Resource-Efficient Field-Programmable Gate Array Implementation of an Integer Fuzzy Inference System for Conveyor Gear Condition Monitoring

Ahmad Sabiq
Department Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia | Informatics Engineering Department, Universitas YARSI, Jakarta, Indonesia
ahmadsabiq@mail.ugm.ac.id

Jazi Eko Istiyanto
Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia
jazi@ugm.ac.id (corresponding author)

Andi Dharmawan
Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia
andi_dharmawan@mail.ugm.ac.id

Rachmat Sriwijaya
Department of Mechanical and Industrial Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia
sriwijaya@gadjahmada.edu

Received: 11 December 2025 | Revised: 18 January 2026, 29 January 2026, and 1 February 2026 | Accepted: 2 February 2026

 DOI: https://doi.org/10.48084/etasr.16872

ABSTRACT

Industrial conveyor systems rely on reliable gearbox drives, where undetected tooth scoring and fractures can lead to downtime and safety risks. Vibration monitoring is well-established, but edge deployment on low-power Field-Programmable Gate Arrays (FPGAs) is constrained by limited logic resources and the absence of floating-point hardware. This study presents a VLSI implementation of an integer fuzzy inference engine for real-time conveyor gear condition monitoring on a small Lattice iCE40 HX8K FPGA. Three time-domain features—peak-to-peak and Willison amplitudes from the y- and z-axes of an ADXL345 accelerometer—are computed over 1 s windows and fed to a 3-class fuzzy system (Normal, Scoring, Damaged) with trapezoidal inputs and triangular outputs, implemented entirely in fixed-point arithmetic. Membership values are quantized from a 64-bit floating-point reference to 16-, 8-, and 4-bit integers. On a labeled vibration dataset, all configurations achieve F1-scores above 90%, and the 8-bit engine matches the floating-point baseline within 0.46 % while using 5256 of 7680 logic cells (68%). The results indicate that integer fuzzy inference is a practical option for gearbox monitoring on low-end FPGAs under strict area and power budgets.

Keywords-field-programmable gate array; fuzzy inference system; gear condition monitoring; vibration; time-domain features

I. INTRODUCTION

Conveyor-based material-handling systems are widely used in industrial plants, and their availability depends on gearbox health (e.g., spur and helical gears) [1, 2]. Wear, overload, and foreign-object damage can cause scoring, chipping, and fractures that appear as changes in vibration signals, making vibration analysis a practical basis for predictive maintenance [3-6].

Driven by IoT and cyber–physical systems, analytics is increasingly pushed toward the sensor to reduce latency and network load [7-9]. Low-power FPGAs are attractive platforms for such edge nodes thanks to their deterministic timing, fine-grained parallelism, and relatively low energy consumption

compared with high-end devices or general-purpose processors [10, 11]. FPGA-based accelerators have been examined for real-time signal analysis and classification, including seizure detection using linear SVM cores mapped to reconfigurable hardware for EEG monitoring [12]; heterogeneous neural-network inference engines for embedded sensing [10, 11]; low-resource vibration-processing implementations such as FPGA-based wireless sensor nodes with on-node vibration analysis and fault diagnosis [13]; and LUT-based DWT denoising architectures [14]. A fuzzy logic-based speed controller for a 3-phase BLDC motor implemented on an Xilinx Spartan-3E FPGA can meet the transient performance requirements in industrial drives [15]. However, deploying advanced analytics on small, LUT-only FPGAs is challenging: floating-point operations are too expensive, dedicated DSP and memory blocks may be absent, and the available logic resources must accommodate sensor interfacing, signal processing, and decision-making concurrently [11, 16, 17]. This is particularly constraining in edge scenarios where the FPGA must share a tight power and cost budget with sensing, communication, and power-management circuitry [11, 18, 19].

Fuzzy Inference Systems (FIS) offer interpretable rule-based reasoning that maps naturally to fixed-point arithmetic [20-22]. Fuzzy logic has also been adopted as a lightweight decision mechanism in resource-constrained sensing (e.g., clustering and aggregation in underwater wireless sensor networks) [23, 24]. Time-domain vibration features with integer-valued definitions are hardware-friendly and can be combined with fuzzy rules to yield interpretable condition labels for rotating machinery [5, 6, 25]. Moreover, FPGA/SoC-FPGA research shows that integer and low-precision arithmetic can effectively support embedded feature extraction and classification [11, 25, 26]. The challenge is to fit an end-to-end pipeline—data acquisition, feature extraction, and fuzzy inference—into a very small FPGA fabric while maintaining multi-class fault-detection performance.

The present study addresses this difficulty by presenting an end-to-end integer-only fuzzy inference pipeline for conveyor gearbox condition monitoring that runs on a small LUT-only FPGA (Lattice iCE40 HX8K) without using floating-point arithmetic, DSP blocks, or block RAM. The main contributions of the study are:

- A compact vibration-based gear monitoring model using three integer time-domain features—z-axis peak-to-peak, z-axis Willison amplitude, and y-axis Willison amplitude—selected via Pearson correlation from a wider feature set.
- An FPGA architecture that integrates SPI-based sensor acquisition, 1 s windowed feature extraction, and a 3-stage fuzzy inference pipeline, mapped entirely to LUT/FF fabric.
- A quantization study of 16-, 8-, and 4-bit fuzzy membership representations, showing that aggressive bit-width reduction provides substantial resource savings while maintaining F1-scores above 90% and high recall for damaged gears.

II. SYSTEM DESCRIPTION AND VIBRATION DATASET

The experimental setup consists of a mini conveyor driven by a gearbox–motor unit instrumented with a 3-axis digital accelerometer (ADXL345), as depicted in Figure 1. The sensor is rigidly mounted on the gearbox housing near the driven gear and connected to a single-board computer via an SPI link for data logging. The axes are fixed as x along the shaft, y forward, and z upward. The accelerometer is mounted on the gearbox housing near the driven gear, and the x–y–z axes follow the directions indicated by the arrows. During acquisition, the shaft speed is 1250 rpm (≈20.8 Hz); the sampling rate is 50 Hz (Nyquist: 25 Hz), which covers the shaft fundamental and low-frequency components used by the selected 1 s windowed time-domain features while keeping edge-node throughput low.

