Proceedings of the ASME 2026 Conference on Smart Materials, Adaptive Structures and Intelligent Systems
SMASIS2026
September 13-16, 2026, Albuquerque, New Mexico

SMASIS2026-191347

HIGH-RATE STATE ESTIMATION OF A PRINTED CIRCUIT BOARD UNDER SHOCK USING AN FPGA-DEPLOYED NEURAL NETWORK

Trotter Roberts[1], Joud N. Satme[1], Tiffany Yu[2], Austin R.J. Downey[1,3,*], Rasha E. Karakchi[2], Jason D. Bakos[2]

[1]Department of Mechanical Engineering, University of South Carolina, Columbia, SC 29208
[2]Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208
[3]Department of Civil and Environmental Engineering, University of South Carolina, Columbia, SC 29208

ABSTRACT

Structures operating in high-rate dynamic environments, such as hypersonic vehicles, orbital infrastructure, and blast mitigation systems, experience extreme stresses where rapid decision-making is essential. In these contexts, delays on the order of milliseconds can result in catastrophic failures, making structural health monitoring a critical function. Recent advancements in high-speed sensing technologies, combined with edge-computing architectures, are enabling the integration of high-rate structural health monitoring techniques into real-time programs previously limited by hardware or computational constraints. This work investigates the use of multi-layer perceptrons (MLPs) trained on windowed acceleration time-series data to predict the health state of electronic assemblies. MLPs are selected for their balance of prediction accuracy, computational efficiency, and suitability for hardware acceleration. The trained model is deployed on a field-programmable gate array (FPGA) to evaluate its feasibility in embedded environments. The methodology is assessed with respect to prediction accuracy, inference latency, and hardware resource utilization, highlighting trade-offs between computational performance and practical deployment. The contributions of this study are: (1) the development of a framework for applying MLPs to high-rate structural health monitoring tasks, and (2) the demonstration of their viability on embedded real-time computing platforms.

Keywords: real-time, multi-layer perceptron, field-programmable gate array, structural health monitoring, time series

1. INTRODUCTION

High-rate structural health monitoring (HR-SHM) refers to systems in which sensing, assessment, and decision-making occur on millisecond timescales [1]. In shock and impact environments, structural responses evolve on sub-millisecond timescales, requiring rapid and deterministic inference for effective state estimation. These high-rate loading conditions can induce both mechanical and electrical failures in electronic systems. Large transient accelerations generate stress concentrations within printed circuit boards (PCBs), solder joints, and semiconductor packages, leading to solder-joint fatigue, pad cratering, interconnect fracture, and chip cracking [2]. Under repeated shock exposure, such damage may accumulate without immediate functional failure, degrading long-term reliability and structural integrity [3]. Early detection of this degradation is therefore critical for aerospace, defense, and other safety-critical systems [4].

Existing survivability strategies primarily emphasize mechanical mitigation or post-failure detection. Hardware-based isolation reduces transmitted stress but does not quantify evolving internal damage states [5]. Software-based fault detection methods rely on deviations in system outputs and typically identify faults only after performance degradation has occurred [6]. Data-driven approaches, including neural networks, provide an alternative by learning discriminative features directly from sensor measurements [7, 8]. However, inference on CPUs or GPUs introduces variability in latency due to operating system scheduling and shared computational resources [9]. For safety-critical applications, deterministic execution in dedicated hardware platforms such as FPGAs is preferred [10, 11].

This work develops and evaluates an MLP classifier for shock-induced health estimation of a PCB and deploys the trained model on an FPGA for deterministic inference. The MLP is selected because the classification task is formulated over fixed-length acceleration windows, enabling a feedforward architecture with bounded computational depth [12]. The trained model is evaluated at the impact level to verify feasibility of the classification approach, and the FPGA deployment is evaluated in terms of resource utilization and deterministic timing. Together, these results quantify the feasibility of using FPGA-deployed neural networks for high-rate structural state estimation.

