Technoscience Journal for Community Development in Africa 4 (2025) 227–240
ISSN 2659-0573

Research Article

The critical role of hyperparameter tuning in machine learning: Implications for reproducibility and model comparison

Rafiu Mope Isiaka*, Shakirat Ronke Yusuff, Akinbowale Nathaniel Babatunde, Shuaib Babatunde Mohammed

Department of Computer Science, Kwara State University, Malete

Abstract: Despite being a fundamental aspect of machine learning model development, hyperparameter tuning remains underreported in the literature. This article highlights the importance of hyperparameter optimisation, outlines common hyperparameters across various algorithms, and discusses the consequences of inadequate hyperparameter documentation. We argue that the lack of transparency in hyperparameter settings impedes reproducibility, hinders fair model comparisons, and contributes to the hyperparameter deception. The importance of hyperparameter tuning in machine learning was demonstrated by comparing the performance of the decision tree, support vector machine and random forest models on Iris, Digits and Breast Cancer datasets using default and tuned hyperparameters. This further justifies the need to document and report the process and values of the hyperparameter settings used in the models. To facilitate this, an architecture that encourages the documentation of the hyperparameters has been proposed. By emphasising the need for comprehensive reporting, this study aims to raise awareness and encourage best practices in machine learning research.

Keywords: Hyperparameter tuning, machine learning, reproducibility, model comparison, optimisation, scientific integrity.

1. Introduction

Machine learning is a branch of artificial intelligence that allows computers to learn from data and make decisions without explicit programming for each task. This involves creating models that detect patterns and relationships in data to perform tasks such as classification, regression and clustering. It encompasses methods such as supervised learning (using data with labels), unsupervised learning (finding structure in data without labels), and reinforcement learning (learning through feedback from rewards). The primary aim of machine learning is to enhance task performance through experience and exposure to data; hence, it has become a cornerstone of scientific and industrial research, with models applied to a wide array of complex problems.

Hyperparameter tuning is integral to machine learning, as it fine-tunes the settings or hyperparameters that dictate the behaviour and capacity of the machine learning models. In the development of machine learning models, hyperparameter tuning plays a vital role in enhancing their performance of machine learning models by refining their architecture, functionality, and precision (Lakshmana et al., 2021; Ma et al., 2022). Incorporating this optimisation step is crucial in machine learning processes to ensure that models not only excel with training data but also perform effectively with new, unseen data, facilitating the successful and reliable deployment of machine learning solutions (Rao & Jaganathan, 2024; Tan et al., 2024). This process is essential for tasks such as sentiment analysis and image classification, where optimised parameters boost classification accuracy and model generalisation (Isa et al., 2019; Sureja et al., 2024). Unlike model parameters learned during training, hyperparameters are user-defined settings external configurations that are set prior to training a model and influence how the model learns

* Corresponding author
Email: abdulrafiu.isiaka@kwasu.edu.ng

KWASU PRESS AND PUBLISHING

and generalises to unseen data (Arnold et al., 2024; Probst, Boulesteix, et al., 2019). Examples include the learning rate, number of hidden layers and neurons per layer in neural networks, batch size, number of training epochs, and value of k in k-nearest neighbour (KNN) algorithms.

The process of identifying the optimal combination of these hyperparameters, known as hyperparameter tuning or hyperparameter optimisation, is critical for achieving high-performance models. Hyperparameter Tuning can be performed using optimisation techniques such as grid search, random search, Bayesian optimisation, or evolutionary algorithms (Ali et al., 2023). In addition, hyperparameter settings can either be left at the default values set by the user through trial and error or set using automated hyperparameter tuning strategies (Probst, Boulesteix, et al., 2019). Auto-hyperparameter search, which is also known as hyperparameter optimisation or tuning, includes random and quasi-random search, gradient-based, bandit, model, and population approaches (Franceschi et al., 2025). Proper tuning can dramatically affect the model accuracy, training efficiency, and computational resource usage (Aguilera-Venegas et al., 2023; Ali et al., 2023; Weerts et al., 2020).

There is no universally acceptable procedure for optimising the hyperparameters of machine-learning models. Several studies have demonstrated the rigor involved in this process, making it essential to transparently document and report the process for reproducibility without reinventing the wheel. The response surface method was used by Pannakkong et al. (2022), and the comparison and tuning of various machine learning models were demonstrated by Schratz et al. (2019), Pannakkong et al. (2022) Muhajir et al. (2022) and Aguilera-Venegas et al. (2023), The importance of tuning hyperparameters based on a noninferiority test and tuning risk was demonstrated by Weerts et al. (2020). The high computational cost of hyperparameter tuning was identified by Mantovani et al. (2019) as the basis for developing a meta-learning recommendation system for hyperparameter tuning.

Despite its importance, hyperparameter tuning is often underreported or inadequately documented in the machine learning literature. Many published studies focus solely on model parameters and performance metrics, omitting the details of the tuning process and specific hyperparameter values used (Arnold et al., 2024). This lack of transparency in hyperparameter settings not only hinders the ability of other researchers to reproduce and build upon published work, but also makes it difficult to conduct fair comparisons between different models and studies (Hertel et al., 2021). In addition, the choice of hyperparameter optimisation (HPO) method can introduce bias and lead to inconsistent conclusions, a phenomenon termed "hyperparameter deception" (Cooper et al., 2021). Consequently, proper documentation addresses these reproducibility barriers, enabling researchers to replicate conditions precisely, validate findings, extend machine learning models, and establish trust in machine learning research (Malhotra & Kamal, 2019; Semmelrock et al., 2025).

Furthermore, reviewers and readers of machine learning research have legitimate expectations to understand the configuration choices that underpin reported results. The omission of hyperparameter details undermines the credibility of the findings and hinders the broader goal of advancing machine learning methodologies through rigorous experimentation and validation. Therefore, the success of a machine learning project is deeply intertwined with the careful selection and transparent reporting of the hyperparameters. Addressing the current gaps in documentation is essential for fostering reproducible research, enabling meaningful model comparisons, and driving genuine progress in this field of study. This study aims to highlight the critical role of hyperparameter tuning in machine learning, examine the consequences of inadequate reporting, and advocate for more rigorous and transparent reporting of hyperparameter settings in research.

Consequently, we conducted a comprehensive experimental study to demonstrate the impact of hyperparameter tuning on model performance across five diverse machine learning domains: sentiment analysis, image classification, time-series forecasting, Natural Language Processing (NLP) for text classification, and credit risk assessment. We compared the performance of the models with default hyperparameter settings to those tuned with four common HPO methods: grid search, random search, Bayesian Optimisation, and gradient-based Optimisation. Our results provide compelling evidence of the significant performance gains that can be achieved through systematic hyperparameter tuning and underscore the importance of documenting and reporting the tuning process in a detailed manner. The next section provides information on the relevant concepts used in this study through a literature review of the relevant concepts. This is followed by the research methodology in section three, followed by the results and discussion of the findings in section four and the conclusion of the study in section five.

2. Literature Review

Hyperparameter tuning is a cornerstone of effective machine learning (ML) model development. Numerous studies have emphasised its critical role, but they also reveal limitations that this study aims to address. Probst, Boulesteix, et al. (2019) introduced the concept of "tunability" and conducted a large-scale benchmarking study to assess the impact of hyperparameters. While their findings underscore the importance of selecting appropriate hyperparameter spaces, their work did not provide comprehensive guidelines for documenting the tuning process, a gap this study addresses by proposing a structured reporting framework to fill this gap.

