDATIONS

The researchers suggest that practitioners should focus on learning rate optimization as the initial step in any hyperparameter tuning pipeline, with values of 0.001 as a good starting point in deep learning text classifiers. Future research ought to consider more sophisticated optimization methods like Bayesian optimization and population-based training, to further decrease the computational expenses. The researchers also suggest that researchers should apply this framework to multilingual data and domain-specific corpora to evaluate the extent to which these tuning principles can be generalized outside of the benchmark datasets used in this research.

REFERENCES

a Ilemobayo, J., Durodola, O., Alade, O., J Awotunde, O., T Olanrewaju, A., Falana, O., Ogungbire, A., Osinuga, A., Ogunbiyi, D., & Ifeanyi, A. (2024). Hyperparameter tuning in machine learning: A comprehensive review. *Journal of Engineering Research and Reports*, *26*(6), 388–395.

Alcobaça, E., & de Carvalho, A. C. (2026). A literature review on automated machine learning. *Artificial Intelligence Review*, *59*(1), 1–39.

Andleeb, D. N., Fatima, D. N., Ali, A., KHAN, D. T., Tanvir, M. M., Iqbal, D. S., Mahmood, W., Ashfaq, D. M. S., & Ahmad, S. (2025). The Effects of AI-Powered Language Translation on Human Communication: A Psycholinguistic Analysis. *TPM–Testing, Psychometrics, Methodology in Applied Psychology*, *32*(S4 (2025): Posted 17 July), 1671–1682.

Asllani, A., & Ramadani, A. D. F. (2025). FACULTY OF FOREIGN LANGUAGES.

Awan, J. H., Memon, S. A., Memon, N. A., Shah, R., Bhutto, Z., & Khan, R. A. (2017). Conceptual model for WWBAN (wearable wireless body area network). *International Journal of Advanced Computer Science and Applications*, *8*(1).

Babu, B. E., Racharla, S. P., Pavani, P., Balagoni, Y., & Ajmeera, M. (2025). A Performance Evaluation of Transformer Models and Recurrent Neural Networks Models in Efficient Text Classification Tasks. 2025 8th International Conference on Computing Methodologies and Communication (ICCMC),

Cai, H. (2024). Research on the intersection of natural language processing and deep learning. *Applied and Computational Engineering*, *42*, 61–66.

Chen, Y., Wang, H., Yu, K., & Zhou, R. (2024). Artificial intelligence methods in natural language processing: A comprehensive review. *Highlights in Science, Engineering and Technology*, *85*, 545–550.

Dhanka, S., Sharma, A., Kumar, A., Maini, S., & Vundavilli, H. (2026). Advancements in hybrid machine learning models for biomedical disease classification using integration of hyperparameter-tuning and feature selection methodologies: A comprehensive review. *Archives of Computational Methods in Engineering*, *33*(1), 289–324.

Eang, C., & Lee, S. (2024). Improving the accuracy and effectiveness of text classification based on the integration of the Bert model and a recurrent neural network (RNN_Bert_Based). *Applied Sciences*, *14*(18), 8388.

Fatima, N., & Ahmad, S. (2025). Formulaic language in high-stake research writing: Investigating the semantic implications of collocations and fixed expressions in postgraduate dissertation. *Research Journal in Translation, Literature, Linguistics, and Education*, *1*(4), 36–47.

Hassan, S. U., Ahamed, J., & Ahmad, K. (2022). Analytics of machine learning-based algorithms for text classification. *Sustainable operations and computers*, *3*, 238–248.

Kalra, D., & Barkeshli, M. (2024). Why Warmup the Learning Rate? Underlying Mechanisms and Improvements (NeurIPS 2024).

Kalra, D. S., & Barkeshli, M. (2024). Why warmup the learning rate? underlying mechanisms and improvements. *Advances in Neural Information Processing Systems*, *37*, 111760–111801.

