

Hyperparameter Optimization of the Perceptron Algorithm for Determining the Feasibility of Research Proposals and Community Service

Lisnawita*¹, Guntoro², Loneli Costaner³

^{1,2,3}Program Studi Teknik Informatika/Universitas Lancang Kuning

E-mail: Lisnawita@unilak.ac.id¹, Guntoro@unilak.ac.id², Lonelcostaner@unilak.ac.id³

Abstrac

Higher education in Indonesia includes diploma, bachelor, master, specialist, and doctoral programmes organised by universities. The Institute for Research and Community Service (LPPM) is in charge of assessing lecturers' proposals. This research aims to optimise the Perceptron algorithm to assess proposal eligibility using Turnitin plagiarism scores and reviewer scores. The optimisation results show that Perceptron accuracy reaches 99.44% to 99.63% at various training data ratios. GridSearchCV achieved 100% accuracy, while RandomisedSearchCV recorded accuracy between 98.89% to 99.63%. GridSearchCV also had the lowest MSE, despite higher Loss values, indicating a sacrifice in generalisation ability. Perceptron Default and RandomisedSearchCV had higher MSE and Loss, but remained low. GridSearchCV's AUC reached 100%, while Perceptron Default and RandomisedSearchCV showed very high AUC, ranging from 99.25% to 99.98%. Overall, the Perceptron algorithm is effective in assessing proposal eligibility with high accuracy.

Keyword: Research, Community Service, Perceptron, LPPM

1. Introduction

Higher education in Indonesia includes education programmes organised by universities after secondary education, including diploma, bachelor, master, specialist, and doctoral programmes. Each university has a Research and Community Service Institute (LPPM) that is tasked with managing research and community service activities conducted by lecturers. These activities are organised efficiently and in accordance with established standards, while the assessment and funding processes are fully managed by LPPM.

Procedures established by LPPM ensure that every research and community service activity complies with applicable regulations. One important step in this process is the evaluation of research and community service proposals, which must fulfil certain criteria to be eligible for funding. Each proposal undergoes a selection process that involves administrative and substantive reviews by a team of reviewers appointed by LPPM. However, this manual process is often time-consuming and inconsistent, leading to delays in proposal evaluation. The Perceptron algorithm, which is based on artificial neural networks, offers a promising solution. It is able to recognise patterns and dependencies between variables in data. During

training, Perceptron adjusts the weights and biases to produce a model that can make predictions with a high degree of accuracy. The advantage of Perceptron lies in its ability to handle classification problems with small data sets and a deep understanding of the variety of variables that affect the eligibility of research and community service proposals. By utilising this algorithm, this research aims to overcome inefficiencies in the proposal evaluation process, speed up, and improve the consistency of the evaluation process.

Various previous studies have shown the effectiveness of the Perceptron algorithm in various fields, such as: Research [1] that uses 40 input data, where Perceptron produces the best results with a certain training-testing ratio, Research [2] that shows the algorithm's ability to predict loan results for cooperatives, Research [3] that analyses the impact of malnutrition on toddler development using artificial neural networks, Research [4] that evaluates Perceptron models with MSE and RMSE metrics for performance assessment, Research [5] which classifies student personality with high accuracy, Research [6] which assesses student knowledge with high accuracy, Research [7] [8] [9] [10] which show that the MultiLayer Perceptron prediction model provides the best results on various tasks, such as predictive analysis and classification in more complex contexts.

Although the application of the Perceptron algorithm has been widely used in various purposes, the use of Perceptron for evaluating research and community service proposals in higher education is still limited.

This research aims to fill the gap by utilising labelled proposal data to optimise the Perceptron algorithm. By using labelled data as training data, this research will automate the proposal evaluation process, improve consistency, reduce the time required, and achieve high accuracy in determining proposal eligibility. This research also aims to optimise the hyperparameters of the Perceptron algorithm using the randomsearchSv and GridSearch techniques, by utilising two main variables, namely the plagiarism score from Turnitin (Turnitin score) and the score from the reviewer, as the main features in the classification model.

