Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)
Vol. 8, No. 1, March 2025, pp. 210 – 218
p-ISSN: 2614-3372 | e-ISSN: 2614-6150

Comparison of Recurrent Neural Network and Naive Bayes Algorithms in Identifying Stunting in Toddlers

[1]Forentina Kerti Pratiwi Sujayanti, [2*]Yisti Vita Via, [3]Muhammad Muharrom Al Haromainy
[1,2,3]Department of Informatics, Faculty of Computer Science,
Universitas Pembangunan Nasional Veteran Jawa Timur Surabaya, Indonesia
Email: [1]200810102333@student.upnjatim.ac.id,
[2]yistivia.if@upnjatim.ac.id, [3]muhammad.muharrom.if@upnjatim.ac.id

Article Info
Article history:
Received Nov 10th, 2024
Revised Jan 23th, 2025
Accepted Feb 2nd, 2025

Keyword:
Gausian Naive Bayes
Long Short-Term Memory
Recurrent Neural Network
Stunting
Toddlers

ABSTRACT
Stunting in toddlers is a health issue that affects their quality of life. This study aims to predict stunting status using three classification methods: Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gaussian Naive Bayes. The dataset from Kaggle was split into 70% for training and 30% for testing to ensure optimal model evaluation. The RNN model was built with three hidden layers of 64 units each, while the LSTM model had four hidden layers with the same number of units. Both models utilized hidden states to capture temporal patterns and employed the tanh activation function to detect complex data patterns. The ADAM optimizer with a learning rate of 0.001 was applied to accelerate convergence. In contrast, the Gaussian Naive Bayes model used a simple probabilistic approach without temporal patterns, making it suitable for simpler datasets. Evaluation using accuracy and RMSE showed that LSTM achieved the highest accuracy (91%), followed by RNN (90%), though both exhibited signs of overfitting. Gaussian Naive Bayes attained 72% accuracy with stable performance. While LSTM and RNN effectively capture complex temporal patterns, they are prone to overfitting, whereas Gaussian Naive Bayes is suitable for initial implementation or simpler datasets, supporting early intervention for stunted toddlers.

Corresponding Author:
Yisti Vita Via,
Informatics , Faculty of Computer Science,
Universitas Pembangunan Nasional Veteran Jawa Timur Surabaya, Indonesia
Email: yistivia.if@upnjatim.ac.id
DOI: http://dx.doi.org/10.24014/ijaidm.v8i1.33946

1. INTRODUCTION

Stunting is a health issue that affects physical growth and brain development in toddlers due to chronic malnutrition. In addition to its impact on physical health, stunting can also reduce learning abilities and future productivity [1]. Therefore, early detection of stunting in children is crucial to support preventive measures. With its high prevalence in various countries, stunting has become a global health issue that requires a comprehensive approach for its management [2].

Advancements in information technology have opened new opportunities in the healthcare sector, including supporting early detection of stunting. One of its applications is the use of artificial intelligence (AI) in health classification to identify risks, diagnose diseases, and plan more effective interventions [3]. This technology can optimize the allocation of healthcare resources, prevent waste, and ensure better healthcare services [4]. In the context of stunting detection, this technology allows for faster and more accurate data analysis, supporting rapid responses to public health trends [5].

Stunting detection, characterized by a height below the standard for a child's age, requires fast and accurate analytical methods to provide valid results. Data mining, as a primary technique, is used to extract patterns from large and complex datasets, providing a foundation for better decision-making [6]. The application of this technique can identify key factors that affect stunting, support more efficient healthcare management, and improve the effectiveness of interventions [7].

This study compares two different algorithms in the classification process: Recurrent Neural Network (RNN) and Naive Bayes. RNN, part of deep learning, is known to excel at analyzing temporal patterns and complex relationships in data [8]. However, this algorithm has the disadvantage of requiring high computational resources. In contrast, Naive Bayes offers a simpler yet effective approach, especially in cases with a large number of independent features [9].

