

A Cost-effective, Stand-alone, and Real-time TinyML-Based Gait Diagnosis Unit Aimed at Lower-limb Robotic Prostheses and Exoskeletons

Zarin Anjum Madhiha
Dept. of EEE, Brac University
Dhaka 1212, Bangladesh
zarin.anjum.madhiha@g.bracu.ac.bd

Antar Mazumder
Dept. of MTE, RUET
Rajshahi 6204, Bangladesh
antar.mte@ieee.org

Sohani Munteha Hiam
Dept. of EEE, RUET
Rajshahi 6204, Bangladesh
sohani@eee.ruet.ac.bd

Abstract—Robotic prostheses and exoskeletons can do wonders compared to their non-robotic counterpart. However, in a cost-soaring world where 1 in every 10 patients has access to normal medical prostheses, access to advanced ones is, unfortunately, extremely limited especially due to their high cost, a significant portion of which is contributed to by the diagnosis and controlling units. However, affordability is often not a major concern for developing such devices as with cost reduction, performance is also found to be deducted due to the cost vs. performance trade-off. Considering the gravity of such circumstances, the goal of this research was to propose an affordable wearable real-time gait diagnosis unit (GDU) aimed at robotic prostheses and exoskeletons. As a proof of concept, it has also developed the GDU prototype which leveraged TinyML to run two parallel quantized int8 models into an ESP32 NodeMCU development board (7.30 USD) to effectively classify five gait scenarios (idle, walk, run, hopping, and skip) and generate an anomaly score based on acceleration data received from two attached IMUs. The developed wearable gait diagnosis stand-alone unit could be fitted to any prosthesis or exoskeleton and could effectively classify the gait scenarios with an overall accuracy of 92% and provide anomaly scores within 95–96 ms with only 3 seconds of gait data in real-time.

Index Terms—Rehabilitation Robotics, Robotic Prostheses, Affordable Robotic Prostheses, Robotic Exoskeletons, Human Gait Classification, TinyML, Embedded AI

I. INTRODUCTION

From birth defects to unexpected accidents, since the dawn of the human race, people have been consistently exposed to the unforgiving consequences of losing their limbs leaving them incapable of standard human mobility. Summarized in figure 1, a 2017 global estimate by the World Health Organization (WHO) stated that up to 40 million people were reported to have been in need of medical prostheses or assistive devices [1], [2] whereas, only 1 in every 10 people could access them due to their high cost and other impediments [1]. Although there is no strong numerical data dedicated to robotic prostheses and exoskeletons expressing such disparity, it is, for certain, much more severe as they are significantly costlier compared to their non-robotic counterpart. A major reason behind such high cost is the cost of the processing units

leveraged for diagnosing the user gaits for any anomalies, and also for controlling the device. Although the cost is the primary impediment to making robotic prostheses accessible, there is an obvious lack of research focus on developing affordable robotic prostheses compared to the whole research domain dedicated to robotic prostheses. A common reason behind such a tendency is the trade-off between cost and performance as to alleviate the cost, less sophisticated equipment is to be used which is not adequate in a sensitive application such as medical prostheses where precision and performance are primary priorities. To address the research gap, this study proposed a wearable cost-effective (7.30 USD) stand-alone GDU based on TinyML. The quantized int8 Artificial Neural Network (ANN) model could successfully classify five types of gait scenarios -*Idle*, *Walk*, *Run*, *Hopping*, and *Skip* with an overall classification accuracy of 92%. Additionally, the device ran a K-means clustering model in parallel that generated an anomaly score for any unexpected movement scenarios. As a proof of concept, the wearable GDU was integrated with a lower-limb exoskeleton prototype where it provided inference within 95-96 ms after taking just 3 seconds of gait data.

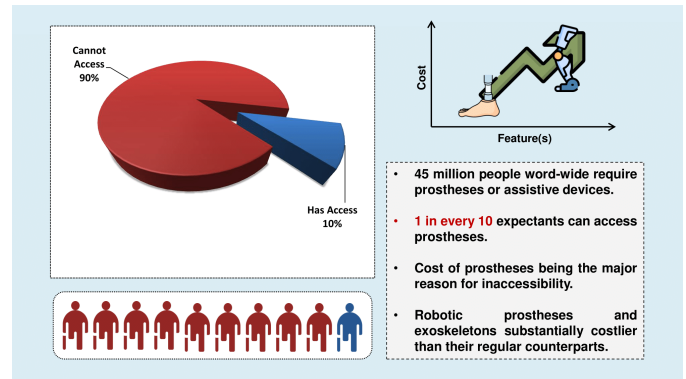


Fig. 1. Disparity in accessibility to prostheses and assistive devices: Demand vs. Access and the major catalyst.