The optimisation process is performed on the hyperparameters of the Perceptron algorithm in order to obtain a more accurate model in evaluating proposals. The randomsearchSv and GridSearch method was chosen due to its ability to explore the hyperparameter. The main variables used in the classification model are the plagiarism score obtained from Turnitin, as well as the score given by the reviewer on the quality of the proposal. By using these two variables as features, the Perceptron algorithm is expected to produce faster, more accurate, and consistent proposal classification.

2. Research Methodology

1. Research Location: This research was conducted at LPPM XYZ
2. Population and Sample: The population in this study is the lecturers of XYZ , with a sample of 209 from the year 2023
3. Research Variables

Research and Community Service Proposal Data: This data includes information submitted in the proposal, including the title, background, objectives, and expected outcomes of each proposal submitted by lecturers.

Funded Proposals: This includes proposals that successfully passed the selection and received funding from LPPM. The eligibility of the proposal is determined based on two main sources, namely the results of Turnitin checking (to check for plagiarism) and the judgement given by the reviewer on the eligibility of the proposal.

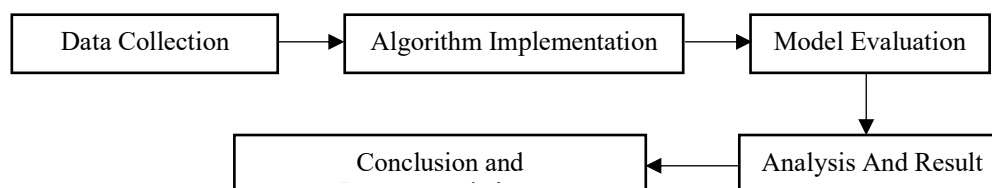


Figure 1. Research Design

1. Data Collection

Collect data related to research and community service proposals, and perform data preprocessing

2. Algorithm Implementation

In the implementation stage of the Perceptron algorithm, a model was created using the processed data, with features taken from the proposal. model is created using the processed data, with features taken from the proposal. The model is trained with the labelled data to optimise the weights in the artificial neural network, in order to predict proposal eligibility. During training, hyperparameter optimisation, such as learning rate, number of hidden layers, number of neurons, and epochs, is performed to improve model performance. This optimisation was performed using grid search and randomised search techniques [11]–[16]. After the model is trained, testing is conducted with test data to evaluate the accuracy and ability of the model to classify proposal eligibility.

3. Model Evaluation

Evaluate the implemented Perceptron model by calculating the accuracy, MSE,, AUC and Loss.

4. Analysis And Result

In this section, we will analyse the evaluation results of the Perceptron model after testing. The analysis will focus on how the factors in the proposal affect the classification results. The discussion will address the comparison of the results obtained with the default algorithm with hyparparameter optimisation, as well as explain the reasons why the Perceptron algorithm can produce good accuracy in this task.

5. Conclusion and Recommendations

Conclusion and Recommendations: Summarize the research findings and provide suggestions for future research development

3. Results and Discussion**3.1 Data Collection**

The data used for this research comes from the LPPM, comprising 209 entries based on the proposal submissions of lecturers within the Unilak environment.

Table 1. Data Collection

No	Score	Turnitin	Average	Label
1	300	100	200	Negatif
2	450	100	275	Negatif
3	450	100	275	Negatif
4	300	200	250	Negatif
5	500	200	350	Negatif
...	...	...	...	...
176	500	600	550	Positif
177	300	605	452	Positif
178	450	615	532	Positif
...	...	...	...	...
206	380	400	380	Negatif
207	350	450	350	Negatif
208	100	500	100	Negatif
209	300	500	300	Negatif

Then the dataset is divided into two parts as learning data (training data) and testing data (testing data). 1) Model testing is carried out at the dataset testing stage using the Perceptron algorithm with the first configuration, which is the default setting of Perceptron. The default configuration refers to the default parameter values determined based on the number of features. The research results can be seen in Table 1

Table 1: Default perceptron model test

Rasio	Accuracy %	MSE	Loss	AUC	Information
70.30	99.63	0.0037	0.1335	99.98	Fitting 3 folds for each of 756 candidates, totalling 2268 fits
80.20	99.44	0.0056	0.2002	99.93	Fitting 3 folds for each of 756 candidates, totalling 2268 fits
90.10	99.44	0.0056	0.2002	100	Fitting 3 folds for each of 756 candidates, totalling 2268 fits

