INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Energy-Efficient Rainfall Prediction Using Support Vector Machine on Edge AI Platforms

Willy Permana Putra [a], Robi Robiyanto [a], Renol Burjulius [a], Alifia Puspaningrum [a], Ahmad Rifa'i [a], A Sumarudin [a,*]

[a] *Department Informatics, Politeknik Negeri Indramayu, Lohbener, Indramayu, Indonesia*

Corresponding author: [*]shumaru@polindra.ac.id

***Abstract*—The integration of AI into various sectors, including agriculture, has been advancing significantly. Implementing AI in the context of IoT and edge AI presents challenges due to resource limitations. Current climate changes affect planting strategies, pest management, and harvest timing. This study explores an SVM-based machine-learning model with multiple kernels to classify weather conditions as rainy or clear. The research includes two phases: model training on a PC-based system and model deployment on an edge AI device. The training phase includes preprocessing with PCA and fine-tuning of parameters, such as kernel types (linear, polynomial, sigmoid, and RBF), C and gamma. The development phase involves deploying the model on an ESP32, where execution time and power consumption are evaluated. The results show that the SVM model with an RBF kernel, C of 0.1, and gamma of 1 achieves a precision of 79.37%. Inference on the ESP32 yields an average execution time of 35.5 ms and a power consumption of 66 mA, showing a 202-fold reduction in power usage compared to the PC-based system and a 59-fold increase in execution time. This reduced power consumption supports the feasibility of edge AI for climate-based agricultural applications, enabling effective rainfall prediction. The findings contribute to the development of precision agriculture by providing insights into climate prediction, which can inform planting decisions, pest management, and harvest timing, thereby advancing the application of edge AI in response to global climate change.**

***Keywords*—Edge AI; SVM; Energy-efficient; rainfall prediction; agriculture.**

Manuscript received 4 Aug. 2024; revised 19 Sep. 2024; accepted 24 Oct. 2024. Date of publication 30 Nov. 2024.

I. Introduction

In recent years, the rapid advancement of artificial intelligence (AI) technology has led to the widespread adoption of applications running on edge devices [1], [2]. Edge AI is deploying AI models directly on edge devices such as smart sensors, mobile devices, and the Internet of Things (IoT) to process data locally [3], [4] without sending it to the cloud [5]. This enables real-time decision-making and reduces latency and bandwidth requirements for data transmission to the cloud. However, implementing AI on edge devices introduces new challenges, particularly concerning resource constraints such as processing power, memory, and, most critically, energy consumption [1], [6]. Energy efficiency has become crucial because many edge devices operate on limited power sources, such as batteries, which must support long-term operations. Therefore, there is an urgent need to develop Edge AI solutions that are computationally efficient and energy-efficient [7]. Several approaches have been developed to achieve energy efficiency in Edge AI [8]. These include optimizing AI model architectures, utilizing more efficient inference algorithms, and applying model compression techniques to reduce computational complexity [9], [10]. Additionally, specialized hardware [11], such as Neural Processing Units (NPUs) designed for AI inference, significantly reduces energy consumption [12], [13]. As a result, energy-efficient Edge AI solutions enable longer operation times, support sustainability, and reduce the carbon footprint in AI technology deployment [14].

The development of energy-efficient Edge AI [7], [14]-[17] has significant potential across various applications, including healthcare[18]-[20], automotive [21], industrial [22], [23], and agricultural sectors [17], [24]-[27], [28], [29]. For example, AI-powered wearable devices in healthcare can continuously monitor patient conditions with low power consumption [27]. In the automotive sector, autonomous vehicles equipped with Edge AI capabilities can make quick and efficient decisions without requiring extensive communication with central servers [5]. Additionally, there is

the issue of energy efficiency related to security [30], [31], [16], which requires higher power consumption.

Climate change has become a significant concern for agriculture, particularly due to the shifts in rainfall patterns, which are critical for crop production. The availability of water, heavily dependent on consistent and predictable rainfall, directly influences agricultural output. As climate change continues to disrupt these patterns, it becomes increasingly vital to develop accurate systems for detecting and predicting rainfall. Such advancements are necessary to maintain agricultural productivity and ensure food security in a changing climate[32]-[35].

This research explores with machine learning model based on SVM for prediction rainfall with any strategies [36-43] that can enhance energy efficiency in Edge AI applications and examines their impact on the overall system performance. Thus, the prediction system can provide weather recommendations that run efficiently on edge AI to address the impact of climate change on agriculture.

II. Materials and Methods

The methodology in this research consists of two steps: training and development, as shown in Figure 1. The model training was conducted on a PC with an i7 processor to obtain the *.kpi model. Using micromlgen, the model was then converted into a *.h file for edge AI development on an embedded device based on a microcontroller. The model's performance was compared by running inference on both the PC and the edge AI device to measure execution time and power consumption required for SVM inference during the development process for the end user.

A. Support Vector Machine Algorithm

A Support Vector Machine (SVM) is a supervised learning algorithm designed for analyzing data and identifying patterns for classification purposes. It processes a set of training data, classifies it into different categories, and then predicts if a test document belongs to one of these categories.