*Corresponding author

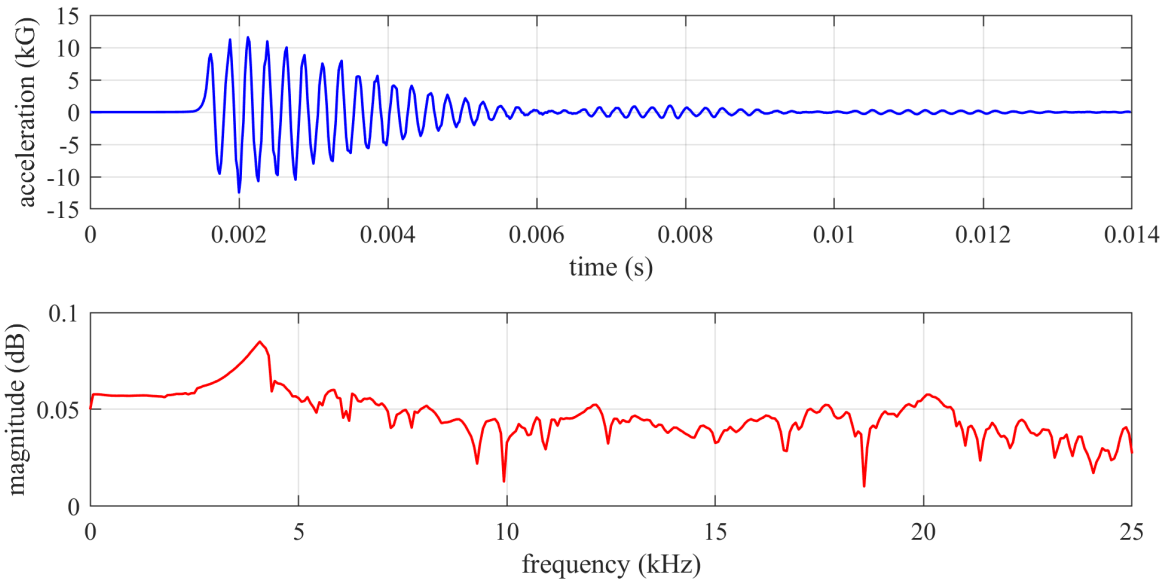

FIGURE 1: Representative PCB acceleration response during a single impact event, shown in (a) the time domain and (b) the frequency domain.

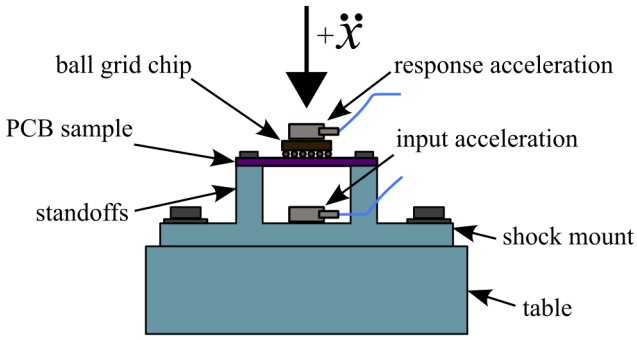

FIGURE 2: Schematic of the shock test configuration and accelerometer placement on the PCB assembly.

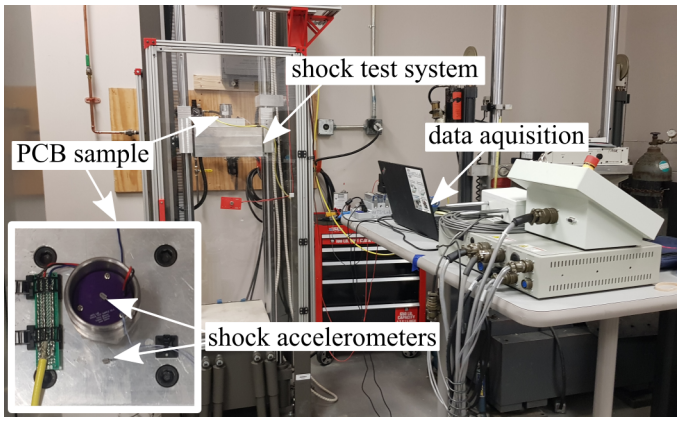

FIGURE 3: Shock test system used to generate controlled impact events on the PCB assembly.

2. METHODOLOGY

2.1. Experimental Setup and Data Preparation

The dataset used in this study was previously developed for online health monitoring of electronic components subjected to repeated high-energy shock [13] and is publicly available at [14]. A ball grid array (CABGA_36, Amkor Technology) was mounted on a PCB and subjected to 30 controlled impact events. Two accelerometers were used during testing: one mounted to the shock mass to measure input acceleration and one mounted to the PCB to measure structural response. The PCB acceleration trace, consisting of 51,200 samples per one-second acquisition, serves as the input signal for classification.