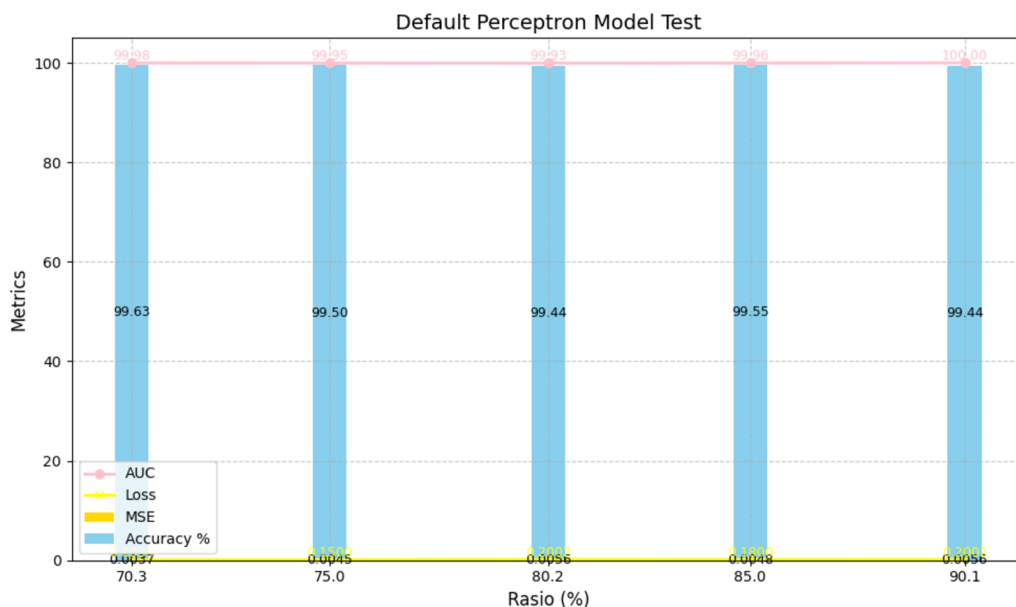


Figure1 : Default perceptron model test

2. Configuration HyperTuning

Testing on the dataset was performed using *hyperparameter tuning* method, with randomsearchSv and GridSearch techniques, This process involves setting parameter values to find the optimal configuration that results in the best performance of the model used. The research results can be seen in Table 2 dan Table 3

Table 2 : Optimisation with GridSearchCV

Rasio	Accuracy %	MSE	Loss	AUC	Best Score	Best Parameters	Information
70.30	100	0.0000	0.3825	100	99.76	{'alpha': 0.001, 'eta0': 0.001, 'max_iter': 1000, 'penalty': 'l1', 'tol': 0.001}	Fitting 3 folds for each of 100 candidates, totalling 300 fits
80.20	99.44	0.0056	0.3933	99.25	99.79	{'alpha': 0.0001, 'eta0': 0.001, 'max_iter': 1000, 'penalty': 'l1', 'tol': 0.0001}	Fitting 3 folds for each of 100 candidates, totalling 300 fits
90.10	100	0.0000	0.0365	100	99.69	{'alpha': 0.0001, 'eta0': 0.1, 'max_iter': 1000, 'penalty': 'l1', 'tol': 0.0001}	Fitting 3 folds for each of 100 candidates, totalling 300 fits

