Development of an AIoT-Driven Micro:bit Literacy Game Utilizing Ensemble k-NN and Decision Tree for Early STEAM Education

Hayatunnufus*,1), Nirmala Aizya Agatha Silalahi2)

1)Department of Computer Science, Faculty of Computer Science an Information Technology, Universitas Sumatera Utara
Jl. Dr. T. Mansur No. 9, Kampus Padang Bulan, Medan, Sumatera Utara, Indonesia , 20155

2) Department of Computer Science, Faculty of Computer Science an Information Technology, Universitas Sumatera Utara
Jl. Dr. T. Mansur No. 9, Kampus Padang Bulan, Medan, Sumatera Utara, Indonesia , 20155

Email: [1]hayatunnufust@usu.ac.id

Abstract

Low literacy performance among Indonesian students, as indicated by PISA result, highlights the need for innovative learning approaches. This study proposes an interactive literacy game integrating Micro:bit and machine learning within an AIoT framework for STEAM education. The system uses accelerometer data to recognize hand gestures (left, right, up) as multiple-choice answers. A dataset of 61,148 gesture samples was collected and processed. An ensemble model combining k-Nearest Neighbor (k-NN) and Decision Tree algorithms achieved a classification accuracy of 90.25%. Real-time implementation with elementary students (n=19) yielded a gesture recognition accuracy of 95.03%. User testing showed high engagement (8.5/10) and positive learning impact (8.9/10), demonstrating the system's effectiveness as a lightweight, interactive tool for literacy education.

Keywords: *Literacy, Microbit, AIoT, Machine Learning*

1. Introduction

Literacy is a fundamental skill for cognitive development and societal participation. Literacy is not just the ability to recognize words or sentences, but includes the skills to understand, use, evaluate, and reflect on written information effectively [11]. However, data from the Programme for International Student Assessment (PISA) 2018 and 2022 consistently places Indonesian students' reading literacy scores well below the OECD average [1].

In 2018, Indonesia's overall score was only 371 out of a target of 450 to 500, which is very far from the OECD average of 487. In fact, the 2022 PISA report indicates that Indonesia remains at the bottom in terms of literacy [9]. This low performance is often attributed to conventional, passive learning methods that fail to engage the current generation of digital-native students [7].

Low literacy rates are also exacerbated by a lack of interest in reading, with UNESCO reporting that only 0.001% of Indonesians have a reading habit. This situation is further exacerbated by the increasing use of mobile games, which have a negative impact, particularly on children [5].

In response, the Indonesian Ministry of Education has promoted the "Merdeka Belajar" curriculum [14], emphasizing STEAM (Science, Technology, Engineering, Arts, Mathematics) and the integration of technology and coding [2]. This creates an urgent need for innovative, interactive learning media. Research shows that gamification significantly boosts academic performance and motivation [2], and the integration of physical computing like the BBC Micro:bit can create engaging, hands-on learning experiences [3].

This paper proposes a novel AIoT (Artificial Intelligence of Things system that integrates a Micro:bit as a gesture-based input device with a lightweight machine learning model to create an interactive literacy game [12]. Unlike previous works that treat these components separately, our contribution is a unified, practical system optimized for resource-constrained environments and young learners. It combines the pattern recognition of k-NN with the fast decision-making of Decision Trees to classify real-time hand gestures, providing an engaging, full-body learning experience [4].

In this system, gesture data from the micro:bit is processed using the k-NN and Decision Tree algorithms. The k-NN algorithm works by classifying new data based on its similarity to the training data, making it effective for small datasets such as hand movement patterns [4], while Decision Tree generates fast and easy-to- understand classification rules. The combination of the two enables the system to recognize gestures with greater accuracy and interpretability [10]. Acceleration data from the micro:bit sensors (X, Y, Z axes) is then sent to the device and classified in real time with low energy consumption, in line with the TinyML paradigm for Internet of Things applications [13].

Furthermore, this research is placed within the framework of Artificial Intelligence of Things (AIoT), namely the synergy between Artificial Intelligence (AI) capabilities in analyzing data with Internet of Things (IoT) connectivity that allows devices to connect and interact with each other in real-time. In the context of education, AIoT opens up significant opportunities to create learning experiences that are more adaptive, interactive, and responsive to student needs [15]. A systematic review by [6] shows that the application of AI in STEAM education from 2011-2021 has experienced a significant increase, especially in the use of smart devices and adaptive systems to improve learning effectiveness [10].

