

Feature-driven Approximate Computing for Wearable Health-Monitoring Systems

Authors

Email

University

City, State, Country

ABSTRACT

Real-time health monitoring systems generate a large volume of sensing data requiring tremendous processing time and storage space. Orthogonal to existing approximate computing mechanisms, this work proposes a feature-driven approximation (FDAPx) to address the pressing need for fast data processing and a limited storage budget in wearable health monitoring devices. The proposed FDAPx method reverses the features interested in the application to derive an approximation threshold for the purpose of retaining feature-critical information, rather than aimlessly saving or transmitting all raw data. Case studies in an insole sensing system for fall risk assessment show that FDAPx can reduce the data size by up to 87% over raw data and up to 85% over 2-bit precision reduction-based approximation. The approximation from FDAPx only results in up to a 2% deviation in swing time; in contrast, the approximation based on precision reduction causes a 30% deviation in the same gait feature.

CCS CONCEPTS

• **Hardware** → Design modules and hierarchy; • **Applied computing** → Health care information systems.

KEYWORDS

Approximate computing, wearable sensor, data processing

ACM Reference Format:

Authors. 2024. Feature-driven Approximate Computing for Wearable Health-Monitoring Systems. In *Proceedings of Proceedings of the Great Lakes Symposium on VLSI 2024 (GLSVLSI '24)*. ACM, Irvine, CA, USA, 6 pages. <https://doi.org/xxx>

1 INTRODUCTION

Health monitoring systems provide a large volume of data for remote diagnosis. For example, the REALDISP dataset [2] shows that a personal wearable sensor could generate 180,000 sampling data for 20ms. A wearable wrist device, which is Bluetooth low energy enabled, has the potential to generate 315TB data per year [9]. For a wearable device performing 24/7 monitoring, it is not affordable to store such a large amount of data in mobile devices or transmit

them to a cloud space within a limited power budget [21]. As Approximate Computing (AC) techniques trade accuracy for speed improvement, memory reduction, and energy efficiency in various data-intensive applications [15], approximate storage and computation are promising candidates for wearable sensing systems with a restricted budget on memory and processing power.

The existing approximation mechanisms are applied to circuit, architecture, algorithm, and software levels [19]. Limited work is available to discuss the approximation at the application level, particularly for health monitoring. Circuit-level techniques [4, 6, 11, 14] such as voltage scaling are not suitable for low-end edge devices for health monitoring due to their narrow voltage swing range. Precision reduction techniques at circuit, storage, and software could lead to the loss of critical information, which may be vital to diagnosing the targeted health problems. Instruction set modification [8, 13] is not affordable for low-end sensors and microcontrollers. Loop iterations [1] based approximation is only useful for computation-extensive applications, rather than for frequent storage elements. As existing approximation methods are not designed for application-oriented use, the deployed approximation either does not satisfy the minimum requirement on accuracy or does not achieve the desired approximation efficiency. For instance, an energy-efficient health monitoring system [7] approximates wavelet coefficients to reduce energy consumption; however, that system still requires processing large amounts of data before transmitting the data to the cloud service.

To address the limitations mentioned above, this work proposes a feature-driven approximation method for health monitoring systems. The main contributions of this work are as follows:

- The proposed feature-driven approximation (FDAPx) method reverses the features interested in the application to derive an approximation threshold for the purpose of retaining feature-critical information, rather than aimlessly saving or transmitting all raw data.
- To the best of our knowledge, this is the first work that implements a high-level approximation mechanism at the hardware level. We evaluate the approximation efficiency of precision reduction, approximate arithmetic, and application-level approximation in a health monitoring scenario.
- The success of the proposed FDAPx relaxes the heavy burden of sensing data transmission from sensing nodes to the cloud and releases a large memory space from the personal mobile device for temporal sensing data storage.

The rest of this work is organized as follows. Section II introduces the background of the health-monitoring application and highlights the key features of the sensing data of interest. Section III presents

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GLSVLSI '24, June 12–14, 2024, Tampa Bay Area, FL, USA

© 2024 Association for Computing Machinery.

