Proceedings of the ASME 2022
International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC/CIE 2022
August 14-17, 2022, St. Louis, United States

DETC2022-90118

PROGRESS TOWARDS DATA-DRIVEN HIGH-RATE STRUCTURAL STATE ESTIMATION ON EDGE COMPUTING DEVICES

JOUD SATME
Department of Mech. Eng.
University of South Carolina
Columbia, South Carolina 29201
Email: Jsatme@email.sc.edu

DANIEL COBLE
Department of Mech. Eng.
University of South Carolina
Columbia, South Carolina 29201
Email: DNCOBLE@email.sc.edu

BRADEN PRIDDY
Department of Mech. Eng.
University of South Carolina
Columbia, South Carolina 29201
Email: bpriddy@email.sc.edu

AUSTIN R.J. DOWNEY
Department of Mech. Eng.
Department of Civil, Const. and Env. Eng.
University of South Carolina
Columbia, South Carolina 29201
Email: austindowney@sc.edu

JASON D. BAKOS
Department of Comp. Sci. and Eng.
University of South Carolina
Columbia, South Carolina 29201
Email: jbakos@cse.sc.edu

GURCAN COMERT
Department of Comp. Sci., Phy., and Eng.
Benedict College
Columbia, SC 29204
Email: gcomert@illinois.edu

ABSTRACT

Structures operating in high-rate dynamic environments, such as hypersonic vehicles, orbital space infrastructure, and blast mitigation systems, require microsecond (µs) decision-making. Advances in real-time sensing, edge-computing, and high-bandwidth computer memory are enabling emerging technologies such as High-rate structural health monitoring (HR-SHM) to become more feasible. Due to the time restrictions such systems operate under, a target of 1 millisecond (ms) from event detection to decision-making is set at the goal to enable HR-SHM. With minimizing latency in mind, a data-driven method that relies on time-series measurements processed in real-time to infer the state of the structure is investigated in this preliminary work. A methodology for deploying LSTM-based state estimators for structures using subsampled time-series vibration data is presented. The proposed estimator is deployed to an embedded real-time device and the achieved accuracy along with system timing are discussed. The proposed approach has shown potential for high-rate state estimation as it provides sufficient accuracy for the considered structure while a time-step of 2.5 ms is achieved. The Contributions of this work are twofold: 1) a framework for deploying LSTM models in real-time for high-rate state estimation, 2) an experimental validation of LSTMs running on a real-time computing system.

NOMENCLATURE

HR-SHM	High-rate structural health monitoring.
DROPBEAR	Dynamic Reproduction of Projectiles in Ballistic Environments for Advanced Research.
LSTM	Long short-term memory.
RNN	Recurrent neural networks.
FEA	Finite element analysis.
$\mathrm{SNR}_{\mathrm{dB}}$	Signal-to-noise ratio in decibels.
RMSE	Root-mean-square eror.