This research received no specific grant from any funding agency.

II. RELATED WORKS

Gait classification has predominantly relied on swing phase analysis [3], where accelerometer-based IMU data are processed as time series. LSTM networks are frequently employed for such analyses. Tan *et al.* trained a modified LSTM on wearable accelerometer data to detect Heel-strikes and Toe-offs in running and walking, outperforming six benchmark models with a median F1-score of 0.98 [4], though without hardware validation. Similarly, Zhen *et al.* used an LSTM-DNN to distinguish stance and swing phases, achieving 91.8% accuracy, but relied on Bluetooth data transfer to a host computer [5]. Lee *et al.* demonstrated LSTM with two 9-axis IMUs, achieving mean absolute error below 0.01 across walking speeds, though requiring expensive CPU/GPU processing [6]. Hong *et al.* proposed a piecewise label-based LSTM for improved accuracy and speed adaptability [7], and later combined torso kinematics with ANN+LSTM/Bi-LSTM layers, yielding MSE of $5.34E^{-4} \pm 7.11E^{-4}$ in mid-stance estimation [8].

Beyond LSTM, other architectures have been applied. DCNNs with seven IMUs classified five gait phases at $\sim 97\%$ accuracy by constructing spatial input patterns [9], though without real-time implementation. Gong *et al.* achieved 98.03% accuracy in six locomotion modes using a BPNN at 0.9 ms inference latency, but relied on costly cRIO hardware [10]. Figueiredo *et al.* deployed seven MPU6050 IMUs for real-time kinematic analysis, outperforming MVN BIOMECH in stairs, with robust $R^2 > 0.90$ for the knee joint, though hardware compactness remained an issue [11]. Li *et al.* attached three IR sensors, achieving 99.62% detection, but average latency reached 34 ms and IR interference was unaddressed [12]. Wu *et al.* proposed a GCN model achieving 97.43% accuracy, surpassing LSTM/DCNN baselines (70–80%), though requiring expensive hardware [13]. Choi *et al.* introduced a Gait Phase Estimation Module (GPEM) with real-time processing, reducing deviations by $\sim 48\%$, yet lacking offline capability [14]. CNN-LSTM-CBAM networks reached $R^2 = 0.9221$ in wearable exosuits [15], while Zhang *et al.* employed LSTM on a Teensy 4.1 microcontroller with 94.60% accuracy, though at higher development cost [16].

III. METHODOLOGY

There are two outputs to our approach- classified gait scenario and anomaly score. Gait scenarios comprise 5 different classes - walk, skip, run, idle, and hopping. Typically, most approaches recognise gait phases to detect gait anomalies. This study, on the other hand, recognises normal gait scenarios and then assigns an anomaly score to scenarios that deviate from the normal. The methodology adopted in this research can be divided into four phases - 1) *development of the gait diagnosis unit(GDU)*, 2) *data collection and processing*, 3) *training of a modified CNN+K-means model*, and 4) *deployment in real-time gait diagnosis*.

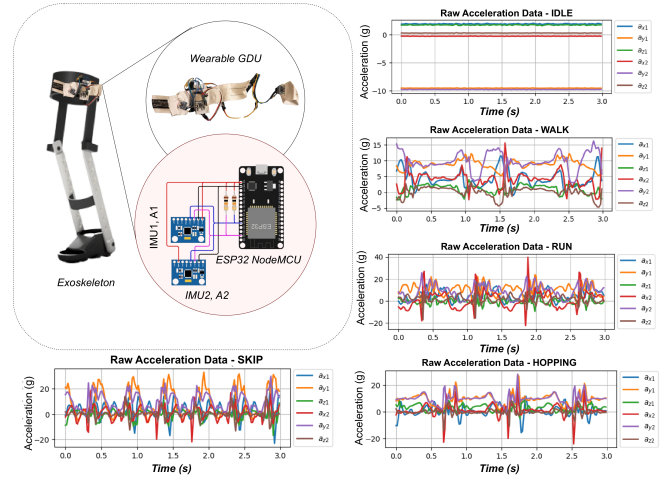


Fig. 2. The GDU physical form and schematics and raw acceleration data from A1 and A2 for the five gait scenarios considered.