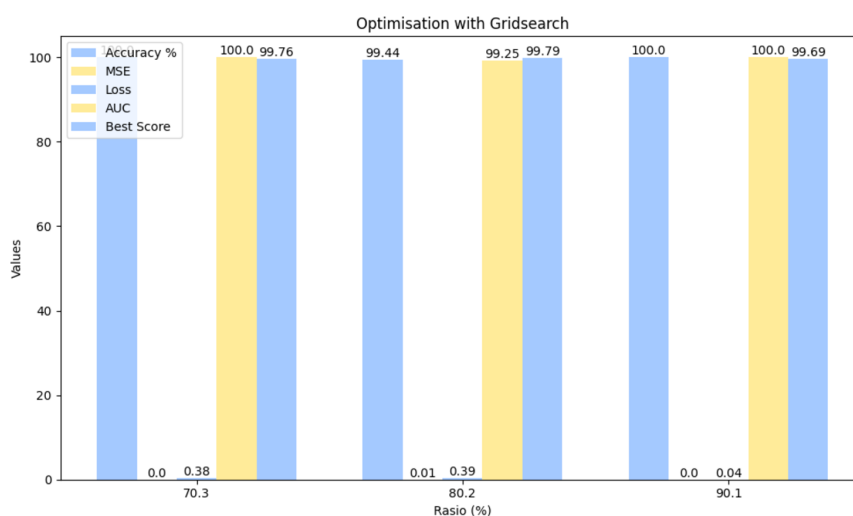


Figure2: Optimisation with GridSearchCV

Optimisation with GridSearchCV showed excellent performance with accuracy reaching 100% at 70:30% and 90:10% ratios, and the lowest MSE of 0.0000. Parameter adjustments such as alpha and eta0 contributed to the optimal results in each training data ratio.

Table 3 : Optimisation with RandomizedSearchCV

Rasio	Accuracy %	MSE	Loss	AUC	Best Score	Best Parameters
70.30	99.63	00.0037	0.1540	1.000	99.76	{'tol': 0.0001, 'penalty': 'l1', 'max_iter': 2000, 'eta0': 0.01, 'alpha': 0.001}
80.20	99.44	0.0056	0.1888	99.25	99.79	{'tol': 0.001, 'penalty': 'l1', 'max_iter': 1000, 'eta0': 0.01, 'alpha': 0.0001}
90.10	98.89	0.0111	0.4005	99.82	99.69	{'tol': 0.0001, 'penalty': 'elasticnet', 'max_iter': 1000, 'eta0': 0.1, 'alpha': 0.0001}

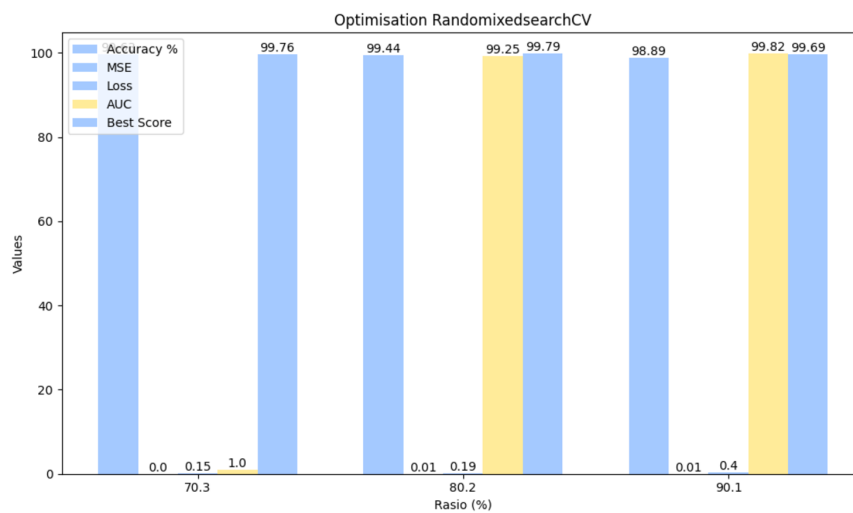


Figure3: Optimisation with RandomizedSearchCV

Optimisation with RandomizedSearchCV showed high accuracy, reaching 99.63% at a 70:30% ratio and an AUC of 1,000. Parameter adjustments such as toll, alpha, and eta0 gave optimal results at each training data ratio.