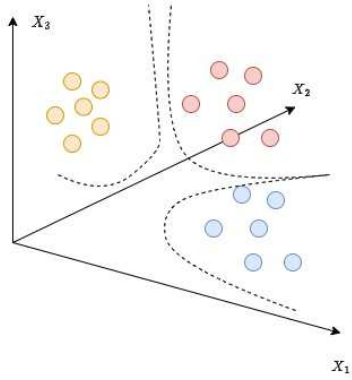

Fig. 1 Support Vector Machine

In SVM, data is represented as points in space, and the classification is determined by a line or hyperplane that separates these points. Additionally, SVM classifiers can use various kernel functions to handle non-linear data samples. By transforming the data into a higher-dimensional space where it becomes linearly separable, these kernels simplify the classification task, effectively turning a previously non-linear decision boundary into a linear one. Some commonly used kernel functions with SVM, namely Polynomial kernel, Gaussian kernel, RBF or Gaussian radial basis function kernel, Laplace RBF kernel, Hyperbolic tangent kernel, Sigmoid kernel, Bessel function kernel, Linear splines kernel, etc.

Support Vector Machine (SVM) is a machine learning technique that utilizes kernels to assist in classifying data into two or more categories, as shown in Figure 2. As the input parameter increases, SVM performs classification comparison between classes using methods such as One-vs-One (OVO), One-vs-All or One-vs-Rest (OVR). The kernel, which serves as the line separating classes, employs approaches such as linear kernel (eq. 1), polynomial kernel (eq. 2), sigmoid kernel (eq. 3), and radial basis function (RBF) kernel (eq. 4) [44].

$$K(x_i, x_j) = x_i \times x_j \quad (1)$$

$$K(x_i, x_j) = (\alpha(x_i \times x_j) + c)^d \quad (2)$$

$$K(x_i, x_j) = \tanh(\alpha(x_i \times x_j) + c) \quad (3)$$

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (4)$$

$$f(x) = sign(\sum_{i=1}^{N} a_i y_i (-\gamma \|x_i - x_j\|^2 + b) \quad (5)$$

In this Equation 5, the class x is determined by calculating a weighted sum of the kernel function values for all training samples. Each term in the sum includes a weight a_i, the class label y_i, and a kernel function value that depends on γ (gamma) and the squared distance between x_i (a training sample) and x_j (the test sample), plus a bias term b. The 'sign' function is then used to convert this sum into a class label.

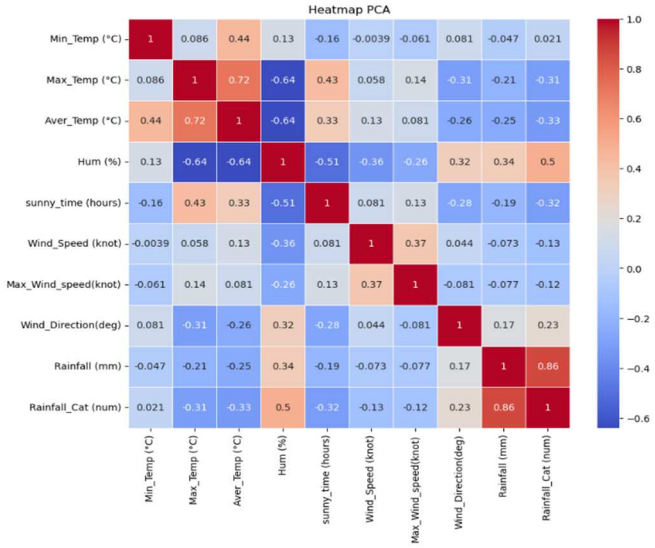

Fig. 2 Principal component analysis rainfall prediction

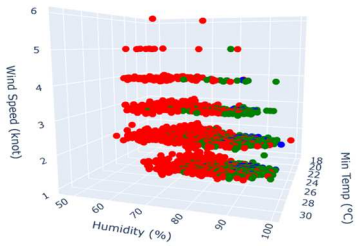

Fig. 3 Plot dataset rainfall based on three classifications (red = sunny, green=rain, and blue = heavy rain)

The input parameters were derived from the BMKG at Majalengka district dataset, covering the period from June 2008 to December 2018. There are 9 attributes collected, namely Min_Temp, Max_Temp, Aver_Temp, Hum, sunny_time, wind_speed, max_wind_speed, wind_direction, rainfall.

After that, Principal Component Analysis (PCA) was conducted to identify the most significant input parameters for the system, as illustrated in Figure 3. The analysis revealed that humidity, minimum temperature, and wind speed were the key factors influencing the classification of rainfall into two categories: clear and rainy, as shown in Figure 4. The preprocessing of the dataset, utilizing PCA, identified minimum temperature (-0.047), humidity (0.34), and wind speed (-0.073) as the parameters most correlated with rainfall (mm) output.

As shown in Figure 4, the visualization of rainfall is primarily determined by humidity, which effectively distinguishes between rainy and non-rainy categories but is insufficient for differentiating types of rain. Therefore, additional parameters such as minimum temperature and wind speed are required. The model will classify these categories using an SVM, and the analysis will employ four different kernels as outlined in Equations 1-4.