ACM ISBN xxx...\$15.00

<https://doi.org/xxx>

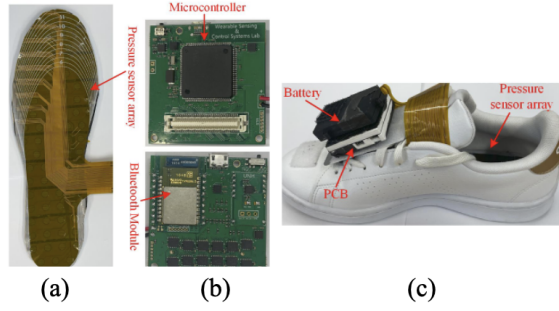


Figure 1: A wearable insole system for fall risk assessment. (a) The pressure sensor array, (b) the front and back side of the printed circuit board that integrates a microcontroller and a Bluetooth module, and (c) the entire sensing system.

the concept of feature-driven approximate computing and the implementation details. Section IV provides the assessment results. This work is concluded in Section V.

2 BACKGROUND AND PRELIMINARIES

2.1 Fall Risk Assessment via Wearable Insole

Falls among older adults are a serious and growing public health problem [17]. In 2018, over one in four reported at least one fall, leading to over 8 million injuries, 32,000 deaths, and \$50 billion in medical costs [3, 12]. Falls, with or without direct injuries, can lead to devastating consequences, including fear of fall, reduced mobility, functional decline, and loss of independence [16, 18]. To avoid the serious consequences of falls, early detection of fall risks has been recognized as an effective measure and recommended by the Centers for Disease Control and Prevention (CDC). To perform fall risk assessment, the work [5] introduces a wearable insole system, which mainly includes a flexible pressure sensor array shown in Fig. 1(a) and a printed circuit board (PCB) shown in Fig. 1(b). To enable a high resolution in the measurement of the ground reaction force (GRF), the flexible pressure sensor array has 96 pressure sensors uniformly distributed on the insole and is placed inside a normal shoe. The microcontroller on the PCB is responsible for collecting data from the pressure sensor array and transmitting the data to a smartphone application via the Bluetooth module. Figure 1(c) shows the overall sensing system in a shoe.

2.2 Key Features Interested in Fall Risk Monitoring

Fall risk assessment relies on the gait parameters extracted from insole sensing data. Among all those parameters, the fundamental Ground Reaction Force (GRF) measured by the insole system is used to derive other gait features, which characterize an individual's walking activity and indicate his/her fall risk. The unit of GRF is body weight (BW). The other key gait parameters we consider in this work are illustrated in Fig. 2. The definition of all gait parameters is summarized in Table 1. The gait cycle time is the sum of stance time and swing time. Figure 3 highlights the typical computation steps that we take to obtain the gait parameters for fall risk assessment. The first two steps *Reciprocal Computation (C)*

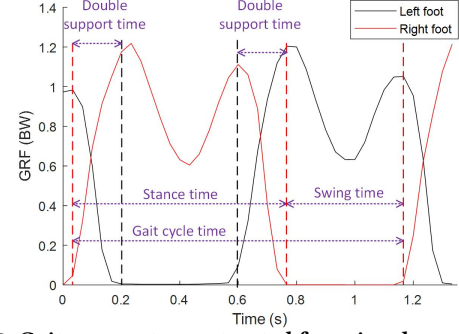


Figure 2: Gait parameters extracted from insole sensing data.

Table 1: Key Features for Proposed Feature-driven Approximate Computing.

No.	Parameter Name	Definition
1	GRF	Summation of the force measured by all the pressure sensors under foot.
2	Gait Cycle Time (GCT)	Time interval between two successive occurrences of one of the repetitive events of the same foot.
3	Stance Time (StT)	Time duration between the initial and last contact of a stride.
4	Swing Time (SwT)	Time duration between the last contact of one stride and the initial contact of the subsequent stride of the same foot.

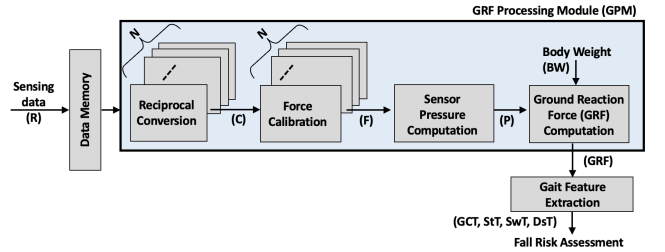


Figure 3: Computation flow for the gait parameters used in fall risk assessment.

and *Force Calibration (F)* are repeated N times, where N is the number of sensors in one insole. The reciprocal computation module converts sensor resistance to a conductance as expressed in Eq. (1).