Table 4. Comparison of Accuracy, MSE, Loss, AUC Results of Default Perceptron with GridSearchCV Optimiser, RandomisedSearchCV

Rasio	Model	Accuracy %	MSE	Loss	AUC	Best Score	Best Parameters	Information
70.30	Perceptron Default	99.63	0.0037	0.1335	99.98	-	-	Fitting 3 folds for each of 756 candidates, totalling 2268 fits
	GridSearchCV	100	0.0000	0.3825	100	99.76	{'alpha': 0.001, 'eta0': 0.001,	Fitting 3 folds for each of 100

Rasio	Model	Accuracy %	MSE	Loss	AUC	Best Score	Best Parameters	Information
80.20	RandomizedSearchCV	99.63	00.0037	0.1540	1.000	99.76	'max_iter': 1000, 'penalty': 'l1', 'tol': 0.001}	candidates, totalling 300 fits
							{ 'tol': 0.0001, 'penalty': 'l1', 'max_iter': 2000, 'eta0': 0.01, 'alpha': 0.001}	
	Perceptron Default	99.44	0.0056	0.2002	99.93	-	-	Fitting 3 folds for each of 756 candidates, totalling 2268 fits
	GridSearchCV	99.44	0.0056	0.3933	99.25	99.79	{ 'alpha': 0.0001, 'eta0': 0.001, 'max_iter': 1000, 'penalty': 'l1', 'tol': 0.0001}	Fitting 3 folds for each of 100 candidates, totalling 300 fits
							{ 'tol': 0.001, 'penalty': 'l1', 'max_iter': 1000, 'eta0': 0.01, 'alpha': 0.0001}	
	RandomizedSearchCV	99.44	0.0056	0.1888	99.25	99.79	{ 'tol': 0.001, 'penalty': 'l1', 'max_iter': 1000, 'eta0': 0.01, 'alpha': 0.0001}	
90.10	Perceptron Default	99.44	0.0056	0.2002	100	-	-	Fitting 3 folds for each of 756 candidates, totalling 2268 fits
	GridSearchCV	100	0.0000	0.0365	100	99.69	{ 'alpha': 0.0001, 'eta0': 0.1, 'max_iter': 1000, 'penalty': 'l1', 'tol': 0.0001}	Fitting 3 folds for each of 100 candidates, totalling 300 fits
							{ 'tol': 0.0001, 'penalty': 'elasticnet', 'max_iter': 1000, 'eta0': 0.1, 'alpha': 0.0001}	
	RandomizedSearchCV	98.89	0.0111	0.4005	99.82	99.69	{ 'tol': 0.0001, 'penalty': 'elasticnet', 'max_iter': 1000, 'eta0': 0.1, 'alpha': 0.0001}	

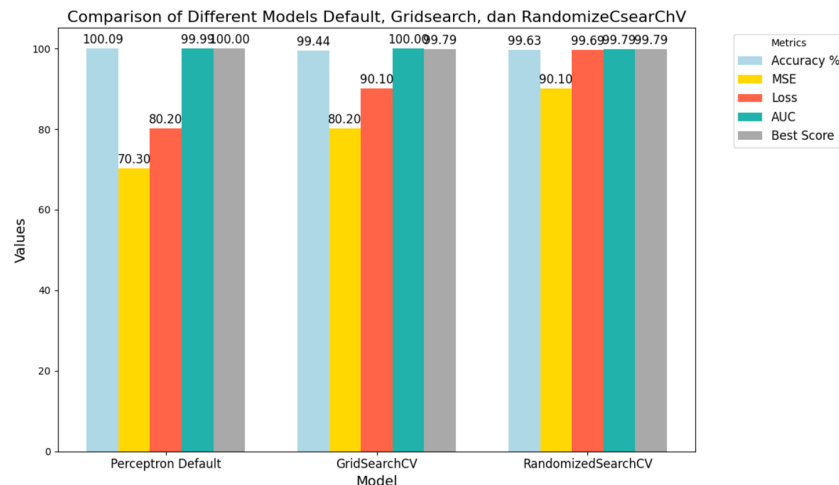


Figure4: Comparison of Accuracy, MSE, Loss, AUC Results of Default Perceptron with GridSearchCV Optimiser, RandomisedSearchCV

Comparison of Default Perceptron Model, GridSearchCV, and RandomisedSearchCV

Accuracy: At all training data ratios (70.30%, 80.20%, 90.10%), the Default Perceptron showed very high accuracy (99.44% to 99.63%). Meanwhile, GridSearchCV produced 100% accuracy across all training data ratios, which shows that a more thorough parameter search can significantly improve accuracy. On the other hand, RandomisedSearchCV has a slightly lower accuracy than GridSearchCV, but is still within the range of 98.89% to 99.63%, which still shows excellent performance.