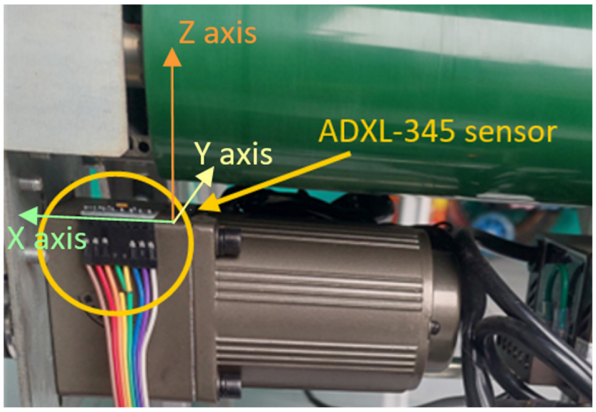

Fig. 1. ADXL345-based vibration data acquisition on the mini conveyor gearbox.

Three gear health conditions are considered: Normal, Scoring, and Damaged. In the normal state, gears show no visible scratches, cracks, or fractures. The scoring condition is characterized by erosion on the top land of the addendum, while the damaged condition exhibits broken teeth extending down to the pitch circle and dedendum [27], as illustrated in Figures 2 and 3. For each condition, 1200 s of 3-axis acceleration data are recorded, yielding 60,000 samples per axis and 1200 non-overlapping 1 s windows (50 samples) per class. The windows are labeled according to the underlying gear condition, and subsequently partitioned into modeling and testing subsets, with 70% used for designing the fuzzy model and 30% forming a held-out test set for performance evaluation.

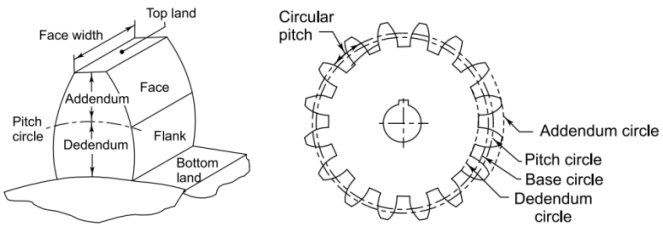

Fig. 2. Gear nomenclature.

Fig. 3. Gears used in conveyor gearboxes: (a) normal gear, (b) scoring gears, (c) damaged gear.

Data were recorded under steady-state operation at a constant conveyor load (no intentional load variation), room temperature (~25–27 °C), and manufacturer-proposed lubrication (no intentional change in lubrication state).

III. INTEGER FUZZY INFERENCE MODEL AND QUANTIZATION

A. Time-Domain Feature Selection

To characterize the gearbox vibration under the three health conditions, a set of simple time-domain features was extracted from the y- and z-axis signals within each 1 s window. The candidate features include peak-to-peak amplitude, Willison amplitude, slope sign change, zero-crossing rate, and wavelength for both axes. All features are computed as integer-valued statistics, which simplifies their hardware implementation.

Feature relevance is quantified using the Pearson correlation coefficient between each feature and an ordinal condition label (Normal = 0, Scoring = 1, Damaged = 2). Table I summarizes the correlation values, demonstrating the dominance of three features: z-axis Willison amplitude ($r \approx 0.90$), z-axis peak-to-peak amplitude ($r \approx 0.86$), and y-axis Willison amplitude ($r \approx 0.78$). In contrast, the remaining features exhibit substantially lower correlations. To reduce the dimensionality of the fuzzy model and simplify the hardware, only these three integer features were retained as inputs to the FIS.

Pearson correlation was chosen as a lightweight, transparent relevance metric aligned with the LUT/FF-only design goal; exploring non-linear relevance measures (e.g., mutual information/mRMR) under broader operating variability is left for future work.

TABLE I. PEARSON CORRELATION VALUES OF FEATURES WITH CONDITION LABELS

Feature	Pearson correlation
willison_amp_z	0.902992
peak_to_peak_z	0.858746
willison_amp_y	0.781255
peak_to_peak_y	0.657617
zero_cross_z	0.353287
zero_cross_y	0.308598
wavelength_z	0.276626
ssc_y	0.255486
ssc_z	0.141203
wavelength_y	0.110063

B. Fuzzy Inference System

The FIS uses the three selected features as inputs and produces a scalar output representing the gear health index, which is subsequently mapped to the three classes (Normal, Scoring, Damaged). Each input feature is associated with three trapezoidal membership functions (LOW, MEDIUM, HIGH) whose parameters are derived from the minimum, maximum, and average values of the feature in each condition and then snapped to integer breakpoints in the feature range. For example, for the 8-bit WAz input (0–30), the trapezoids are LOW (0, 0, 5, 15), MEDIUM (4, 12, 15, 25), and HIGH (10, 20, 30, 30), where (a, b, c, d) are the breakpoints. The output variable is defined on an integer domain and uses three triangular membership functions corresponding to the three gear conditions.

Rules are extracted in a data-driven manner from the labeled windows by first assigning a discrete fuzzy label to each feature using the maximum membership degree, yielding a triplet $F_j = (\ell_{P2Pz}, \ell_{WAz}, \ell_{WAy})$ per sample (Algorithm 1). For each label triplet and true condition C_i, the empirical co-occurrence frequency was computed using (1), and the rule consequent was selected by majority vote (argmax over C_i). Redundant rules were then compacted by replacing weakly informative antecedents with a wildcard "–" (don't-care) and merging rules with identical consequents, resulting in the compact Mamdani rule base reported in Tables II and III.

$$freq(C_i, F_j) = \sum_{k=i}^{N} 1(Cond_k = C_i \wedge FuzzyLabels_k = F_j) \quad (1)$$

```
Algorithm 1: Rule Extraction and
Compaction from Labelled Dataset
Input: df with {P2Pz,WAz,WAy,Condition};
membership functions MFp2p,MFwz,MFwy
Output: compact Mamdani rule base R
1. Assign fuzzy labels by max-membership:
     df.fuzzy_z ← label(df.P2Pz,MF_p2p)
     df.fuzzy_wz ← label(df.WAz,MF_wz)
     df.fuzzy_wy ← label(df.WAy,MF_wy)
2. Compute co-occurrence counts
   C[ℓ_z, ℓ_wz, ℓ_wy, C_i] from df using(1).
```

3. For each distinct triplet $(\ell_z, \ell_{wz}, \ell_{wy})$: $C \leftarrow \arg\max_{Ci} C[\ell_z, \ell_{wz}, \ell_{wy}, C_i]$. Add rule r: $(\ell_z, \ell_{wz}, \ell_{wy}) \rightarrow C$ to R.
4. Compact R: for same consequents, replace one differing antecedent with "–" and merge.
5. Return R.