The processing of the dataset identifies key parameters that influence the desired classification, which then serve as inputs to the SVM algorithm. The outcome of this process is used to generate the model, as shown in Table 1. The SVM input (X) includes three parameters: minimum temperature, daily average humidity, and wind speed in knots, which are used to map the output classes (Y) into two categories: sunny, and rain, labeled [0, 1].

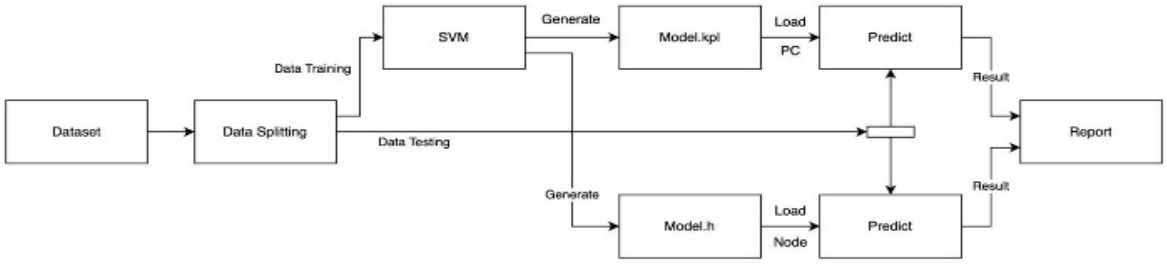

Fig. 4 Method development edge AI

The key parameters in the SVM model are C and gamma, which are critical for the kernel and classification function. A smaller C value provides a larger margin, while a larger C value results in a smaller margin, thereby influencing the separation boundary between output classes. The gamma value determines the influence of each support vector in shaping the classification boundary. A grid search will be conducted to tune the C and gamma parameters, and the performance accuracy will be compared to identifying the best parameter settings.

TABLE I
PSEUDOCODE SVM FOR RAINFALL PREDICTION

Input:
Training data: X = {min temp, humidity, wind speed}
Label: Y = {'sunny', 'rain'}
Regularization parameter: C
Kernel parameter: gamma

Initialize:
Lagrange multiplier
Bias term: b = 0

SVM compute:
Normalization input parameter and label
Choose method comparison with one vs all
Grid search with best parameter SVM:
  Compute kernel Metrix using eq. 1/2/3/4
  Optimize model SVM with eq. 5
Choose best parameter SVM
Compute the decision function for predict with eq. 5

OUTPUT:
SVM model with decision function classification

B. Setting parameter

Table 2 lists the parameter settings used in this research. The kernel parameters were compared across four types: linear, sigmoid, polynomial, and RBF. The C value was tested with three different parameters: 0.01, 0.1, and 1, while the gamma value was evaluated with 1, 0.1, 0.001, 0.0001, and 0.00001.

TABLE II
PARAMETER SVM

Parameter	Value
Kernel	Linear, Sigmoid, Poly, RBF
C	[0.01, 0.1, 1]
Gamma [γ]	[1, 0.1, 0.01, 0.001, 0.0001]

C. Design edge AI platform

The SVM model was trained on a CPU with an i7 processor, and the development was carried out using an ESP32, as shown in Table 3. The device used for the edge AI implementation was an ESP32 WROM, with its specifications detailed in Table 3. The ESP32 Wroom operates with a core frequency of 40 MHz and a bit rate powered by a 3.3 V supply with a minimum current of 500 mA, and it features 4 MB of flash RAM. These limited resources pose a significant challenge when running AI algorithms on edge computing devices. The parameters obtained from machine learning training are used to generate the model as a library (in .h format), which is then loaded into the flash RAM. The size of the machine learning parameters directly influences the amount of flash RAM required. Edge AI enables predictions to be made directly on the device without relying on cloud-

based AI, thereby increasing processing speed and reducing communication with the cloud AI.

TABLE III
DEVICE TRAINING AND DEVELOPMENT EDGE AI

Parameters	CPU i7 Training	ESP32 Development
Core	4	2
Frequency	1.9 GHz	40 MHz
bit width	64 bits	32 bits
Flash RAM	SSD	4 MB
SRAM	16 GB DDR	520 KB

Figure 5 illustrates the implementation of edge AI on the ESP32. Weather conditions in the edge environment are sensed using a wind sensor to measure wind speed in knots, minimum temperature in Celsius, and average humidity. The sensors communicate via the RS485 protocol. After training, the SVM model is loaded into the flash RAM to perform rainfall predictions on the edge AI. The predicted results are then displayed on an OLED screen.