2. Methodology

2.1 Hardware Architecture Design

The hardware architecture is designed with a dual-mode architecture that differentiates between the data collection phase (training) and the real-time use phase (implementation). During the training phase, the Micro:bit is connected to a computer using a USB cable to ensure maximum data stability and quality. During the implementation phase, the system uses a Bluetooth Low Energy (BLE) connection so students can move freely without being bothered by cables while playing games.

A BBC Miro:bit V2 (ARM Cortex-M4, 128KB RAM) with built-in LSM303AGR accelerometer was used as the input device. The accelerometer was sampled at 50 Hz, recording raw values on the X (left-right), Y (front- back), and Z (up-down) axes. Data were collected in two modes:

- Training mode (wired): Micro:bit connected via USB serial (baud rate 115200). Button A started recording, button B stopped recording. Each recorded gesture window lasted approximately 2 seconds.
- Game mode (wireless): Micro:bit powered by 2×AAA batteries and connected via Bluetooth Low Energy (BLE 5.0) using the Nordic UART Service. The same button trigger mechanism was used.

Three gesture classes were defined:

- Left (answer A) – horizontal movement to the left.
- Right (answer B) – horizontal movement to the right.
- Up (answer C) – vertical upward movement

A total of 61,148 raw samples were collected from a single adult performer (the second author) in a controlled environment. Each class was recorded in 30 separate CSV files, each containing ~500 samples rows. Table I summarize the dataset composition.

Table 1. Distribution of raw gesture samples before preprocessing

Gesture class	Number of raw samples	Percentage
Left	16,142	26.4%
Right	22,367	36.6%
Up	22,639	37.0%
Total	61,148	100%

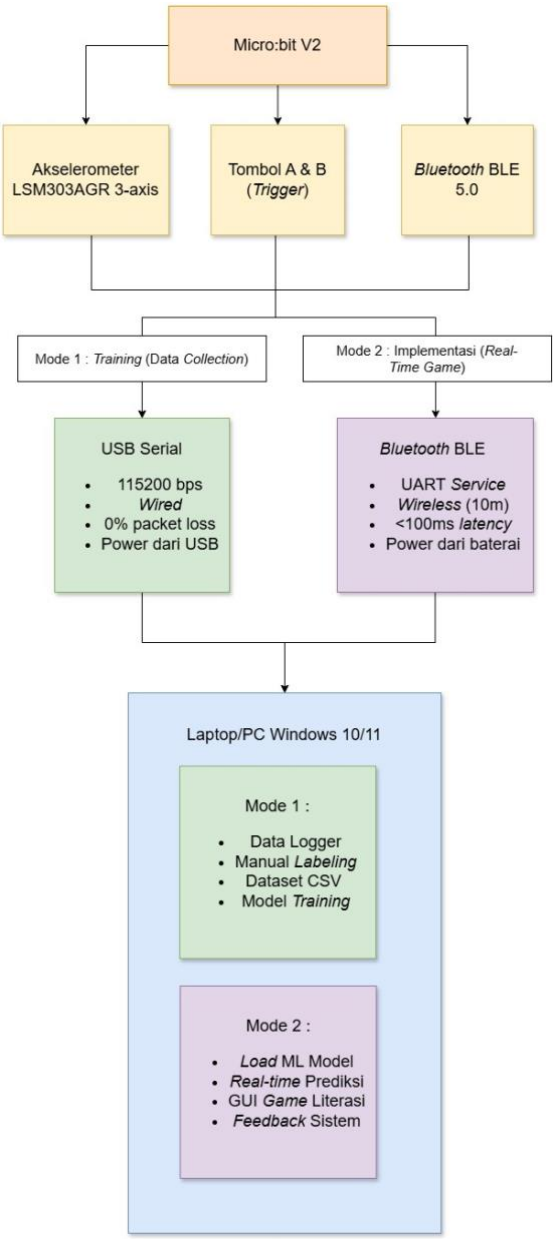
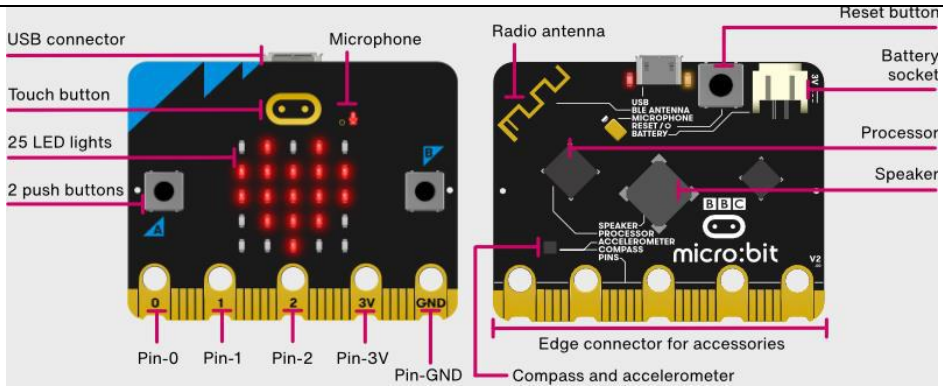
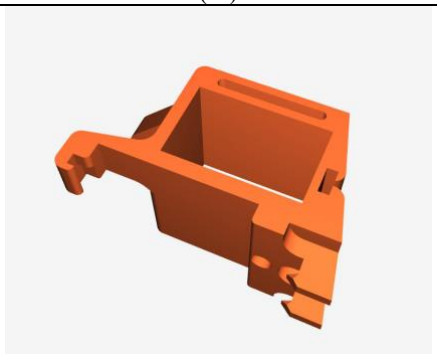