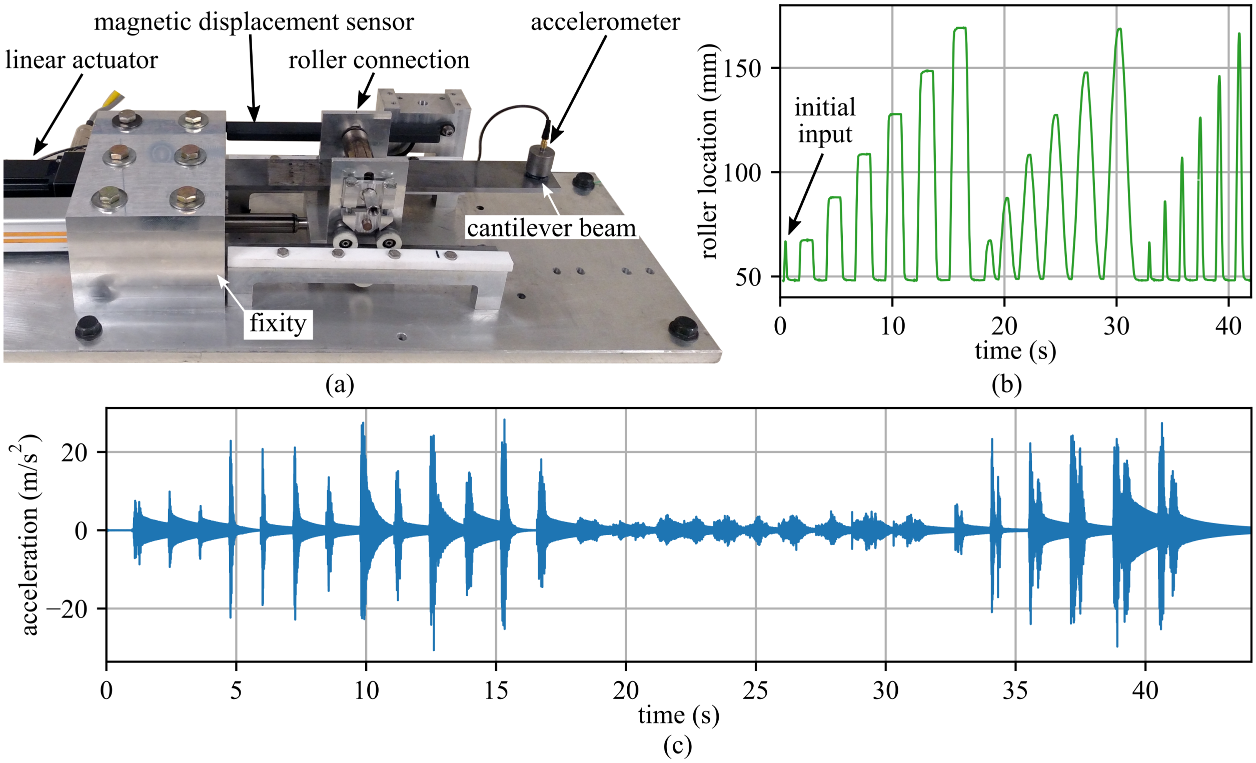
ADC Analog to digital convertor.
DAC Digital to analog convertor.

1 INTRODUCTION

Structures subjected to impact loading that results in accelerations of greater than 100 g_n during time periods of less than 100 ms are considered to be structures operating in high-rate dynamic environments [1]. High rate structural health monitoring (HR-SHM) and prognostics is an emerging field focused on highly dynamic engineering systems that are being enabled through the recent introduction of real-time sensing, edge-computing, and high-bandwidth computer memory [2]. The goal of HR-SHM is to target the 1 ms timescales from event detection to decision-making. The timing deadline of 1 ms does not include the time it takes to execute decisions, such as the energizing of active structural components, changes in desired outcomes, or mission cancellation and device termination. Potential high-rate applications that would benefit from the development of HR-SHM include super and hyper-sonic vehicles, orbital space infrastructures, and active blast mitigation systems [3–5].

Real-time state estimation of structural state is essential to enabling high-rate decision-making for structures operating in extreme dynamic environments. However, traditional model-based state-estimation techniques are slow and operate on the order of seconds to hours [6, 7]. Prior work by the authors has demonstrated that finite element analysis (FEA) can be used to track the state of a simple cantilever beam-based setup [8]. This work demonstrated that a 40 element 1-D Euler-Bernoulli beam model could be solved for every 4.04 ms while providing an overall error of 2.9%. These results were obtained using parallel-optimized code on a 2.3 GHz eight-core PXI controller (PXIe-8880 manufactured by NI). Solving the generalized eigenvalue problem to obtain the systems frequency components is the most time-consuming aspect of the presented model-based approach and accounts for over 3.6 ms of the total 4.04 ms. It is envisioned that the FEA model used for a real system experiencing high-rate dynamics will be a relatively complex 3D model with thousands of FEA nodes, however, due to its $O(n^3)$ complexity, the generalized eigenvalue formulation scales poorly for larger FEA models. It becomes evident that model-based approaches are poorly suited for obtaining state-estimations on the μs timescale.