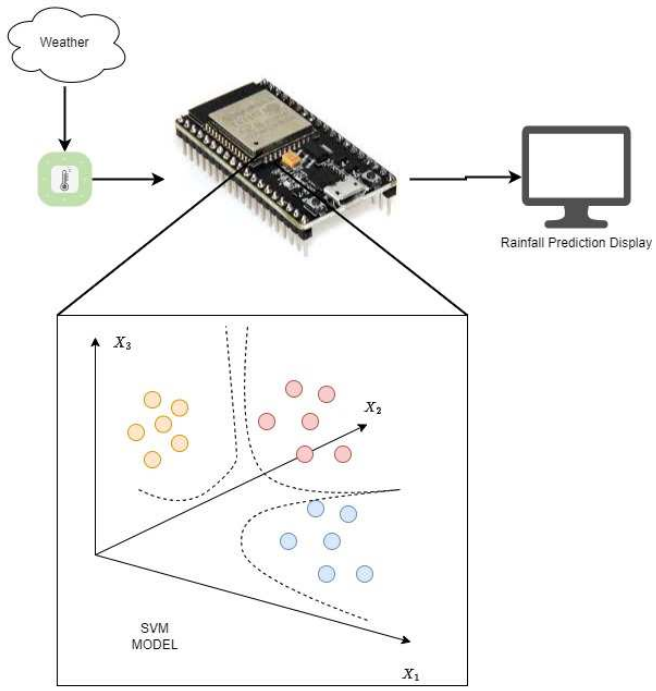

Fig. 5 Edge SVM implementation process for an edge AI

III. RESULTS AND DISCUSSION

A. Results Implementation SVM Training

As designed in Table 1, the SVM model was trained on a computer with an i7 CPU and 16 GB of RAM. The training aimed to identify the optimal parameters for the SVM, including the kernel type (linear, sigmoid, polynomial, and RBF) and the values of C and gamma. To achieve the best parameter values, 300 fits were conducted, requiring a computation time of 52 seconds. The best performance for each Kernel is shown in Table 4. The grid search results identified the RBF kernel with a C value of 0.1 and a gamma value of 1, yielding an accuracy of 79.37%. The classification boundary of the SVM, using the RBF kernel with C set to 0.1 and gamma to 1, is illustrated in Figure 6. This boundary effectively separates the classifications of rainy and non-rainy conditions, which is essential for predicting whether it will rain on agricultural land—critical for water supply and pest prediction.

TABLE IV
PERFORMANCE SVM

Kernel	C	Gamma [γ]	Precision [%]
Sigmoid	1	0.1	74.36
Linier	0.1	1	75.23
Poly	0.1	1	71.56
RBF	**0.1**	**1**	**79.37**

The confusion matrix of the performance parameters obtained with the best precision in Table 4 is shown in Table 5. The testing data used consists of 2,543 data points.

TABLE V
CONFUSION MATRIX

Data testing = 2543	Actual sunny (0)	Actual rain (1)
Actual sunny (0)	1237	271
Actual Rain (1)	299	736

TABLE VI
PERFORMANCE MODEL WITH RBF KERNEL

Parameter	Value
Accuracy	78%
Precision	82%
Recall	81%
F1 score	81%

The best parameters obtained from this process were used to generate a library as a .h file using micromlgen. This library was then loaded onto the ESP32 for the inference model during the edge AI development phase to predict rainfall.

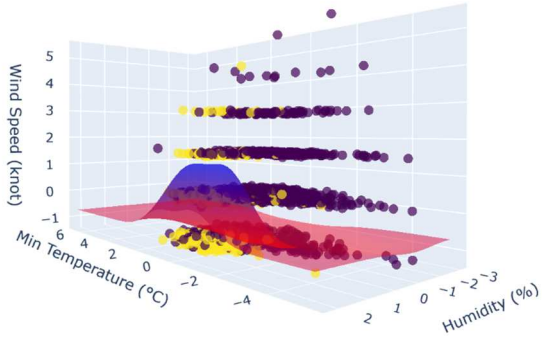

Fig. 6 Classification using best parameter

B. Implementation SVM Development Inference on edge AI

The trained model was loaded onto the edge AI, as shown in Figure 7, marking the development phase of the edge AI. The generated library file in `.h` format was 304 KB, comfortably fitting within the edge AI device's 2 MB flash RAM. This `.h` file encapsulates the computations from Equations 1-5 using the parameters obtained from the training process. The edge AI takes input from weather sensors with meteorological multi-element shutter connected via the RS485 protocol. These sensors have been calibrated beforehand to ensure data validity. The prediction relies on three sensors: a wind sensor, a temperature sensor, and a humidity sensor.

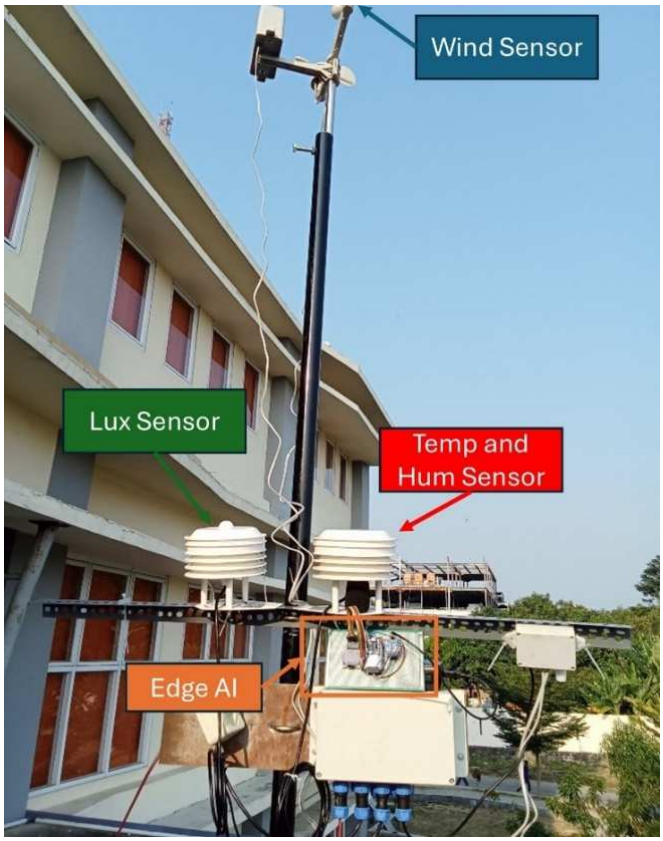

Fig. 7 Overview Rainfall Prediction Application with weather sensor and edge AI for this implementation