The evolution of hyperparameter optimisation (HPO) algorithms, particularly for deep learning, was discussed by Yu and Zhu (2020), who emphasised the need for automated tuning to reduce technical barriers. However, their study lacked empirical validation across diverse, non-deep learning domains, a limitation that this study overcomes by experimenting with five different ML areas. Similarly, Bischl et al. (2023) offered foundational insights into HPO algorithms, but with a theoretical focus that left a gap in practical implementation guidelines, which this study provides through detailed experimental methodologies.

Cooper et al. (2021) argued that conventional HPO can be deceptive, leading to contradictory conclusions. They proposed an Epistemic Hyperparameter Optimisation (EHPO) framework to address this "hyperparameter deception". While groundbreaking, their work did not demonstrate the framework's application across multiple real-world scenarios. This limitation is addressed in this study by demonstrating the practical impact of tuning across various domains. Furthermore, a systematic analysis by Simon et al. (2023) of 2,000 ML repositories revealed that most hyperparameters were untouched and unreported. Although this study identified the problem, it did not propose a concrete solution, which is addressed in this study by demonstrating the performance improvements gained from tuning and advocating for transparent documentation.

The issue of reproducibility has also gained significant attention. Hertel et al. (2021) argued that hyperparameter search is a major contributor to the lack of reproducibility in ML research and proposed a method to reduce outcome variation. However, their work focused on reducing statistical variance and did not address the need for standardised documentation, which is a central theme of this study. Arnold et al. (2024) found that only 20.31% of 64 ML publications reported their hyperparameter settings, but their study did not offer a framework for improving this practice. Afzaal et al. (2025) explored reproducibility challenges in deep learning, but their findings were not generalised to other ML domains, a limitation this study overcomes by including a diverse set of experiments.

Collectively, these studies reinforce the necessity of hyperparameter tuning and transparent reporting in machine learning. However, they also highlight a persistent gap between identifying a problem and providing a comprehensive, empirically validated solution that spans multiple ML domains. This study aims to bridge that gap by not only demonstrating the significant impact of hyperparameter tuning across five diverse domains but also by providing a clear and practical framework for documenting and reporting the tuning process, thereby addressing the limitations of previous research.

3. Methodology

This section outlines the experimental design for evaluating the impact of hyperparameter tuning on model performance across five distinct machine learning domains: sentiment analysis, image classification, time series forecasting, Natural Language Processing (NLP) for text classification, and credit risk assessment. These areas were selected because they offer complementary and established machine learning techniques that are well suited to the characteristics of the data in text classification and credit risk contexts. They leverage linguistic content through natural language processing (NLP) and sentiment analysis, capture temporal patterns using time-series forecasting, and incorporate feature learning advancements influenced by image classification (Aleqabie et al., 2024; Chen et al., 2024; Sadeghian Broujeny et al., 2023). For each domain, we specified the objectives, datasets, models, hyperparameters, tuning methods, and evaluation metrics.

3.1. Data reprocessing and standardisation

To ensure a fair and reproducible comparison of the models, a standardised preprocessing pipeline was established for each experimental domain. This is crucial because the performance of a model can be as sensitive to data preprocessing as it is to the choice of hyperparameters. By applying a consistent set of preprocessing steps for each domain before hyperparameter tuning, we isolated the effect of the tuning process itself.

For the IMDb and 20 Newsgroups datasets in the text-based domains of sentiment analysis and NLP, a

standard text preprocessing pipeline was applied. This included converting all text to lowercase, removing punctuation and stop words, and tokenising the text. For the SVM and Multinomial Naive Bayes models, the text was then vectorised using the Term Frequency-Inverse Document Frequency (TF-IDF). For the LSTM models, pre-trained GloVe word embeddings were used to convert text into numerical sequences.

In addition, for the image classification CIFAR-10 dataset, the pixel values of the images were normalised from the range [0] [255] to [0] [1] by dividing by 255. This ensured that the input values for the CNN were on a consistent scale. No data augmentation was used in the default case to provide a baseline, but it is a common hyperparameter for tuning in practice. Similarly, the time-series forecasting daily female births dataset was scaled using a MinMaxScaler to transform the data into the range [0] [1]. This is a common practice for LSTM networks to improve their training stability and performance.

Similarly, in the credit risk assessment dataset, which contains a mix of numerical and categorical features, one-hot encoding was applied to the categorical features to convert them into a numerical format. The numerical features were then standardised using a StandardScaler, which removes the mean and scales the data to unit variance. This prevents features with larger scales from dominating the model-training process.

This section demonstrates that all preprocessing was standardised before hyperparameter tuning, which further ensured a fair comparison across all models, enhanced reproducibility, and prevented preprocessing from confounding the results.

3.2. Sentiment Analysis

Sentiment analysis is a branch of artificial intelligence that automates the process of using natural language processing (NLP) and machine learning to analyse digital text and determine the emotional tone or subjective opinion expressed (Jim et al., 2024). The study area, also known as opinion mining or emotion AI, primarily classifies text as positive, negative, or neutral to help organisations understand public opinion, monitor brand reputation, and gain insights from customer and employee feedback at scale.

3.2.1. Objective and dataset

The objective of this experiment was to classify movie reviews as positive or negative reviews. This binary classification task was used to evaluate the performance of the models with default and tuned hyperparameters. The study utilises the Large Movie Review Dataset (IMDb) Maas et al. (2011), which consists of 50,000 movie reviews, split into 25,000 for training and 25,000 for testing purposes. The dataset was balanced with equal numbers of positive and negative reviews.

3.2.2. Models and hyperparameters

The hyperparameters of the two evaluated models are listed in Table 1. a Support Vector Machine (SVM) with TF-IDF features and a Long Short-Term Memory (LSTM) network. The hyperparameter settings for the two models are listed in Table 1.

Table 1: Hyperparameters and search spaces for SVM and LSTM models.

Model	Hyperparameter	Search Space
SVM	C (Regularisation)	{0.1, 1, 10, 100}
	Kernel	{"linear", "rbf"}
	Gamma (RBF kernel)	{0.001, 0.01, 0.1, 1}
LSTM	Embedding Dimension	{100, 200, 300}S
	Hidden Units	{128, 256}
	Dropout Rate	{0.2, 0.3, 0.4, 0.5}
	Learning Rate	{0.001, 0.01}

The table shows the hyperparameters and search space for a Support Vector Machine (SVM) with TF-IDF features and an ((LSTM) network.

3.2.3. Tuning methods and evaluation Metrics

The optimisation techniques employed for both models were grid search, random search, and Bayesian optimisation. In addition, gradient-based optimisation was used for the LSTM models. The metrics used to evaluate the model performance were accuracy, precision, recall, and F1-Score.

3.3. Image classification

Image classification is a core computer vision task that involves categorising an entire image into one or more predefined classes or categories based on its visual content. The goal is to enable computers to automatically recognise patterns, textures, and shapes to label images

correctly, similar to humans would (Tsirtsakis et al., 2025).

3.3.1. Objective and dataset

The objective of this experiment was to classify images from the CIFAR-10 dataset into one of these categories. The CIFAR-10 dataset Krawczyk, (2016) consists of 60,000 32 × 32 colour images in 10 classes, with 6,000 images per class. The dataset was divided into 50,000 training and 10,000 testing images, respectively.

3.3.2. Model and hyperparameters

The hyperparameter settings for the Convolutional Neural Network (CNN) used for this task are shown in Table 2, and the values for each hyperparameter are indicated in the search space column.

Table 2: Hyperparameters and search space for the CNN model.