Data-driven approaches offer the potential to link complex time-series measurements obtained from a structure to an estimation of the structure's state in real-time. In particular, the long short-term memory (LSTM) is well suited for inferring a state



(a)
(b)
(c)

FIGURE 1. The Dynamic Reproduction of Projectiles in Ballistic Environments for Advanced Research (DROPBEAR) experimental testbench, showing: (a) the physical setup, (b) the measured roller movement profile which induces vibrations into the cantilever beam, and (c) the measured acceleration data taken during the test.

from time-series data due to its ability to keep a memory of previous inputs within the model's framework. For an LSTM, the time and storage complexity is $O(n^2)$ where n is the number of LSTM units [9, 10]. Where the number of needed LSTM units is a model hyperparameter that must be set. The de-coupling of the model complexity from the structure's physical models offers the potential to greatly improve high-rate state estimation, moving it closer to the stated 1 ms time-step for a system of moderate complexity.

This work reports on the deployment of an LSTM to track the state of a benchtop structural testbed in real-time. To achieve this, an LSTM model is deployed to an edge-scale computing device (1.33 GHz Dual-Core Intel Atom) running a Linux-based real-time operating system. A methodology for developing a multi-cell LSTM and determining the number of units needed to accurately track the state of the structure is presented. An experimental investigation is undertaken using an edge-computing device demonstrating an achieved time step of 2.5 ms. The contributions of this work are twofold: 1) a framework for deploying LSTM models to real-time computing systems for tracking the state of structures operating in high-rate dynamic environments is introduced, and 2) an experimental validation of LSTMs running on a real-time computing system is undertaken. The code has been made available in a public repository.

2 Background

This section presents background on the experimental testbench and a mathematical formulation of LSTM models.

2.1 DROPBEAR experimental testbed

The Dynamic Reproduction of Projectiles in Ballistic Environments for Advanced Research (DROPBEAR), introduced and modeled by Joyce et al. [11], was used to generate the experimental data used in this work. DROPBEAR is presented in figure 1(a) and in its present configuration is a cantilever beam featuring controllable roller support that moves to alter the "condition" or "state" of the beam. The automated roller can produce a repeatable change in the system's state that is analogous to damage in the structural system. The cantilever beam measures 51 × 6 × 350 mm beam with a single accelerometer (PCB Piezotronics - 393B04) mounted near the end of the beam and digitized using a 24-bit IEPE ADC (NI-9234). As shown in figure 1(b), the roller followed a profile ranging from 48 mm (closest to the fixity) to 175 mm.

The beam is self-excited by the roller's movements and therefore no extraneous inputs are required, however, this does require an initial input to the beam as annotated in figure 1(b) to initiate vibrations in the beam. The test profile consists of square, sinusoidal, and impulse inputs each in sets of six with increasing amplitude. The maximum roller movement was limited by the experimental setup to 250 mm/s. The time-series acceleration response of the beam is presented in figure 1(c). The data used in this work is available through a public repository [12].

2.2 Long-Short Term Memory

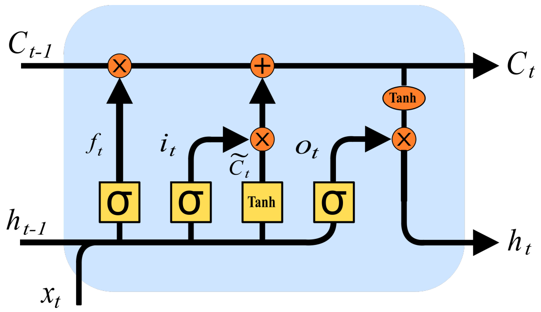


FIGURE 2. Data flow through an LSTM cell with a forget gate.