Given a feature vector, the FIS performs fuzzification, rule evaluation using MIN–MAX operators, and defuzzification via a center of sums method. The latter computes a crisp output as the weighted average of the centroids of the activated output membership functions, with weights equal to their aggregated firing strengths. The crisp output is finally partitioned into three intervals by two thresholds, assigning each decision to Normal, Scoring, or Damaged.

C. Quantization of Fuzzy Membership Values

The fuzzy membership functions are initially designed and evaluated in a 64-bit floating-point environment using the scikit-fuzzy library. However, floating-point arithmetic is prohibitively expensive on low-power FPGAs, so all membership degrees are quantized into integer representations before hardware implementation. Specifically, 16-, 8-, and 4-bit unsigned fixed-point formats covering the [0, 1] membership range are considered. In each case, the floating-point membership degree $\mu(x)$ is mapped to an integer by linear scaling and rounding, and all FIS operations are expressed in terms of scaled integers.

IV. FPGA ARCHITECTURE AND IMPLEMENTATION

A. System-Level Architecture

The complete hardware architecture, shown in Figure 4, implements the full processing chain from sensor acquisition to fuzzy decision on a Lattice iCE40 HX8K FPGA. A clock-management block generates the required clocks, including a high-speed core clock and divided clocks at 50 Hz and 1 Hz. The SPI interface block controls the ADXL345, issuing configuration commands and retrieving 10-bit axis samples at 50 Hz via the SCLK, SDI, SDO, and SS signals.

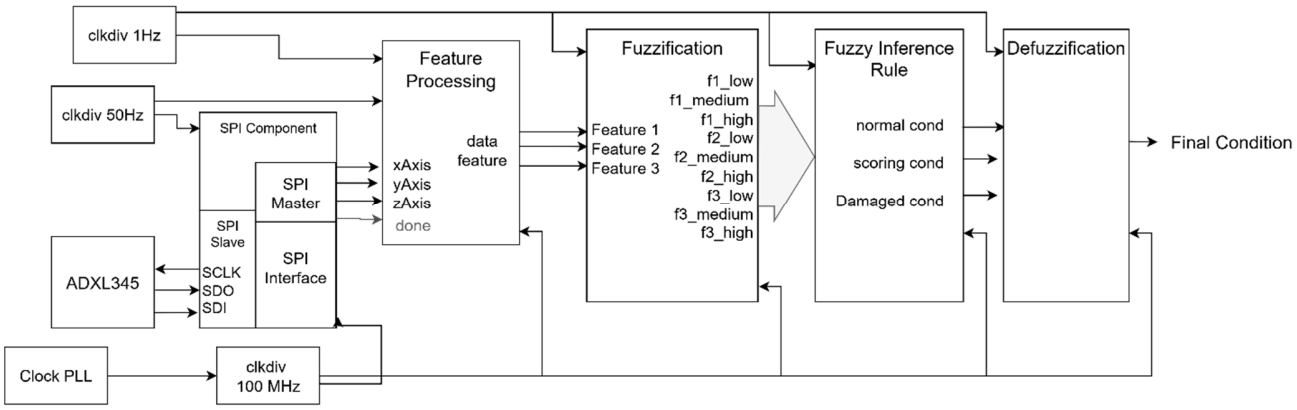

Fig. 4. FPGA architecture design for 3-axis digital accelerometer sensor ADXL345 reading and FIS for gear condition monitoring.

The feature extraction block processes the y- and z-axis samples in streaming mode, computing the three selected time-domain features over non-overlapping 1 s windows. Within each window, the block maintains the minimum and maximum of the z-axis signal for peak-to-peak calculation and accumulates thresholded absolute differences on the y and z axes to obtain the Willison amplitudes. At the end of the window, the final feature values are latched and forwarded to the fuzzy inference module. The FIS block then performs fuzzification, rule evaluation, and defuzzification using fixed-point arithmetic, and outputs a crisp health index and a 2-bit class label that can be monitored via on-board LEDs or exported through a communication peripheral.

B. Micro-Architecture of the Fuzzy Engine

The fuzzy inference engine is implemented as a 3-stage pipeline comprising fuzzification, inference, and defuzzification submodules. Each stage is internally time-multiplexed to reuse arithmetic resources and minimize logic usage. In the fuzzification stage, a single arithmetic unit evaluates trapezoidal membership functions for one input–set pair at a time, under the control of a small finite-state machine. The resulting membership degrees are stored in registers and made available to the next stage. The inference stage iterates through the rule base, using multiplexers to select the relevant membership degrees and a shared min/max core to compute rule firing strengths and aggregate contributions for the three output conditions. Instead of allocating a dedicated accumulator per rule, a small register bank and a feedback network are used to update per-class supports sequentially across rules.

In the defuzzification stage, the aggregated supports are converted into a crisp output by computing the weighted-sum numerator and the sum-of-weights denominator in widened fixed-point accumulators. The final normalization is implemented, with a synthesizable integer divider inferred in the RTL and realized as a radix-2 shift–subtract (restoring) division datapath in LUT/FF fabric; the quotient is taken with integer truncation, which is consistent across all bit-width configurations. Whenever the divisor is a power of two (e.g., the factor ½ in trapezoidal area terms), the operation is implemented as a binary right shift to avoid a general divider and reduce logic cost.

To improve robustness, feature inputs are clamped to the predefined integer domains used by the membership functions, and membership degrees are saturated to the valid [0, 2b−1] range for the selected bit-width. Accumulators in the inference and defuzzification stages use widened registers (guard bits) to prevent overflow during per-rule aggregation and during numerator/denominator accumulation. All computations in the fuzzy engine use fixed-point integers with a configurable bit-width (16, 8, or 4 bits). The Verilog code is written in a bit-width-parametric style, so switching between quantization levels only requires changing a few parameters and resynthesizing the design.

C. Implementation and Synthesis Flow

The design is written in synthesizable Verilog and implemented on an iCE40 HX8K FPGA using an open-source flow. Synthesis is performed with Yosys and place-and-route with nextpnr-ice40; bitstreams are generated with icepack and programmed via iceprog. No BRAM or DSP blocks are used: storage relies on flip-flops/LUT shift registers, and arithmetic is mapped to LUTs and carry chains.

V. EXPERIMENTAL RESULTS

A. Time-Domain Feature Behavior

Figure 5 depicts representative 3-axis vibration signals over 1200 s for the three gear conditions. Normal operation exhibits low-amplitude, stable waveforms, whereas scoring and damaged gears show higher amplitude and variability, with the damaged case producing the most pronounced peaks. Converting the raw signals into time-domain features makes these inter-condition differences more explicit.