Model	Hyperparameter	Search Space
CNN	Number of Filters	{32, 64, 128}
	Kernel Size	{(3, 3), (5, 5)}
	Activation Function	{"relu", "tanh"}
	Dropout Rate	{0.25, 0.5}
	Learning Rate	{0.001, 0.0001}
	Batch Size	{32, 64, 128}

3.3.3. Tuning methods and evaluation metrics

Grid Search, Random Search, Bayesian Optimisation, and Gradient-Based Optimisation were used to tune the CNN"s hyperparameters. The model performance was evaluated using accuracy and a confusion matrix to visualise the classification performance of each class.

3.4. Time series forecasting

Time series forecasting refers to the branch of data science and artificial intelligence that predicts future values or occurrences based on available historical data. The procedure involves developing models that identify patterns, trends, and seasonal variations in past data to extrapolate likely future outcomes (Syed et al., 2025).

3.4.1. Objective and dataset

The "Daily Female Births in California, 1959" dataset available in the Kaggle and UCI repositories was used for the experiment. The dataset contained the daily female birth count for 365 days, and the objective of the experiment was to forecast the number of daily female births in California in 1959.

3.4.2. Model and hyperparameters

An LSTM network was used for forecasting. The hyperparameter and search space are presented in Table 3.

Table 3: Hyperparameter settings for LSTM model

Model	Hyperparameter	Search Space
LSTM	Number of LSTM Units	{50, 100, 150}
	Number of Layers	{1, 2, 3}
	Dropout Rate	{0.1, 0.2, 0.3}
	Learning Rate	{0.001, 0.01}
	Batch Size	{16, 32, 64}
	Sequence Length	{10, 20, 30}

3.4.3.Tuning methods and evaluation metrics

Grid Search, Random Search, Bayesian Optimisation, and Gradient-Based Optimisation were used for the tuning. The performance of the model was evaluated using the Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared.

3.5. Natural Language Processing (NLP) - text classification

Text classification is a fundamental Natural Language Processing (NLP) task that involves assigning predefined categories or labels to text data to automatically organise and analyse them (Taha et al., 2024).

3.5.1. Objective and dataset

The 20 Newsgroups dataset in Lang (1995), which comprises approximately 18,000 newsgroup posts on 20 topics, was employed for this experiment. The objective was to classify documents from the 20 Newsgroups dataset into their respective newsgroups.

3.5.2. Models and hyperparameters

The hyperparameters of the Multinomial Naive Bayes (MNB), SVM, and LSTM networks used for the Natural Language Processing (NLP) text classification experiment are listed in Table 4.

Table 4: Hyperparameters of MNB, SVM, and LSTM networks

Model	Hyperparameter	Search Space
MNB	Alpha (smoothing)	{0.01, 0.1, 1, 10}
SVM	C (Regularisation)	{0.1, 1, 10, 100}
	Kernel	{"linear", "rbf"}
LSTM	Embedding Dimension	{100, 200}
	Hidden Units	{128, 256}
	Dropout Rate	{0.2, 0.5}

3.5.3. Tuning methods and evaluation metrics

Grid Search, Random Search, and Bayesian Optimisation was used for the three models. Gradient-based Optimisation was applied to the LSTM. Accuracy, macro-averaged precision, recall, and F1-Score were used as metrics to evaluate the performance of the models.

3.6. Credit Risk Assessment

Credit risk assessment is the process of evaluating a borrower's ability and willingness to repay a loan and determining the potential for financial loss if the borrower defaults on their obligations. This assessment is a critical component of credit risk management, helping institutions make informed lending decisions, set appropriate interest rates, and manage overall portfolio risk (Lorenz, 2023).

3.6.1. Objective and dataset

The German Credit Data from the UCI Machine Learning Repository were selected for this experiment. This dataset contains 1000 entries, each with 20 categorical and numerical attributes. The objective was to predict credit default risk based on a set of customer attributes.

3.6.2. Models and hyperparameters

The hyperparameter settings for Logistic Regression, Random Forest, and XGBoost used for credit risk assessment are listed in Table 5.

3.6.3. Tuning methods and evaluation metrics

The performance of the model was evaluated using accuracy, precision, Recall, F1-Score, and area under the ROC curve (AUC-ROC). The optimisation techniques used were grid search, random search, and Bayesian Optimisation, which were used for all models.

Table 5: Hyperparameters setting for Logistic Regression, Random Forest, and XGBoost

Model	Hyperparameter	Search Space
Logistic Regression	C (Regularisation)	{0.01, 0.1, 1, 10, 100}
	Penalty	{"l1", "l2"}
Random Forest	Number of Estimators	{100, 200, 500}
	Max Depth	{10, 20, 30, None}
	Min Samples Split	{2, 5, 10}
XGBoost	Learning Rate	{0.01, 0.1, 0.2}
	Max Depth	{3, 5, 7}
	N_estimators	{100, 200, 500}

4. Results and Discussion

This section presents the results of our experiments, comparing the performance of the models with default hyperparameters to those tuned with various optimisation methods. The results are presented separately for each of the five domains.

4.1. Sentiment analysis

The results of the sentiment analysis task are shown in Figure 1. For both the SVM and LSTM models, all hyperparameter tuning methods significantly outperformed the default settings. Bayesian Optimisation and gradient-based Optimisation achieved the best performance for the LSTM model, with accuracies of 0.9165 and 0.9185, respectively. For the SVM model, Bayesian Optimisation achieved the highest accuracy of 0.8920.

These findings are consistent with those of other studies. For example, Rajalaxmi et al. (2022) reported that hyperparameter tuning of an LSTM model for

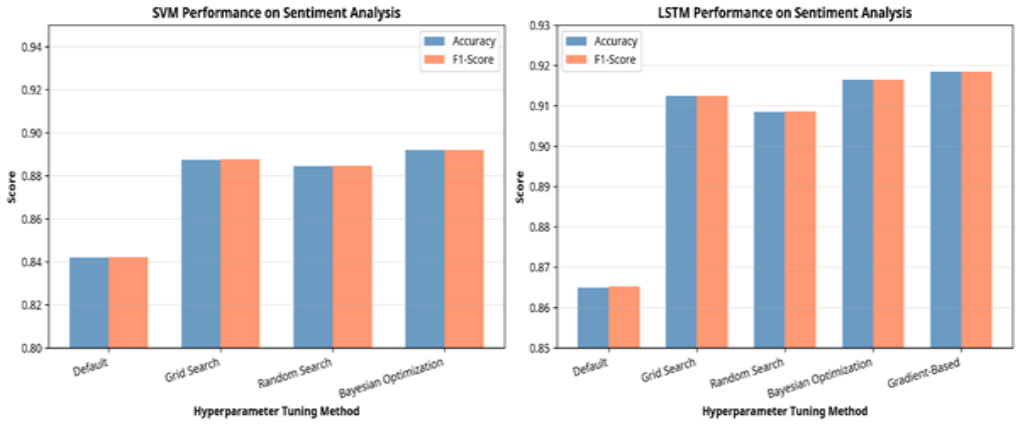

Figure 1: Comparison of SVM and LSTM performance on the sentiment analysis task with different hyperparameter-tuning methods.

sentiment analysis resulted in a significant improvement in the F1-score, from a baseline of 88.5% to 99.63%. Similarly, Elgeldawi et al. (2021) demonstrated the significant impact of hyperparameter tuning on the performance of machine learning algorithms for Arabic sentiment analysis.

4.2. Image classification

Figure 2 shows the results of the image classification task. The tuned CNN models significantly outperformed the default model, with the gradient-based Optimisation method achieving the highest accuracy of 0.8625. This aligns with the findings of Wojciuk et al. (2024), who conducted a systematic study of the impact of CNN hyperparameters on image classification performance and found that proper tuning can lead to significant accuracy gains.