Traditional recursive neural networks (RNN) are deep learning algorithms designed to deal with a variety of complex tasks where the data of interest is a sequence of events that occur in succession. In these applications, each new data point is tied to the previous data point through some underlying information. For example, a classic use for RNNs is speech recognition where the information that ties the individual words together forms a complete thought [13]. Ideally, deeper RNNs have a longer memory period and better capabilities. However, this is not achievable in real-world applications due to the challenge of backpropagating through the network. This is known as the vanishing gradient problem. Consequently, RNNs are typically only useful for shorter data sequences as the depth of their useful memory is limited.

LSTMs overcome the vanishing gradient problem by maintaining a constant error, allowing them to learn over sequential data. Consider the typical formulation for an LSTM with forget gates, as depicted in Fig. 2. The compact form of the equations for a forward pass of an LSTM is [9, 10]:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{1}$$
$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{2}$$
$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{3}$$
$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c) \tag{4}$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \tag{5}$$
$$h_t = o_t \circ c\sigma_h(c_t) \tag{6}$$

where h_t is the output vector of the LSTM unit at time t. Note that the input vector x_t is transformed to the output vector (h_t) through a series of point-wise operations, sigmoid (σ_g) and hyperbolic tangent (σ_h) activation functions, and the Hadamard product $(\circ)$. W and U represent the weights of the input and recurrent connections while b represents the bias term. W, U, and b are learned through backpropagation. The subscript denotes the input gate (i), output gate (o), forget gate (f), or the memory cell (c).

3 METHODOLOGY

This section presents the development of an LSTM for deployment on a real-time edge device and presents the setup for its experimental validation.

3.1 LSTM Model Development

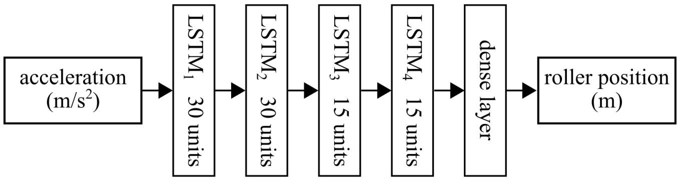


FIGURE 3. LSTM algorithm architecture used in this work and deployed onboard the real-time edge device.

The deployment of LSTMs to edge computing devices for high-rate state estimation results in significant constraints in the size of the model that can be deployed and the speed at which data can be fed to the model. Figure 3 shows the general LSTM model architecture being investigated in this paper. The first layer is a sequence input layer, which takes sequence input data and splits it into the correct vector size. To fit the number of hidden units within the subsequent LSTM cell(s). This is followed by a regression output layer. This LSTM model architecture was chosen after an ad hoc investigation into various model architectures and hyper-parameters while considering hardware limitations. All model configurations were simulated before implementing them onto a real-time system.

The LSTM models were trained on the data produced by DROPBEAR as shown in figure 1(c). The acceleration data is used to estimate the state (position) of the roller, therefore acceleration is the input to the model, and predicted pin location is the output. The acceleration data is digitized at 25.6 kS/s, which is considered over-sampled. Sub-sampling this data without losing information on the dynamics of interest would greatly reduce the computational load. To sub-sample the data, the time series data was mapped into the frequency domain to find the important frequency components and to calculate the max down-sample factor to fit the Nyquist sampling theorem. A down-sample factor of 64 was chosen for both the experimental and simulated models.

The LSTM cell within the model architecture has many hyperparameters that can be adjusted. Optimization of the hyperparameters was performed using a grid search that was run over select parameters for 1000 epochs on one LSTM cell. The parameters considered were hidden unit size, initial learning rate, learning rate drop factor, and batch size. Each run was trained using the Adam optimizer. Figure 4 reports the root mean squared error (RMSE) measured between the real and predicted roller locations for LSTM architectures with 1 to 4 cells over an increasing number of hidden units per cell. Note that while an increase in the number of cells and units tends to help the model learn the complexities of the data set and return predictions with less error; they require greater system resources that may make them infeasible for deployment on edge devices.