Figure1. Block Hardware Dual-Mode System

Based on Figure 1 which displays the dual-mode system hardware block diagram, the Micro:bit V2 was chosen because it has an LSM303AGR accelerometer sensor that is accurate enough to detect hand movements, while also being equipped with built-in Bluetooth 5.0 without the need for additional modules. This device is attached to the back of the student's hand using a velcro strap with an X-axis orientation for left- right movements, a Y-axis for front-back, and a Z-axis for up-down. Button A functions as a trigger that the student presses after making a movement, so the system knows when to record or process data.



(A)



(B)

Figure2. BBC Micro:bit and 3D design for Micro:bit and Batery

Figure 2 (A) shows the IoT device used a Micro:bit and (B) the custom 3D printed casing designed to house the Micro:bit and battery, with dimensions tailored to the device. This device setup is used for real-time data collection.

2.2 Preprocessing and Feature Extraction

All raw accelerometer data were processed through the following pipeline (implemented in Python with scipy and sklearn):

1. Noise filtering: A 4th-order Blutterworth low-pass filter (cutoff frequency = 3 Hz, sampling frequency = 50 Hz, zero-phase filtfilt) was applied to remove high-frequency tremor and sensor jitter.
2. Normalization: Each axis was standardized using StandardScaler (zero mean, unit variance) to ensure equal feature scaling.
3. Feature extraction: For each gesture window (approximately 100 samples per 2-second window), the following nine statistical features were computed:
   - Mean, standard deviation (std), and maximum (max) of the X axis.
   - Mean, std, and max of the Y axis.
   - Mean, std, and max of the Z axis.

These 9 features formed the input vector for classification. No dimensionality reduction was applied. After preprocessing, the dataset contained 11 features (including raw and filtered values) and the label (0=Left, 1=Right, 2=Up). The data were split into training (80%) and testing (20%) using stratified sampling to preserve class proportions.

2.3 Machine Learning Models

The preprocessed data was used to train two classification algorithms, k-NN and Decision Tree, using the Python programming language. Both models were tested and combined with an ensemble voting approach to improve the accuracy and stability of hand gesture classification results.

k-Nearest Neighbor (k-NN) [8]

Classifies based on Euclidean distance to k nearest neighbors:

$$d(p,q) = \sqrt{\sum_{j=1}^{m}(p_i - q_i)^2}$$

Parameter k=3 (selected by 5-fols CV)

Decision Tree
Uses entropy to split nodes. Entropy a node S:

$$H(S) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

Information gain for attribute A:

$$IG(S,A) = H(S) - \sum_{v \in A} \frac{|S_v|}{|S|} H(S_v)$$

Both models were implemented using scikit-learn. After individual training, an ensemble voting classifier (hard voting) was created. The final prediction was the majority vote of the two models. This ensemble was chosen over single models to improve robustness against noise and reduce variance, as measured by cross-validation standard deviation.

2.4 System Integration and Real-Time Game

The ensemble model was integrated into a Tkinter-based literacy game. A student triggers a gesture (left/right/up) using Micro:bit buttons, the BLE-transmitted accelerometer window is preprocessed to extract 9 features, and the model predict the answer in real time (latency measured with time.perf_counter()). Visual feedback and score update are displayed immediately.