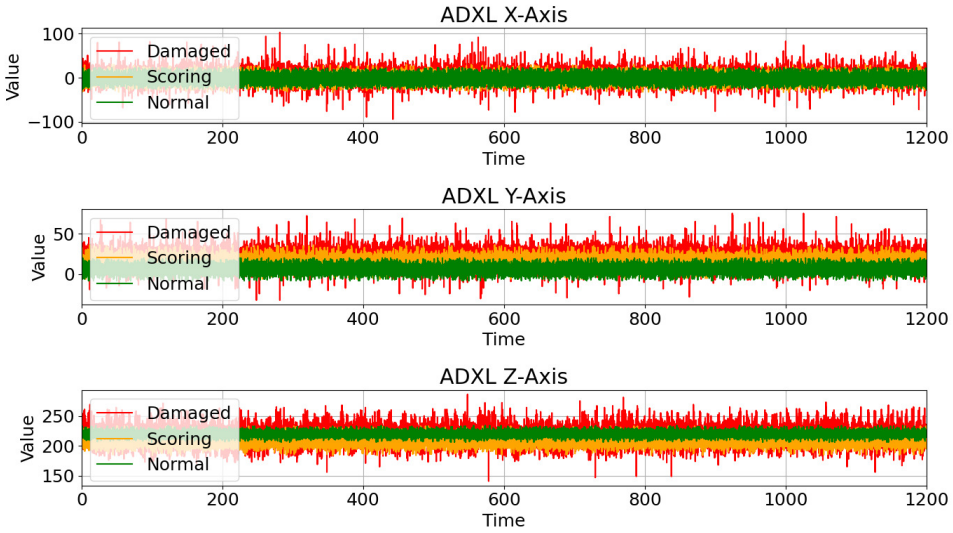

Fig. 5. Plot of the accelerometer data of the three axes.

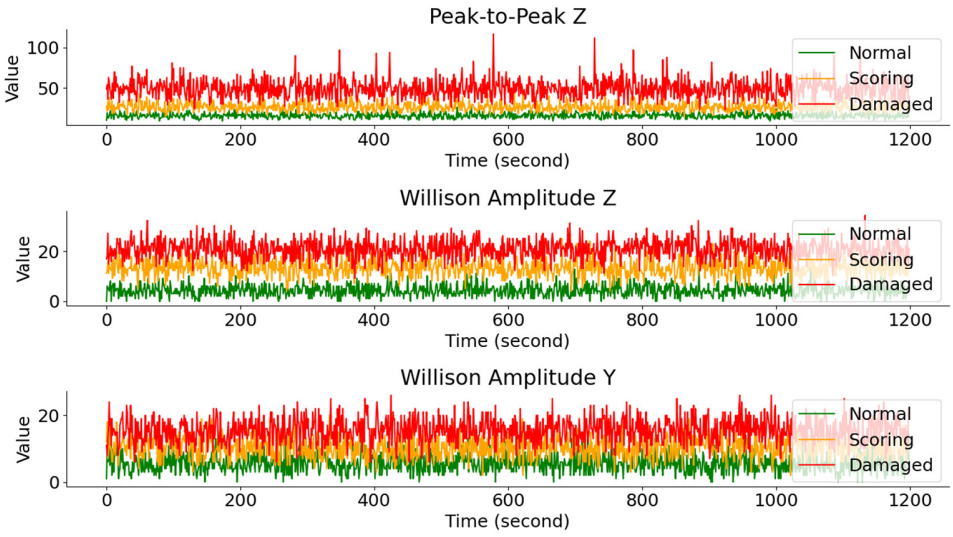

Fig. 6. Plot of selected time-domain features.

Figure 6 presents the plots of the three selected features—z-axis peak-to-peak, z-axis Willison amplitude, and y-axis Willison amplitude—across the dataset. The feature trajectories reveal clear separation between normal, scoring, and damaged windows: damaged gears consistently exhibit higher peak-to-peak and Willison amplitudes, while scoring conditions occupy an intermediate range. These observations are consistent with the high Pearson correlation coefficients reported in Table I, and justify the use of these features as inputs to the fuzzy model.

B. Fuzzy Membership, Rule Base, and Classification Performance

The input and output membership functions used in the 8-bit configuration are exhibited in Figure 7, and their integer parameters are outlined in Table II. The trapezoidal input sets partition the feature domains into low, medium, and high regions, while the triangular output sets span the condition axis from normal to damaged. Descriptive analysis of the fuzzified dataset yields the dominant combinations of input labels and gear conditions listed in Table II, from which the 14-rule Mamdani rule base in Table III is derived.

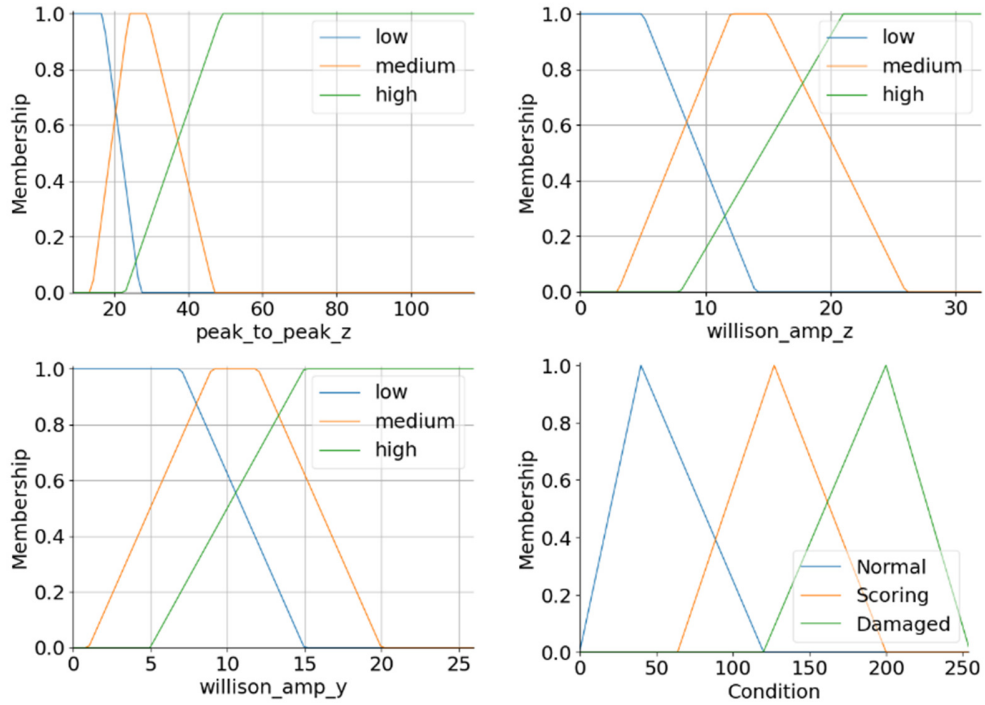

Fig. 7. Fuzzy membership functions for input and output variables in the gear-condition model.