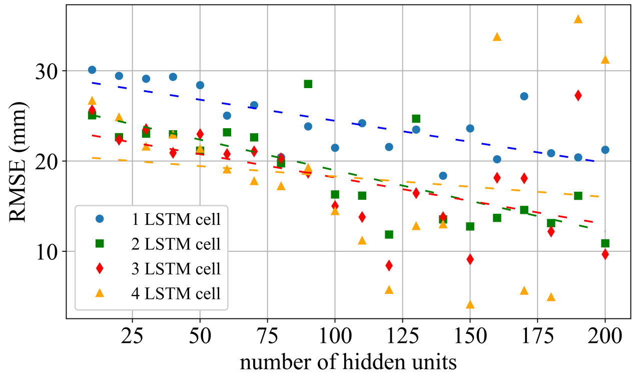


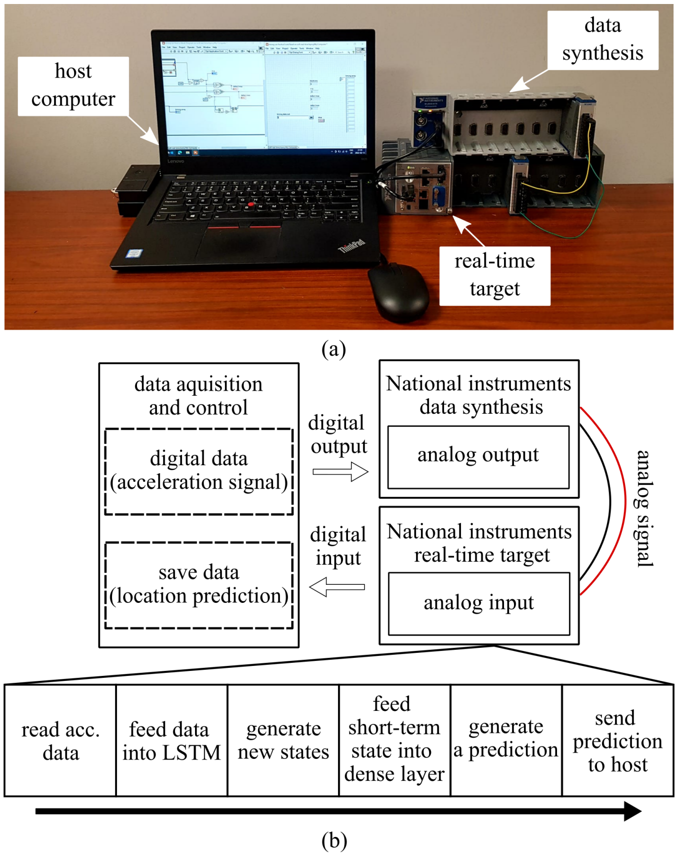
FIGURE 4. RMSE values of various model architectures with different numbers of cells and units.

The model configuration chosen for hardware implementation was a four-stacked LSTM with cell units of 30, 30, 15, and 15, as well as a densely connected top with no activation function, as shown in figure 3. This architecture was chosen as it was shown to have high accuracy compared while still maintaining an efficient model size. The cell architecture was programmed in the LabVIEW visual programming environment, with LSTM weights stored as constants.

3.2 Real-time Edge Implementation

To gauge the algorithms performance, an experiment is constructed with two subsystems, a host machine, and a real-time target machine, as presented in figure 5. The test setup is composed of two devices, a real-time target, and a data synthesis device. The purpose of the data synthesis device is to reproduce the data obtained by Downey et al. [8, 12] as an analog voltage. This setup allows the real-time target to function as if it was directly deployed on a structure obtaining measurements from an accelerometer, without simulating data buffers. The DROPBEAR acceleration data is imported from a file located on the host PC. Preprocessing of this data includes scaling the data by the same factor used in training, keeping the same sampling rate. This data is streamed using a cDAQ-9178 and NI-9262 digital-to-analog (DAC) module to produce an analog signal simulating the accelerometer's signal shown in figure 1(c).