Previous research was conducted by Desi Efriyani and Febriyanti Panjaitan in 2021. In this study, the authors discussed the increasing number of internet users often exploited for various types of crimes. The results indicated that the RNN method achieved an accuracy rate of 86% and an F1 score of 85% in classifying types of Malware. The second study, conducted by Eko Arip Winanto, Kurnia Budi, Sharipuddin, Ibnu Sani Wijaya, and Dodi Sandra (2022), applied the RNN method to improve the effectiveness of intrusion detection systems in IoT networks. The study found that the RNN model successfully enhanced the performance of the intrusion detection system in IoT networks, achieving an accuracy rate of 87%. Further, a study by Rachmad et al. (2022) compared K-Nearest Neighbor and Gaussian Naive Bayes methods for stroke disease classification. The study concluded that stroke detection with GNB performed better, with an accuracy of 74.45%, precision of 74.01%, and recall of 75.71%.

In relation to these studies, this research aims to compare the RNN method with the developed LSTM approach and Gaussian Naive Bayes in detecting stunting in toddlers [12]. Unlike previous studies, this research specifically focuses on stunting data classification, with an in-depth analysis of relevant risk factors. By utilizing a larger and more complex dataset, this study is expected to provide an efficient solution to significantly reduce the prevalence of stunting [13].

2. RESEARCH METHOD

In this study, a classification model using RNN, LSTM, and Gaussian Naive Bayes for Stunting Toddlers is developed, and the system design is structured as Figure 1.

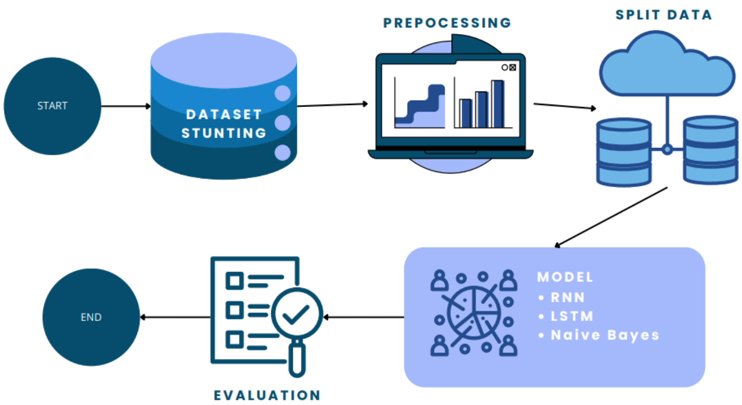

Figure 1. Flowchart System

2.1. Dataset

The data used in this study is the "Stunting Toddler" dataset from the Kaggle platform. This dataset consists of 6,500 entries with 8 variable columns, as described in Table 1.

Table 1. Dataset

Variable	Data Type
Gander	String
Age	Numeric
Birth Weight	Numeric
Birth Length	Numeric
Body Weight	Numeric
Body Length	Numeric
Breastfeeding	Kategoric
Stunting	Kategoric

2.2. Prepocessing

Data preprocessing in this study is the initial stage of processing the dataset. This ensures that the data used is more easily applied in the research process [14]. Preprocessing also involves checking for missing or unrecorded data, making the research process smoother. Several steps are undertaken during the preprocessing stage.

1. Label Encorder
   Where three variables are converted into Boolean data types, can view table 2.

Table 2. Label encorder

Gander Variabel	Breastfeeding Variabel	Stunting Variabel	Data Type of Bolean
Female	Yes	Yes	1
Male	No	No	0

2. Calculation of statistical values for the stunting variable in relation to other variables, resulting in average values, can view table 3.

Table 3. Class average analysis

Variable	Mean Stunting Toodler '1'	Mean Stunting Toodler '0'
Age	24.4	26
Birth Weight	2.86	3.1
Birth Length	49	49.3
Body Weight	2.9	10.8
Body Length	75.5	53

3. Performing a Missing Value filter to ensure that the dataset variables have no missing (empty, unrecorded) values and all data is properly recorded.