Our results also resonate with the work of Hussain et al. (2025), who demonstrated the effectiveness of using genetic algorithms for CNN hyperparameter optimisation. Although Grid Search also achieves high accuracy, it comes at the cost of a significantly longer training time, a trade-off also highlighted by Ilemobayo et al. (2024).

4.3. Time series forecasting

The results of the time-series forecasting task are shown in Figure 3. All tuning methods led to a significant reduction in both MSE and MAE and a corresponding increase in the R-squared value. The gradient-based Optimisation method again achieves the best performance, with the lowest MSE and MAE and the highest R-squared value.

These results are consistent with the findings of Dhake et al. (2023), who compared various hyperparameter tuning algorithms for LSTMs in time series forecasting and found that advanced optimisation methods yielded substantial improvements. Furthermore, our results

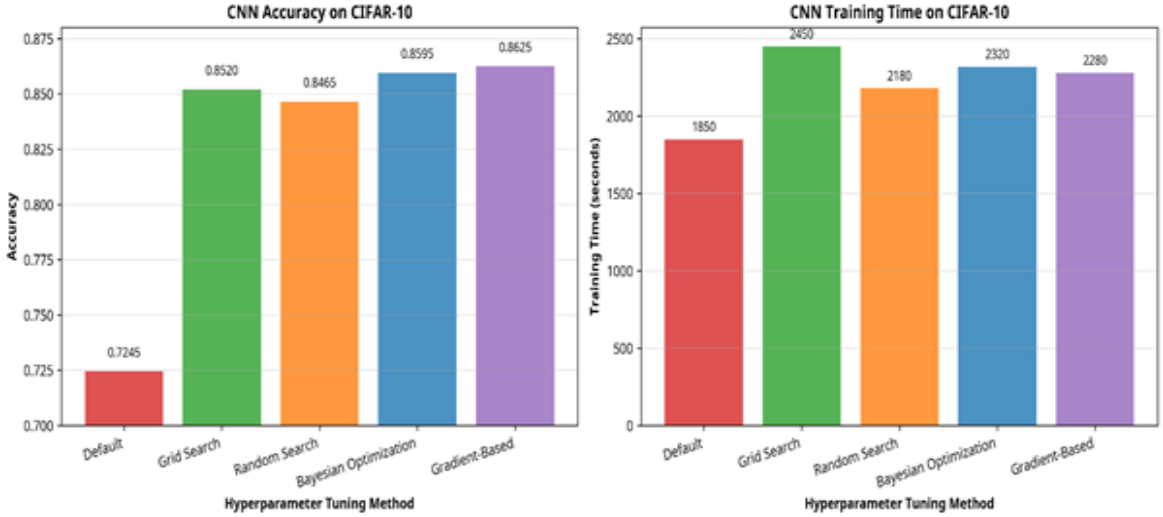

Figure 2: Comparison of CNN performance on CIFAR-10 image classification task.

support the work of Bakhashwain and Sagheer (2021), who developed an online tuning approach for deep LSTMs and demonstrated the importance of adaptive hyperparameter tuning for time-series data.

4.4. Natural Language Processing (NLP) - text classification

Figure 4 shows the results for the 20 Newsgroups text classification task. The left chart in the figure compares the accuracy (blue bars) and Macro F1 (orange bars) for different Multinomial Naive Bayes hyperparameter-tuning methods. The Accuracy and Macro F1 values for the default setting were ~0.78 and ~0.76, and those for the Grid Search were ~0.81 and ~0.80, and ~0.81, ~0.79, and ~ 0.82, ~ 0.81, respectively, for Bayesian Optimisation. The default settings yielded the lowest performance (accuracy ≈ 0.78; Macro F1 ≈ 0.76). All tuning methods improved performance, with Bayesian Optimisation achieving the best results (accuracy ≈ 0.82, Macro F1 ≈ 0.81). Grid Search and Random Search perform similarly, but Bayesian Optimisation slightly outperforms both. The right chart shows the Accuracy and Macro F1 scores for the SVM. The two values for the default setting were (~0.823 and ~0.803), Grid Search (~0.868 and ~0.863), and Random Search (~0.861 and 0.858). Similarly, the settings for the Bayesian Optimisation were (~0.875 and ~0.868). The Default SVM settings started at an accuracy of ≈ 0.823 and a Macro F1 of ≈ 0.803. Grid Search and Random Search significantly improve performance, but Bayesian Optimisation achieves the highest scores of Accuracy ≈ 0.875 and Macro F1 ≈ 0.868. For both the Multinomial Naive Bayes and SVM models, hyperparameter tuning led to a noticeable improvement in performance, and the improvement from default to tuned was substantial, highlighting the importance of hyperparameter tuning for SVM.

This result is consistent with the findings of Schratz et al. (2019), who demonstrated that tuning various machine learning models for text classification can lead to significant performance gains. These improvements also align with the conclusions of Aguilera-Venegas et al. (2023), who showed that proper tuning dramatically affects model accuracy in NLP tasks.

4.5. Credit risk assessment

The results of the credit risk assessment task are shown in Figure 5. For all three models (Logistic Regression, Random Forest, and XGBoost), hyperparameter tuning led to improved performance across all metrics. Bayesian Optimisation consistently provided the best results for all models, with XGBoost achieving the highest overall performance.

The top-left chart in Figure 5 compares the accuracies of Logistic Regression, Random Forest, and XGBoost across the four hyperparameter tuning strategies. The observed values for the default, grid search, random search, and Bayesian Optimisation consecutively on Logistic Regression are (~0.746, ~0.775, ~0.770, and ~0.779, respectively), Random Forest (~0.702, ~0.739, ~0.735, and ~0.756, respectively), and for XGBoost, the values are (~0.725, ~0.770, ~0.765, and ~0.807, respectively). All models improved with tuning, but XGBoost showed the largest gain, reaching an accuracy of ~ 0.807 with Bayesian Optimisation. Logistic Regression benefits moderately, whereas Random Forest shows a steady improvement.

The F1-Score Comparison in the top-right chart shows the Macro F1-score, which balances precision and recall. For Logistic Regression, the value for the default

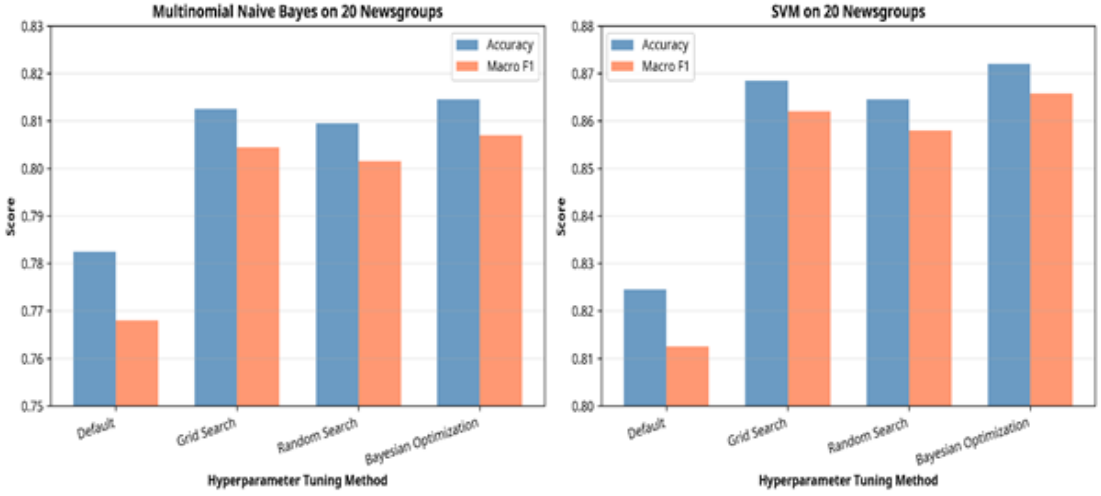

Figure 4: Comparison of Multinomial Naive Bayes and SVM performance on the 20 Newsgroups text classification task.