The wind speed using APRS Weather Station is measured using an anemometer-based sensor. Temperature and humidity are measured using a temperature chip sensor, which offers a humidity accuracy of 3% over a 0-99% range and a temperature accuracy of ±0.5°C over a -40 to 125°C range.

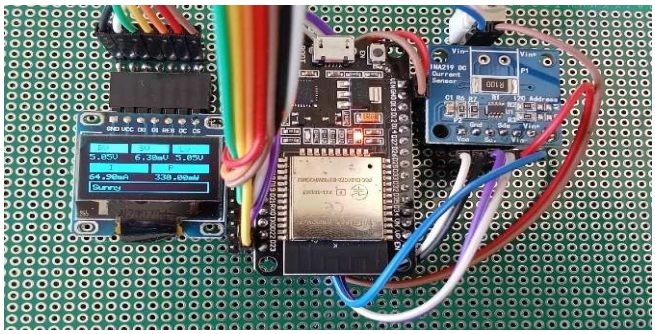

Fig. 8 Edge AI Implementation Prediction Results

The prediction results are displayed on an OLED screen, allowing the user to view the predicted weather conditions. Additionally, these predictions can be sent to the cloud for further analysis. This approach enables predictive computations to be performed at the edge, enhancing the speed of AI processes on the edge device. The cloud serves as a storage medium for big data, facilitating more complex analyses such as climate analysis and pest prediction on agricultural lands. Figure 8 illustrates the prediction results on the edge AI, based on input from the sensors. The prediction classifies the weather into two categories: rainy or sunny.

C. Performance comparison SVM inference on edge AI

The performance measurements focused on execution time and power consumption on the edge AI. A comparison was made between the edge AI running on the ESP32 and a PC-based CPU with an i7 processor. Computational time was measured by applying a timestamp in the inference model program: Python-based timing was used for the i7 CPU, while the `millis()` function was employed for the ESP32. The results, shown in Table 7, indicate a computational time difference of approximately 30 ms, meaning the ESP32 is 59 times slower in execution compared to the i7 CPU. However, despite this 30 ms difference in execution time, the edge AI can implement rainfall prediction.

TABLE VII
TIME EXECUTION

Device	Average time consumption [ms]
CPU i7	0.6
ESP32	35.5
Slower than	**59x**

Figure 9 shows the computation time on a PC with an i7 processor using a Python-based program. The test was conducted with 727 data points (data testing). As illustrated in Figure 9, the average execution time is approximately 0.6 ms. However, there are instances where the execution time reaches 1.3 ms and 2.2 ms, which can be attributed to variations influenced by the operating system's processes.

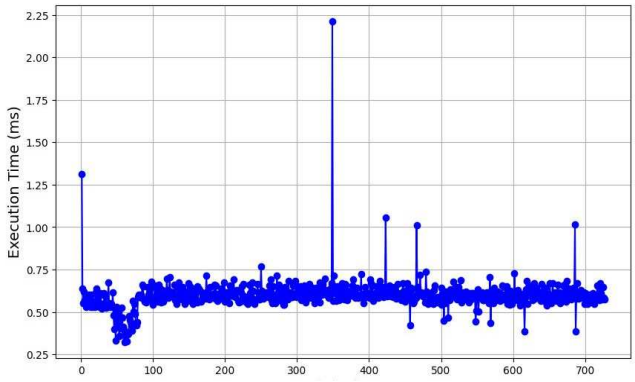

Fig. 9 Time execution on PC on CPU i7

Figure 10 illustrates the execution times for each test data point on the edge AI. These times vary due to differences in input data and the process of converting the input into the model format. On average, the edge AI has an execution time of approximately 35.5 ms.

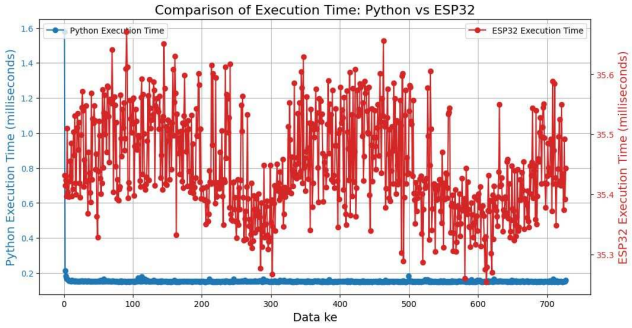

Fig. 10 Time execution on ESP32

Figure 11 compares the execution times between the PC-based system and the ESP32-based edge AI, highlighting a difference of about 35 ms between the two platforms. Power consumption was measured on the ESP32 using current and voltage sensors during program execution.