2.3. Splitting Data

The next step is to split the data into two parts: training data and testing data [15]. The training data is used to train the classification algorithm, while the testing data is used to evaluate the performance of the trained algorithm, especially when it encounters new data.

2.4. Model Classification

The first model proposed in this study is the Recurrent Neural Network (RNN). The process in this model will involve three units: the input unit, hidden unit, and output unit. RNN is a special type of artificial neural network designed to work with sequential data, such as text or time sequences [16]. The RNN formula involves using internal state or "memory" to cope with understanding the context of sequential data and also matching the context of the data to make accurate predictions. RNN is a special type of artificial neural network designed to work with sequential data, such as text or time sequences [17]. The RNN formula involves using internal state or "memory" to cope with understanding the context of sequential data and also matching the context of the data to make accurate predictions (Figure 2).

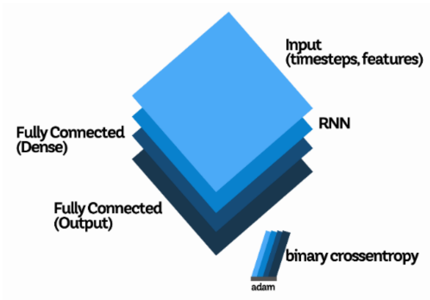

Figure 2. Algorithm RNN

The model is built sequentially, where the input has the format ('timesteps', 'features'). The RNN cell processes the input sequentially, calculating the hidden state at each timestep and storing temporal information . Each hidden state is passed to the next timestep, forming a continuous memory flow. After the final hidden state is calculated, the information is passed to a Dense layer, which transforms it into a lower representation and generates probabilities for binary classification using a sigmoid activation [18]. The formula used involves the hidden layer and output layer, can view on equation 1 and 2.

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

$$y_t = \sigma\left(W_{hy}h_t + b_y\right) \tag{2}$$

Finally, optimization is performed using the Adam optimizer to update the network weights based on the gradients calculated from the predictions and the target. The selection of the Adam optimizer is based on its advantages in faster convergence and adaptive adjustment of the learning rate, making Adam optimizer suitable for networks with sequential data.

The second model in this study uses Long Short Term Memory (LSTM). The LSTM model is an advanced version of the RNN model (Figure 3).

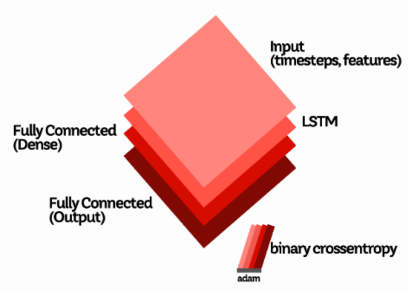

Figure 3. Algorithm LSTM

What differentiates the LSTM model is its more complex architecture with gates, consisting of the forget gate, input gate, and output gate. Each gate functions to control the flow of information within the network, allowing it to capture long-term dependencies in sequential data. The weights for input (W), hidden state (U), and bias (b) are also included for each gate. With these layers, LSTM can address the vanishing gradient problem, allowing the model to learn patterns involving long-term dependencies without losing crucial information. LSTM is highly effective for tasks involving sequential data, such as time series analysis, speech recognition, and text classification, including the detection of stunting patterns in toddlers.

$$i_t = \sigma\,(\,W_i * x_t + U_i * h_{t-1} + b_i) \tag{3}$$

$$f_t = \sigma\,(W_f . [h_{t-1}, x_t] + b_f) \tag{4}$$

$$\tilde{c}_t = \tanh\,(W_c * x_t + U_c * h_{t-1} + b_c) \tag{5}$$

$$O_t = \sigma\,(\,W_o * x_t + U_o * h_{t-1} + b_o) \tag{6}$$

The final model, unlike the previous two models, is a machine learning technique called Naive Bayes. Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. The Naive Bayes formula uses probabilities to predict the class of a data point, assuming independence between each pair of features [19]. The main advantage of Naive Bayes lies in its simplicity and efficiency in handling large datasets, and it is also very fast in both training and prediction because it only requires simple probability estimates. However, its main weakness is the assumption of independence between features, which can lead to a decrease in accuracy when the features in the dataset are significantly dependent on each other. The Naïve bayess algorithm can view Figure 4.

