Feature Extraction and Ensemble Learning for the Detection of Dengue Fever

Gen Kasahara [0009-0001-9057-2760] and Tad Gonsalves [0000-0001-9424-3078]

Department of Information and Communication Sciences
Sophia University, Tokyo, Japan
r-nisimura-8f9@eagle.sophia.ac.jp
t-gonsal@sophia.ac.jp

Abstract

In this study, we propose a new machine learning model to identify dengue fever patients. Specifically, we convert actual medical records into text data using Optical Character Recognition (OCR) technology and extract features from this data. Subsequently, we generate high-dimensional features using unsupervised learning and construct a training dataset. Using this training data, we adopted seven types of Deep Neural Networks (DNN), FT-Transformer, and ResNet as base models. Additionally, we performed ensemble learning (stacking) by using Support Vector Machine, Random Forest, DNN, and FT-Transformer as meta-models, and compared and verified each stacking model. To evaluate the performance of the models, we used precision, recall, F1 score, and Matthews correlation coefficient (MCC) as the main indicators, and conducted hyperparameter optimization using Optuna to evaluate and verify the models from multiple perspectives. Comparing the performance of the 9 base models, DNN2 shows the highest accuracy, while FT-Transformer shows the highest recall.

Keywords: OCR, Unsupervised Learning, SVM, Random Forest, Deep Nural Network, ResNet, FT-Transformer, Ensemble Learning, Stacking

1 Introduction

Dengue is a highly endemic infectious disease mostly found in tropical countries. It is caused by any of the 4 serotypes of dengue virus and is transmitted within humans through female Aedes mosquitoes (Khetarpal & Khanna, 2016). The global incidence of dengue has markedly increased over the past two decades, posing a substantial public health challenge. From 2000 to 2019, the World Health Organization (WHO) documented a ten-fold surge in reported cases worldwide (WHO, 2023). Rapid urbanization and climate change have significantly contributed to the spread of the disease. It is mainly concentrated in tropical and subtropical regions, putting nearly a third of the human population, worldwide, at risk of infection (Wu, et al., 2007; Chen and Vasilakis, 2011). Patients infected with the dengue virus show a wide spectrum of clinical

manifestations, ranging from silent infections with no symptoms to a mild flu-like syndrome, dengue fever (DF), or severe dengue disease (SDD), including dengue haemorrhagic fever (DHF) and dengue shock syndrome (DSS), which can be fatal (Chevillon and Failloux, 2003; Malavige, et al., 2024). Therefore, early diagnosis and appropriate treatment interventions are crucial for patient prognosis.

However, diagnosing dengue fever presents challenges in the field. For instance, its symptoms are very similar to other febrile illnesses such as malaria, Zika virus, and chikungunya, making accurate diagnosis difficult. Additionally, standard diagnostic methods like PCR and ELISA tests are expensive and require specialized equipment and expertise, making them less accessible in low-resource areas. This background highlights the need for rapid, affordable, and accurate diagnostic technologies.

Recent advancements in machine learning in the medical field have enabled the construction of systems that learn patterns from large datasets and perform high-precision classification and prediction. Diagnostic support systems utilizing clinical data and patient symptom data have high applicability in resource-limited environments and hold the potential to improve healthcare quality. Some of the latest machine learning algorithms for dengue fever prediction (Ruban et al., 2022, 2024) and custom-built datasets (Ruban & Rai, 2021; Riya, et al., 2024) are found in the literature on dengue fever. Previous studies have shown some success in disease classification models using machine learning (Jayakody, et al., 2015; Perez, et al., 2019; Vannavong, et al.; 2019), but improving model accuracy and verifying applicability in clinical settings remain challenges, especially for diseases with overlapping symptoms like dengue fever.

This study aims to develop a machine learning model to distinguish dengue fever patients from those with other febrile illnesses and verify its effectiveness. Specifically, we will design a model that efficiently extracts features from clinical data and uses multivariate analysis techniques to learn dengue-specific characteristics. Furthermore, we will evaluate the developed model using various performance indicators and discuss the potential for accuracy improvement. By doing so, we aim to expedite and enhance the accuracy of dengue fever diagnosis, contributing to reducing the burden on regional healthcare.

2 Data Sets and Preprocessing

We were provided with a total of 4,386 medical records in PDF format, including those diagnosed as positive and negative for dengue fever. All these data are raw data from individuals who were actually diagnosed in hospitals in India. This section describes the various methods used in data preprocessing and feature extraction.