Rules 7 and 8 are retained because they capture two frequent, distinct label patterns (with Pz = Medium) that both map to the Scoring class; further merging would broaden the antecedent coverage and could increase rule conflicts.

The fuzzy system is evaluated on the held-out test set using the 64-bit floating-point reference implementation. With appropriately chosen decision thresholds, the model achieves a precision of 92.30%, a recall of 92.28%, and an F1-score of 92.20%. These results confirm that the 3-feature fuzzy model provides consistent and reliable discrimination among the considered gear conditions.

C. Impact of Quantization on F1-Score

Table IV compares floating-point memberships with 16-, 8-, and 4-bit integer quantizations. All integer configurations maintain F1-scores above 90%. Both the 16-bit and 8-bit cases achieve F1-scores of 91.74%, whereas the 4-bit representation reaches 90.94%.

TABLE II. COMBINATIONS OF FUZZY LABELS AND CONDITIONS

Peak-to-peak z-axis	Willison amplitude z-axis	Willison amplitude y-axis	Condition
High	High	–	Damaged
High	–	High	Damaged
High	Medium	–	Damaged
High	Medium	Low	Scoring
Medium	High	High	Damaged
Medium	High	Low	Scoring
Medium	Medium	–	Scoring
Medium	–	Medium	Scoring
Medium	Low	High	Scoring
Low	High	–	Scoring
Low	Medium	High	Scoring
Low	Medium	Medium	Scoring
Low	–	Low	Normal
Low	Low	–	Normal

"–" denotes a don't-care antecedent (any label matches).

TABLE III. INFERENCE RULE OF THE FIS

No	Inference rule
1	IF Pz = High ∧ Wz = High, THEN Condition = Damaged
2	IF Pz = High ∧ Wy = High, THEN Condition = Damaged
3	IF Pz = High ∧ Wy = Medium, THEN Condition = Damaged
4	IF Pz = High ∧ Wz = Medium ∧ Wy = Low, THEN Condition = Scoring
5	IF Pz = Medium ∧ Wz = High ∧ Wy = High, THEN Condition = Damaged
6	IF Pz = Medium ∧ Wz = High ∧ Wy = Low, THEN Condition = Scoring
7	IF Pz = Medium ∧ Wz = Medium, THEN Condition = Scoring
8	IF Pz = Medium ∧ Wy = Medium, THEN Condition = Scoring
9	IF Pz = Medium ∧ Wz = Low ∧ Wy = High, THEN Condition = Scoring
10	IF Pz = Low ∧ Wz = High, THEN Condition = Scoring
11	IF Pz = Low ∧ Wz = Medium ∧ Wy = High, THEN Condition = Scoring
12	IF Pz = Low ∧ Wz = Medium ∧ Wy = Medium, THEN Condition = Scoring
13	IF Pz = Low ∧ Wy = Low, THEN Condition = Normal
14	IF Pz = Low ∧ Wz = Low, THEN Condition = Normal

Pz represents the Peak-to-Peak on the z-axis, Wz represents the Willison Amplitude on the z-axis, and Wy represents the Willison Amplitude on the y-axis.

Moving from 64-bit floating point to 8-bit integers incurs only a 0.46 percentage point/% loss in F1-score, whereas the 4-bit case shows a 1.26-point degradation. Considering that an F1-score above 85% is often regarded as robust for predictive maintenance and scores above 90% as desirable for critical real-time diagnostics, these results indicate that 8-bit quantization is a safe choice for the proposed application, and that even 4-bit membership values can be acceptable when resource savings are prioritized.

TABLE IV. EVALUATION METRICS

Fuzzy membership data type	Precision	Recall	F1-score
Floating point 64-bit	92.30%	92.28%	92.20%
Integer 16-bit	91.85%	91.77%	91.74%
Integer 8-bit	91.85%	91.77%	91.74%
Integer 4-bit	91.03%	90.95%	90.94%

D. Timing and Resource Utilization of FPGA

Functional verification is carried out by simulating the RTL design with realistic accelerometer traces to confirm cycle-accurate behavior across the acquisition, feature extraction, and inference stages. Figure 8 shows the SPI waveforms, confirming that the ADXL345 axis registers are updated correctly every 20 ms at the 50 Hz sampling rate and that the chip-select and clock timing follow the intended transaction framing. Figure 9 illustrates the evolution of the feature-accumulation registers over a 1 s window, including the running min/max tracking for peak-to-peak and the thresholded difference accumulation for Willison amplitudes, followed by a clean reset and latching event at the window boundary. Figure 10 presents internal signals of the FIS, demonstrating correct stage sequencing (fuzzification → rule evaluation/aggregation → defuzzification) under finite-state control, and showing that both the 8-bit and 4-bit quantized implementations produce stable, consistent class labels for the same input windows despite the reduced membership precision.

Table V summarizes the post-synthesis resource usage of the complete 8-bit design on the iCE40 HX8K device. The implementation uses 5256 logic cells out of 7680 available (68%), 13 I/O blocks (5%), all 8 global buffers (100%), and one of the two PLLs (50%). No DSP blocks or embedded block RAMs are employed. The heavy utilization of global buffers reflects the need for distributing multiple clocks and significant control signals across the device.

Despite the 68% logic-cell occupancy, the design still meets the target clock frequency with comfortable timing margins, which is consistent with the common observation that timing closure problems typically arise above 80–85% utilization on similar FPGA families. The remaining logic cells are sufficient for moderate extensions, such as communication interfaces or supervisory control blocks, although the saturation of global buffers may limit the addition of further independent clock domains. Overall, the resource results confirm that the proposed architecture fits within the constraints of a low-end FPGA while leaving some headroom for system-level integration.