The real-time target digitizes the analog voltage and treats it as acceleration data. Data is digitized using a NI-9201 12-Bit analog-to-digital (ADC) module with a maximum sampling



(a)

(b)

FIGURE 5. Experimental validation showing: (a) physical hardware used, and (b) block diagram of data flow within the physical hardware.

rate of 500 kS/s. No clock synchronization between the data synthesis and data acquisition systems was used. The rate of data acquisition is set to 1/64th that of data synthesis so that the downsampling performed in model training is replicated. In this initial work, the rates of synthesis and sampling are varied to investigate timing and noise, however, the downsampling factor remains a constant 64 throughout the duration of this work to provide a true metric of performance. The acceleration signal is then fed into the LSTM architecture deployed onboard the real-time processor. The LSTM model runs within a timed loop and will return the state prediction within the period of the timed loop. State predictions (e.g., roller locations) are returned to a first-in-first-out buffer back to the host PC for analysis and error calculation.

The real-time target edge device used in this work is a cRIO-9035 manufactured by National Instruments which allows for the easy integration of a 1.33 GHz dual-Core Intel Atom (E3825) with data acquisition. In this work, a NI-9201 12-bit ADC with a maximum sampling rate of 500 kS/s is used to re-digitize the analog signal. The real-time time step was set to 2.5 ms, therefore, every 400 times per second data digitization and a forward pass of the LSTM is performed. The real-time target utilizes the LabVIEW Real-Time environment and used NI Linux Real-Time, a proprietary real-time OS developed by NI from open-source code. Mainly, NI Linux Real-Time is built on PREEMPT_RT [14, 15] which aims to improve the determinism of the Linux kernel itself.

Limitations of this experiment revolve around the DAC's ability to reproduce signals with high slew rates and therefore, provide a true reconstruction of the voltage signal. Simulating the vibration signature of the test structure was shown to be challenging as the acceleration signals are rich in high-frequency components that could not be reproduced on the current hardware. Furthermore, discretization and reconstruction with a zero-order hold introduce disturbances that deviate from the desired reference signal. To expand, when the synthesis and prediction are performed at the same speed as during the DROPBEAR experiment, a complete pass through the data takes approximately 45 s.

4 RESULTS AND DISCUSSION

In this section, the findings of the real-time deployment of the LSTM architecture is presented. Performance was examined using signal-to-noise ratio measured in decibels ($\mathrm{SNR}_{\mathrm{dB}}$), RMSE, and execution time.

The position of the roller pin predicted by the LSTM model is presented in figure 6. Figure 6(a) reports the predicted pin locations at 400S/s. Note that the LSTM has difficulties reproducing the peaks of the roller movement shapes as this is where the acceleration signals include the highest frequency components, and the DAC used lacked the adequate bandwidth to accurately synthesize such signals. The accuracy demonstrated by the LSTM model for state estimation shows the promise of data-driven approaches for tracking system health.

An investigation into the consistency of the real-time operation was performed and is reported in figure 7. Results reported here are for the LSTM at the 2.5 ms time step as the real-world processing speed is the performance threshold of interest. Recorded results demonstrate an average time step of 2.5 ms with a standard deviation of 0.004 ms and a skew of -1.277×10^{-14}. Where the standard deviation and skew are measurements of jitter in the real-time operating system. Importantly, the max recorded time is 2.519 ms which results in a timing overshoot of 0.019 ms over the desired time step of 2.5 ms, as shown in table 1.