$$C(i,j) = \frac{1}{R(i,j)} \quad (1)$$

where i and j stand for the sensor ID and sampling time index, respectively. The force applied to each sensor node $F(i,j)$ is equivalent to a conditionally weighted conductance, as expressed in Eq. (2).

$$F(i,j) = \begin{cases} 32.99 \cdot C(i,j), & \text{if } (C(i,j) \leq 0.02) \\ 56.51 \cdot C(i,j) - 0.4547, & \text{if } (0.02 < C(i,j) \leq 0.04) \\ 93.41 \cdot C(i,j) - 2.004, & \text{if } (C(i,j) > 0.04) \end{cases} \quad (2)$$

The coefficients in Eq. (2) are empirical values obtained from a large number of clinic measurement [5]. We sum up the calibrated forces $F(i,j)$ and subtract the offset noise, modeled as the minimum force $F(:,j)$ observed at time stamp j , to obtain the final pressure value P represented by Eq. (3).

$$P(j) = \sum_{i=1}^N F(i,j) - \min(F(:,j)) \quad (3)$$

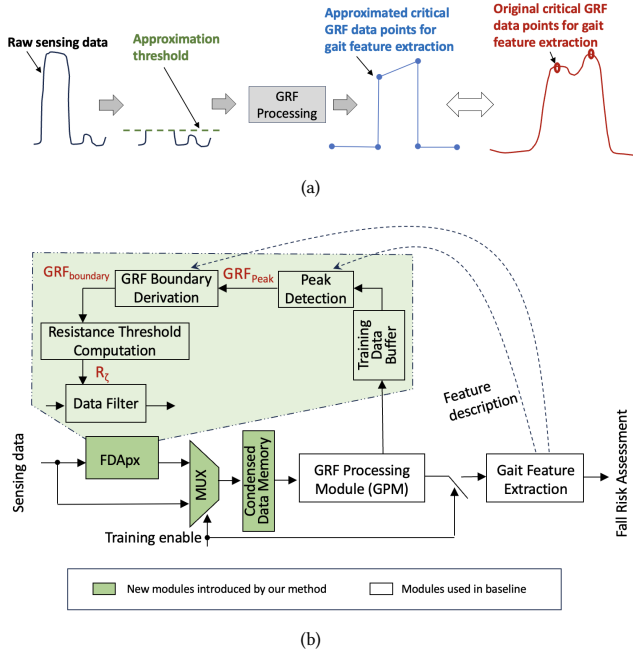


Figure 4: Overview of proposed FDAPx method. (a) Concept and (b) block diagram.

To normalize the final pressure, we define $GRF(j)$ for the sampling time stamp j as the ratio of P at that particular moment over the body weight BW . The expression is as shown in Eq. (4).

$$GRF(j) = \frac{P(j)}{BW} \quad (4)$$

3 PROPOSED FEATURE-DRIVEN APPROXIMATION (FDAPX)

3.1 Method Overview

Due to the real-time monitoring, the sensing data could quickly accumulate to a large volume, which is not favorable for storage and wireless communication in the real-time monitoring scenario. The ultimate goal of our proposed Feature-Driven Approximation (FDAPx) is to significantly improve the approximation degree without losing the accuracy of critical information. The core concept of the proposed method is to exploit the data features interested in applications to design the feature-driven approximation mechanism. As shown in Fig. 4(a), the peak values of the original GRF are critical for gait feature extraction and the rest of the data points do not contribute to gait analysis. Our method simplifies the raw data profile by applying a data filter with a customized approximation threshold. After the standard GRF processing, our method will only provide critical data points for storage and transmission without losing the information useful for gait feature extraction.

The approximation threshold for the sensing data filter is automatically generated by a hardware module FDAPx shown in Fig. 4(b). By using FDAPx, we can significantly reduce the size of data memory before the GRF processing module. Inside FDAPx, a buffer is used to save a small set of training data, which facilitates

determining the approximation threshold through the sequence of *Peak Detection*, *GRF Boundary Derivation* and *Resistance Threshold Computation*. As gait feature analysis relies to the peak value of GRF, the gait feature extraction module provides specific feature descriptions to the peak detection module and the GRF boundary derivation module.