A representative PCB acceleration response during impact is shown in Figure 1. A representative experimental configuration is shown in Figure 2, with the physical shock test system presented in Figure 3. The impacts produced an average peak acceleration of approximately 10,000 g_n with an average half-sine duration of 322 µs, and the PCB exhibits a dominant resonance near 4 kHz.

Aligned traces were trimmed to a fixed length and segmented into overlapping windows of 128 samples with a stride of 64 samples. No input normalization or scaling was applied, reflecting the intended deployment environment, where acceleration signals are streamed directly from a data acquisition system and global statistical properties of future inputs are not known a priori. This design choice prioritizes consistency between training and deployment conditions, though it may limit robustness to variations in signal magnitude.

2.2. Neural Network Architecture and Training

The classifier is implemented as a feedforward multilayer perceptron that maps a fixed-length acceleration window $x \in \mathbb{R}^{128}$ to a binary health-state estimate. The network consists of three hidden layers with widths 32, 16, and 8 neurons, followed by a two-neuron output layer. The model operates on windowed segments of the PCB acceleration signal and produces a window-level health-state prediction.

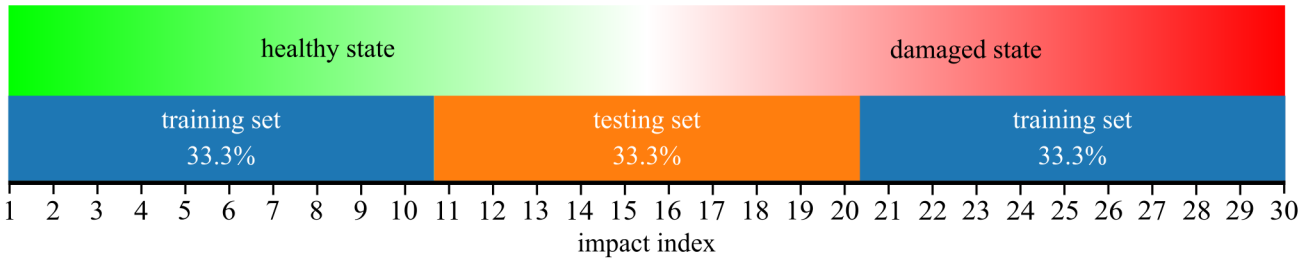

FIGURE 4: Training and validation split across the 30 impact events.

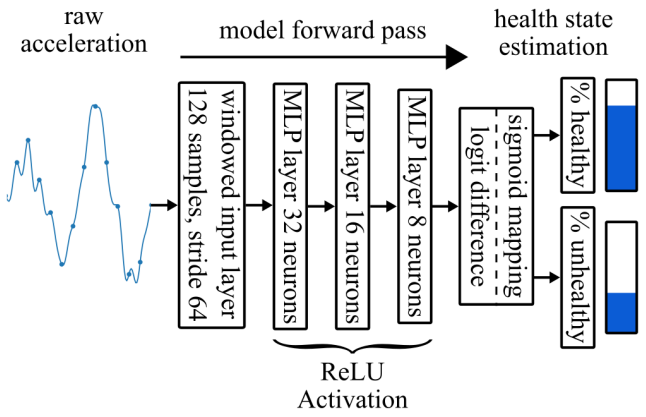

FIGURE 5: Inference pipeline for window-based PCB health-state estimation using an MLP classifier.

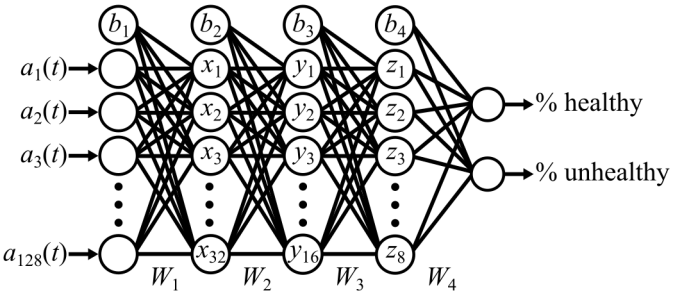

FIGURE 6: MLP architecture used for binary health-state classification.