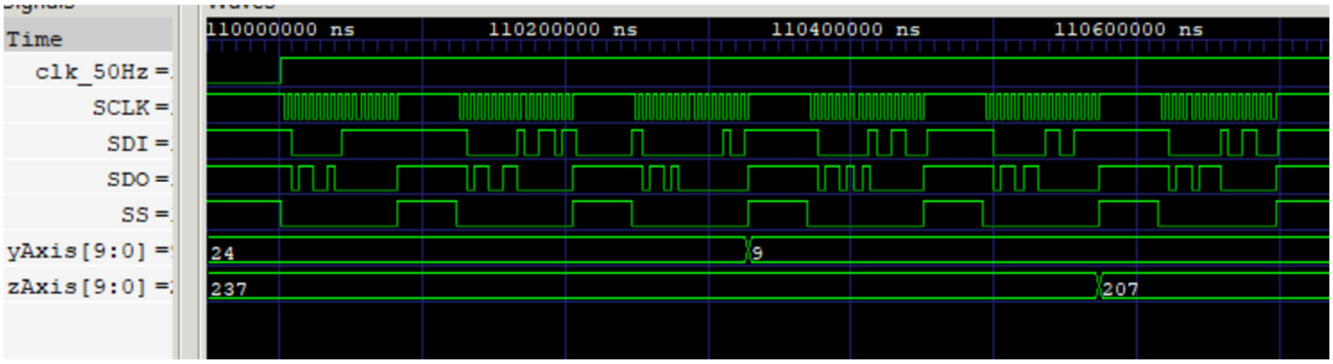

Fig. 8. Simulation of sensor data reading via SDI signals using the SPI protocol.

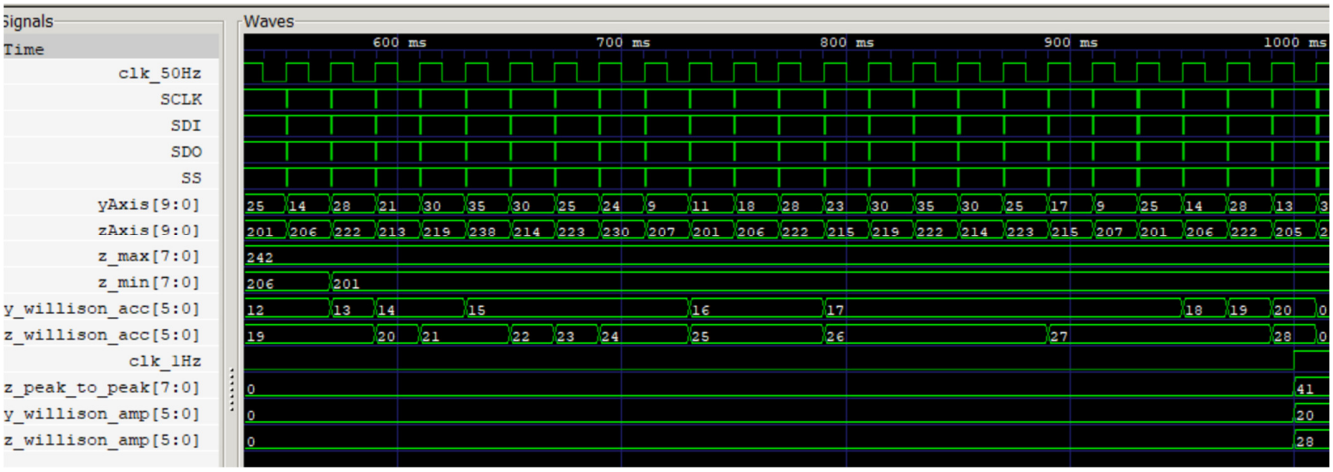

Fig. 9. Simulation of sensor data reading and feature calculation results.

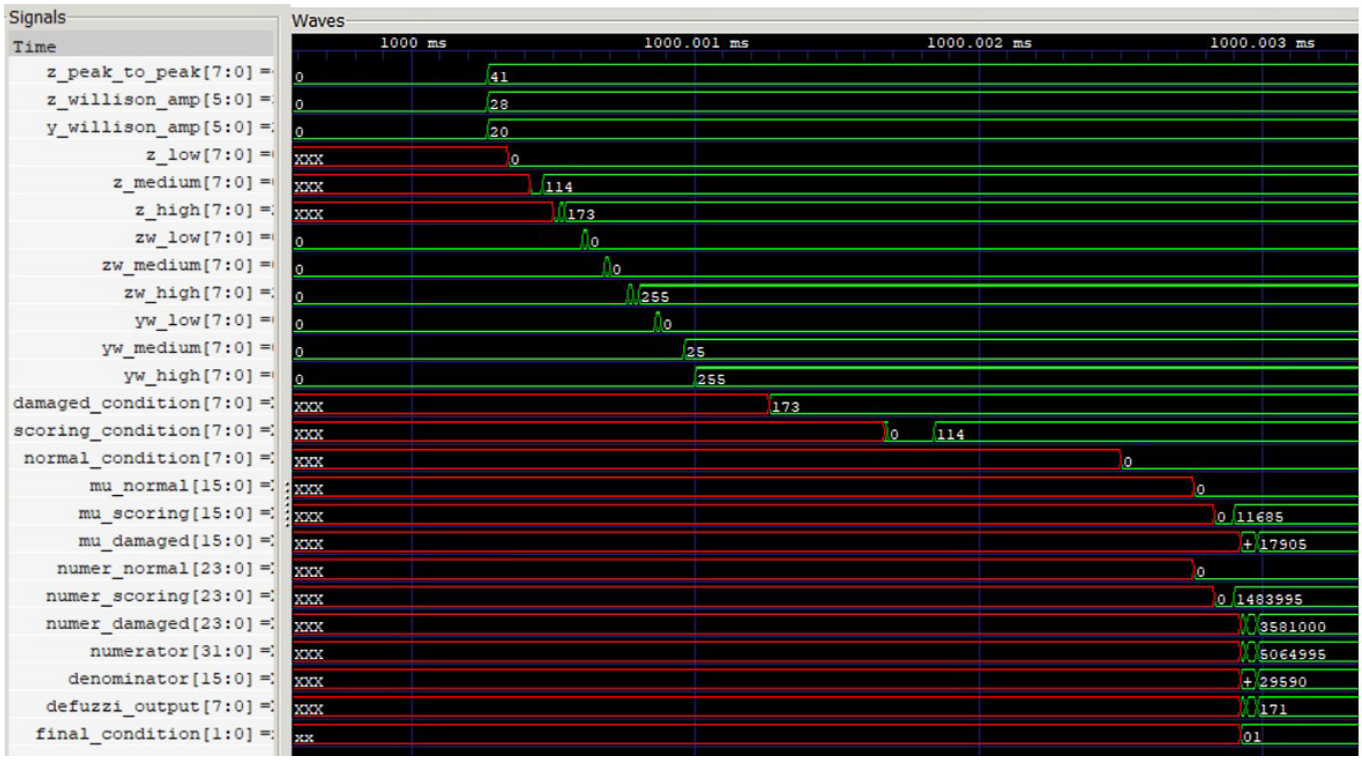

Fig. 10. Simulation of FIS value calculation from three vibration signal features.