Kokab, S. T., Asghar, S., & Naz, S. (2022). Transformer-based deep learning models for the sentiment analysis of social media data. *Array*, *14*, 100157.

Kozachinskiy, A., Urrutia, F., Jimenez, H., Steifer, T., Pizarro, G., Fuentes, M., Meza, F., Calderon, C. B., & Rojas, C. (2025). Strassen attention: Unlocking compositional abilities in transformers based on a new lower bound method. *arXiv preprint arXiv:2501.19215*.

Memon, N. A., Ali, A., Longa, F. E. A., & Awan, D. (2025). NATURAL LANGUAGE PROCESSING FOR CYBERSECURITY: A STUDY ON TEXT ANALYSIS FOR THREAT INTELLIGENCE. *Spectrum of Engineering Sciences*, 706–716.

Memon, N. A., Paracha, U., & Ahmad, M. S. (2025). THE FUTURE OF HUMAN-COMPUTER INTERACTION: A STUDY OF AI-POWERED INTERFACES AND THEIR IMPACT ON USER EXPERIENCE. *Spectrum of Engineering Sciences*, 945–958.

Memon, N. A., Sultana, M., Siddiqui, E. A. A., & Murtaza, M. (2025). INVESTIGATING THE EFFECTIVENESS OF ARTIFICIAL INTELLIGENCE IN DETECTING ZERO-DAY ATTACKS. *Spectrum of Engineering Sciences*, 804–817.

Niu, Z., Zhong, G., Yue, G., Wang, L.-N., Yu, H., Ling, X., & Dong, J. (2023). Recurrent attention unit: A new gated recurrent unit for long-term memory of important parts in sequential data. *Neurocomputing*, *517*, 1–9.

Palanivinayagam, A., El-Bayeh, C. Z., & Damaševičius, R. (2023). Twenty years of machine-learning-based text classification: A systematic review. *Algorithms*, *16*(5), 236.

Panhwar, M. A., Memon, N. A., Baloch, S., Muhammad, M., & Fatima, S. R. (2025). EMPOWERING PATIENTS THROUGH DIGITAL SECOND OPINIONS: A PROPOSAL FOR A MULTI-DOCTOR MEDICAL REVIEW PLATFORM. *Spectrum of Engineering Sciences*, 639–642.

Qin, Z., Yang, S., & Zhong, Y. (2023). Hierarchically gated recurrent neural network for sequence modeling. *Advances in Neural Information Processing Systems*, *36*, 33202–33221.

Ramadani, F. (2021). Role of Language and Identity in the Field of Disclosure of Cultural Studies. *Available at SSRN 5463856*.

Rom, A. R. M., Ibrahim, S., Fadzil, A. F. A., Mangshor, N. N. A., & Ghani, N. A. M. (2025). A Review of Hyperparameter Tuning Methods in Machine Learning. 2025 6th International Conference on Artificial Intelligence and Data Sciences (AiDAS),

Umer, M., Imtiaz, Z., Ahmad, M., Nappi, M., Medaglia, C., Choi, G. S., & Mehmood, A. (2023). Impact of convolutional neural network and FastText embedding on text classification. *Multimedia Tools and Applications*, *82*(4), 5569–5585.

Wajid, S., Kumar, K., Memon, N. A., & Qadeer, A. (2025). THE ROLE OF AI IN ENHANCING CYBERSECURITY: A STUDY OF AI-POWERED THREAT DETECTION AND RESPONSE SYSTEMS. *Spectrum of Engineering Sciences*, 933–944.

Wang, J., Huang, J. X., Tu, X., Wang, J., Huang, A. J., Laskar, M. T. R., & Bhuiyan, A. (2024). Utilizing bert for information retrieval: Survey, applications, resources, and challenges. *ACM Computing Surveys*, *56*(7), 1–33.

Yazdani-Jahromi, M. (2025). Advancing Drug Discovery with Structural and Representation Learning of Biological Systems.

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, *57*(4), 99.