Forward propagation through layer l is defined as

$$z^{(l)} = W^{(l)} x^{(l-1)} + b^{(l)}, \tag{1}$$

$$x^{(l)} = \max(0,\, z^{(l)}), \tag{2}$$

where the rectified linear unit introduces nonlinearity in the hidden layers. The final layer produces two logits corresponding to the healthy and unhealthy states. For inference, the scalar logit difference

$$d = z_{\text{unhealthy}} - z_{\text{healthy}} \tag{3}$$

is mapped to class probability via

$$P_{\text{unhealthy}} = \frac{1}{1 + e^{-d}}, \qquad P_{\text{healthy}} = 1 - P_{\text{unhealthy}}. \tag{4}$$

Training was performed offline using aligned acceleration traces segmented into overlapping windows of 128 samples with stride 64. Windows extracted from the first ten aligned impacts (healthy) and the final ten aligned impacts (damaged) formed the training dataset, while the middle ten impacts were reserved for validation. The training/validation split follows the chronological progression of the experimental impacts and is therefore best interpreted as a feasibility study rather than a fully generalizable validation of damage classification performance. The dataset partitioning strategy is illustrated in Figure 4, which shows the separation of healthy and damaged states and the corresponding allocation of training and testing subsets across the 30 impact events. Optimization was carried out using the Adam algorithm with mini-batch updates and early stopping. Dropout regularization was applied to the first two hidden layers during training and removed during deterministic FPGA inference.

The resulting trained parameters were exported for FPGA deployment. The trained model parameters and FPGA implementation used in this study are publicly available at [15], enabling reproducibility and further investigation of the proposed approach. Figures 5 and 6 summarize the inference framework used for window-based health-state estimation, including the progression from windowed acceleration input through neural network inference to binary health-state prediction, as well as the multilayer perceptron structure used to perform the classification task.

2.3. FPGA Implementation

The trained network parameters were exported in FPGA-compatible form, including transposed weight matrices and bias vectors. These parameters are stored in on-chip Block RAM (BRAM) and indexed during inference. The FPGA processes one window at a time using a window-triggered execution model: the host transfers a single window, the FPGA executes a bounded forward pass, and a probability value is returned. No continuously running loop is used, ensuring deterministic per-window latency.

Each fully connected layer is implemented using a memory-driven processing loop that maximizes parallelism within the resource constraints of the FPGA. Weight memories are aggressively unrolled, allowing multiple weights to be read simultaneously and paired with the corresponding input features. These input-weight pairs are processed in parallel by multiple floating-point multipliers, and the resulting partial products are accumu-