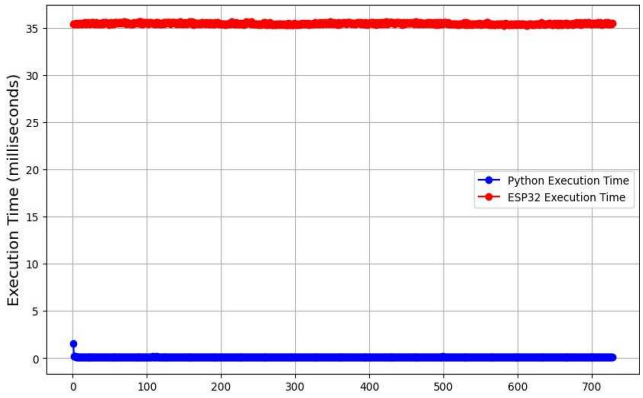

Fig. 11 Comparison time execution CPU i7 vs ESP32

As illustrated in Figure 12, the power required for executing the edge AI is 330 mW. In comparison, power measurements for the PC, obtained through operating system tools, show a consumption of 77 watts on the i7 CPU.

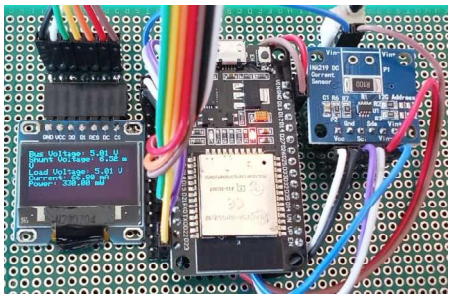
Fig. 12 Power consumption measurement

Table VIII compares power usage, revealing that the edge AI is approximately 202 times more energy-efficient than the PC-based system.

TABLE VIII
POWER CONSUMPTION COMPARISON

Device	Power Source	Power [W]
CPU i7	Regulator	72
ESP32	Battery Li-Ion	0.330
Less Power consumption		**202 x**

With a Li-Ion battery capacity of 4800 mAh and current ede AI 330mA, the device can operate for approximately 72.73 hours, as calculated using Equation 6.

$$Life\ time\ edge\ AI = \frac{Bat\_Capacity}{I(edgeAI)} \quad (6)$$

IV. CONCLUSION

The application of edge AI for rainfall prediction utilizes an SVM model with inputs including wind speed, minimum temperature, and humidity, and produces classification outcomes of rainy or sunny. The research decided that an SVM with an RBF kernel, C set to 0.1, and gamma set to 1 achieved a precision of 79.37%. This model was implemented on an edge AI device based on the ESP32 in a library format. The implementation showed a power consumption that is 202 times more efficient, with an average SVM inference execution time of approximately 35.5 ms. These findings show that the model is effective for deployment on edge AI devices with limited resources, making it suitable for agricultural applications related to climate change and rainfall prediction.

ACKNOWLEDGMENT

We thank Politeknik Negeri Indramayu under grant PUKTI JIP. The authors fully acknowledged Politeknik Negeri Indramayu for the approved fund, making this research can effective and can implementation.

REFERENCES

[1] K. Jewani and S. Abimannan, "Edge Intelligence in IoT: Architecture and Applications," 2023 8th International Conference on Communication and Electronics Systems (ICCES), pp. 337–345, Jun. 2023, doi: 10.1109/icces57224.2023.10192608.

[2] Y. Shi, K. Yang, T. Jiang, J. Zhang, and K. B. Letaief, "Communication-Efficient Edge AI: Algorithms and Systems," IEEE Communications Surveys & Tutorials, vol. 22, no. 4, pp. 2167–2191, 2020, doi: 10.1109/comst.2020.3007787.

[3] J. Zhang et al., "AntiConcealer: Reliable Detection of Adversary Concealed Behaviors in EdgeAI-Assisted IoT," IEEE Internet of Things Journal, vol. 9, no. 22, pp. 22184–22193, Nov. 2022, doi:10.1109/jiot.2021.3103138.

[4] A. Katiyar and P. Kumar, "A Review of Internet of Things (IoT) in Construction Industry: Building a Better Future," International Journal of Advanced Science Computing and Engineering, vol. 3, no. 2, pp. 65–72, Aug. 2021, doi: 10.62527/ijasce.3.2.53.

[5] S. Rane, P. Shah, and R. Sekhar, "Survey of Technologies for Industry 4.0," 2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA, pp. 1–6, Aug. 2022, doi: 10.1109/iccubea54992.2022.10010837.

[6] S. K. Jagatheesaperumal, Q.-V. Pham, R. Ruby, Z. Yang, C. Xu, and Z. Zhang, "Explainable AI Over the Internet of Things (IoT): Overview, State-of-the-Art and Future Directions," IEEE Open Journal of the Communications Society, vol. 3, pp. 2106–2136, 2022, doi: 10.1109/ojcoms.2022.3215676.

[7] N. K. Manjunath, A. Shiri, M. Hosseini, B. Prakash, N. R. Waytowich, and T. Mohsenin, "An Energy Efficient EdgeAI Autoencoder Accelerator for Reinforcement Learning," IEEE Open Journal of Circuits and Systems, vol. 2, pp. 182–195, 2021, doi:10.1109/ojcas.2020.3043737.