2.1 OCR

OCR (Optical Character Recognition) is a technology that automatically reads character information from images or scanned documents and converts them into digital data. It is used to digitize text from paper media, making it editable and searchable. For example, it can scan printed materials such as books, newspapers, contracts, or handwritten notes and extract them as text data. The OCR process generally progresses through several steps. First, the document is scanned to obtain image data. The obtained image usually undergoes preprocessing such as noise removal and contrast adjustment to facilitate character recognition. Then, segmentation processing is performed to extract character regions and identify each character or word. The recognized characters are output as text data by matching them with known patterns. In this study, we used the Python library *pytesseract* to convert the medical records in PDF format into text data.

2.2 Feature Extraction from Text Data

From the text data generated by OCR, we labeled each word in Table 1 with a 1 if the word was present and a 0 if it was not, creating a table of data.

Table 1. Features extracted from text

Fever	Muscle pain	Joint pain	rash
abdominal pain	lymphadenopathy	chills	diarrhea
Vomiting	eye pain	nausea	fatigue

2.3 Feature Extraction using Unsupervised Learning

We used k-nearest neighbors (k-NN), Principal Component Analysis (PCA), t-SNE, and Cosine Similarity for dimensionality reduction and high-dimensional feature generation. Specifically, we re-evaluated the importance of features by compressing and reconstructing information from the features (table data) obtained from the text data. This method reduced redundant features and created a compact training dataset suitable for classification models.

k-Nearest Neighbors (k-NN)

k-NN can extract new features based on the distance and neighborhood information between data points. For each data point, we created features based on the distance to the k-nearest data points and the class distribution within that neighborhood. In this study, we added the distance and clusters of the neighborhood of a data point as new features. This allows the model to better understand the relationships and local distributions between data points. We used 10 clusters for training.

Principal Component Analysis (PCA)

PCA not only reduces dimensionality but also generates new axes that explain variance in the data. The principal components obtained by applying PCA are expressed as linear combinations of the original features, removing redundant information and noise. Using these principal components as new features provides a more concise and interpretable data representation. We used 10 principal components for training.

t-distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE generates new features by embedding high-dimensional data into lower dimensions, resulting in 2D or 3D coordinates. These features reflect the local structure of the original high-dimensional data, capturing the relationships and distribution characteristics between clusters. We used 3D for training.

Cosine Similarity

Cosine similarity generates new features by quantifying the relationships between data points. We calculated the cosine similarity with a reference vector and added it as a feature. Using cosine similarity allows us to extract similarities and relationships within the data as new features.

2.4 Training data

The final constructed training dataset includes data from 4,386 patients, integrating features extracted from text via OCR and latent variables generated through unsupervised learning. This resulted in a multidimensional dataset with 30 features. For model training, 80% of this data is used as training data, and 20% is used as test data.

3 Proposed ML model

3.1 3.1 Ensemble Learning

The ensemble learning method used in this study is stacking. Stacking involves training multiple machine learning models (*base models*) and using the accuracy obtained from each as features for the final model (*meta models*). In this study, we used nine deep learning models as base models and integrated them with SVM, Random Forest, DNN, and FT-Transformer as meta models. In the first layer of stacking, the base models made predictions based on the features, and their output accuracies were used as inputs for the second layer. Additionally, unsupervised learning similar to the preprocessing stage was performed on the second layer's inputs to generate features. The hyperparameters used were six clusters for k-NN, six principal components for PCA, and 3D for t-SNE.

Support Vector Machine (SVM)

SVM is a machine learning algorithm specialized for classification problems. Its basic concept is to find the optimal hyperplane that separates data points of different classes. This hyperplane is designed to maximize the margin between the data points, and in cases where linear separation is difficult, kernel methods are used to map the data into a higher-dimensional space, enabling nonlinear separation. SVM performs particularly well with high-dimensional data.

Random Forest

Random Forest is a representative example of ensemble learning algorithms widely used for classification and regression problems. It constructs multiple decision trees and outputs the final result based on majority voting or averaging the predictions. The *randomness* in this method comes from the process of randomly selecting subsets of data and features, which increases model diversity and prevents overfitting. Additionally, it can evaluate feature importance and performs well with high-dimensional data.

Deep Neural Networks (DNN)

DNNs are an advanced form of artificial neural networks characterized by having multiple hidden layers. Each layer extracts and transforms data features hierarchically, allowing the network to learn complex patterns and nonlinear relationships. Training involves backpropagation to minimize the loss function, optimizing the model parameters. This flexibility and high representational power make DNNs suitable for learning complex patterns from high-dimensional data.