TABLE V. RESOURCE USAGE OF THE ICE40 HX8K FPGA DEVICE

Utilization	Resource usage	Resource	Percentage (%)
ICESTROM_LC	5256	7680	68%
SB IO	13	256	5%
SB GB	8	8	100%
DSP	N/A	N/A	N/A
PLL	1	2	50%

VI. DISCUSSION

The main objective of the present study is to demonstrate that an end-to-end fuzzy-inference gear monitoring pipeline can be realized on a low-end FPGA using integer arithmetic and modest logic resources. Compared to ML/DL approaches, which typically require larger memory and compute footprints, the proposed integer FIS provides interpretable rule-based reasoning tailored to three informative time-domain features.

Quantization to 8-bit membership values has a negligible impact on the F1-score compared to the 64-bit baseline while enabling a fully fixed-point design; even 4-bit maintains F1-scores above 90% with a slightly larger loss in damaged-gear recall. Hardware results show that the 8-bit configuration uses 68% of the iCE40 HX8K logic cells and avoids DSP/embedded

memories via time-multiplexing, making it suitable for area- and cost-constrained edge nodes.

Nevertheless, the current prototype has limitations. The feature set is limited to three time-domain metrics and has only been tested on one conveyor gearbox at a set speed. Extending the model to multiple speeds, additional fault types, or different mechanical configurations may require richer feature sets and a larger fuzzy rule base. Long-term in-field testing would be necessary to assess robustness under varying temperatures, mechanical noise, and other environmental factors.

VII. CONCLUSION

This study presented a VLSI implementation of an end-to-end, integer-only fuzzy inference pipeline for real-time conveyor gearbox condition monitoring on a small Lattice iCE40 HX8K Field-Programmable Gate Array (FPGA). The proposed system integrates SPI-based acquisition from an ADXL345 accelerometer, non-overlapping 1 s windowed integer feature extraction at 50 Hz sampling, and a 3-stage fuzzification–inference–defuzzification engine using trapezoidal input and triangular output membership functions. Three integer time-domain features (z-axis peak-to-peak, z-axis Willison amplitude, and y-axis Willison amplitude) were selected from a broader set and used to classify three gear conditions (Normal, Scoring, and Damaged).

Fuzzy membership degrees were quantized from a 64-bit floating-point reference to 16-, 8-, and 4-bit integer representations to enable efficient LUT/FF-based hardware mapping. On the labeled vibration dataset, all integer configurations achieved F1-scores above 90%, with the 8-bit engine matching the floating-point baseline within 0.46%. The complete pipeline was synthesized without floating-point arithmetic, DSP blocks, or block RAM, and the 8-bit configuration used 5256 out of 7680 logic cells (68%). RTL simulations with realistic accelerometer traces confirm deterministic operation, including correct SPI timing at 50 Hz and consistent internal sequencing across quantization levels.

Overall, the results indicate that compact integer fuzzy inference can deliver practical diagnostic performance under the tight area constraints of LUT/FF-only low-end FPGAs, making it suitable for cost- and power-constrained edge monitoring nodes. Future work will evaluate robustness under broader operating conditions (e.g., varying speed/load/temperature) and extend the feature/rule set to additional fault types while preserving the resource-efficient hardware profile.

ACKNOWLEDGMENT

The authors would like to thank the Indonesian Education Scholarship (BPI), the Center for Higher Education Funding and Assessment (PPAPT), the Ministry of Higher Education, Science, and Technology of the Republic of Indonesia, and the Indonesia Endowment Fund for Education (LPDP) for supporting this research.

DATA AVAILABILITY STATEMENT

The labeled vibration dataset was collected by the authors using the experimental setup. The dataset is private but can be made available upon reasonable request.

REFERENCES

[1] N. J. Parmar, A. T. James, and M. Asjad, "Analysis of Maintenance Outsourcing Challenges for Belt Conveyors in the Industry 4.0 Era," *Journal of Global Operations and Strategic Sourcing*, vol. 16, no. 3, pp. 718–744, Aug. 2023, https://doi.org/10.1108/JGOSS-06-2022-0050.

[2] M. Elahi, S. O. Afolaranmi, W. M. Mohammed, and J. L. Martinez Lastra, "Energy-Based Prognostics for Gradual Loss of Conveyor Belt Tension in Discrete Manufacturing Systems," *Energies*, vol. 15, no. 13, Jun. 2022, Art. no. 4705, https://doi.org/10.3390/en15134705.

[3] S. E. Kramti, J. B. Ali, E. Bechhoefer, K. Takrouni, A. Darghouthi, and M. Sayadi, "Toward an Online Strategy for Mechanical Failures Diagnostics Inside the Wind Turbine Generators Based on Spectral Analysis," *Wind Engineering*, vol. 45, no. 4, pp. 782–792, Aug. 2021, https://doi.org/10.1177/0309524X211028759.

[4] Q. Fan, Q. Zhou, C. Wu, and M. Guo, "Gear Tooth Surface Damage Diagnosis Based on Analyzing the Vibration Signal of an Individual Gear Tooth," *Advances in Mechanical Engineering*, vol. 9, no. 6, Jun. 2017, Art. no. 168781401770435, https://doi.org/10.1177/1687814017704356.

[5] L. Wang, H. Li, T. Xi, and S. Wei, "Fault Feature Extraction Method for Rolling Bearings Based on Complete Ensemble Empirical Mode Decomposition with Adaptive Noise and Variational Mode Decomposition," *Sensors*, vol. 23, no. 23, Nov. 2023, Art. no. 9441, https://doi.org/10.3390/s23239441.

[6] M. Altaf, T. Akram, M. A. Khan, M. Iqbal, M. M. I. Ch, and C.-H. Hsu, "A New Statistical Features Based Approach for Bearing Fault Diagnosis Using Vibration Signals," *Sensors*, vol. 22, no. 5, Mar. 2022, Art. no. 2012, https://doi.org/10.3390/s22052012.