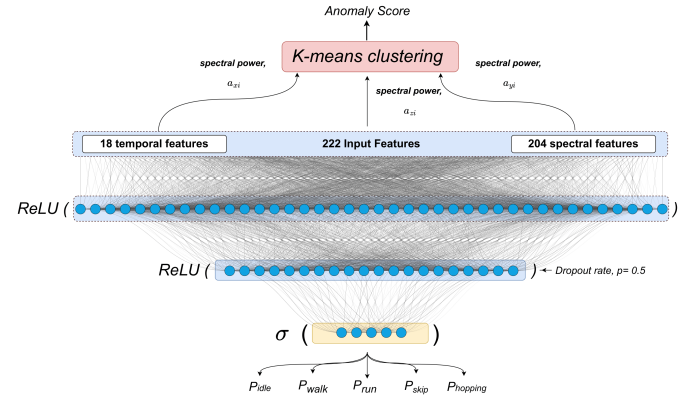


Fig. 3. The parallel models leverage ANN to classify gait scenarios and K-means clustering to throw an anomaly score.

A. Development of GDU

The GDU, as demonstrated in figure 2, is a wearable unit with two inexpensive six-axis MPU6050 accelerometers attached to an ESP32S 30P NodeMCU development board via two 1k Ohm pull-up resistors. The two accelerometers are used to send acceleration data to the MCU over I2C connections with addresses 0x68 and 0x69 respectively. One of the accelerometers, A1 is strapped to the mid-thigh segment of the user while the other one, A2 is attached to the ankle near the shank. These two have been found among the most optimal positions for gait classification [17]. The wearable unit can be placed on any exoskeleton or prosthesis due to its modular design. In this study, it was strapped to an aluminium-alloy exoskeleton prototype where a servo motor was utilised and controlled by the same MCU using PWM signals. The primary emphasis of this study was the development of an affordable GDU to minimize processing unit costs. Thus, *it is imperative to acknowledge that the exoskeleton used in this study was an arbitrary choice.*

B. Data Collection and Preprocessing

Each of the accelerometers provided three acceleration values along the axes X, Y, and Z in 59 Hz. This resulted in the six primary inputs, which were labelled as $a_{x1}, a_{y1}, a_{z1}, a_{x2}, a_{y2},$ and a_{z2} in units of g, where $1 \text{ g} = 9.8 \text{ ms}^{-2}$. These six parameters were recorded for all five gait scenarios among five volunteers between the ages of 21 to 55, two of whom were above the age of 30. Each candidate was instructed to perform the five gait scenarios with three minutes allotted to each gait scenario with varying speeds, resulting in 15 minutes of gait data for a single user for five gait scenarios. For example, figure 2 shows the raw acceleration pattern differences among different gait scenarios, where it is evident that some had more spikes, especially along the Z axis between the same time-frame. The gait data was fed into the edge impulse CLI framework via the data forwarder [18]. For each axis for both A1 and A2, a 3-second window was considered to extract acceleration values along the time axis. The total number of data instances or points for each sample can be expressed as in equation 1, where N_T is the total number of data points for each sample, w is the size of the window, c is the number of acceleration axes, and f is the frequency in Hz. Thus, for the 6 axes with a frequency of 59 Hz, a total of 1062 data points were collected for each gait scenario sample or 177 data points for each axis.