3.2 Derivation of Approximation Threshold

3.2.1 GRF boundary obtained from training data set. The peak values in GRF are critical data points for gait feature extraction. In the training phase, it is possible to capture all peaks by employing peak detection techniques as illustrated in Fig. 4. $GRF_{boundary}$ can be calculated by finding the minimum value of the peak ground reaction force in a training period. Certainly, a longer training time will enable us to find a more stable $GRF_{boundary}$. However, as indicated in Fig. 5(a), the $GRF_{boundary}$ can be stabilized within 30 seconds in our insole sensing system.

3.2.2 Approximation Threshold. After rearranging Eq.(4), we can obtain the pressure $P(j)$ that satisfies the relation defined in Eq.(5).

$$\frac{P(j)}{BW} \geq GRF_{boundary} \quad (5)$$

Next, we replace $P(j)$ with Eq. (3) and use $F(i,j)$ within the range of $C(i,j)$ less than 0.2 to determine the maximum limit of $R(i,j)$ in Eq. (1). This substitution allows us to have the relationship between conductance $C(i,j)$ and $GRF_{boundary}$, as expressed in Eq. (6). The parameter $GRF_{boundary}$ is produced by a small set of training data recorded per insole user.

$$\sum_{i=1}^N 32.99 \cdot C(i,j) \geq GRF_{boundary} \cdot BW \quad (6)$$

If we assume that the body weight is uniformly distributed over all 96 pressure sensors in the sole, the summation of $C(i,j)$ can be converted to $N \cdot C(i,j)$. Thus, the approximation threshold R_{ζ} for our FDAPx can be represented in the form of the number of sensors N , body weight BW , and $GRF_{boundary}$, as expressed in Eq. (7).

$$R_{\zeta} = \frac{N \cdot 32.99}{GRF_{boundary} \cdot BW} \quad (7)$$

When $GRF_{boundaries}$ varies due to different training periods, the fluctuation of the approximation threshold R_{ζ} is minor. As shown in Fig. 5(b), the right foot R_{ζ} remains unchanged and R_{ζ} and the left foot R_{ζ} varies in a range of 247.85 to 255.02. Once the training period is determined, we use the approximation threshold R_{ζ} to filter out the sensing data that do not carry gait features using Eq. (8).

$$R_{FDAPx} \approx \begin{cases} R & (R \leq R_{\zeta}) \\ 0 & (R > R_{\zeta}) \end{cases} \quad (8)$$

3.3 Hardware Implementation of FDAPx

The proposed FDAPx has been implemented on a Kintex-7 FPGA board. Initially, we implemented the GRF Processing Module (GPM) to generate training data. The training data ($S_1, S_2, \dots, S_8$) go through a peak detection module as shown in Fig. 6. The peak detection module obtains training data as input in every clock cycle and stores it in three registers (previous, current, and next). These registers shift

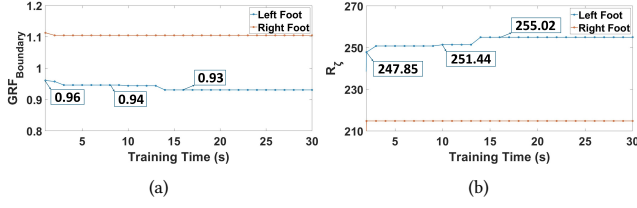


Figure 5: The variation of (a) $GRF_{boundaries}$ and (b) R_{ζ} in different lengths of training time.

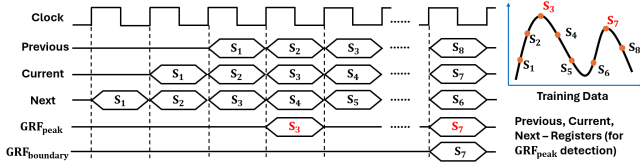


Figure 6: Timing diagram for the FDAPx hardware implementation.

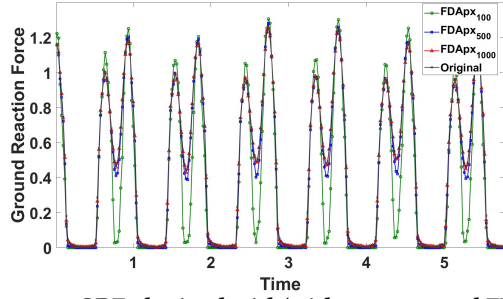
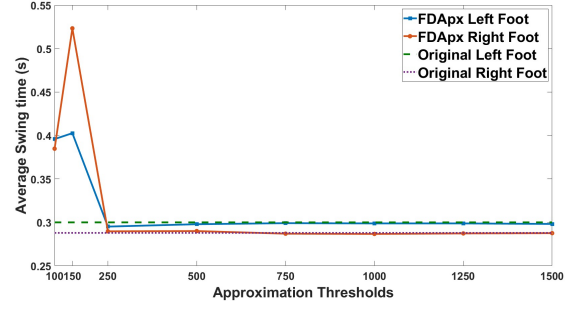


Figure 7: GRF obtained with/without proposed FDAPx.