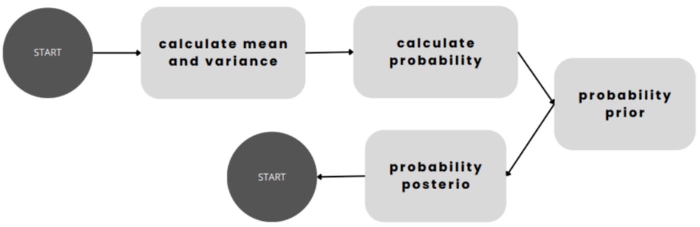

Figure 4. Algorithm Naive Bayes

Naive Bayes assumes that the features follow a normal (Gaussian) distribution. Each feature in the Gaussian Naive Bayes model is calculated based on the mean (μC) and variance for each feature to determine the probability (P) in the prior probability (P(C)) and posterior probability ($\sigma^2_{c,i}$) [14]. The main advantage of Gaussian Naive Bayes is its simplicity in implementation and computational speed, as it only requires the estimation of parameters (μC and ($\sigma^2_{c,i}$)) for each feature. However, this assumption can affect accuracy when applied to datasets where features are highly correlated or do not follow a normal distribution.

$$\mu C, i = \frac{1}{N_c} \sum_{j=1}^{N_C} X_{j,i} \tag{7}$$

$$P(X_i|C) = \frac{1}{\sqrt{2\pi\sigma^2 C}} \exp\left(\frac{(X_i - \mu c)^2}{2\sigma^2 c}\right) \tag{8}$$

$$P(C) = \frac{\text{number of class C samples}}{\text{total number of samples}} \tag{9}$$

$$\sigma^2_{c,i} = \frac{1}{N_C} \sum_{j=1}^{N_C} (X_{j,i} - \mu_{C,i})^2 \tag{10}$$

2.5. Evaluation

Model evaluation for classification can be performed using a confusion matrix, which provides a comprehensive overview of the model's performance in classifying data. The confusion matrix consists of four main components: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP represents the number of correct predictions for the positive class, TN represents the number of correct predictions for the negative class, FP represents the number of incorrect predictions for the positive class, and FN represents the number of incorrect predictions for the negative class. The confusion matrix allows for the identification of specific errors made by the model. Additionally, evaluation can also be performed using Root Mean Square Error (RMSE) [20]. RMSE calculates the square root of the mean of squared errors, where the error is the difference between the values predicted by the model and the actual observed values (equation 11).

$$RMSE = \sqrt{\frac{I}{n} \sum_{i=1}^{n} (y_i - \hat{\ } y_i)^2} \tag{11}$$

3. RESULTS AND ANALYSIS

The test results comparing the performance of Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) based on deep learning, and Naive Bayes, a simple probabilistic method, in detecting stunting in toddlers. The testing was conducted using a toddler-related dataset, which was split into 70% training data and 30% testing data.

In building the RNN and LSTM models, several hyperparameters were used to optimize the model’s performance. First, the model uses 16 'tanh' activations, referring to the number of neurons in the hidden layer with the tanh activation function. This function produces output values between -1 and 1, which helps the model capture non-linear patterns in the data. With 16 units, the model has enough capacity to efficiently learn from the data without the risk of overfitting. In the output layer, 1 'sigmoid' activation is used. The sigmoid function provides output values between 0 and 1, representing the probability of each class. Additionally, the ADAM optimizer is used to efficiently update the model's weights. ADAM combines the benefits of the Momentum and RMSProp optimizers by adjusting parameter updates based on the average and variance of the gradients, making it highly effective in achieving stable convergence. A learning rate of 0.001 was selected, allowing for smaller and more refined updates to the parameters. With this combination of hyperparameters, both RNN and LSTM models are capable of learning data effectively, particularly in problems that require understanding complex patterns and rely on temporal or sequential data contexts. Meanwhile, in the Naïve Bayes model, calculations involve the class mean and variance, followed by probability calculations.