$$N_T = w \times c \times f \quad (1)$$

$$\bar{a}_{ki} = \frac{1}{N} \sum_{n=1}^N a_{ki}(n) \quad (2)$$

$$a_{ki}(n) = a_{ki}(n) - \bar{a}_{ki} \quad (3)$$

However, the raw data spikes as exhibited in figure ?? were inconsistent and also the influence of gravity had to be neutralised. So first, the mean value for each axis was subtracted, as expressed in equation 2 and equation 3, from individual data points to center the charts around zero-g. Here $a_{ki}(n)$ is the acceleration in g at the data point, n with k as the axis (x,y,z) and i as the IMU number(1,2) for each sample. N is the total number of data points for each axis i.e., $N = N_T/6$. $\bar{a}_{ki}$ is the mean acceleration for the gait scenario sample for all its data points in the 3-second window (i.e., $N=177$). Then a 6-order low-pass Butterworth filter with a cut-off frequency of 3 Hz was applied to get rid of the unreliable spikes in the higher frequency regions. The filtered signal determined 3 types of features for each sample which were RMS (equation 4), Kurtosis (equation 5), and Skewness [19] (equation 6). The kurtosis value used was Kurtosis excess [20] with 3 subtracted so that for normal distribution the Kurtosis value would be zero. Since these terms are used interchangeably in practice, this study will refer to the feature as Kurtosis for ease.

$$a_{ki}(RMS) = \sqrt{\frac{1}{N} \sum_{n=1}^N a_{ki}(n)^2} \quad (4)$$

$$a_{ki}(Kurtosis) = \frac{\frac{1}{N} \sum_{n=1}^N (a_{ki}(n) - \bar{a}_{ki})^4}{\left(\frac{1}{N} \sum_{n=1}^N (a_{ki}(n) - \bar{a}_{ki})^2\right)^2} - 3 \quad (5)$$

$$a_{ki}(Skewness) = \frac{\frac{1}{N} \sum_{n=1}^N (a_{ki}(n) - \bar{a}_{ki})^3}{\left(\frac{1}{N} \sum_{n=1}^N (a_{ki}(n) - \bar{a}_{ki})^2\right)^{3/2}} \quad (6)$$

3 types of spectral features were obtained for each sample Spectral Skewness (equation 8), Spectral Kurtosis [21] (equation 9), and Spectral Power [22] (equation 10). Here K represents the frequency of the selected bin. Fast Fourier Transform (FFT) [23] was performed (as in equation 7) in the signal $a_{ki}(n)$ of 177 samples. The length of FFT, N_L was 64, which implies within the frequency range of 0-59 Hz, 64 bins had been considered resulting in 64×3 or 192 bins for the 3-second window which then provided 192 spectral power [22](as in equation 10) values used as features.

$$A_{ki}(K) = \sum_{n=0}^{N-1} a_{ki}(n) \cdot e^{-j \frac{2\pi K n}{N}} \quad (7)$$

$$A_{ki}(Spectral \ Skewness) = \frac{\frac{1}{N} \sum_{k=1, i=1}^N |A_{ki}(K)|^3}{\left(\frac{1}{N} \sum_{k=1, i=1}^N |A_{ki}(K)|^2\right)^{3/2}} \quad (8)$$

$$A_{ki}(Spectral \ Kurtosis) = \frac{\frac{1}{N} \sum_{k=1, i=1}^N |A_{ki}(K)|^4}{\left(\frac{1}{N} \sum_{k=1, i=1}^N |A_{ki}(K)|^2\right)^2} - 3 \quad (9)$$

$$P_{ki}(Spectral \ Power) = \frac{1}{N_L \times f} |A_{ki}(K)|^2 \quad (10)$$

C. Model Architecture

As evident in figure 3, this approach leveraged two models in parallel - an ANN to classify gait scenarios and a K-means clustering model for generating an anomaly score. First, the ANN model comprises the input layer with 222 nodes representing the 222 inputs. It is a simple structure with two fully connected hidden layers with *ReLU* activation [24] and a dropout rate, $p=0.5$ applied to the second hidden layer. A Softmax activation [25] was applied to the output layer, which then provided the probability of the gait scenarios within a range (0,1). The shallow network was intended to reduce the computation load on the MCU adopted as it had limited resources. The training loop was executed for 500 epochs. After each forward pass, Categorical Cross Entropy (CCE) loss was calculated as equation 11 where y_i is the true label probability and $\hat{y}_i$ is the predicted label probability [26]. Adam optimizer [27] was leveraged to adjust the weights with a learning rate of 0.0005 for training the network.

$$\text{Categorical Cross Entropy} = - \sum_{i=1} y_i \cdot \log(\hat{y}_i) \quad (11)$$

Confusion Matrix in %						
True Labels	HOPPING	IDLE	RUN	SKIP	WALK	
	80.0	0.0	0.0	20.0	0.0	
	0.0	100.0	0.0	0.0	0.0	
	0.0	0.0	87.5	12.5	0.0	
	0.0	0.0	0.0	100.0	0.0	
	0.0	0.0	0.0	0.0	100.0	
		Predicted Labels				
		HOPPING	IDLE	RUN	SKIP	WALK

Fig. 4. Confusion matrix for the Neural Network-based classifier.