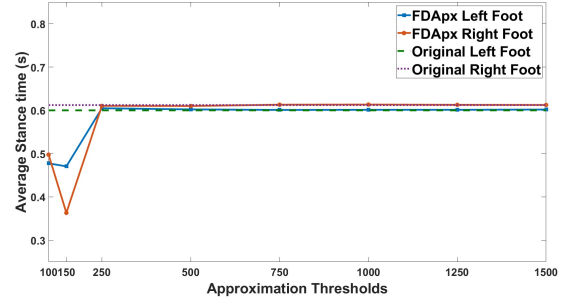
the data when new training data comes. Thus, the peak detection module compares three consecutive data points and detects the GRF_{peak} . For instance, we detected the peaks S_3 and S_7 as shown in Fig. 6. All the detected GRF_{peak} values proceed through a GRF boundary derivation block. $GRF_{boundary}$ block provides the minimum value of the detected peaks. Thus, the resistance threshold of our proposed method is calculated using the minimum value of all the detected peaks.

4 EXPERIMENTAL RESULTS

The raw sensing data in the following experiments were obtained from the insole system shown in Fig. 1. There were 96 pressure sensors in total for the foot force measurement. Each sensing trail contains 2995 samples per minute. All sensor data files were imported and fed to the prototype of the proposed approximation algorithms implemented in Verilog HDL and the Kintex-7 FPGA board. The hardware cost of all methods under comparison was reported by the FPGA design software Vivado 2023.1.



(a)



(b)

Figure 8: Impact of approximation threshold on extracted gait features (a) swing time and (b) stance time.

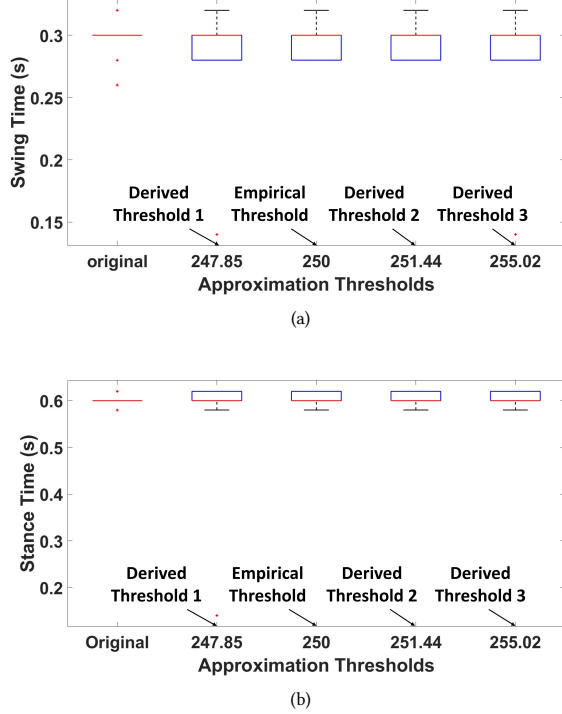
4.1 Impact of Approximation on Accuracy of Gait Feature Extraction

4.1.1 Relation Between Approximation Threshold and Accuracy in Extracted Gait Features. The proposed feature-driven approximation only retains the data that contribute to the features interested in the specific application. In the fall risk assessment, only the peak GRF values in each gait period are useful. Thus, we examine the accuracy loss of our FDAPx method by comparing the retrieved peak GRF values. To investigate the impact of the approximation threshold, we varied R_{ζ} with three numbers (100, 500, and 1000) in our GRF computation. As shown in Fig. 7, although the different approximation thresholds lead to slightly different peak GRF values, our FDAPx method captures all GRF peaks at the correct timing (as it is supposed to be in the baseline).