Table 4. Result Skenario

Dataset	Epoch	Optimizer	Accuracy (%)	RMSE
Recurrent Neural Network (RNN)	20	Adaptive Moment Estimation	86%	0.44
	30		88%	0.44
	50		89%	0.44
Long Short-Term Memory (LSTM)	20		87%	0.45
	30		88%	0.45

Dataset	Epoch	Optimizer	Accuracy (%)	RMSE
	50		90%	0.45
Naive Bayes	-	-	72%	0.53

Table 4 presents the results of the tested scenarios. After conducting the tests, an analysis of the developed model is required. Therefore, it continues with checking for overfitting by displaying the results from the training data, testing data, and validation data used. The models of algorithm can view Figure 5-7.

Figure 5. Model of RNN

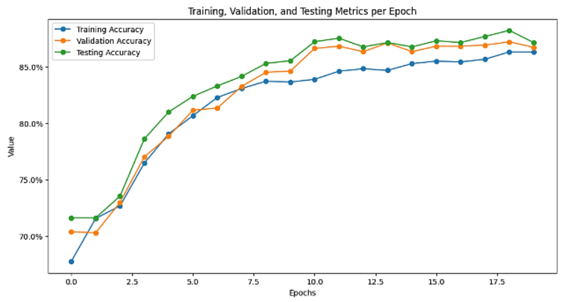

Figure 6. Model of LSTM

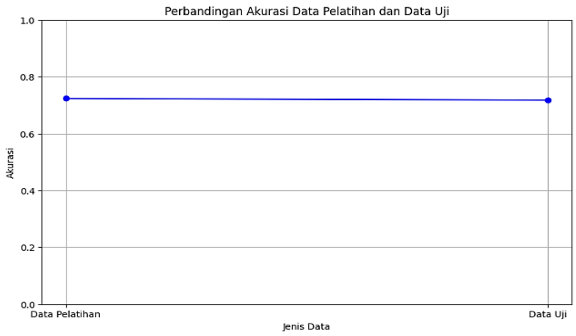

Figure 7. Model of Naive Bayes

The results of the overfitting analysis indicate that each model has different characteristics and capabilities in handling data. The RNN model shows an increase in accuracy on the test data as the epochs progress but starts to experience overfitting after the 15th epoch, indicating that the model struggles to generalize on new data. The LSTM model also shows signs of overfitting, where the accuracy on the test data increases significantly in the beginning but plateaus around 85% at the 10th epoch. This suggests that the LSTM starts memorizing the training data. In contrast, the Naive Bayes model does not show signs of overfitting, with consistent accuracy performance between the training and test data. This indicates that Naive Bayes generalizes well to new data, making it a stable model to use with this dataset. The RMSE evaluation of algorithm can view Figure 8-10.