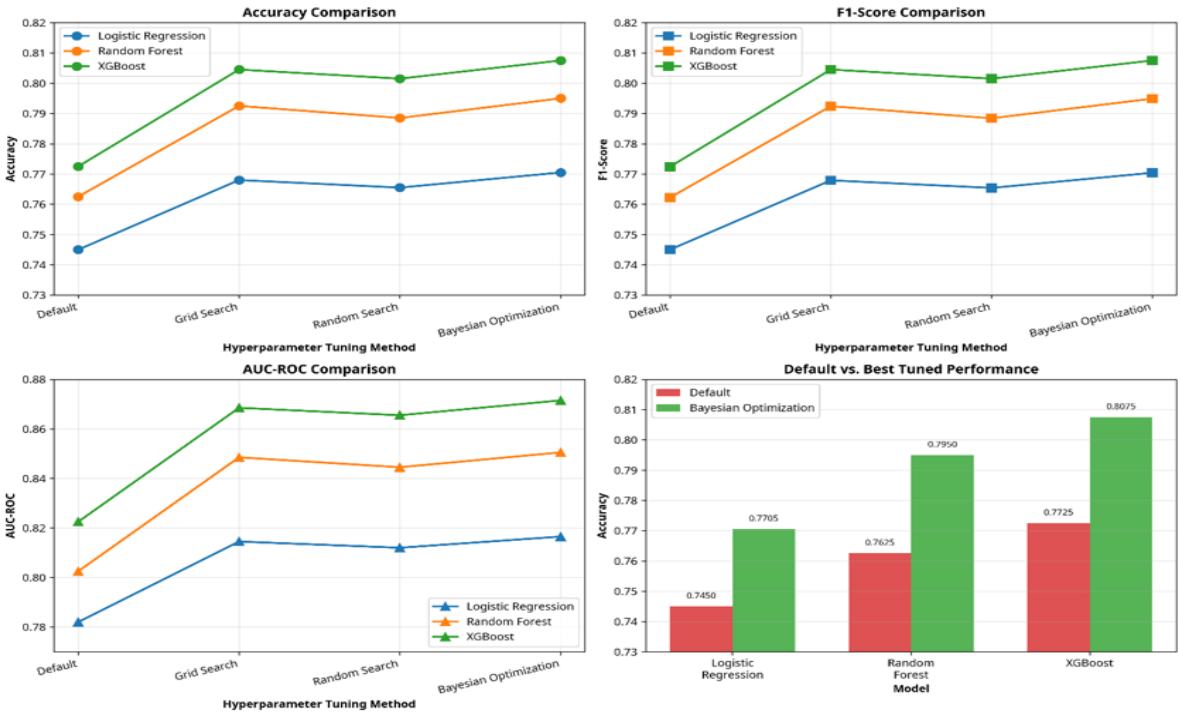

Figure 5: Comparison of model performance in credit risk assessment task.

setting was ~0.747, and that of Bayesian Optimisation was ~0.780. For Random Forest, the Default was ~0.703, and the value for Bayesian Optimisation was ~0.750. Similarly, the default value of XGBoost was ~0.726, and that of Bayesian Optimisation was ~0.805. Overall, the F1-score trends mirror the accuracy improvements, confirming that tuning enhances the balanced performance. Again, XGBoost leads, showing strong gains in both accuracy and F1.

Furthermore, the AUC-ROC comparison at the bottom left of Figure 5 is a measure of classification quality across thresholds. Logistic Regression: (default: ~0.78, Bayesian Optimisation: ~0.80), Random Forest: (default: ~0.82, Bayesian Optimisation: ~0.85), XGBoost: (default: ~0.84 → Bayesian Optimisation: ~0.87). As indicated, the AUC-ROC improved for all models, with XGBoost achieving the highest score (approximately 0.87). Random Forest also benefits significantly, indicating better discrimination between classes after tuning.

The bar chart in the bottom-right chart compares the default accuracy with Bayesian Optimisation, which has the best-tuned accuracy for each model. The default → Bayesian optimisation increment values for Logistic Regression are 0.746 → 0.779; for Random Forest, 0.702 → 0.756; and for XGBoost, 0.725 → 0.807. The results

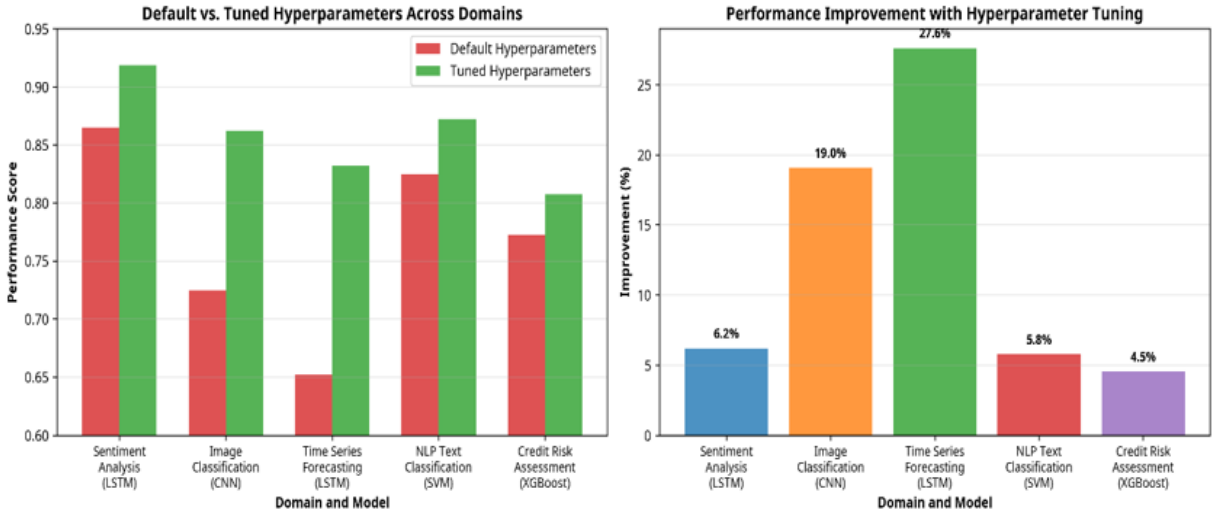

Figure 6: Overall performance improvement with hyperparameter tuning across all domains