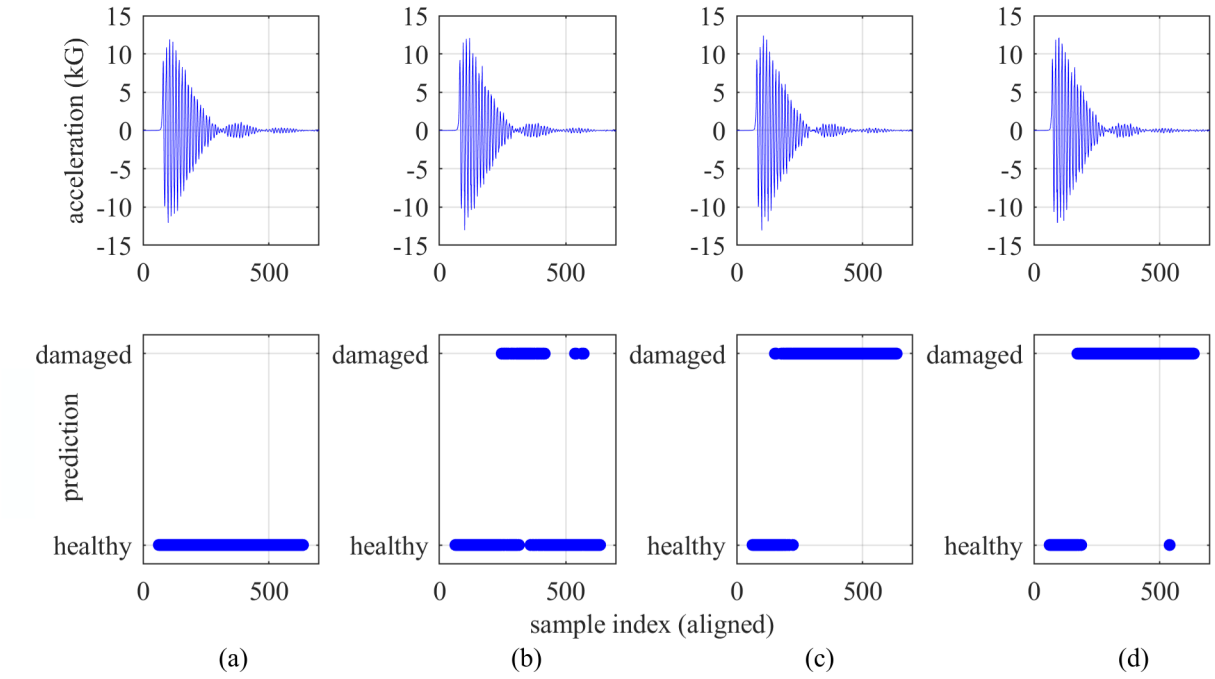

FIGURE 7: Acceleration time series (top row) and corresponding window-level winner vote (bottom row) for Impacts 2, 12, 18, and 29 (left to right).

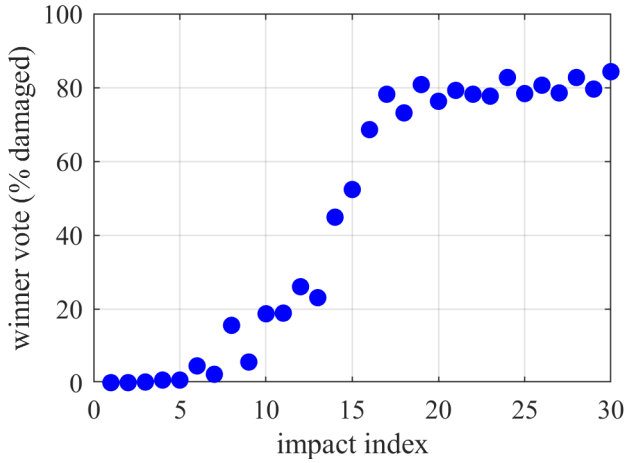

FIGURE 8: Average winning class vote percentage across all 30 impacts, showing progression from healthy to damaged states.

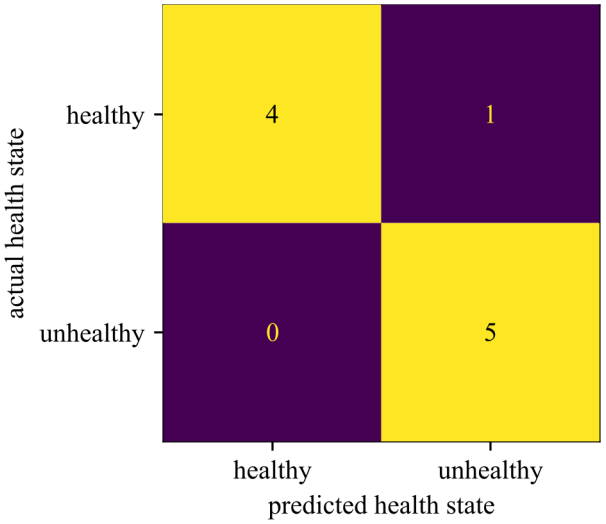

FIGURE 9: Impact-level confusion matrix for the validation impacts.

lated to form each neuron's dot product. The degree of parallelism is selected to maximize throughput while ensuring the complete neural network fits within the available FPGA resources. After accumulation, the neuron bias is added and the ReLu activation function is applied. The resulting activations are stored in local memory and used as inputs to the subsequent layer. All computations are performed using single-precision floating-point (SGL) arithmetic to maintain numerical consistency with the trained software model.

The final stage computes the logit difference and maps it to a scaled probability using a 512-entry sigmoid lookup table stored in BRAM. Because BRAM access incurs one-cycle latency, probability evaluation is implemented using a two-cycle address-read control sequence and a validity register to ensure stable outputs. The FPGA returns a scaled integer probability, which the host converts to floating-point form for interpretation. The resulting datapath consists exclusively of bounded arithmetic operations and deterministic memory accesses, enabling repeatable inference timing suitable for real-time deployment.