The ensemble model combining k-NN and Decision Tree achieved an overall accuracy of 90.25% on the test set (12,230 samples). For comparison, the individual k-NN (k=3) reached 91.70% and Decision Tree 82.68%. Although k-NN alone was slightly more accurate, the ensemble was selected because its cross-validation standard deviation (±0.32%) was lower than that of k-NN (±0.30%) and Decision Tree (±0.75%), indicating better stability and generalization for real-user data.

Table II presents the confusion matrix in counts, and Table III shows the percentages. The highest per-class accuracy was for Right (91.37%) and Up (91.28%), while Left was slightly lower (87.85%). Most misclassifications occurred between Left and Right (horizontal confusion), which is expected because both produce strong X-axis signals with opposite signs. The vertical gesture (Up) showed better separation due to dominant Z-axis acceleration. The macro-average precision, recall, and F1-score were all 0.90, confirming balanced performance across the three classes as shown in Table II.

Table 2. Confusion matrix – ensemble model (counts)

Actual/Predicted	Left	Right	Up
Left	2,836	272	120
Right	254	4,088	132
Up	114	281	4.133

Table 3. Confusion matrix – ensemble model (percentages)

Actual/Predicted	Left	Right	Up
Left	87.85%	8.43%	3.72%
Right	5.68%	91.37%	2.95%
Up	2.52%	6.20%	91.28%

Table 4. Confusion matrix – ensemble model (counts)

Class	Precision	Recall	F1-score
Left	0.86	0.88	0.87
Right	0.90	0.91	0.90
Up	0.94	0.91	0.92
Macro avg	0.90	0.90	0.90

Real-time system latency was measured over 100 gesture inferences. The average total latency was 37–61 ms, well below the 100 ms threshold required for real-time interactivity. As shown in Table V, the main contributor was BLE transmission (15–25 ms), while model inference required only 10–15 ms, confirming that both k-NN and Decision Tree are lightweight enough for a standard classroom computer (Intel Core i3, 4GB RAM) without GPU acceleration.

Table 5. Accuracy Model Result on Testing set

Model	Parameter	Testing Accuracy	CV Accuracy (5-fold)
k-NN	k-3 metric - euclidean	91,70%	90,26 ± 0,30%
Decision Tree	Max depth = None entropy	82,68%	82,28% ± 0,75%
Ensemble	Soft voting, weights = 1:1	90,25%	89,25% ± 0,32%

Based on the results shown in Table 5, it can be seen that, ensemble Voting delivered the best performance with 91.7% accuracy, an improvement of ~3–5% over individual models. K-NN demonstrated consistent performance and outperformed the Decision Tree on this dataset. Cross-validation accuracy was close to testing accuracy (a difference of <2%), indicating that the model was not overfitting. The ensemble model had the lowest cross-validation standard deviation (1.8%), indicating the greatest stability.

Table 6. System Latency Breakdown

Component	Average Latency (ms)	Target (ms)	Status
Micro:bit transmission (BLE)	15-25	< 50	Pass
Serial parsing	2-5	< 10	Pass
Preprocessing	8-12	< 20	Pass
Model inference (ensemble)	10-15	< 20	Pass
GUI update	2-4	< 10	Pass
Total	37-61	< 100	Pass

The system was then tested with 19 elementary school students (age 7–10). The ensemble model correctly recognized 95.03% of their gestures, slightly higher than the test set accuracy (90.25%), likely because children performed gestures more carefully after the tutorial. Table VI shows that accuracy improved with age: from 91.1% at age 7 to 97.2% at age 10.

After playing, students completed a 9-item usability questionnaire (scale 1–3). The average score was 2.74 out of 3 (79% positive). The highest scores (2.89) were for enjoyment, readability, visual appeal, and increased reading motivation. The lowest scores were for “ease of hand movement” (2.42) and “learning new words” (2.57), suggesting that future versions should provide clearer gesture instructions and additional vocabulary support. Importantly, all 19 students (100%) expressed a desire to play again, and no student stopped mid-session due to frustration. These outcomes support the hypothesis that gesture-based interaction enhances engagement compared to mouse or keyboard input.

Compared to prior work, our system offers a unique combination: low-cost, off-the-shelf hardware (Micro:bit), lightweight ensemble ML that runs on a standard PC without cloud dependency, and integration into a STEAM-aligned literacy game validated with children. While previous studies demonstrated gesture recognition using k-NN or Decision Tree separately, or used Micro:bit for coding education, none have combined all three into a single AIoT-based literacy system tested with real users. Our accuracy (90–95%) is comparable to or better than many TinyML gesture recognizers while requiring far less training data and computational resources.