MSE (Mean Squared Error): GridSearchCV has the lowest MSE value (0.0000) at every training data ratio, which indicates a very small prediction error. Meanwhile, Perceptron Default and RandomisedSearchCV had higher, but still relatively low MSE values, ranging from 0.0037 to 0.0111.

Loss: GridSearchCV produced higher Loss values compared to Default Perceptron and RandomisedSearchCV, especially at training data ratios of 70.30% and 90.10%. This shows that although GridSearchCV achieves 100% accuracy, it tends to sacrifice a bit of the model's generalisation ability (higher Loss values). Meanwhile, Perceptron Default and RandomisedSearchCV showed lower Loss values, with RandomisedSearchCV showing better results at training data ratios of 70.30% and 80.20%.

AUC (Area Under the Curve): GridSearchCV achieved 100% AUC at every training data ratio, demonstrating its ability to distinguish classes perfectly. Perceptron Default showed a very high AUC (99.93% to 99.98%), with a slight decrease at higher training data ratios (90.10%). Meanwhile, RandomisedSearchCV has a slightly lower, but still very high AUC, ranging from 99.25% to 99.82%.

Best Score and Best Parameters: GridSearchCV shows a consistently high Best Score, close to 100%, with more precise parameters for each training data ratio. Meanwhile, RandomisedSearchCV also gave a high Best Score, although slightly lower than GridSearchCV. The Default Perceptron had no best parameters found, as it is a model without parameter search.

A comparison of the results showed that although this study [6] used a Perceptron Artificial Neural Network with an accuracy of 96%, another study using GridSearchCV produced higher accuracy (99.44%-99.63%) with more in-depth parameter optimisation, but with a slight increase in Loss

4. Conclusion

GridSearchCV provided the most optimal results in terms of accuracy, MSE, and AUC, albeit at the slight sacrifice of higher loss values. This shows that careful parameter search can significantly improve performance, but can sometimes lead to overfitting, as seen from the large difference between accuracy and loss. Meanwhile, the Default Perceptron performs very well with high accuracy and low loss, but without parameter search, so it may not be as optimal as GridSearchCV. On the other hand, RandomisedSearchCV gives almost equivalent performance to GridSearchCV on some training data ratios, with lower Loss, but slightly lower in terms of AUC and accuracy compared to GridSearchCV.