Simultaneously, four of the inputs (in this case, spectral powers: $P_{x1}, P_{y1}, P_{z1}, P_{z2}$) were taken and paired against each other along the X and Y axis denoted by the point (x_{test}, y_{test}) on the XY plane. The selection of these features was performed inherently by the Ege Impulse studio based on its feature importance analysis [28]. Now, based on which of the parameters were chosen, similar feature points with cluster size, N_c of 32 were considered in the XY plane. For example, if P_{x1} was taken along the X-axis and P_{z1} along the Y-axis, then 32 points, where the X-coordinates of the same belonged to P_{x1} and the Y-coordinates belonged to P_{z1} were considered. A_x and A_y represent these points of the cluster along the X and Y axes respectively. C_x (as in equation 12) and C_y (as in equation 13) are the X, Y coordinates of the centroid of the clustered points.

$$C_x = \frac{1}{N_c} \sum_{i=1}^{N_c} A_{xi} \quad (12)$$

$$C_y = \frac{1}{N_c} \sum_{i=1}^{N_c} A_{yi} \quad (13)$$

$$\text{Euclidean Distance (d)} = \sqrt{(C_x - x_{test})^2 + (C_y - y_{test})^2} \quad (14)$$

Next, the Euclidean distance [29] (equation 14) was calculated from the centroid (C_x, C_y) to the test point (x_{test}, y_{test}) . based on the combination of the four features, there were a minimum distance and a maximum distance, the average of these distances represented the anomaly score. In this study, a tolerance of 3 was initially considered as the threshold. Therefore, any gait sample producing a higher score was considered as an anomaly.

D. Deployment in Real-time Gait Diagnosis

The original model developed was a float32 model. However, such a model, although typically used in most applications, is not suitable to run in a resource-limited development

board such as the ESP32 NodeMCU. Therefore, the original model was first downgraded or quantized to an int8 profile [30] and then turned into an Arduino library based on c-array via the Edge Impulse studio framework. The quantization process reduced the number of bits of the weights and biases from 32-bit to 8-bit. Then using the library in the Arduino IDE environment, the model embedded with the necessary conditions was uploaded to the ESP32s NodeMCU. The GDU was attached to the designated areas of the lower limb on top of the exoskeleton prototype and the users were instructed to perform various gait scenarios. First, the output of the model as well as latency values were displayed using a serial monitor. Later the serial monitor was replaced with a 0.96" I2C OLED display. In the later cases, the latency information was excluded and only the gait classification information was displayed to utilise the small display size.

IV. RESULTS

TABLE I
PERFORMANCE METRICS FOR PRECISION, RECALL, AND F1 SCORE ACROSS DIFFERENT CLASSES.

Class	Precision	Recall	F1-Score	Support
HOPPING	1.00	0.80	0.89	10
IDLE	1.00	1.00	1.00	7
RUN	1.00	0.88	0.93	8
SKIP	0.70	1.00	0.82	7
WALK	1.00	1.00	1.00	7
Accuracy			0.92	39
Macro avg	0.94	0.93	0.93	39
Weighted avg	0.95	0.92	0.93	39
Latency Performance:				
Feature Extraction: 92 ms				
Classification (Prediction): 3 ms				
Anomaly Detection: 1 ms				

There were three criteria for result analysis- 1) Performance of the model (precision, recall, accuracy, accuracy, f1-score, and average anomaly score), 2) Cost vs. performance trade-off, and 3) Data acquisition and inference latencies. There was no specific performance parameter for the k-means-based anomaly detection as the anomaly threshold was arbitrary and subject to a variable for different users. Nevertheless, the max value of this threshold was kept below 3.0.