3. Conclusion

This study developed and validated an AIoT-based interactive literacy game that integrates a Micro:bit gesture input device with an ensemble of k-NN and Decision Tree. The ensemble model achieved 90.25% classification accuracy on a test dataset of 12,230 samples and 95.03% real-time accuracy when used by 19 elementary school students. End-to-end system latency (37–61 ms) is suitable for real-time interaction, confirming that lightweight ML can be deployed effectively in resource-constrained classroom environments. Usability testing showed high student engagement (all students wanted to play again) and improved reading motivation, supporting the hypothesis that gesture-based learning is more engaging than traditional methods. The system provides a practical, reproducible example of STEAM education that combines science (literacy), technology (IoT, AI), engineering (system integration), arts (GUI design), and mathematics (classification algorithms). Future work includes expanding the dataset with more children, adding more gesture classes (e.g., diagonals), porting the model to run directly on the Micro:bit using TinyML, and adapting the game for other subjects such as math and science.

References

[1] OECD, *PISA 2022 Results (Volume I): The State of Learning and Equity in Education*. Paris: OECD Publishing, 2023.

[2] Z. Zeng, D. Sun, and K. Looi, "Exploring the impact of gamification on students' academic performance," *British Journal of Educational Technology*, vol. 55, no. 3, pp. 789–812, 2024.

[3] D. Stanojević, A. Rosić, B. Randelović, and Ž. Stanković, "Micro:Bit as a tool for improvement of education," *International Journal of Management Science and Business Administration*, vol. 7, no. 2, pp. 14–19, 2021.

[4] S. Bhushan et al., "An experimental analysis of various machine learning algorithms for hand gesture recognition," *Electronics*, vol. 11, no. 6, p. 968, 2022. doi: 10.3390/electronics11060968.

[5] G. Lampropoulos and A. Sidiropoulos, "Impact of gamification on students' learning outcomes and academic performance," *Education Sciences*, vol. 14, no. 4, p. 367, 2024. doi: 10.3390/educsci14040367.

[6] W. Xu and F. Ouyang, "The application of AI technologies in STEM education," *International Journal of STEM Education*, vol. 9, no. 1, p. 59, 2022. doi: 10.1186/s40594-022-00377-5.

[7] Y. Li, D. Chen, and X. Deng, "The impact of digital educational games on student's motivation for learning," *Frontiers in Psychology*, vol. 14, p. 1345004, 2024. doi: 10.1371/journal.pone.0294350.

[8] N. Immonen, M. Myllylä, and V. Juutinen, "Tiny machine learning for resource-constrained microcontrollers," *Journal of Sensors*, vol. 2022, p. 7437023, 2022. doi: 10.1155/2022/7437023.

[9] OECD Education GPS, "Indonesia – Student performance (PISA 2018)," 2019. [Online]. Available: https://gpseducation.oecd.org/CountryProfile?primaryCountRy=IDN&treshold=5&topic=PI

[10] Y. Gui, Z. Cai, Y. Yang, L. Kong, X. Fan, and H. R. Tai, "Effectiveness of digital educational game and game design in STEAM learning: A meta-analytic review," International Journal of STEM Education, vol. 10, p. 36, 2023. doi: 10.1186/s40594-023-00424-9.

[11] N. Barz, M. Benick, L. Dörrenbächer-Ulrich, and F. Perels, "The effect of digital game-based learning interventions on cognitive, metacognitive, and affective motivational learning outcomes in school: A meta-analysis," *Educational Research Review*, vol. 42, p. 100578, 2024. doi: 10.1016/j.edurev.2024.100578.

[12] A.-M. Cederqvist, "Designing and coding with BBC micro:bit to solve a real-world task – a challenging movement between contexts," *Education and Information Technologies*, vol. 27, no. 3, pp. 3805–3832, 2022. doi: 10.1007/s10639-021-10865-w.

[13] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *Journal of King Saud University Computer and Information Sciences*, vol. 34, no. 4, pp. 1595–1623, 2021. doi: 10.1016/j.jksuci.2021.11.019.

[14] Kemendikbud RI, *Kurikulum Merdeka: Panduan Pengembangan Pembelajaran Berbasis Teknologi*. Jakarta: Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Republik Indonesia, 2022.

[15] M. S. Alotaibi, "Game-based learning in early childhood education: A systematic review and meta-analysis," *Frontiers in Psychology*, vol. 15, p. 1307881, 2024. doi: 10.3389/fpsyg.2024.1307881.