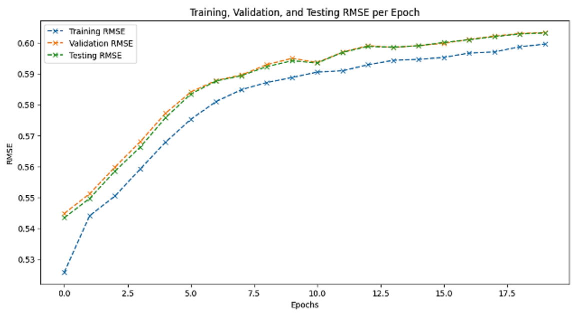

Figure 8. RMSE RNN

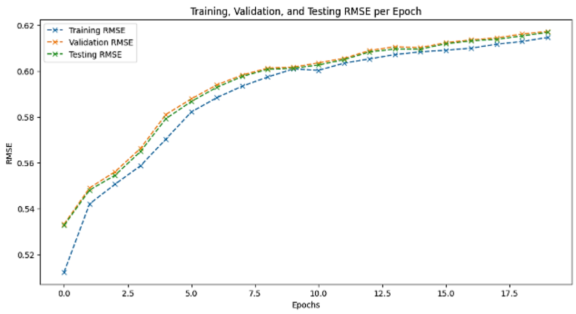

Figure 9. RMSE LSTM

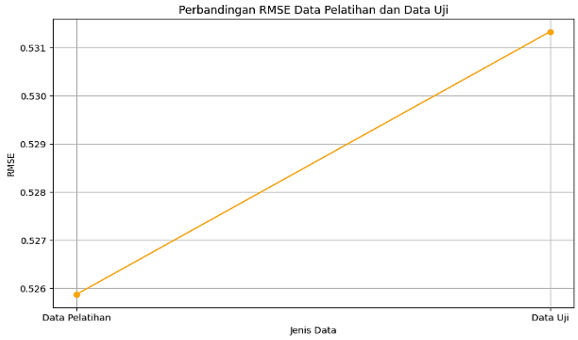

Figure 10. RMSE Naive Bayes

Then, based on the RMSE analysis of the three models, it can be concluded that the generalization ability of each model differs significantly. The RNN model shows a decrease in RMSE at the beginning of the training, but after several epochs, the RMSE starts to increase, indicating that the model experiences overfitting on the training data and struggles to handle the test data effectively. The LSTM model also shows a rapid decrease in RMSE at the beginning of the training, but the RMSE tends to increase gradually after several epochs, suggesting that the model begins to lose its generalization ability as the epochs progress. In contrast, the Naive Bayes model demonstrates stable performance, with RMSE values that are almost identical between the training and test data. This indicates that Naive Bayes can handle new data effectively without overfitting, making it a more reliable model for generalization.

4. DISCUSSION

RNN exhibits performance similar to LSTM, with slightly lower accuracy and higher RMSE. While RNN can also handle data with complex patterns, it tends to be less efficient in minimizing prediction errors compared to LSTM, especially on test data. On the other hand, Gaussian Naive Bayes demonstrates much lower performance, with an accuracy of only 72% and a higher RMSE. This reflects that the model is less effective in handling data with complex patterns, such as in the case of predicting stunting in toddlers.

Technically, the probabilistic Naive Bayes method is based on the main assumption that each feature in the dataset is conditionally independent of the target class. This approach ignores any dependencies or relationships between features. In the case of complex datasets, such as stunting data that includes various correlated variables like height, weight, age, and other factors, this assumption becomes invalid. As a result, Naive Bayes often fails to capture non-linear relationships or hidden patterns in the data, ultimately leading to a decrease in model performance.

In contrast, RNNs are designed to process data with temporal or sequential dependencies and can capture complex relationships between features through dynamic information propagation mechanisms within their recurrent network structure. RNNs have the ability to retain information from previous steps through hidden states, enabling the model to recognize complex patterns and feature dependencies in the dataset. This advantage makes RNN more adaptive to handling data. LSTM, as a variant of RNN, addresses these limitations by introducing long-term and short-term memory mechanisms, thereby improving prediction accuracy and overall stability.

Overall, LSTM is the best model for classifying stunting status in toddlers, given its superior ability to handle complex data and provide accurate, stable results. RNN can still be considered as a lighter computational alternative, but Naive Bayes should only be used for simpler data, where basic probabilistic approaches remain relevant and effective.

5. CONCLUSION

Based on the evaluation of model performance, the Long Short-Term Memory (LSTM) model proved to be superior in predicting stunting status in toddlers, achieving the highest accuracy of 91% and a low and stable Root Mean Square Error (RMSE) ranging from 0.53 to 0.54 across training, validation, and testing data. This model demonstrated excellent generalization capability by minimizing prediction errors. The Recurrent Neural Network (RNN) also exhibited good performance with an accuracy of 90% and a slightly higher RMSE of 0.54 to 0.56, still reflecting its ability to handle complex data patterns effectively. In contrast, the Gaussian Naive Bayes model achieved only 72% accuracy with a higher RMSE, indicating that this simpler approach is less effective in handling complex data patterns. Therefore, the LSTM model is the best choice for classifying stunting status in toddlers, followed by RNN, while Gaussian Naive Bayes is more suitable for data with simpler patterns.