[8] S. Pal, M. Umair, W.-H. Tan, and Y.-L. Foo, "Practical Evaluation of Federated Learning in Edge AI for IoT," JOIV : International Journal on Informatics Visualization, vol. 7, no. 3–2, p. 2115, Nov. 2023, doi:10.30630/joiv.7.3-2.2329.

[9] H. Kong et al., "EdgeCompress: Coupling Multidimensional Model Compression and Dynamic Inference for EdgeAI," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 42, no. 12, pp. 4657–4670, Dec. 2023, doi:10.1109/tcad.2023.3276938.

[10] J. Li et al., "A Lightweight Deep Learning-Based Cloud Detection Method for Sentinel-2A Imagery Fusing Multiscale Spectral and Spatial Features," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–19, 2022, doi: 10.1109/tgrs.2021.3069641.

[11] C. Åleskog, H. Grahn, and A. Borg, "Recent Developments in Low-Power AI Accelerators: A Survey," Algorithms, vol. 15, no. 11, p. 419, Nov. 2022, doi: 10.3390/a15110419.

[12] F. Wang, M. Zhang, X. Wang, X. Ma, and J. Liu, "Deep Learning for Edge Computing Applications: A State-of-the-Art Survey," IEEE Access, vol. 8, pp. 58322–58336, 2020, doi:10.1109/access.2020.2982411.

[13] D. M. Satyavathi, B. V. Mala, Ch. V. Vamsi, Ch. C. Chiranjeevi, and Ch. N. Neeraj, "Real-Time Hidden Data Transmission Using Lora," International Journal of Advanced Science Computing and Engineering, vol. 4, no. 2, pp. 130–137, Aug. 2022, doi:10.62527/ijasce.4.2.88.

[14] M. Zawish, N. Ashraf, R. I. Ansari, and S. Davy, "Energy-Aware AI-Driven Framework for Edge-Computing-Based IoT Applications," IEEE Internet of Things Journal, vol. 10, no. 6, pp. 5013–5023, Mar. 2023, doi: 10.1109/jiot.2022.3219202.

[15] M. Shafique, A. Marchisio, R. V. Wicaksana Putra, and M. A. Hanif, "Towards Energy-Efficient and Secure Edge AI: A Cross-Layer Framework ICCAD Special Session Paper," 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), pp. 1–9, Nov. 2021, doi: 10.1109/iccad51958.2021.9643539.

[16] D. Katare, D. Perino, J. Nurmi, M. Warnier, M. Janssen, and A. Y. Ding, "A Survey on Approximate Edge AI for Energy Efficient Autonomous Driving Services," IEEE Communications Surveys & Tutorials, vol. 25, no. 4, pp. 2714–2754, 2023, doi:10.1109/comst.2023.3302474.

[17] A. K. Bashir et al., "Federated Learning for the Healthcare Metaverse: Concepts, Applications, Challenges, and Future Directions," IEEE Internet of Things Journal, vol. 10, no. 24, pp. 21873–21891, Dec. 2023, doi: 10.1109/jiot.2023.3304790.

[18] F. Deeba and S. R. Patil, "Implementation of Artificial Intelligence in Disease Prediction and Healthcare System- A Survey," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), pp. 1–8, Nov. 2021, doi: 10.1109/i-pact52855.2021.9696698.

[19] Y. A. Qadri, A. Nauman, Y. B. Zikria, A. V. Vasilakos, and S. W. Kim, "The Future of Healthcare Internet of Things: A Survey of Emerging Technologies," IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1121–1167, 2020, doi:10.1109/comst.2020.2973314.

[20] W. Tong, A. Hussain, W. X. Bo, and S. Maharjan, "Artificial Intelligence for Vehicle-to-Everything: A Survey," IEEE Access, vol. 7, pp. 10823–10843, 2019, doi: 10.1109/access.2019.2891073.

[21] O. Elijah et al., "A Survey on Industry 4.0 for the Oil and Gas Industry: Upstream Sector," IEEE Access, vol. 9, pp. 144438–144468, 2021, doi: 10.1109/access.2021.3121302.

[22] I. Ahmad et al., "Communications Security in Industry X: A Survey," IEEE Open Journal of the Communications Society, vol. 5, pp. 982–1025, 2024, doi: 10.1109/ojcoms.2024.3356076.

[23] R. K. Singh, R. Berkvens, and M. Weyn, "AgriFusion: An Architecture for IoT and Emerging Technologies Based on a Precision Agriculture Survey," IEEE Access, vol. 9, pp. 136253–136283, 2021, doi: 10.1109/access.2021.3116814.

[24] C. Sun et al., "Advancing UAV Communications: A Comprehensive Survey of Cutting-Edge Machine Learning Techniques," IEEE Open Journal of Vehicular Technology, vol. 5, pp. 825–854, 2024, doi:10.1109/ojvt.2024.3401024.

[25] P. McEnroe, S. Wang, and M. Liyanage, "A Survey on the Convergence of Edge Computing and AI for UAVs: Opportunities and Challenges," IEEE Internet of Things Journal, vol. 9, no. 17, pp. 15435–15459, Sep. 2022, doi: 10.1109/jiot.2022.3176400.