[7] S. Eichstädt, M. Gruber, A. P. Vedurmudi, B. Seeger, T. Bruns, and G. Kok, "Toward Smart Traceability for Digital Sensors and the Industrial Internet of Things," *Sensors*, vol. 21, no. 6, Mar. 2021, Art. no. 2019, https://doi.org/10.3390/s21062019.

[8] M. Uppal *et al.*, "Fault Pattern Diagnosis and Classification in Sensor Nodes Using Fall Curve," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1799–1814, 2022, https://doi.org/10.32604/cmc.2022.025330.

[9] A. Ikpehai *et al.*, "Low-Power Wide Area Network Technologies for Internet-of-Things: A Comparative Review," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2225–2240, Apr. 2019, https://doi.org/10.1109/JIOT.2018.2883728.

[10] B. Liang, S. Wang, Y. Huang, Y. Liu, and L. Ma, "F-LSTM: FPGA-Based Heterogeneous Computing Framework for Deploying LSTM-Based Algorithms," *Electronics*, vol. 12, no. 5, Feb. 2023, Art. no. 1139, https://doi.org/10.3390/electronics12051139.

[11] M. Rosół and W. Kula, "An Energy-Efficient Field-Programmable Gate Array Rapid Implementation of a Structural Health Monitoring System," *Energies*, vol. 17, no. 11, May 2024, Art. no. 2626, https://doi.org/10.3390/en17112626.

[12] S. S. Rafiammal, D. N. Jamal, and S. K. Mohideen, "Reconfigurable Hardware Design for Automatic Epilepsy Seizure Detection using EEG Signals," *Engineering, Technology & Applied Science Research*, vol. 10, no. 3, pp. 5803–5807, Jun. 2020, https://doi.org/10.48084/etasr.3419.

[13] B. Bengherbia, R. Kara, A. Toubal, M. O. Zmirli, S. Chadli, and P. Wira, "FPGA Implementation of a Wireless Sensor Node with a Built-in ADALINE Neural Network Coprocessor for Vibration Analysis and Fault Diagnosis in Machine Condition Monitoring," *Measurement*, vol. 163, Oct. 2020, Art. no. 107960, https://doi.org/10.1016/j.measurement.2020.107960.

[14] M. H. El Yousfi Alaoui, A. Jilbab, and S. El Hani, "A New Approach to DWT Design for Real Time De-noising of Vibration Signatures Related to the Induction Machine Defects," in *2016 International Conference on Electrical and Information Technologies*, Tangiers, Morocco, May 2016, pp. 197–202, https://doi.org/10.1109/EITech.2016.7519589.

[15] S. Raja and M. Rathinakumar, "Transient Analysis of the Fuzzy Logic-Based Speed Control of a Three-phase BLDC Motor," *Engineering, Technology & Applied Science Research*, vol. 13, no. 1, pp. 9855–9860, Feb. 2023, https://doi.org/10.48084/etasr.5419.

[16] C. Pham-Quoc, T. Pham-Dinh, and B. Kieu-Do-Nguyen, "Efficient Random Forest Acceleration for Edge Computing Platforms with FPGA Technology," *Journal of Advances in Information Technology*, vol. 15, no. 2, pp. 195–201, 2024, https://doi.org/10.12720/jait.15.2.195-201.

[17] Y. Luo and Y. Chen, "FPGA-Based Acceleration on Additive Manufacturing Defects Inspection," *Sensors*, vol. 21, no. 6, Mar. 2021, Art. no. 2123, https://doi.org/10.3390/s21062123.

[18] V.-K. Wong *et al.*, "Active Ultrasonic Structural Health Monitoring Enabled by Piezoelectric Direct-Write Transducers and Edge Computing Process," *Sensors*, vol. 22, no. 15, Jul. 2022, Art. no. 5724, https://doi.org/10.3390/s22155724.

[19] P. Song *et al.*, "Resource-Saving Customizable Pipeline Network Architecture for Multi-Signal Processing in Edge Devices," *Sensors*, vol. 22, no. 15, Jul. 2022, Art. no. 5720, https://doi.org/10.3390/s22155720.

[20] H. Ansaf, B. K. Ansaf, and S. S. Al Samahi, "A Neuro-Fuzzy Technique for the Modeling of β-Glucosidase Activity from Agaricus bisporus," *BioChem*, vol. 1, no. 3, pp. 159–173, Oct. 2021, https://doi.org/10.3390/biochem1030013.

[21] N. F. Idris and M. A. Ismail, "Breast Cancer Disease Classification Using Fuzzy-ID3 Algorithm with FUZZYDBD Method: Automatic Fuzzy Database Definition," *PeerJ Computer Science*, vol. 7, May 2021, Art. no. e427, https://doi.org/10.7717/peerj-cs.427.

[22] L. T. Giang *et al.*, "Adaptive Spatial Complex Fuzzy Inference Systems with Complex Fuzzy Measures," *IEEE Access*, vol. 11, pp. 39333–39350, 2023, https://doi.org/10.1109/ACCESS.2023.3268059.

[23] N. Goyal, M. Dave, and A. K. Verma, "Fuzzy Based Clustering and Aggregation Technique for Under Water Wireless Sensor Networks," in *2014 International Conference on Electronics and Communication Systems*, Coimbatore, India, Feb. 2014, pp. 1–5, https://doi.org/10.1109/ECS.2014.6892804.

[24] A. M. Pandith *et al.*, "Fuzzy Decision-Based Clustering for Efficient Data Aggregation in Mobile UWSNs," *Computers, Materials & Continua*, vol. 83, no. 1, pp. 259–279, 2025, https://doi.org/10.32604/cmc.2025.062608.

[25] S. Hadiyoso, A. Z. Ramdani, I. D. Irawati, and I. Wijayanto, "Implementation of First Order Statistical Processor on FPGA for Feature Extraction," *International Journal of Reconfigurable and Embedded Systems*, vol. 13, no. 2, Jul. 2024, Art. no. 234, https://doi.org/10.11591/ijres.v13.i2.pp234-243.

[26] A. Rizal, S. Hadiyoso, and A. Z. Ramdani, "FPGA-Based Implementation for Real-Time Epileptic EEG Classification Using Hjorth Descriptor and KNN," *Electronics*, vol. 11, no. 19, Sept. 2022, Art. no. 3026, https://doi.org/10.3390/electronics11193026.

[27] V. B. Bhandari, *Design of Machine Elements*, 5th ed. New Delhi, India: McGraw-Hill Education (India), 2021.