A. Performance Metrics

In Figure 4, the provided confusion matrix represents the performance of a classification model with five classes: Hopping, Idle, Run, Skip, and Walk. Each row represents the actual class, while each column represents the predicted class. The diagonal elements (from top-left to bottom-right) show that the majority of instances have been correctly classified, with a perfect 100% accuracy for the Idle, Skip, and Walk classes. However, the classes, "Hopping" and "Run" have been predicted as "Skip" in around 20% and 12.5% of instances respectively, indicating some misclassification between these three classes. Overall, the model demonstrates near-perfect performance for most classes, resulting in an overall accuracy of 92%. More details have been summed up in table I,

TABLE II
COMPARISON BASED ON COST VS PERFORMANCE TRADE-OFF [Note: The costs of the equipment used in developing GDUs were estimated based on the contemporary pricing of the year 2025. These prices are subject to change with time.]

Reference	Year	Gait Diagnosis Unit	Classification Accuracy (%)	Cost (USD)	Stand Alone
Zhen <i>et al.</i>	2019	PC/Work Station	91.8	500+	X
Gong <i>et al.</i>	2020	cRIO9082-based Controller	98.03	1,000+	✓
Su <i>et al.</i>	2020	Lenovo Thinkpad T470p	97	500+	X
Negi <i>et al.</i>	2021	Arduino Nano 33 BLE	97.8	40.50	✓
Isam <i>et al.</i>	2022	PC/Work Station	97.15	500+	X
Zhang <i>et al.</i>	2024	Teensy 4.1	94.60	94	X
This Study	—	ESP32s NodeMCU	92 (int8), 97.14 (float32)	7.30	✓

where it can be observed that all classes demonstrated high-performance metric values with a slight exception in the case of the "Hopping" and "Run" classes.

As for the K-means-based anomaly score, it was found that the average anomaly scores, especially the average highest anomaly scores varied across different age groups. There was a clear decline in the average anomaly score among the older age groups compared to the youth. The highest average maximum anomaly value was found to be around 2.38, which was well below the set threshold of 3.

B. Cost vs Performance Trade-off Comparison with Related Approaches

As evident from the table II, most similar approaches have leveraged expensive hardware to process gait data. These hardware components were either full-fledged desktop or laptop computers [5], [9], [31] that received gait data via Bluetooth or other sensors in real time. Otherwise, controllers with high computation abilities [10] were utilized for providing wearable stand-alone features. Despite having very high classification accuracy, these hardware components even came at over 1,000 USD, which was quite expensive. The closest competitor to our approach was found in [32] with a 97.8% classification accuracy, which leveraged an Arduino Nano33 BLE development board to run real-time inference. However, our approach did the same with only 5.8% less accuracy but almost at one-sixth of the cost. Furthermore, the float32 version of the model attained a similarly high accuracy score as the top-performing ones.

C. Latency Analysis

After the 3-second data collection window, latency has been distributed to three applications - feature generation, classification and anomaly score generation. From the serial monitor output taken from the Arduino IDE as also added in the table I, it can be observed that for our system took 92 ms for feature generation, 3 ms for gait scenario classification and 0-1 ms for generating the anomaly score. Thus, this approach could successfully detect gait scenarios along with gait anomalies within 95-96 ms after taking 3 sec of gait data.

V. DISCUSSION

The proposed method took a different approach, in which it recognized the most common gait scenarios instead of gait events, unlike most contemporary studies. It provided a simple

alternative to rigorous analysis based on gait events from which gait anomaly is typically derived. However, some limitations should be mentioned. The first apparent downside of the model is its slightly lower classification accuracy compared to the hardware-intensive processes, especially for the cases Hopping and Run. These activities show very similar high and rapid acceleration patterns, the difference of which becomes harder to distinguish when the model is quantized, losing information. Nevertheless, given that it was a quantized int8 model derived from a float32 variant to run on a resource-scare platform such as Esp32 NodeMCU with minimal latency, the accuracy reduction is understandable. There is also a visible low support resulting in class imbalance, but given this was a self-funded research with human experiment, it remained as a limitation due to financial and logistical constraints subject to future improvements. Another concern can be considered regarding the absence of an algorithmic threshold determination for the K-means-based anomaly score. However, as found in the results, different people from different age groups would required different thresholds as the youth were able to perform more frequent and accelerated movements. However, considering the threshold was determined by trial and error, an adaptive thresholding process would be subject to future studies with a bigger population sample.

Despite the shortcomings, given its stellar modularity, low training data requirement of only 15 minutes of gait data for each individual, and the feasibility to rapidly device custom GDUs for individual patients, the benefits far outweighed the limitations. Most importantly, with only a 4-6% accuracy reduction compared to the state-of-the-art models, the GDU diagnoses user gait at a fraction of the cost, which makes it a viable step towards affordable robotic prostheses and exoskeleton development.