However, this study has several limitations, including a limited dataset size, which may reduce the generalizability of the evaluation results to a broader population. Additionally, the study evaluated only three algorithms without comparing them to more advanced methods, such as Transformer-based models, and did not employ automatic hyperparameter optimization. The study also did not assess the importance of each feature in the predictions. For future research, it is recommended to use a larger and more diverse dataset, explore more advanced algorithms, and implement automatic hyperparameter optimization techniques to improve model performance. Furthermore, feature analysis using explainable AI (XAI) techniques can be conducted to identify key factors influencing stunting status in toddlers. Integrating the model with clinical decision support systems is also recommended to support practical applications in the healthcare field.

REFERENCES

[1] H. Herlianty, A. Setyawati, A. Lontaan, T. Limbong, I. A. Tyarini, dan S. Z. Putri, “Determinants Influence the Incidence of Stunting in Toddlers Aged 6-59 Months,” J. Edukasi Ilm. Kesehat., vol. 1, no. 2, hal. 73–79, 2023, doi: 10.61099/junedik.v1i2.18.

[2] M. R. U. Hutabarat, I. Irwanto, dan S. Sulistiawati, “Risk factors of stunting in toddler aged 24-59 month,” J. Kebidanan, vol. 10, no. 2, hal. 119, 2021, doi: 10.26714/jk.10.2.2021.119-128.

[3] P. Pamungkas dan A. P. Hidayathillah, “Pengembangan Model Kasiting Berbasis Artificial Intelligence Terhadap Deteksi Dini Stunting,” vol. 8, hal. 5829–5835, 2024.

[4] Siti Masrichah, “Ancaman Dan Peluang Artificial Intelligence (AI),” Khatulistiwa J. Pendidik. dan Sos. Hum., vol. 3, no. 3, hal. 83–101, 2023, doi: 10.55606/khatulistiwa.v3i3.1860.

[5] R. B. Ramadhan dan M. A. Romli, “Mobile Health Monitoring Application As An Effort To Detect Stunting In Early Childhood Based On Android Aplikasi Mobile Health Monitoring Sebagai Upaya Deteksi Stunting Pada Anak Usia Dini,” hal. 679–689, 2024.

[6] R. M. Sari, A. Rizka, N. A. Putri, dan A. Efriana, “Penerapan Data Mining Untuk Analisis Stunting Pada Balita,” vol. 13, no. November, hal. 1717–1728, 2024.

[7] S. Ramadoan, “Model Intervensi Terpadu dalam Mengatasi Prevalensi Stunting di Kota Bima,” hal. 229–239, 2024.

[8] A. Farhah, A. L. Prasasti, dan M. W. Paryasto, “Implementasi Recurrent Neural Network dalam Memprediksi Kepadatan Restoran Berbasis LSTM,” J. Media Inform. Budidarma, vol. 5, no. 2, hal. 524, 2021, doi: 10.30865/mib.v5i2.2916.

[9] A. Nugroho dan Y. Religia, “Analisis Optimasi Algoritma Klasifikasi Naive Bayes menggunakan Genetic Algorithm dan Bagging,” J. RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 5, no. 3, hal. 504–510, 2021, doi: 10.29207/resti.v5i3.3067.

[10] A. Jatmiko, N. D. Hendrawan, A. S. Affandi, dan D. S. Susanto, “Sistem Pendeteksi Jenis Penyakit Pernafasan Menggunakan Metode Gabungan GRU dan LSTM Melalui Suara Pasien dengan Gangguan Pernafasan,” J. Inform. Polinema, vol. 9, no. 4, hal. 445–450, 2023, doi: 10.33795/jip.v9i4.1407.