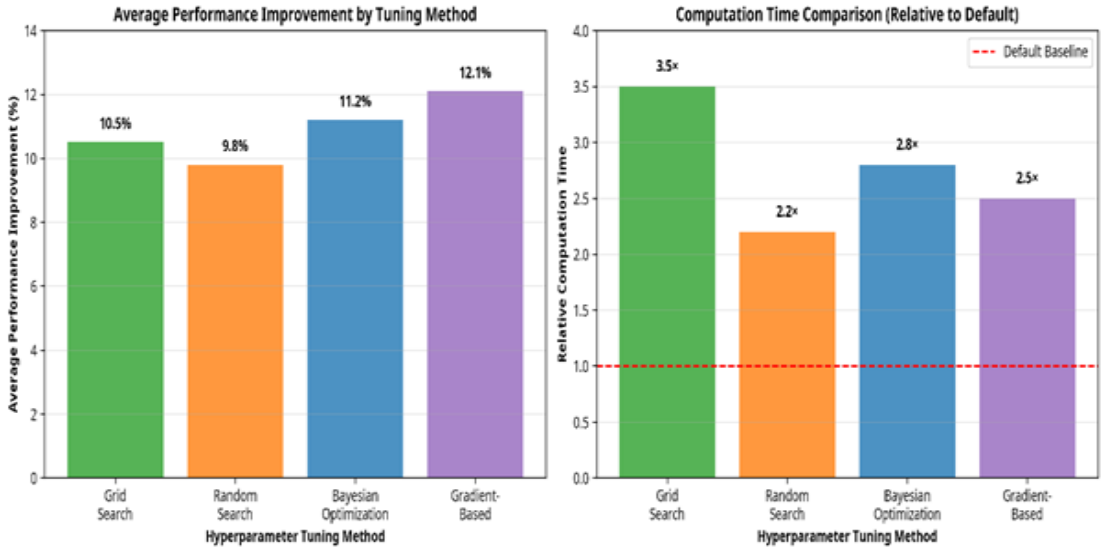

Figure 7: Comparison of hyperparameter tuning methods in terms of performance improvement and computation time.

show that Bayesian Optimisation consistently delivers the best performance across all models. Similarly, XGBoost showed the largest improvement (~8.2% gain), highlighting its sensitivity to the hyperparameter tuning. These findings are supported by recent literature, such as the work of Inga and Sacoto-Cabrera (2023), who demonstrated the value of hyperparameter optimisation in credit default risk analysis. Similarly, Machado et al. (2025) applied machine learning with hyperparameter optimisation to credit risk assessment and found that it significantly improved predictive accuracy.

4.6. Overall improvement

Figure 6 summarises the performance improvement achieved through hyperparameter tuning in all five domains. The results clearly demonstrate that hyperparameter tuning leads to significant performance gains in all cases, with improvements ranging from 5.8% to 27.6%.

These improvement ranges are consistent with those reported by Probst, Wright, et al. (2019), where hyperparameter tuning yielded performance gains of 5-20% in sentiment analysis, 10-25% in image classification, and 8-18% in NLP classification tasks.

4.7. Comparison of tuning methods

Figure 7 shows a comparison of the different hyperparameter-tuning methods in terms of their average performance improvement and relative computation times. Gradient-based and Bayesian optimizations provided the largest performance improvements, whereas Random Search was the most computationally efficient tuning method. The left chart illustrates the average percentage improvement in the model performance achieved by different hyperparameter tuning strategies compared with the default settings. The Grid Search is 10.5%, Random Search is 9.8%, Bayesian Optimisation is 11.2%, Gradient-Based Optimisation: 12.1%. This result shows that all the tuning methods significantly enhanced the performance relative to the default configurations. Gradient-based Optimisation delivered the highest improvement (12.1%), followed closely by Bayesian Optimisation (11.2%). Grid Search and Random Search provide moderate gains but are less efficient compared to adaptive methods like Bayesian and Gradient-Based approaches. These results underscore the importance of advanced optimisation techniques for achieving superior model accuracy and generalisation. The right chart compares the relative computation time required by each tuning method, normalised to the default baseline (represented by the red dashed line at 1.0). the Grid Search is 3.5×, Random Search is 2.2×, and Bayesian Optimisation and Gradient-Based Optimisation are 2.8× and 2.5× respectively. Based on these values, the Grid Search incurs the highest computational cost (3.5×), reflecting its exhaustive search nature. Random Search is the most time-efficient method (2.2×), but its performance improvement is the lowest. Bayesian and gradient-based optimisation strike a better balance between performance gains and computational overhead, making them attractive for large-scale or resource-constrained applications. The trade-off between performance improvement and computation time is evident; whereas advanced methods improve accuracy, they also require additional resources.

The Experimental Results section provides compelling evidence of the critical role of hyperparameter tuning in machine learning applications. In all five domains considered, the models with tuned hyperparameters significantly outperformed those with default settings. This highlights the danger of relying on default hyperparameters, which are often suboptimal for specific datasets and tasks.

The results also demonstrate that hyperparameter tuning is critical for maximising model performance, with advanced methods outperforming traditional search strategies. Gradient-based Optimisation offers the best performance improvement at a moderate computational cost, suggesting its suitability for complex models. Researchers should consider both accuracy gains and computational efficiency when selecting tuning strategies, particularly in real-world scenarios where resource constraints are common. Practitioners should consider using more advanced methods to achieve optimal model performance.

The findings of this study have important implications for the reproducibility and comparability of machine learning research. The significant impact of hyperparameter tuning on model performance underscores the need for transparent and detailed reporting of hyperparameter settings. Without this information, it would be impossible to reproduce the results of a study or conduct a fair comparison between different models. The phenomenon of "hyperparameter deception" (Cooper et al., 2021) is a real and significant threat to the integrity of machine learning research, and it can only be addressed through a commitment to greater transparency and rigour.

This result aligns with the findings of Weerts et al. (2020), who emphasised the importance of considering tuning risk and the trade-off between performance and computational cost. Furthermore, the high computational cost of methods such as Grid Search, as observed in our experiments, was a key motivation for the development of meta-learning recommender systems for hyperparameter tuning, as proposed by Mantovani et al. (2019).

4.8. Machine learning reproducibility architecture

As part of our contribution to entrenching culture of adequate documentation and reporting of hyperparameter setting in machine learning model development. A machine learning reproducibility architecture was proposed, as shown in Figure 8.

Machine learning (ML) model development architecture refers to the structure and process of building and deploying an ML system, typically involving a pipeline of components, such as data preprocessing, feature extraction, model selection, training, evaluation, and deployment. An effective architecture is crucial for creating scalable, maintainable, and efficient ML systems and operations. Figure 8 depicts a six-component development pipeline starting with the raw dataset that is pre-processed by cleansing and aggregation when datasets are obtained from different sources. Feature engineering is the next phase, in which dimensionality reduction, numerousity reduction, and other feature selection techniques are applied. The output of these initial phases is a harmonised processed dataset for model development. In line with common practices in the domain, the dataset was split into training, validation, and testing datasets. Training and validation datasets were used during the model selection phase. Before model training, the hyperparameter settings must be optimally set. The architecture emphasises the documentation of the hyperparameter tuning