VI. CONCLUSION

Affordability is a major concern for patients requiring advanced prostheses or exoskeletal systems. However, due to the cost vs. performance trade-off, it was observed to be a challenging task to develop high-accuracy gait diagnosis devices that are stand-alone, affordable and can perform in real-time. To address this issue, this research proposed and, as a proof of concept, developed a TinyML-based wearable GDU that can perform gait scenario classification with an overall 92% accuracy and generate a gait anomaly score based on only two IMUs and an inexpensive ESP32 NodeMCU development

board which exhibited potential solution at a fraction of the state-of-the-art cost margins. The authors fervently believe that despite the minor limitations, this is the right step towards making robotic prostheses and exoskeletons more available to those in need. In the future, it is intended to increase the classification accuracy even higher and run inference on a smaller data collection window. It is also intended to use the IoT features of the NodeMCU to seamlessly integrate the exoskeleton model into a digital twin.

ETHICS AND CONSENT

The five volunteers who participated in this experiment provided full verbal and written consent to contribute their gait data for the study. The research was approved by the Research & Extension Committee of Rajshahi University of Engineering & Technology, which was the ethical committee of the institute. No minor was part of this research.

REFERENCES

- [1] W. H. Organization, *Global vector control response 2017-2030*. Geneva: World Health Organization, 2017.
- [2] A. Chadwell, L. Diment, M. Micó-Amigo, D. Z. Morgado Ramírez, A. Dickinson, M. Granat, L. Kenney, S. Kheng, M. Sobuh, R. Ssekitoileko, *et al.*, "Technology for monitoring everyday prosthesis use: a systematic review," *Journal of neuroengineering and rehabilitation*, vol. 17, pp. 1–26, 2020.
- [3] U. Trivedi and A. Y. Joshi, "Advances in active knee brace technology: A review of gait analysis, actuation, and control applications," *Heliyon*, 2024.
- [4] H. X. Tan, N. N. Aung, J. Tian, M. C. H. Chua, and Y. O. Yang, "Time series classification using a modified lstm approach from accelerometer-based data: A comparative study for gait cycle detection," *Gait & posture*, vol. 74, pp. 128–134, 2019.
- [5] T. Zhen, L. Yan, and P. Yuan, "Walking gait phase detection based on acceleration signals using lstm-dnn algorithm," *Algorithms*, vol. 12, no. 12, p. 253, 2019.
- [6] J. Lee, W. Hong, and P. Hur, "Continuous gait phase estimation using lstm for robotic transfemoral prosthesis across walking speeds," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 1470–1477, 2021.
- [7] W. Hong, J. Lee, and P. Hur, "Piecewise linear labeling method for speed-adaptability enhancement in human gait phase estimation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, pp. 628–635, 2022.
- [8] W. Hong, J. Lee, and P. Hur, "Effect of torso kinematics on gait phase estimation at different walking speeds," *Frontiers in Neurobotics*, vol. 16, p. 807826, 2022.
- [9] B. Su, C. Smith, and E. Gutierrez Farewik, "Gait phase recognition using deep convolutional neural network with inertial measurement units," *Biosensors*, vol. 10, no. 9, p. 109, 2020.
- [10] C. Gong, D. Xu, Z. Zhou, N. Vitiello, and Q. Wang, "Bpnn-based real-time recognition of locomotion modes for an active pelvis orthosis with different assistive strategies," *International Journal of Humanoid Robotics*, vol. 17, no. 01, p. 2050004, 2020.
- [11] J. Figueiredo, S. P. Carvalho, J. P. Vilas-Boas, L. M. Gonçalves, J. C. Moreno, and C. P. Santos, "Wearable inertial sensor system towards daily human kinematic gait analysis: benchmarking analysis to mvn biomech," *Sensors*, vol. 20, no. 8, p. 2185, 2020.
- [12] C. Li, Y. He, T. Chen, X. Chen, and S. Tian, "Real-time gait event detection for a lower extremity exoskeleton robot by infrared distance sensors," *IEEE Sensors Journal*, vol. 21, no. 23, pp. 27116–27123, 2021.