ResNet

ResNet is a type of deep learning model developed to facilitate the training of deep networks. It introduces a mechanism called “skip connections” which propagate residuals between layers, significantly mitigating the vanishing gradient problem. This mechanism allows for effective training of very deep networks with over 100 layers, achieving higher accuracy than shallower networks. While ResNet has shown remarkable results in image recognition tasks such as ImageNet, it is also effective for high-dimensional tabular data.

FT-Transformer

FT-Transformer is a model based on the transformer architecture, specifically designed to handle structured tabular data. This model treats each feature of the data as a “token” and uses the transformer's self-attention mechanism to learn the relationships between features. This approach excels at capturing complex feature interactions more efficiently compared to traditional methods like XGBoost or Random Forest.

3.2 Feature Extraction Using Unsupervised Learning on Base Model Outputs

We performed feature extraction using unsupervised learning on the outputs (accuracies) obtained from each base model. This process extracted correlations and common information between the base models, generating high-dimensional features suitable for the meta-model's input. Ultimately, this resulted in a training dataset with 20 features. The entire machine learning pipeline from preprocessing to the execution of the ensemble meta model is shown in Fig. 1.

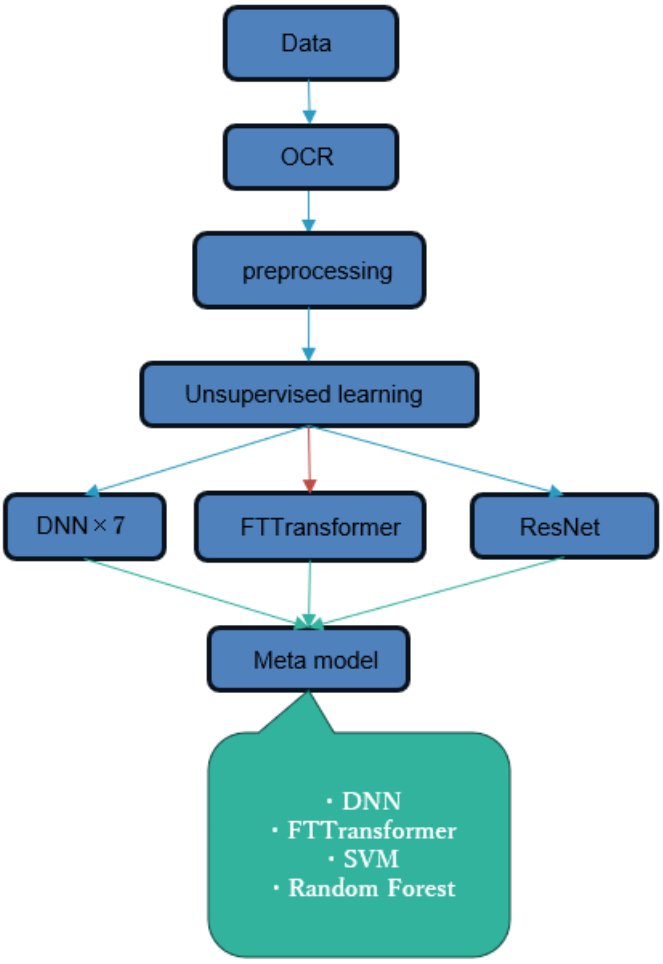

Fig. 1. Ensemble Learning Pipeline used in predicting dengue fever

4 Experimental Results

4.1 Experiment settings

The experimental settings are shown in Table 2.

Table 2. Experimental settings

OS	Ubuntu 24.04.1 LTS
CPU	Intel core i7 ×15
GPU	Nvidia GeForce GTX 1080 Ti ×2
Language	Python

4.2 Evaluation metrics

The standard precision, recall and F1 scores were used to evaluate the performance of the models used in this study. In addition, The Matthews Correlation Coefficient (MCC) was also used. MCC is a metric used to evaluate the performance of a classification model, taking into account true positives, true negatives, false positives, and false negatives. MCC values that range from -1 to 1, where a value close to 1 indicates perfect classification, 0 indicates performance close to random prediction, and -1 represents a perfect inverse prediction. MCC is given by the following equation:

$$MCC = \frac{TP \cdot TN + FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{4}$$

The MCC metric is used to evaluate the overall balance of a model and is particularly robust to class imbalance. This makes it useful for compensating for cases where simple accuracy or F1 score may overestimate model performance due to skewed class distributions. For example, MCC emphasizes the model's ability to correctly predict both True Positives and True Negatives, and thus gives a more stringent evaluation to models that only classify one class accurately. This characteristic makes it well-suited for performance evaluation in situations of class imbalance.