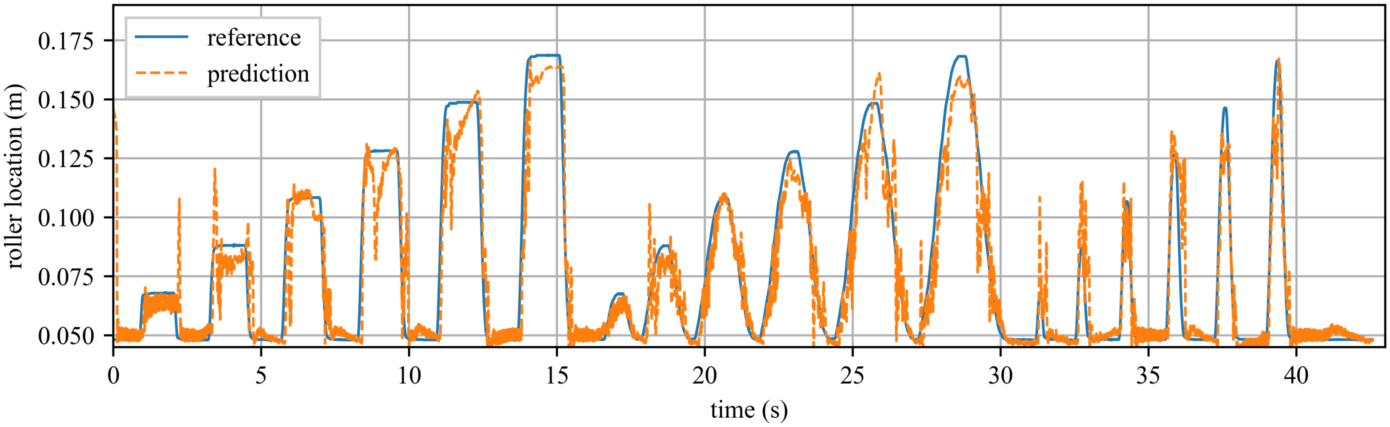


FIGURE 6. Results for the high-rate LSTM-based state estimator showing the predicted values at 400 S/s.

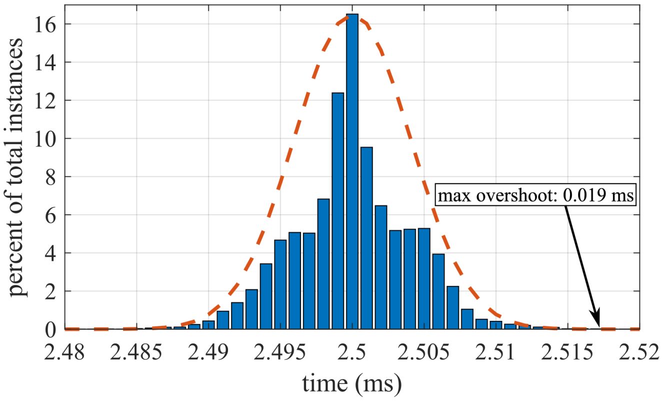


FIGURE 7. Timing distribution of the LSTM forward path at a 2.5 ms time step.

TABLE 1. Algorithm execution timing report.

Mean	2.5 ms
Standard deviation	0.004 ms
Max overshoot	0.019 ms

5 CONCLUSION

In this preliminary stage, the design and implementation of an LSTM-based state-estimation framework were presented. This method demonstrates the feasibility of deployment on a real-time edge device and an ad hoc methodology for selecting the size of the hyperparameters for the LSTM architecture. In order to achieve the rate of 400 Hz, the acceleration signal was downsampled to reduce computation. This study shows that LSTMs offer a viable path forward for high-rate state estimation as they can achieve accurate state estimations for structures subjected to dynamic environments. Results demonstrated that a $\mathrm{SNR}_{\mathrm{dB}}$ of 43.2 and an RMSE of 12.8 mm could be achieved. Moreover, a time-step of 2.5 ms was demonstrated with a maximum overshoot of 0.019 ms. Future work will involve validating the accuracy of the signal replication. Additionally, decreasing the time from receiving the measurement to issuing a prediction will be investigated. The algorithm has been made available in a public repository [16].