[26] Ms. R. Aishwarya and G. Mathivanan, "AI Strategy for Stake Cloud Computing and Edge Computing: A State of the art survey," 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 920–927, Dec. 2021, doi:10.1109/iceca52323.2021.9676013.

[27] V. Hayyolalam, M. Aloqaily, O. Ozkasap, and M. Guizani, "Edge-Assisted Solutions for IoT-Based Connected Healthcare Systems: A Literature Review," IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9419–9443, Jun. 2022, doi: 10.1109/jiot.2021.3135200.

[28] W. Bismi, D. Riana, and A. S. Hewiz, "Disease Identification on Fig Leaf Images Using Deep Learning Method," International Journal of Advanced Science Computing and Engineering, vol. 6, no. 2, pp. 57–63, Jul. 2024, doi: 10.62527/ijasce.6.2.203.

[29] R. Hidayat, H. Amnur, A. Alanda, Yuhefizar, and D. Satria, "Capacity Building for Farming System Digitalization Using Farming Management System," International Journal of Advanced Science Computing and Engineering, vol. 5, no. 3, pp. 323–327, Dec. 2023, doi: 10.62527/ijasce.5.3.186.

[30] J. Chang, Z. Li, M. Kaveh, Y. Zhang, J. Li, and Z. Yan, "A Survey on AI-Enabled Attacks and AI-Empowered Countermeasures in Physical Layer," 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), pp. 1–7, Oct. 2023, doi: 10.1109/wf-iot58464.2023.10539554.

[31] M. Shen et al., "Blockchains for Artificial Intelligence of Things: A Comprehensive Survey," IEEE Internet of Things Journal, vol. 10, no. 16, pp. 14483–14506, Aug. 2023, doi: 10.1109/jiot.2023.3268705.

[32] I. P. on Climate Change (IPCC), *Climate Change 2021: The Physical Science Basis*. Cambridge University Press, 2021.

[33] Food and A. O. of the United Nations (FAO), *Climate Change and Food Security: Risks and Responses*. Rome, Italy: FAO, 2016.

[34] D. B. Lobell and S. M. Gourdji, "The Influence of Climate Change on Global Crop Productivity," Plant Physiology, vol. 160, no. 4, pp. 1686–1697, Oct. 2012, doi: 10.1104/pp.112.208298.

[35] C. Rosenzweig and D. Hillel, *Climate Change and the Global Harvest: Potential Impacts of the Greenhouse Effect on Agriculture*. New York, NY: Oxford University Press, 2008.

[36] X. Chen, W. Huang, M. C. Haller, and R. Pittman, "Rain-Contaminated Region Segmentation of X-Band Marine Radar Images With an Ensemble of SegNets," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 141–154, 2021, doi: 10.1109/jstars.2020.3043739.

[37] Q. Zhao, Y. Liu, W. Yao, and Y. Yao, "Hourly Rainfall Forecast Model Using Supervised Learning Algorithm," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1–9, 2022, doi:10.1109/tgrs.2021.3054582.

[38] X. Chen, W. Huang, C. Zhao, and Y. Tian, "Rain Detection From X-Band Marine Radar Images: A Support Vector Machine-Based Approach," IEEE Transactions on Geoscience and Remote Sensing, vol. 58, no. 3, pp. 2115–2123, Mar. 2020, doi:10.1109/tgrs.2019.2953143.

[39] Md. M. Hassan et al., "Machine Learning-Based Rainfall Prediction: Unveiling Insights and Forecasting for Improved Preparedness," IEEE Access, vol. 11, pp. 132196–132222, 2023, doi:10.1109/access.2023.3333876.

[40] K. Song, X. Liu, M. Zou, D. Zhou, H. Wu, and F. Ji, "Experimental Study of Detecting Rainfall Using Microwave Links: Classification of Wet and Dry Periods," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 13, pp. 5264–5271, 2020, doi: 10.1109/jstars.2020.3021555.

[41] X. Zhang, S. N. Mohanty, A. K. Parida, S. K. Pani, B. Dong, and X. Cheng, "Annual and Non-Monsoon Rainfall Prediction Modelling Using SVR-MLP: An Empirical Study From Odisha," IEEE Access, vol. 8, pp. 30223–30233, 2020, doi: 10.1109/access.2020.2972435.

[42] Y. Wei, Y. Liu, H. Song, and Z. Lu, "A Method of Rainfall Detection From X-Band Marine Radar Image Based on the Principal Component Feature Extracted," IEEE Geoscience and Remote Sensing Letters, vol. 20, pp. 1–5, 2023, doi: 10.1109/lgrs.2023.3235714.

[43] Y. Jiang, J. Yao, and Z. Qian, "A Method of Forecasting Thunderstorms and Gale Weather Based on Multisource Convolution Neural Network," IEEE Access, vol. 7, pp. 107695–107698, 2019, doi: 10.1109/access.2019.2932027.

[44] T. Fletcher, "Support Vector Machines Explained," 2008. [Online]. Available: www.cs.ucl.ac.uk/staff/T.Fletcher/