3. RESULTS AND DISCUSSION

3.1. Classification Performance

Classifier performance was evaluated at the impact level using the reserved middle ten impacts. Fig. 7 illustrates representative examples from four impacts: one healthy training impact, one from the first half of the validation set, one from the second half of the validation set, and one damaged impact. For each case, the trimmed acceleration time series is shown above the corresponding window-level predicted probabilities, highlighting how classification confidence evolves throughout the impact event. To assess classification behavior across the full dataset, Fig. 8 shows the average winning class vote percentage for all 30 impacts as a function of impact index. The transition from healthy to damaged states is reflected in the increasing dominance of the damaged class prediction beyond approximately the midpoint of the impact sequence.

The trained MLP achieved an impact-level accuracy of 90%, corresponding to 9 correct classifications out of 10 validation impacts. The distribution of impact-level predictions is summarized in the confusion matrix shown in Fig. 9. While this result indicates that the model captures the progression from healthy to damaged behavior using windowed acceleration data, it should be interpreted as a feasibility indicator rather than a comprehensive evaluation of classification performance. The validation is based on a small, progression-based dataset, and the chronological train/validation split may introduce sensitivity to impact ordering rather than fully generalizable damage features. Misclassifications are concentrated near transitional impacts, suggesting that additional data diversity and alternative validation strategies may improve robustness near the damage onset region.

3.2. Hardware Performance

Post-synthesis resource utilization and timing results are shown in Figure 10 and summarized in Table 1. The deployed MLP fully utilizes available slices on the target Kintex-7 device, with 69.5% of lookup tables and 20.8% of registers consumed, as well as total slice usage reaching 84.7%. Block RAM (BRAM) utilization remains moderate at 4.6%, and DSP48 usage is limited to 14.0%, reflecting the sequential multiply-accumulate implementation strategy.

The design met timing constraints at a 40 MHz clock frequency. Direct measurement on the FPGA indicates that inference of a single 128-sample window requires approximately 609 µs (0.609 ms). This latency is comparable to the duration of the shock event, indicating that inference can be completed within a timescale relevant for event-level structural assessment. Because the datapath consists exclusively of fixed-depth loops and bounded memory access patterns, the per-window latency is deterministic and constant across executions, with no variability introduced by operating system scheduling or processor-level resource contention. These results emphasize that the primary contribution of this work is the demonstration of deterministic, bounded-latency inference suitable for high-rate embedded deployment.

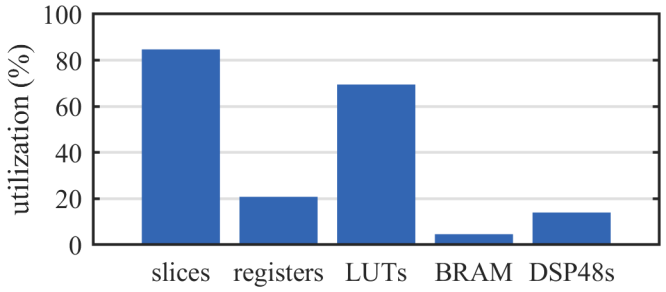

FIGURE 10: Post-synthesis FPGA resource utilization for the deployed MLP.

TABLE 1: Post-synthesis FPGA resource utilization for the deployed MLP.

	slices	registers	LUTs	BRAM	DSP48s
utilized	21479	42230	70488	15	84
total	25350	202800	101400	325	600
percent	84.7%	20.8%	69.5%	4.6%	14.0%

4. CONCLUSION

This study demonstrates the feasibility of deploying a multilayer perceptron on FPGA hardware for shock-induced health-state estimation with deterministic sub-millisecond latency. The implemented system achieved a fixed per-window inference latency of approximately 0.609 ms at a 40 MHz clock frequency, placing it within the high-rate SHM regime. This latency is on the same order as the shock event duration (hundreds of microseconds), indicating suitability for near-real-time, event-level structural state assessment. While the trained model achieved 90% impact-level classification accuracy on the validation impacts, this result should be interpreted as a feasibility indicator due to the limited dataset size and progression-based train/validation split. The primary contribution of this work is the demonstration of predictable, bounded-latency neural network inference for high-rate embedded applications.