- [13] X. Wu, Y. Yuan, X. Zhang, C. Wang, T. Xu, and D. Tao, "Gait phase classification for a lower limb exoskeleton system based on a graph convolutional network model," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 5, pp. 4999–5008, 2021.
- [14] S. Choi, C. Ko, and K. Kong, "Walking-speed-adaptive gait phase estimation for wearable robots," *Sensors*, vol. 23, no. 19, p. 8276, 2023.
- [15] F. Yang, C. Chen, Z. Wang, Y. Xu, Y. Liu, and X. Wu, "Real-time terrain recognition based on transformer and gait prediction based on cnn-lstm-attention for exosuit," in *2023 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 743–748, IEEE, 2023.
- [16] S. Zhang, J. Zhu, T.-H. Huang, S. Yu, J. S. Huang, I. Lopez-Sanchez, T. Devine, M. Abdelhady, M. Zheng, T. C. Bulea, *et al.*, "Actuator optimization and deep learning-based control of pediatric knee exoskeleton for community-based mobility assistance," *Mechatronics*, vol. 97, p. 103109, 2024.
- [17] S. K. Manna, M. H. B. Azhar, and A. Greace, "Optimal locations and computational frameworks of fsr and imu sensors for measuring gait abnormalities," *Heliyon*, 2023.
- [18] E. Impulse, "Edge impulse cli - edge impulse documentation," 2024.
- [19] J. Schulte, F. Policelli, and B. Zaitchik, "A waveform skewness index for measuring time series nonlinearity and its applications to the enso-indian monsoon relationship," *Nonlinear Processes in Geophysics*, vol. 29, no. 1, pp. 1–15, 2022.
- [20] J. M. Van Zyl and S. Van der Merwe, "An empirical study of the behaviour of the sample kurtosis in samples from symmetric stable distributions," *South African Statistical Journal*, vol. 54, no. 2, pp. 255–265, 2020.
- [21] J. Taylor, N. Denman, K. Bandura, P. Berger, K. Masui, A. Renard, I. Tretyakov, and K. Vanderlinde, "Spectral kurtosis-based rfi mitigation for chime," *Journal of Astronomical Instrumentation*, vol. 8, no. 01, p. 1940004, 2019.
- [22] S. V. Vaseghi, "Power spectrum and correlation," *Advanced Digital Signal Processing and Noise Reduction*, Saeed V. Vaseghi Copyright© 2000 John Wiley & Sons Ltd ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic), p. 263, 2000.
- [23] Y. Fu, Z. Gao, Y. Liu, A. Zhang, and X. Yin, "Actuator and sensor fault classification for wind turbine systems based on fast fourier transform and uncorrelated multi-linear principal component analysis techniques," *Processes*, vol. 8, no. 9, p. 1066, 2020.
- [24] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.
- [25] S. Sharma, S. Sharma, and A. Athaiya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, no. 12, pp. 310–316, 2017.
- [26] M. Martinez and R. Stiefelhagen, "Taming the cross entropy loss," in *German Conference on Pattern Recognition*, pp. 628–637, Springer, 2018.
- [27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [28] "Anomaly detection (k-means) edge impulse documentation." <https://docs.edgeimpulse.com/docs/edge-impulse-studio/learning-blocks/anomaly-detection>, 2024. Accessed: March 12, 2024.
- [29] A. Aldino, D. Darwis, A. Prastowo, and C. Sujana, "Implementation of k-means algorithm for clustering corn planting feasibility area in south lampung regency," in *Journal of Physics: Conference Series*, vol. 1751, p. 012038, IOP Publishing, 2021.
- [30] N. Schizas, A. Karras, C. Karras, and S. Sioutas, "Tinyml for ultra-low power ai and large scale iot deployments: A systematic review," *Future Internet*, vol. 14, no. 12, p. 363, 2022.
- [31] M. Islam, M. Tabassum, M. M. Nishat, F. Faisal, and M. S. Hasan, "Real-time clinical gait analysis and foot anomalies detection using pressure sensors and convolutional neural network," in *2022 7th International Conference on Business and Industrial Research (ICBIR)*, pp. 717–722, IEEE, 2022.
- [32] S. Negi and N. Sharma, "A standalone computing system to classify human foot movements using machine learning techniques for ankle-foot prosthesis control," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 25, no. 12, pp. 1370–1380, 2022.