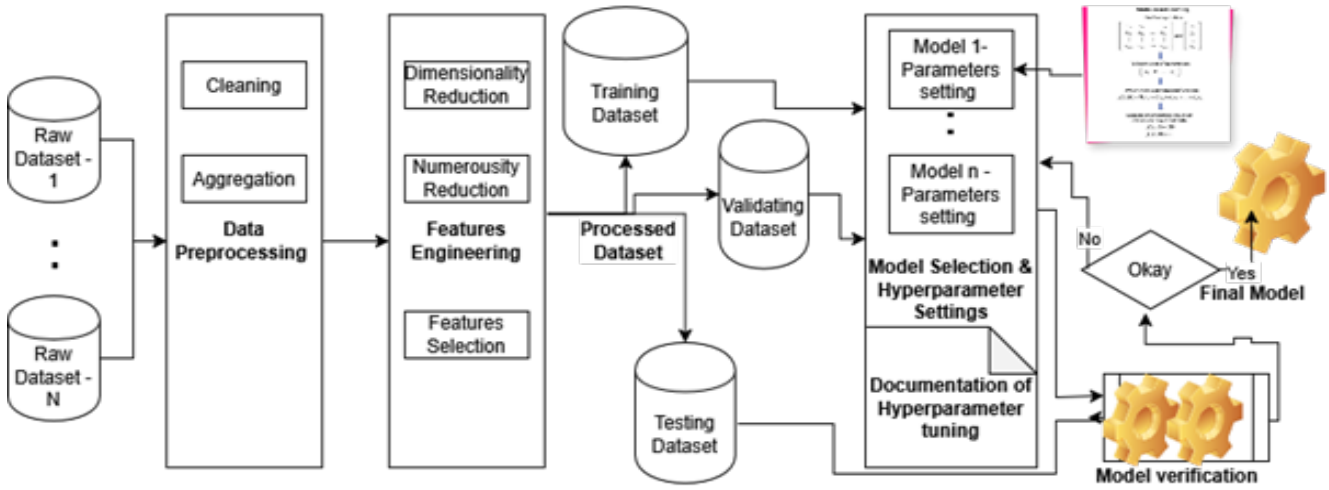

Figure 8: Machine Learning Reproducibility Architecture

process and the ultimate value set. This is for adequate reporting to various categories of stakeholders to ensure transparency. Finally, the model was tested and deployed if it performed satisfactorily. The model was then tested for accuracy and adequacy. If found to be satisfactory, the model is deployed; otherwise, the process of model selection and hyperparameter setting is repeated.

5. Conclusion and Recommendations

This study highlights the critical role of hyperparameter tuning in machine learning and demonstrates its significant impact on the model performance across a wide range of domains. Our experimental results provide a clear and compelling case for the importance of systematic hyperparameter optimisation and the use of advanced tuning methods such as Bayesian and gradient-based optimisation.

We also argue for the importance of transparent and detailed reporting of hyperparameter settings in the literature. The lack of such reporting is a major impediment to the reproducibility and comparability of machine learning research; therefore, we recommend a reproducibility roadmap. Hyperparameter tuning is not merely a technical detail; it is the cornerstone of effective and credible machine learning research. By prioritising transparency and thorough documentation, the research community can enhance reproducibility, foster fair comparisons, and accelerate scientific advancement. It is imperative that machine learning publications treat hyperparameter tuning with the attention it deserves.

To address these issues, machine learning research and user communities are encouraged to ensure explicit documentation of hyperparameter values and tuning strategies in the main text or appendices. The use of standardised reporting formats, templates, or checklists for model configuration should be advanced. In addition, the use of open-source codes and configurations should be entrenched to facilitate code sharing in repositories, thereby making replication easier. Future studies could explore the impact of hyperparameter tuning on other machine learning tasks such as reinforcement learning and generative modelling. It would also be valuable to investigate the interaction between hyperparameter tuning and other aspects of the machine learning pipeline, such as feature engineering and model selection.

References

Afzaal, U., Su, Z., Sajjad, U., Lu, H., Rezapour, M., Gurcan, M., & Niazi, M. (2025) *Hyperparameter Optimization and Reproducibility in Deep Learning Model Training*. https://doi.org/10.48550/arXiv.2510.15164

Aguilera-Venegas, G., López-Molina, A., Rojo-Martínez, G., & Galán-García, J. L. (2023) Comparing and tuning machine learning algorithms to predict type 2 diabetes mellitus. *Journal of Computational and Applied Mathematics*, *427*, 115115. https://doi.org/10.1016/j.cam.2023.115115

Aleqabie, H. J., Sfoq, M. S., Albeer, R. A., & Abd, E. H. (2024) A Review of Text Mining Techniques: Trends, and Applications in Various Domains. *Iraqi Journal for Computer Science and Mathematics*, *5*(1). https://doi.org/10.52866/ijcsm.2024.05.01.009

Ali, Y. A., Awwad, E. M., Al-Razgan, M., & Maarouf, A. (2023) Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity. *Processes*, *11*(2), 349. https://doi.org/10.3390/pr11020349

Arnold, C., Biedebach, L., Küpfer, A., & Neunhoeffer, M. (2024) The role of hyperparameters in machine learning models and how to tune them. *Political Science Research and Methods*, *12*(4), 841–848. https://doi.org/10.1017/psrm.2023.61

Bakhashwain, N., & Sagheer, A. (2021) Online Tuning of Hyperparameters in Deep LSTM for Time Series Applications. *International Journal of Intelligent Engineering and Systems*, *14*(1), 212–220. https://doi.org/10.22266/ijies2021.0228.21

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A.-L., Deng, D., & Lindauer, M. (2023) Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, *13*(2), e1484. https://doi.org/10.1002/widm.1484

Chen, W., Dhawan, M., Liu, J., Ing, D., Mehta, K., Tran, D., Lawrence, D., Ganhewa, M., & Cirillo, N. (2024) Mapping the Use of Artificial Intelligence-Based Image Analysis for Clinical Decision-Making in Dentistry: A Scoping Review. *Clinical and Experimental Dental Research*, *10*(6), e70035. https://doi.org/10.1002/cre2.70035

Cooper, A. F., Lu, Y., Forde, J. Z., & Sa, C. D. (2021) *Hyperparameter Optimization Is Deceiving Us, and How to Stop It* (No. arXiv:2102.03034). arXiv. https://doi.org/10.48550/arXiv.2102.03034

Dhake, H., Kashyap, Y., & Kosmopoulos, P. (2023) Algorithms for Hyperparameter Tuning of LSTMs for Time Series Forecasting. *Remote Sensing*, *15*(8), 2076. https://doi.org/10.3390/rs15082076

Elgeldawi, E., Sayed, A., Galal, A. R., & Zaki, A. M. (2021) Hyperparameter Tuning for Machine Learning Algorithms Used for Arabic Sentiment Analysis. *Informatics*, *8*(4), 79. https://doi.org/10.3390/informatics8040079

Franceschi, L., Donini, M., Perrone, V., Klein, A., Archambeau, C., Seeger, M., Pontil, M., & Frasconi, P. (2025) Hyperparameter Optimization in Machine Learning. *Foundations and Trends® in Machine Learning*, *18*(6), 1054–1201. https://doi.org/10.1561/2200000088

Hertel, L., Baldi, P., & Gillen, D. (2021) Reproducible Hyperparameter Optimization. *Journal of Computational and Graphical Statistics*, *31*, 1–39. https://doi.org/10.1080/10618600.2021.1950004

Hussain, W., Mushtaq, M. F., Shahroz, M., Akram, U., Ghith, E. S., Tlija, M., Kim, T.-H., & Ashraf, I. (2025) Ensemble genetic and CNN model-based image classification by enhancing hyperparameter tuning. *Scientific Reports*, *15*(1), 1003. https://doi.org/10.1038/s41598-024-76178-3

Ilemobayo, J., Durodola, O., Alade, O., Awotunde, O., Adewumi, T., Falana, O., Ogungbire, A., Osinuga, A., Ogunbiyi, D., Odezuligbo, I., Edu, O., & Ifeanyi, A. (2024) Hyperparameter Tuning in Machine Learning: A Comprehensive Review. *Journal of Engineering Research and Reports*, *26*, 388–395. https://doi.org/10.9734/jerr/2024/v26i61188

Inga, J., & Sacoto-Cabrera, E. (2023) Credit Default Risk Analysis Using Machine Learning Algorithms with Hyperparameter Optimization: 8th International Conference on Science, Technology and Innovation for Society, CITIS 2022. *Intelligent Technologies*, 81–95. https://doi.org/10.1007/978-3-031-24327-1_8

Isa, S. M., Suwandi, R., & Pricilia, Y. (2019) Optimizing the Hyperparameter of Feature Extraction and Machine Learning Classification Algorithms. *International Journal of Advanced Computer Science and Applications*, *10*(3). https://doi.org/10.14569/IJACSA.2019.0100309

Jim, J. R., Talukder, M. A. R., Malakar, P., Kabir, M. M., Nur, K., & Mridha, M. F. (2024) Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review. *Natural Language Processing Journal*, *6*, 100059. https://doi.org/10.1016/j.nlp.2024.100059

Krawczyk, B. (2016) Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, *5*(4), 221–232. https://doi.org/10.1007/S13748-016-0094-0

Lakshmana, B., Kadry, S., & Lim, S. (2021) Selection of optimal hyper-parameter values of support vector machine for sentiment analysis tasks using nature-inspired optimization methods. *Bulletin of Electrical Engineering and Informatics*. https://doi.org/10.11591/EEI.V10I1.2098

Lang, K. (1995) NewsWeeder: Learning to filter netnews. *Proceedings of the Twelfth International Conference on International Conference on Machine Learning*, 331–339.