4.3 Hyperparameter tuning

For the deep learning models used in both the base model and meta-model, hyperparameters (such as dropout rate, learning rate, and L2 regularization) were optimized based on the MCC using the Python library *Optuna*. Once the optimal hyperparameters were determined, the final model was trained using these settings. Below are the structures and hyperparameters for the two DNN models, FT-Transformer, and ResNet used in the meta model.

Table 3. DNN1 and DNN7 hyperparameters

DNN1	DNN7
Linear (input, 64)	Linear (input, 64)
Dropout (0.35)	Linear (64, 32)
Linear (64,32)	Dropout (0.15)
Dropout (0.35)	Linear (64, 32)
Linear (32,16)	Linear (16, 8)
Dropout (0.35)	Dropout (0.3)
Linear (16,8)	Linear (8, 1)
Dropout (0.35)	Optimizer: Adam
Linear (8,1)	Learning rate: 0.000264
Optimizer: Adam	Weight decay (L2): 0.000011164
Learning rate: 0.000159	
Weight decay (L2): 0.000431	

Table 4. FT-Transformer hyperparameters

FT-Transformer
Embedding (input, 128)
Dropout (0.5)
Dropout (0.5)
Linear (128,1)
Optimizer: Adam
Learning rate: 0.000226
Weight decay (L2): 0.000353
Weight decay (L2): 0.000011164

Table5. ResNet hyperparameters

ResNet
Hidden dimension = 64
Number of blocks = 4
First linear dropout = 0.35
Second linear dropout = 0.4
Leaning rate = 0.000309
Weight decay(L2) = 0.000403

4.4 Training results

In this experiment, 80% of the created training data were used as training data, with 10% of that set aside for validation data. The training was conducted with a batch size of 128. The training and test results are shown in the following sub-sections.

Base Model Training Results

DNN, FT-Transformer and ResNet were used as base models to predict dengue fever. They were trained for 100, 20 and 100 epochs, respectively. The training progress is shown in Fig. 2, Fig. 3, and Fig. 4.