Next, we evaluate the gait features defined in Table 1. As shown in Fig. 8(a), the average swing time slightly decreases with the decreasing approximation threshold until 250. The accuracy loss in the average swing time is 0.4% - 1.6% compared with the baseline without approximation. However, if the approximation threshold is lower than 250, the average swing time will not reflect the true feature value and the accuracy loss increases to 31.96%. Figure 8(b) indicates that the FDAPx method only leads to a 0.3% loss in the accuracy of stance time compared to the baseline. We summarize the approximation-induced accuracy loss in Table 2. As can be seen, our FDAPx with an approximation threshold above 250 achieves over 98.38% accuracy in swing time, stance time, and gait cycle time. The worst-case accuracy drop on swing time is 1.62%. As the

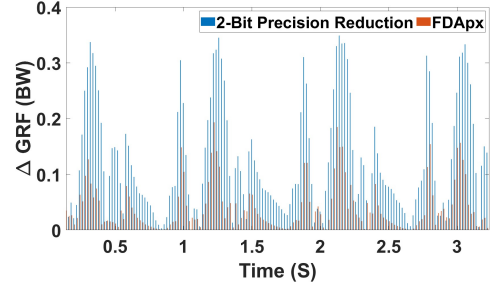
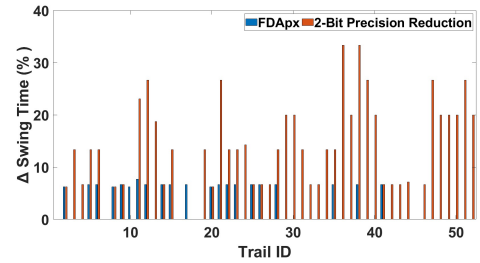
Table 2: Accuracy loss in gait parameters due to FDAPx

Features	Approximation Thresholds				
	100	250	500	750	1000
Swing Time	31.96%	1.62%	0.71%	0.30%	0.40%
Stance Time	18.71%	0.30%	0.35%	0.15%	0.20%
Gait Cycle Time	1.96%	0.00%	0.00%	0.00%	0.00%

**Figure 9: Impact of theoretical and empirical approximation thresholds on gait features (a) swing time and (b) stance time.**

approximation threshold reduces to 100, the accuracy of gait features computed based on our approximation method significantly decreases. This is because when the approximation threshold is set too low, too many $C(i,j)$ data points are filtered out. As a result, the corresponding gait features carried by $C(i,j)$ and GRF are sabotaged.

4.1.2 Comparison of Theoretical and Empirical Approximation Thresholds. Our simulation results indicate that the optimal approximation threshold R_{χ} is around 250. The theoretical analysis in Eq.(7) concludes the best range of R_{χ} is 247.85 and 255.02. We compare the swing and stance cycle time based on the empirical and theoretical R_{χ} in Fig. 9. As shown in Fig. 9(a), the variation of swing time is 6%. There is only a 3% deviation from the mean value in the stance time shown by Fig. 9(b). This result confirms that the approximation threshold derived by the proposed theoretical analysis is close to the empirical approximation threshold. As mentioned in Section 3.2, our training time is only a few ten seconds and thus it is affordable.

**Figure 10: Difference between ΔGRF yielded by using FDAPx and 2-bit precision reduction approximation methods.****Figure 11: Deviation in swing time after applying FDAPx and 2-bit precision reduction approximation methods.**

4.2 Comparison with Other Approximation Methods

In this subsection, we compare the baseline system [5], precision-reduction-based approximation[10] and the proposed FDAPx in terms of accuracy of extracted gait parameters, approximation efficiency, and hardware cost implemented on FPGAs.

4.2.1 Accuracy of Gait Features. Approximate computing may lead to information loss. In the fall risk assessment, approximation will result in the deviation of GRF from the baseline without approximation. We used the metric defined in Eq. (9) to evaluate the impact of approximation on the accuracy of extracted gait features.

$$\Delta GRF = |GRF_{\text{without approximation}} - GRF_{\text{with approximation}}| \quad (9)$$

Figure 10 shows ΔGRF processed by two different approximation methods. Only 10.68% of ΔGRF achieved by our proposed FDAPx method exceeds 0.1; in contrast, the approximation method that ignores the two least important bits leads to 76.52% of ΔGRF being greater than 0.1. The variance in GRF affects the swing time assessed in the gait analysis. The multiple trail results shown in Fig. 11 indicate that, on average, the proposed FDAPx only leads to 0% - 5% change in the swing time. In contrast, the precision reduction-based approximation could cause a 5% - 35% increase/decrease in the swing time. In summary, our FDAPx is 30% more reliable than a precision reduction-based approximation.