[11] D. U. Rachmad, H. Oktavianto, dan M. Rahman, “Perbandingan Metode K-Nearest Neighbor Dan Gaussian Naive Bayes Untuk Klasifikasi Penyakit Stroke,” J. Smart Teknol., vol. 3, no. 4, hal. 405–412, 2022.

[12] V. Utomo, Agusta Praba Ristadi Pinem, dan Bernadus Very Christoko, “Pengenalan Karakter Optis untuk Pencatatan Meter Air dengan Long Short Term Memory Recurrent Neural Network,” J. RESTI (Rekayasa Sist. dan Teknol. Informasi), vol. 5, no. 1, hal. 132–138, 2021, doi: 10.29207/resti.v5i1.2807.

[13] M. Bukhsh, M. S. Ali, M. U. Ashraf, K. Alsubhi, dan W. Chen, “An Interpretation of Long Short-Term Memory Recurrent Neural Network for Approximating Roots of Polynomials,” IEEE Access, vol. 10, hal. 28194–28205, 2022, doi: 10.1109/ACCESS.2022.3157306.

[14] M. I. I. A. N. Kartini, Karakteristik Sumber Daya Manusia Era Digital (Perspektif KepeMimpinan, Motivasi dan Etos Kerja). 2021.

[15] R. Ardianto dan S. K. Wibisono, “Analisis Deep Learning Metode Convolutional Neural Network Dalam Klasifikasi Varietas Gandum,” J. Kolaboratif Sains(JKS), vol. 6, no. 12, hal. 2081–2092, 2023, doi: 10.56338/jks.v6i12.4938.

[16] G. R. Kanagachidambaresan, A. Ruwali, D. Banerjee, dan K. B. Prakash, “Klasifikasi Malware Menggunakan MetodeRecurrent Neural Network,” EAI/Springer Innov. Commun. Comput., vol. 23, no. 3, hal. 53–61, 2021.

[17] E. A. Winanto, K. Kurniabudi, S. Sharipuddin, I. S. Wijaya, dan D. Sandra, “Deteksi Serangan pada Jaringan Kompleks IoT menggunakan Recurrent Neural Network,” JURIKOM (Jurnal Ris. Komputer), vol. 9, no. 6, hal. 1996, 2022, doi: 10.30865/jurikom.v9i6.5298.

[18] D. Yolanda, K. Gunadi, dan E. Setyati, “Pengenalan Alfabet Bahasa Isyarat Tangan Secara Real-Time dengan Menggunakan Metode Convolutional Neural Network dan Recurrent Neural Network,” J. Infra, vol. 8, no. 1, hal. 203–208, 2020, [Daring]. Tersedia pada: https://publication.petra.ac.id/index.php/teknik-informatika/article/view/9791

[19] Y. Naufal, R. Putro, A. Afriansyah, dan R. Bagaskara, “Penggunaan Algoritma Gaussian Naïve Bayes & Decision Tree Untuk Klasifikasi Tingkat Kemenangan Pada Game Mobile Legends,” JUKI J. Komput. dan Inform., vol. 6, no. 1, hal. 10–26, 2024.

[20] R. Setiawan dan A. Triayudi, “Klasifikasi Status Gizi Balita Menggunakan Naïve Bayes dan K-Nearest Neighbor Berbasis Web,” J. Media Inform. Budidarma, vol. 6, no. 2, hal. 777, 2022, doi: 10.30865/mib.v6i2.3566.

BIBLIOGRAPHY OF AUTHORS

Forentina Kerti Pratiwi Sujayanti is an active student in the informatics Study Program, Faculty of Science and Technology, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya.

Yisti Vita Via is a lecturer at the Department of Informatics, Faculty of Computer Science, Universitas Pembangunan Nasional Veteran Jawa Timur Surabaya, Indonesia. Her research interest includes the development of intelligent systems by applying nature-inspired algorithms and optimization algorithms.

M. Muharrom Al Haromainy is a lecturer at the Department of Informatics, Faculty of Computer Science, Universitas Pembangunan Nasional Veteran Jawa Timur Surabaya, Indonesia. He research interest includes data mining, fore casting. And software engineering.