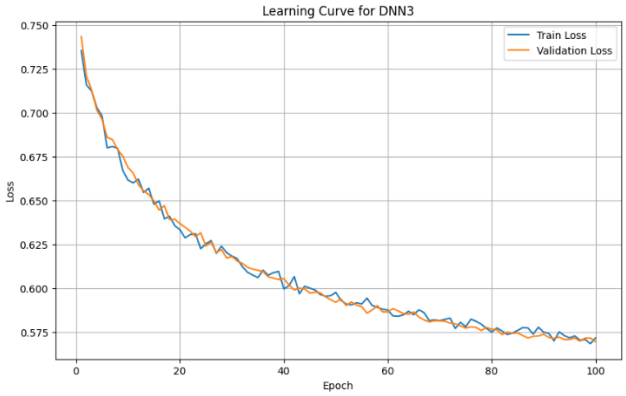

Fig. 2. Learning Curve for DNN

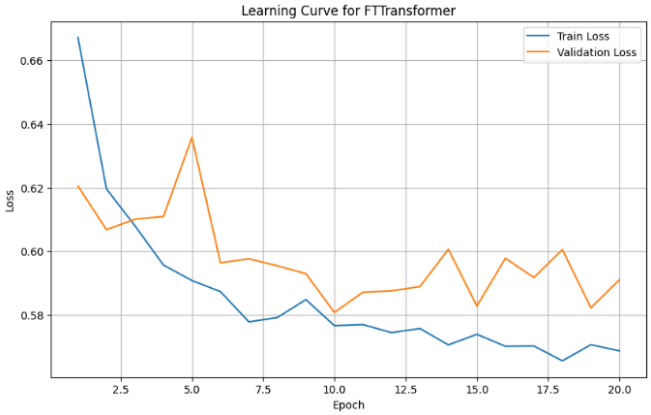

Fig. 3. Learning Curve for FT-Transformer

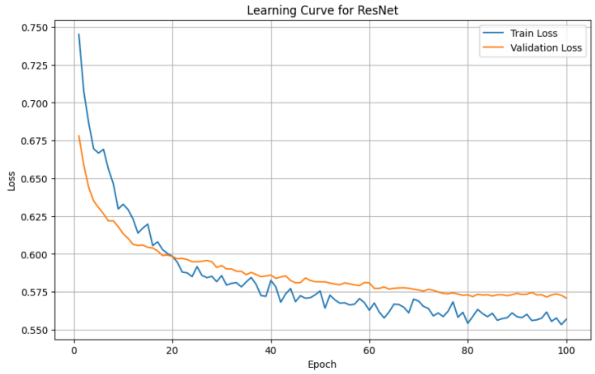

Fig. 4. Learning Curve for ResNet

Meta Model Training Results

DNN and FT-Transformer metamodels are trained for 60 and 50 epochs, respectively. The training progress is shown in Fig. 5 and Fig. 6.

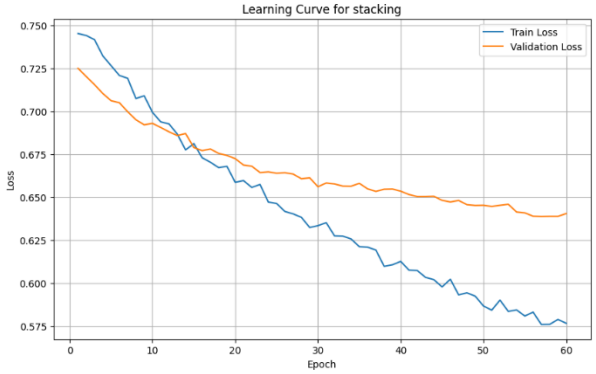

Fig. 5. Learning Curve for DNN (meta model)

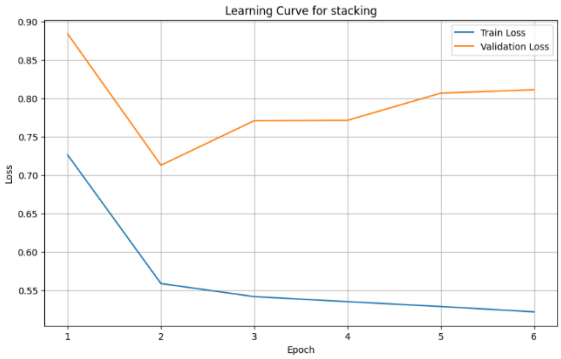

Fig. 6. Learning Curve for FT-Transformer (meta model)

4.5 Evaluation and Validation

For model evaluation, 20% of the created training data were used. The results are shown in Table 7 and Table 8.

Table 7. Evaluation of the base model

model	Accuracy	Precision	Recall	F1 score	MCC
DNN1	59.45%	0.4867	0.9204	0.6367	0.3399
DNN2	61.16%	0.4976	0.6106	0.5483	0.2173
DNN3	58.77%	0.4809	0.8535	0.6149	0.2867
DNN4	57.86%	0.4756	0.8909	0.6201	0.2971
DNN5	55.92%	0.4639	0.9086	0.6142	0.2813
DNN6	57.18%	0.4699	0.8525	0.6059	0.2633
DNN7	59.11%	0.4777	0.6313	0.5438	0.192
FT-Transformer	57.63%	0.4763	0.9764	0.6402	0.3609
ResNet	58.09%	0.475	0.8112	0.5991	0.254

Table 8. Evaluation of the metamodel

model	Accuracy	Precision	Recall	F1 score	MCC
SVM	61.39%	0.4845	0.5719	0.5345	0.2052
Random Forest	59.11%	0.6637	0.6018	0.5319	0.1813
DNN	58.20%	0.4707	0.9725	0.6341	0.3707
FT-Transformer	61.28%	0.4756	0.9235	0.6279	0.3449

Comparing the performance of the 9 base models, DNN2 shows the highest accuracy at 61.16%. On the other hand, FT-Transformer stands out as the model with particularly high recall (0.9764), but its accuracy is only 57.63%, indicating a clear trade-off between accuracy and recall. Additionally, DNN1 has a relatively high F1 score (0.6367) and a high MCC (0.3399), but its accuracy is only 59.45%. Among the base models, there are models with high accuracy (like DNN2) and models with high recall or F1 score (like FT-Transformer and DNN1). In conclusion, the superior model depends on the choice of the evaluation metric.

5 Conclusion and Future Research

In the base model, all the learning is carried out without overfitting, whereas in the meta model, it is observed from the learning curve that FT-Transformer shows slight overfitting. One possible factor for this is an issue with the input data of the meta model. It is necessary to review the hyperparameters for unsupervised learning and conduct repeated validation of the model. For example, re-evaluating the number of clusters in clustering, or changing the number of dimensions removed in PCA, would improve the quality of the data. This should also be considered for the input data of the base model.