4.2.2 Approximation Efficiency. Approximation computing enables us to lower the need for data storage and transmission. The application of different approximation thresholds in the proposed FDAPx achieves different amounts of reduction (i.e., approximation

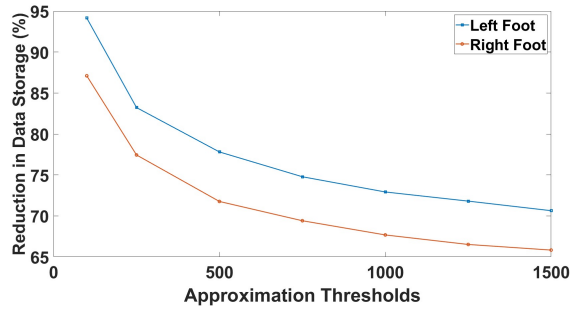


Figure 12: Reduction in data storage for different approximation thresholds in the proposed FDAPx method.

Table 3: Hardware utilization for the FPGA prototype of different approximation methods.

Approximation Method	FPGA Resource				Critical Path Delay (ns)
	LUTs	Register	DSPs	BRAM	
Baseline [5]	0.55%	0.12%	6.07%	71.91%	25.38
Apx. Adder [20]	0.55%	0.12%	6.07%	71.91%	25.73
Precision Reduction [10]	0.49%	0.11%	5.83%	69.66%	25.92
Proposed FDAPx	0.90%	0.24%	13.45%	14.38%	32.57

efficiency) in storage space or transmission bandwidth. As shown in Fig. 12, when the approximation threshold decreases from 1500 to 100, the approximation efficiency of our FDAPx increases to 87% (95%) for the processing of right (left) foot sensing data analysis. In contrast, for the same application with a resolution of 64-bit data, the 2-bit precision reduction-based approximation will only achieve the approximation efficiency of 3.125% ($=2/64$).

4.2.3 FPGA Cost. To support the stable extraction of gait features, it is necessary to collect sensing data for a period of 30 seconds and send those data as a batch to the GRF processing module. Without the approximation mechanism before the GRF processing module, that batch of data requires 71.91% of BRAM blocks in the Kintex-7 FPGA, which is shown in Table 3. Our FDAPx exploits the prior knowledge on the gait features to filter out the non-critical data, reducing the need for BRAM to 14.38%. Although our proposed method has a slight increase in the lookup tables (LUT), registers, and DSPs, the significant reduction in BRAM outweighs the minor increase in other FPGA resources. Due to the approximation before the GRF processing module, the critical delay of FDAPx is about 7ns longer than the baseline and other approximation methods. This limitation can be mitigated by applying a pipeline design in the critical propagation path.

5 CONCLUSION

Feature-driven approximation (FDAPx) is a new approximation mechanism that is desirable for application-level approximation. The proposed approximation inverts the feature extraction process in the application of interest and then derives the approximation threshold to filter out non-critical data. As the proposed approximation aims for the specific features, the loss in accuracy can be

well managed. We evaluated the proposed FDAPx in a fall risk assessment application. Experimental results show that the proposed FDAPx only degrades the accuracy by up to 1.62% in swing time and 0.3% in stance time for the approximation threshold of 250. The deviation in the extracted gait features obtained by the proposed FDAPx is 30% less than that induced by 2-bit precision reduction based approximation. As the significant reduction in raw sensing data happens before the application-oriented feature analysis, the proposed FDAPx can save more than 87% of storage space over the baseline. The prototype of FDAPx consumes 48% less FPGA resource utilization rate than the existing approximation methods.