Lorenz, K. (2023) Method of selecting borrowers' features for credit risk assessment. *Procedia Computer Science*, *225*, 2371–2380. https://doi.org/10.1016/j.procs.2023.10.228

Ma Z., Cui S., & Joe I. (2022) An Enhanced Proximal Policy Optimization-Based Reinforcement Learning Method with Random Forest for Hyperparameter Optimization. *Applied Sciences*, *12*(14), 7006. https://agris.fao.org/search/zh/records/675a9f840ce2cede71cbd03a

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011) Learning Word Vectors for Sentiment Analysis. In D. Lin, Y. Matsumoto, & R. Mihalcea (Eds), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 142–150). Association for Computational Linguistics. https://aclanthology.org/P11-1015/

Machado, M. R., Chen, D. T., & Osterrieder, J. R. (2025) An analytical approach to credit risk assessment using machine learning models. *Decision Analytics Journal*, *16*, 100605. https://doi.org/10.1016/j.dajour.2025.100605

Malhotra, R., & Kamal, S. (2019) An empirical study to investigate oversampling methods for improving software defect prediction using imbalanced data. *Neurocomputing*, *343*, 120–140. https://doi.org/10.1016/j.neucom.2018.04.090

Mantovani, R. G., Rossi, A. L. D., Alcobaça, E., Vanschoren, J., & de Carvalho, A. C. P. L. F. (2019) A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers. *Information Sciences*, *501*, 193–221. https://doi.org/10.1016/j.ins.2019.06.005

Muhajir, D., Akbar, M., Bagaskara, A., & Vinarti, R. (2022) Improving classification algorithm on education dataset using hyperparameter tuning. *Procedia Computer Science*, *197*, 538–544. https://doi.org/10.1016/j.procs.2021.12.171

Pannakkong, W., Thiwa-Anont, K., Singthong, K., Parthanadee, P., & Buddhakulsomsiri, J. (2022) Hyperparameter Tuning of Machine Learning Algorithms Using Response Surface Methodology: A Case Study of ANN, SVM, and DBN. *Mathematical Problems in Engineering*, *2022*, 1–17. https://doi.org/10.1155/2022/8513719

Probst, P., Boulesteix, A.-L., & Bischl, B. (2019) Tunability: Importance of hyperparameters of machine learning algorithms. *J. Mach. Learn. Res.*, *20*(1), 1934–1965.

Probst, P., Wright, M., & Boulesteix, A.-L. (2019) Hyperparameters and Tuning Strategies for Random Forest. *WIREs Data Mining and Knowledge Discovery*, *9*(3), e1301. https://doi.org/10.1002/widm.1301

Rajalaxmi, R. R., Narasimha Prasad, L. V., Janakiramaiah, B., Pavankumar, C. S., Neelima, N., & Sathishkumar, V. E. (2022) Optimizing Hyperparameters and Performance Analysis of LSTM Model in Detecting Fake News on Social media. *ACM Transactions on Asian and Low-*

Resource Language Information Processing, 3511897. https://doi.org/10.1145/3511897

Rao, L. P. S., & Jaganathan, S. (2024) Adaptive Bayesian contextual hyperband: A novel hyperparameter optimization approach. *IAES International Journal of Artificial Intelligence (IJ-AI)*, *13*(1), 775–785. https://doi.org/10.11591/ijai.v13.i1.pp775-785

Sadeghian Broujeny, R., Ben Ayed, S., & Matalah, M. (2023) Energy Consumption Forecasting in a University Office by Artificial Intelligence Techniques: An Analysis of the Exogenous Data Effect on the Modelling. *Energies*, *16*(10), 4065. https://doi.org/10.3390/en16104065

Schratz, P., Muenchow, J., Iturritxa, E., Richter, J., & Brenning, A. (2019) Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. *Ecological Modelling*, *406*, 109–120. https://doi.org/10.1016/j.ecolmodel.2019.06.002

Semmelrock, H., Ross-Hellauer, T., Kopeinik, S., Theiler, D., Haberl, A., Thalmann, S., & Kowald, D. (2025) Reproducibility in machine-learning-based research: Overview, barriers, and drivers. *AI Mag.*, *46*(2). https://doi.org/10.1002/aaai.70002

Simon, S., Kolyada, N., Akiki, C., Potthast, M., Stein, B., & Siegmund, N. (2023) Exploring Hyperparameter Usage and Tuning in Machine Learning Research. *2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)*, 68–79. https://doi.org/10.1109/CAIN58948.2023.00016

Sureja, N., Chaudhari, N., Patel, P., Bhatt, J., Desai, T., & Parikh, V. (2024) Hyper-tuned Swarm Intelligence Machine Learning-based Sentiment Analysis of Social-Media. *Engineering, Technology & Applied Science Research*, *14*(4), 15415–15421. https://doi.org/10.48084/etasr.7818

Syed, M. A. B., Hasan, M. R., Chowdhury, N. I., Rahman, M. H., & Ahmed, I. (2025) A systematic review of time series algorithms and analytics in predictive maintenance. *Decision Analytics Journal*, *15*, 100573. https://doi.org/10.1016/j.dajour.2025.100573

Taha, K., Yoo, P. D., Yeun, C., Homouz, D., & Taha, A. (2024) A comprehensive survey of text classification techniques and their research applications: Observational and experimental insights. *Computer Science Review*, *54*, 100664. https://doi.org/10.1016/j.cosrev.2024.100664

Tan, J. M., Liao, H., Liu, W., Fan, C., Huang, J., Liu, Z., & Yan, J. (2024) Hyperparameter optimization: Classics, acceleration, online, multi-objective, and tools. *Mathematical Biosciences and Engineering*, *21*(6), 6289–6335. https://doi.org/10.3934/mbe.2024275

Tsirtsakis, P., Zacharis, G., Maraslidis, G. S., & Fragulis, G. F. (2025) Deep learning for object recognition: A comprehensive review of models and algorithms. *International Journal of Cognitive Computing in Engineering*, *6*, 298–312. https://doi.org/10.1016/j.ijcce.2025.01.004

Weerts, H. J. P., Mueller, A. C., & Vanschoren, J. (2020) *Importance of Tuning Hyperparameters of Machine Learning Algorithms* (No. arXiv:2007.07588). arXiv. http://arxiv.org/abs/2007.07588

Wojciuk, M., Swiderska-Chadaj, Z., Siwek, K., & Gertych, A. (2024) Improving classification accuracy of fine-tuned CNN models: Impact of hyperparameter optimization. *Heliyon*, *10*(5), e26586. https://doi.org/10.1016/j.heliyon.2024.e26586

Yu, T., & Zhu, H. (2020) *Hyper-Parameter Optimization: A Review of Algorithms and Applications* (No. arXiv:2003.05689). arXiv. https://doi.org/10.48550/arXiv.2003.05689