Regarding evaluation, using DNN as the meta-model has successfully improved the MCC. Although Precision and Recall have decreased, the improvement in MCC suggests that the classification performance for False Positives (FP) has improved.

In the future, to further improve the model's MCC, it will be essential to use data from other infectious disease patients or to amplify the training data through generative models like GANs. Additionally, while deep learning models were used as the base model in this study, it is necessary to add machine learning models such as Random Forest, SVM, and XGBoost, as well as deep learning models not used in this study, to enhance the model's performance. Moreover, it will be necessary to construct other ensemble learning methods aside from stacking and build more multi-layered stacking models for repeated comparative validation.

References

1. Chen R. and Vasilakis N., Dengue-Quo Tu et Quo Vadis?, Viruses. (2011) 3, no. 9, 1562–1608, https://doi.org/10.3390/v3091562, 2-s2.0-80053359626.
2. Chevillon C. and Failloux A.-B., Questions on viral population biology to complete dengue puzzle, Trends in Microbiology. (2003) 11, no. 9, 415–421, https://doi.org/10.1016/S0966-842X(03)00206-3, 2-s2.0-0042825891.
3. Jayakody, A. et al. (2015). Predicting Dengue Fever Incidence Using Machine Learning Techniques: A Case Study from Sri Lanka. International Journal of Environmental Research and Public Health, 12(7), 8184–8197.
4. Khetarpal, N., & Khanna, I. (2016). Dengue fever: causes, complications, and vaccine strategies. Journal of immunology research, 2016(1), 6803098.
5. Malavige G. N., Fernando S., Fernando D. J., and Seneviratne S. L., Dengue viral infections, Postgraduate Medical Journal. (2004) 80, no. 948, 588–601, https://doi.org/10.1136/pgmj.2004.019638, 2-s2.0-6344233465.
6. Perkins, T. A., et al. (2015). Modeling the Impact of Weather and Climate on Dengue Transmission in Singapore. Environmental Health Perspectives, 123(5), 443-448.
7. Perez, L. et al. (2019). A Real-Time Dengue Prediction Model Using Machine Learning Approaches. Journal of Applied Mathematics, 2019, Article 1726271.
8. Riya, N. J. , Chakraborty, M. and Khan, R. "Artificial Intelligence-Based Early Detection of Dengue Using CBC Data," in IEEE Access, vol. 12, pp. 112355-112367, 2024, doi: 10.1109/ACCESS.2024.3443299.
9. Ruban, S., & Rai, S. (2021). Enabling data to develop an AI-based application for detecting malaria and dengue. Computational intelligence and predictive analysis for medical science: a pragmatic approach, De Gruyter, Berlin, Boston, 115-138.
10. Ruban, S., Naresha, Rai, S. (2022). Detecting Dengue Disease Using Ensemble Classification Algorithms. In: Shukla, S., Gao, XZ., Kureethara, J.V., Mishra, D. (eds) Data Science and Security. Lecture Notes in Networks and Systems, vol 462. Springer, Singapore. https://doi.org/10.1007/978-981-19-2211-4_4
11. Ruban, S., Jabeer, M.M., Rai, S. (2024). Daily Platelet Count Prediction in Treating Dengue Patients Using Deep Learning Algorithm. In: Shetty, N.R., Prasad, N.H., Nagaraj, H.C. (eds) Advances in Communication and Applications. ERCICA 2023. Lecture Notes in Electrical Engineering, vol 1105. Springer, Singapore. https://doi.org/10.1007/978-981-99-7633-1_38

12. Vannavong, N. et al. (2019). Machine Learning for Predicting Dengue Outbreaks in Laos. Journal of Infectious Diseases, 219(10), 1583-1591.
13. WHO: Dengue - Global situation. https://www.who.int/emergencies/disease-outbreak-news/item/2023-DON498, Dec. 21, 2023 (accessed on Dec 10, 2024).
14. Wu, J., et al. (2007). Climatic and Environmental Factors Associated with Dengue Fever Transmission in Taiwan. Science of the Total Environment, 373(1), 235-242.
15. Yang, Z. et al. (2016). Predicting the Spatial Spread of Dengue Fever in Southeast Asia Using Unsupervised Learning. Journal of Geographical Systems, 18(3), 255-273.