REFERENCES

- [1] Woongki Baek and Trishul M. Chilimbi. 2010. Green: A Framework for Supporting Energy-Conscious Programming using Controlled Approximation. In *Proc. PLDI'2010*. ACM SIGPLAN.
- [2] Toth Mate Banos, Oresti and Oliver Amft. 2014. REALDISP Activity Recognition Dataset. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5GP6D>.
- [3] Gwen Bergen, Mark R Stevens, and Elizabeth R Burns. 2016. Falls and fall injuries among adults aged 65 years—United States, 2014. *Morbidity and Mortality Weekly Report* 65, 37 (2016), 993–998.
- [4] I. Bhati, Z. Chishti, S. Lu, and B. Jacob. 2015. Flexible auto-refresh: Enabling scalable and energy-efficient DRAM refresh reductions. In *Proc. ISCA*. 235–246.
- [5] Diliang Chen, Yi Cai, and Ming-Chun Huang. 2018. Customizable pressure sensor array: Design and evaluation. *IEEE Sensors Journal* 18, 15 (2018), 6337–6344.
- [6] F. Frustaci, D. Blaauw, D. Sylvester, and M. Alioto. 2015. Better-than-voltage scaling energy reduction in approximate SRAMs via bit dropping and bit reuse. In *Proc. PATMOS*. 132–139.
- [7] Avrajit Ghosh, Arnab Raha, and Amitava Mukherjee. 2020. Energy-Efficient IoT-Health Monitoring System using Approximate Computing. *Internet of Things* 9 (2020), 100166. <https://doi.org/10.1016/j.iot.2020.100166>
- [8] E. Hadi, S. Adrian, C. Luis, and B. Doug. 2012. Architecture Support for Disciplined Approximate Programming. *SIGARCH Comput. Archit. News* 40, 1 (March 2012), 301–312.
- [9] Ping Jiang, Jonathan Winkley, Can Zhao, Robert Munnoch, Geyong Min, and Laurence T. Yang. 2016. An Intelligent Information Forwarder for Healthcare Big Data Systems With Distributed Wearable Sensors. *IEEE Systems Journal* 10, 3 (2016), 1147–1159. <https://doi.org/10.1109/JSYST.2014.2308324>
- [10] A. B. Kahng and S. Kang. 2012. Accuracy-configurable adder for approximate arithmetic designs. In *DAC Design Automation Conference 2012*. 820–825.
- [11] C. B. Kushwah and S. K. Vishvakarma. 2014. A sub-threshold eight transistor (8T) SRAM cell design for stability improvement. In *Proc. ICICDT*. 1–4.
- [12] Briana Moreland, Ramakrishna Kakara, and Ankita Henry. 2020. Trends in nonfatal falls and fall-related injuries among adults aged 65 years—United States, 2012–2018. *Morbidity and Mortality Weekly Report* 69, 27 (2020), 875.
- [13] Adrian Sampson, Werner Dietl, Emily Fortuna, Danushen Gnanapragasam, Luis Ceze, and Dan Grossman. 2011. EnerJ: Approximate Data Types for Safe and General Low-Power Computation. *SIGPLAN Not.* 46, 6 (June 2011), 164–174.
- [14] A. Sampson, J. Nelson, K. Strauss, and L. Ceze. 2013. Approximate storage in solid-state memories. In *Proc. 2013 46th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 25–36.
- [15] M. Sparsh. 2016. A Survey of Techniques for Approximate Computing. *ACM Comput. Surv.* 48, 4, Article 62 (March 2016), 33 pages.
- [16] Daniel A. Sterling, Judith A. O'connor, and John Bonadies. 2001. Geriatric falls: injury severity is high and disproportionate to mechanism. *Journal of Trauma and Acute Care Surgery* 50, 1 (2001), 116–119.
- [17] Judy A. Stevens. 2013. The STEADI tool kit: a fall prevention resource for health care providers. *The IHS primary care provider* 39, 9 (2013), 162.
- [18] Bruno J Vellas, Sharon J Wayne, Linda J Romero, Richard N Baumgartner, and Philip J Garry. 1997. Fear of falling and restriction of mobility in elderly fallers. *Age and ageing* 26, 3 (1997), 189–193.
- [19] Pruthvy Yellu, Landon Buell, Miguel Mark, Michel A. Kinsy, Dongpeng Xu, and Qiaoyan Yu. 2021. Security Threat Analyses and Attack Models for Approximate Computing Systems: From Hardware and Micro-Architecture Perspectives. *ACM Trans. Des. Autom. Electron. Syst.* 26, 4, Article 32 (Apr 2021), 31 pages.
- [20] Pruthvy Yellu, Nishanth Chennagouni, and Qiaoyan Yu. 2022. Leveraging Intermediate Node Evaluation to Secure Approximate Computing for AI Applications. In *2022 IEEE International Symposium on Technologies for Homeland Security (HST)*. 1–8. <https://doi.org/10.1109/HST56032.2022.10025430>
- [21] Wiebren Zijlstra and At L Hof. 2003. Assessment of spatio-temporal gait parameters from trunk accelerations during human walking. *Gait & posture* 18, 2 (2003), 1–10.