Post-synthesis analysis revealed full slice utilization on the target Kintex-7 device, with 83.8% slice usage, highlighting the hardware cost associated with fully connected network architectures. While the implementation confirms the practicality of FPGA-deployed neural networks for high-rate state estimation, logic utilization limits scalability on the selected device. Future work will focus on reducing hardware footprint through architectural compression, reduced-precision arithmetic, and alternative network configurations, as well as expanding the training dataset to improve classification robustness and generalization across a broader range of shock conditions.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation grant numbers CCF - 1956071, CCF-2234921, and CPS - 2237696. Any opinions, findings conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Dodson, Jacob, Downey, Austin, Laflamme, Simon, Todd, Michael D., Moura, Adriane G., Wang, Yang, Mao, Zhu, Avitabile, Peter and Blasch, Erik. *High-Rate Structural Health Monitoring and Prognostics: An Overview*. Springer International Publishing (2021): pp. 213–217. DOI 10.1007/978-3-030-76004-5_23.

[2] Lall, Pradeep, Panchagade, Dhananjay, Iyengar, Deepti, Shantaram, Sandeep, Suhling, Jeff and Schrier, Hubert. "High Speed Digital Image Correlation for Transient-Shock Reliability of Electronics." *2007 Proceedings 57th Electronic Components and Technology Conference*: pp. 924–939. 2007. IEEE. DOI 10.1109/ectc.2007.373908.

[3] Roberts, Trotter, Luck, Hugo, Satme, Joud N., Gol Zardian, Mohsen, Navidi, Sina, Yount, Ryan, Downey, Austin R. J. and Hu, Chao. "Test-to-Failure Characterization and Acceleration Signal Feature Analysis of Electronic Systems Under Repeated Shock Loading." *IMAC XLIV Proceedings*. 2026. In Proceedings of IMAC-XLIV.

[4] Farrar, Charles R. and Worden, Keith. *Structural Health Monitoring: A Machine Learning Perspective*. Wiley (2012). DOI 10.1002/9781118443118.

[5] Amy, Robin Alastair, Aglietti, Guglielmo S. and Richardson, Guy. "Reliability Analysis of Electronic Equipment Subjected to Shock and Vibration – A Review." *Shock and Vibration* Vol. 16 No. 1 (2009): pp. 45–59. DOI 10.1155/2009/546053.

[6] Isermann, Rolf. *Fault-Diagnosis Systems*. Springer Berlin Heidelberg (2006). DOI 10.1007/3-540-30368-5.

[7] Fisco, N.R. and Adeli, H. "Smart structures: Part I—Active and semi-active control." *Scientia Iranica* Vol. 18 No. 3 (2011): pp. 275–284. DOI 10.1016/j.scient.2011.05.034.

[8] Zhong-Ping, Jiang, Tao, Bian and Weinan, Gao. "Learning-Based Control: A Tutorial and Some Recent Results." *Foundations and Trends in Systems and Control* Vol. 8 No. 3 (2020): pp. 176–284. DOI 10.1561/2600000023.

[9] Shawahna, Ahmad, Sait, Sadiq M. and El-Maleh, Aiman. "FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review." *IEEE Access* Vol. 7 (2019): pp. 7823–7859. DOI 10.1109/access.2018.2890150.

[10] Lee, Edward A. "Cyber Physical Systems: Design Challenges." *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*: pp. 363–369. 2008. IEEE. DOI 10.1109/isorc.2008.25.

[11] Kastner, R., Matai, J. and Neuendorffer, S. "Parallel Programming for FPGAs." *ArXiv e-prints* (2018)DOI https://doi.org/10.48550/arXiv.1805.03648. URL 1805.03648.

[12] Goodfellow, Ian, Bengio, Yoshua and Courville, Aaron. *Deep learning*. Adaptive computation and machine learning, The MIT Press, Cambridge, Massachusetts (2016).

[13] Satme, Joud N., Coble, Daniel, Downey, Austin R. J. and Goshorn, Devon. "Online Health Monitoring of Electronic Components Subjected to Repeated High-Energy Shock." *Proceedings of The ASME 2024 Conference on Smart Materials, Adaptive Structures and Intelligent Systems (SMASIS2024)*. 2024. American Society of Mechanical Engineers. DOI 10.1115/smasis2024-139950.

[14] Downey, Austin, Dodson, Jacob, Vereen, Alex, Satme, Joud and Nelson, Chris. "Dataset 5 Extended Impact Testing." GitHub (2022). URL https://github.com/High-Rate-SHM-Working-Group/Dataset-5-Extended-Impact-Testing.

[15] Roberts, Trotter, Yu, Tifffany and Downey, Austin R.J. "Paper 2026 Ultra High-rate State Estimation." GitHub (2026). URL https://github.com/ARTS-Laboratory/Paper-2026-Ultra-High-rate-State-Estimation.