ACKNOWLEDGMENT

This material is based upon work partially supported by the Air Force Office of Scientific Research (AFOSR) through award no. FA9550-21-1-0083. This work is also partly supported by the National Science Foundation Grant numbers 1850012 and 1956071. The support of these agencies is gratefully acknowledged. 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 or the United States Air Force.

REFERENCES

[1] Hong, J., Laflamme, S., Dodson, J., and Joyce, B., 2018, “Introduction to state estimation of high-rate system dynamics,” *Sensors,* **18**(2), jan, p. 217.

[2] Dodson, J., Downey, A., Laflamme, S., Todd, M. D., Moura, A. G., Wang, Y., Mao, Z., Avitabile, P., and Blasch, E., 2021, “High-rate structural health monitoring and prognostics: An overview,” In *Data Science in Engineering, Volume 9*. Springer International Publishing, oct, pp. 213–217.

[3] Stein, C., Roybal, R., Tlomak, P., and Wilson, W., 2000, "A review of hypervelocity debris testing at the air force research laboratory," *Space Debris,* **2**(4), pp. 331–356.

[4] Hallion, R. P., Bedke, C. M., and Schanz, M. V., 2016, *Hypersonic Weapons and US National Security: A 21st Century Breakthrough* Mitchell Institute for Aerospace Studies.

[5] Wadley, H., Dharmasena, K., He, M., McMeeking, R., Evans, A., Bui-Thanh, T., and Radovitzky, R., 2010, "An active concept for limiting injuries caused by air blasts," *International Journal of Impact Engineering,* **37**(3), mar, pp. 317–323.

[6] Rainieri, C., Fabbrocino, G., and Cosenza, E., 2011, "Near real-time tracking of dynamic properties for standalone structural health monitoring systems," *Mechanical Systems and Signal Processing,* **25**(8), nov, pp. 3010–3026.

[7] Astroza, R., Ebrahimian, H., and Conte, J. P., 2019, "Performance comparison of kalman-based filters for nonlinear structural finite element model updating," *Journal of Sound and Vibration,* **438**, jan, pp. 520–542.

[8] Downey, A., Hong, J., Dodson, J., Carroll, M., and Scheppegrell, J., 2020, "Millisecond model updating for structures experiencing unmodeled high-rate dynamic events," *Mechanical Systems and Signal Processing,* **138**, apr, p. 106551.

[9] Hochreiter, S., and Schmidhuber, J., 1997, "Long short-term memory," *Neural Computation,* **9**(8), nov, pp. 1735–1780.

[10] Gers, F., 1999, "Learning to forget: continual prediction with LSTM," In 9th International Conference on Artificial Neural Networks: ICANN '99, IEE.

[11] Joyce, B., Dodson, J., Laflamme, S., and Hong, J., 2018, "An experimental test bed for developing high-rate structural health monitoring methods," *Shock and Vibration,* **2018**, jun, pp. 1–10.

[12] Downey, A., Hong, J., Dodson, J., Carroll, M., and Scheppegrell, J., 2020, Dataset-2-dropbear-acceleration-vs-roller-displacement, Apr.

[13] Kim, C., Gowda, D., Lee, D., Kim, J., Kumar, A., Kim, S., Garg, A., and Han, C., 2020, "A review of on-device fully neural end-to-end automatic speech recognition algorithms," In 2020 54th Asilomar Conference on Signals, Systems, and Computers, IEEE.

[14] McKenney, P. A realtime preemption overview. lwn.

[15] Reghenzani, F., Massari, G., and Fornaciari, W., 2020, "The real-time linux kernel," *ACM Computing Surveys,* **52**(1), jan, pp. 1–36.

[16] Satme, J., Coble, D., Priddy, B., Downey, A., Bakos, J., and Comert, G., 2022, Progres towards data-driven high-rate structural state estimation on edge computing devices Tech. rep., 3 https://github.com/ARTS-Laboratory/Paper-Progress-towards-data-driven-high-rate-structural-state-estimation-on-edge-computing-devices.