References

- [1] F. Ayu, "Implementasi Jaringan Saraf Tiruan Untuk Menentukan Kelayakan Proposal Tugas Akhir," *It J. Res. Dev.*, Vol. 3, No. 2, Pp. 44–53, 2019, [https://doi.org/10.25299/Itjrd.2019.Vol3\(2\).2271](https://doi.org/10.25299/Itjrd.2019.Vol3(2).2271).
- [2] S. Arlis, D. S. Ekajaya, And M. Yanto, "Pola Penentuan Status Peminjaman Dengan Algoritma Perceptron", *Sebatik*, Vol. 23, No. 2, Pp. 619–623, Dec. 2019
- [3] Rini Sovia And Musli Yanto, "Jaringan Syaraf Tiruan Analisa Pengaruh Gizi Buruk Terhadap Perkembangan Balita Dengan Algoritma Perceptron," *J. Ilm. Media Sisfo*, Vol. 12, No. 1, Pp. 1003–1011, 2018.
- [4] A. Prasetya Wibawa, W. Lestar, A. Bella Putra Utama, I. Tri Saputra, And Z. Nabila Izdihar, "Multilayer Perceptron Untuk Prediksi Sessions Pada Sebuah Website Journal Elektronik," *Indones. J. Data Sci.*, Vol. 1, No. 3, Pp. 57–67, 2020, <https://doi.org/10.33096/Ijodas.V1i3.15>.
- [5] L. L. Van Fc, "Implementasi Jaringan Syaraf Tiruan Untuk Menentukan Kepribadian Mahasiswa Menggunakan Algoritma Perceptron," Vol. X, No. X, Pp. 144–158, 2020. <https://journal.unilak.ac.id/index.php/dz/article/view/4019>
- [6] S. N. Kapita, S. Mahdi, And F. Tempola, "Penilaian Pengetahuan Siswa Dengan Jaringan Syaraf Tiruan Algoritma Perceptron," No. 09, Pp. 372–381, 2020. <https://ejournal.unkhair.ac.id/index.php/Techno/article/view/1712>
- [7] E. Fitri And S. N. Nugraha, "Optimasi Kinerja Linear Regression, Random Forest Regression Dan Multilayer Perceptron Pada Prediksi Hasil Panen," *Inti Nusa Mandiri*, Vol. 18, No. 2, Pp. 210–217, 2024, <https://doi.org/10.33480/Inti.V18i2.5269>.
- [8] A. Dwi Sripamuji, I. Ramadhanti, R. Riski Amalia, J. Saputra, And B. Prihatnowo, "Penerapan Algoritma Support Vector Machine Dan Multi-Layer Perceptron Pada Klasifikasi Topik Berita," *Janapati*, Vol. 11, No. 2, Pp. 84–91, 2022, [Online]. Available: <https://doi.org/10.23887/Janapati.V11i2.44151>
- [9] Y. M. Nimas Ratna Sari1*, "Penerapan Multilayer Perceptron Untuk Identifikasi Kanker Payudara," *Cakrawala Ilm.*, Vol. 2, No. 8, Pp. 3261–3268, 2023, <https://doi.org/10.31862/9785426311961>.
- [10] W. I. Sabilla, C. B. Vista, And D. S. Hormansyah, "Implementasi Multilayer Perceptron Untuk Memprediksi Harapan Hidup Pada Pasien Penyakit Kardiovaskular," *J. Sains Komput. Inform.*, Vol. 6, No. 1, Pp. 57–68, 2022. <http://dx.doi.org/10.30645/j-sakti.v6i1.425>
- [11] N. F. Khusna, A. Rahmah, And R. K. Nur, "Implementasi Random Forest Dalam Klasifikasi Kasus Stunting Pada Balita Dengan Hyperparameter Tuning Grid Search," Vol. 2024, No. Senada, Pp. 791–801, 2024. <https://doi.org/10.33005/senada.v4i1.334>
- [12] F. O. Awalullaili, D. Ispriyanti, And T. Widiharih, "Klasifikasi Penyakit Hipertensi Menggunakan Metode Svm Grid Search Dan Svm Genetic Algorithm (Ga)," *J. Gaussian*, Vol. 11, No. 4, Pp. 488–498, 2023,

- <https://doi.org/10.14710/J.Gauss.11.4.488-498>.
- [13] A. Nadroh, D. N. Triwibowo, And R. B. B. Sumantri, “Klasifikasi Status Gizi Balita Menggunakan Algoritma Support Vector Machine Dengan Optimasi Grid Search Cross-Validation,” Vol. 8, No. 2, Pp. 250–257, 2024.
<https://doi.org/10.46880/jmika.Vol8No2.pp250-257>
- [14] G. Abdurrahman, H. Oktavianto, And M. Sintawati, “Optimasi Algoritma Xgboost Classifier Menggunakan Hyperparameter Gridsearch Dan Random Search Pada Klasifikasi Penyakit Diabetes,” *Informal Informatics J.*, Vol. 7, No. 3, P. 193, 2022,
<https://doi.org/10.19184/Isj.V7i3.35441>.
- [15] H. Wijaya, D. P. Hostiadi, And E. Triandini, “Meningkatkan Prediksi Penjualan Retail Xyz Dengan Teknik Optimasi Random Search Pada Model Xgboost,” *Spinter (Prosiding Semin. Has. Penelit. Inform. Dan Komputer)*, Vol. 1, No. 2, Pp. 829–833, 2024.
- [16] M. D. Y. Fordana And N. Rochmawati, “Optimisasi Hyperparameter Cnn Menggunakan Random Search Untuk Deteksi Covid-19 Dari Citra X-Ray Dada,” *J. Informatics Comput. Sci.*, Vol. 4, No. 01, Pp. 10–18, 2022,
<https://doi.org/10.26740/Jinacs.V4n01.P10-18>.
-