Online back-propagation of recurrent neural network for forecasting nonstationary structural responses

Zhymir Thompson[1], Alex Vereen[2], Austin R.J. Downey[3,4], and Jason D. Bakos[1], Jacob Dodson[4], Adriane G. Moura[5]

[1]Department of Mechanical Engineering
[2]Department of Computer Science and Engineering
[3]Department of Civil & Environmental Engineering
[4]Air Force Research Laboratory
[5]Applied Research Associates

UNIVERSITY OF SOUTH CAROLINA

High-rate Dynamic Events

- High-rate dynamic events [1]
  - time scale of less than 100 milliseconds
  - high amplitude exceeding 100 gn
- High-rate Structural Health Monitoring[3]
  - “Monitor functional integrity and remaining life”
  - “Maximize function, minimize risk”

Car collision [1]

South Carolina

High-rate Dynamic Events Challenge

- Structural changes
  - Structure can change significantly during an event
  - Changes are often permanent and change dynamics of structure in varying degrees
- Model prediction of structures
  - Imagine the King in checkers
  - Dynamics used by model are altered, model must relearn dynamics

Car collision [1]

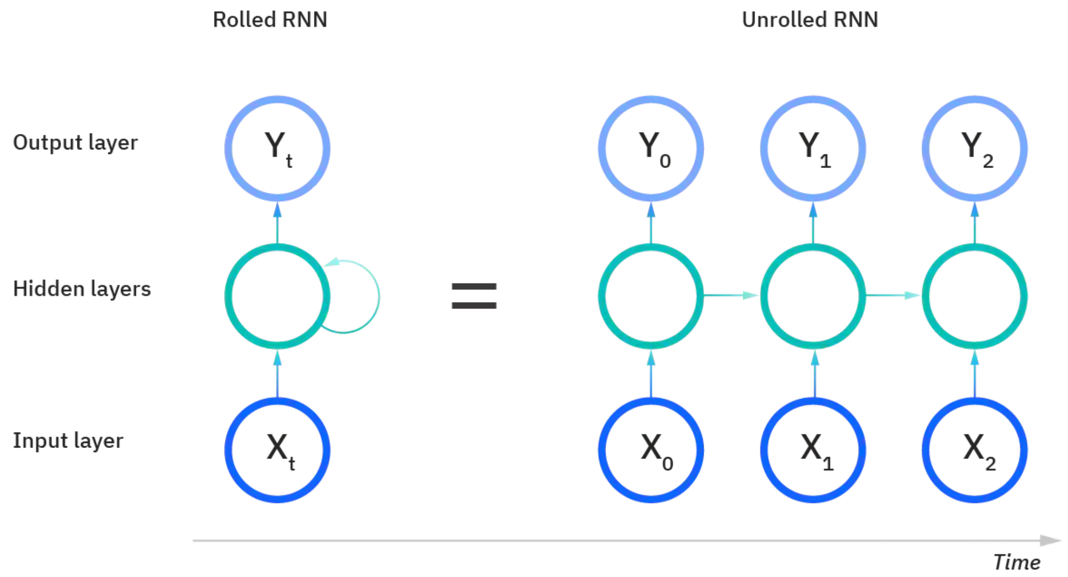
South Carolina

RNN: Recurrent Neural Network

A machine learning model that maintains state through time:

- Takes in sequence of inputs in sequential order instead of independent batches
- Maintains state in hidden layer to learn with short term memory

A typical neural network takes one input at a time or batches where each sample is independent of others

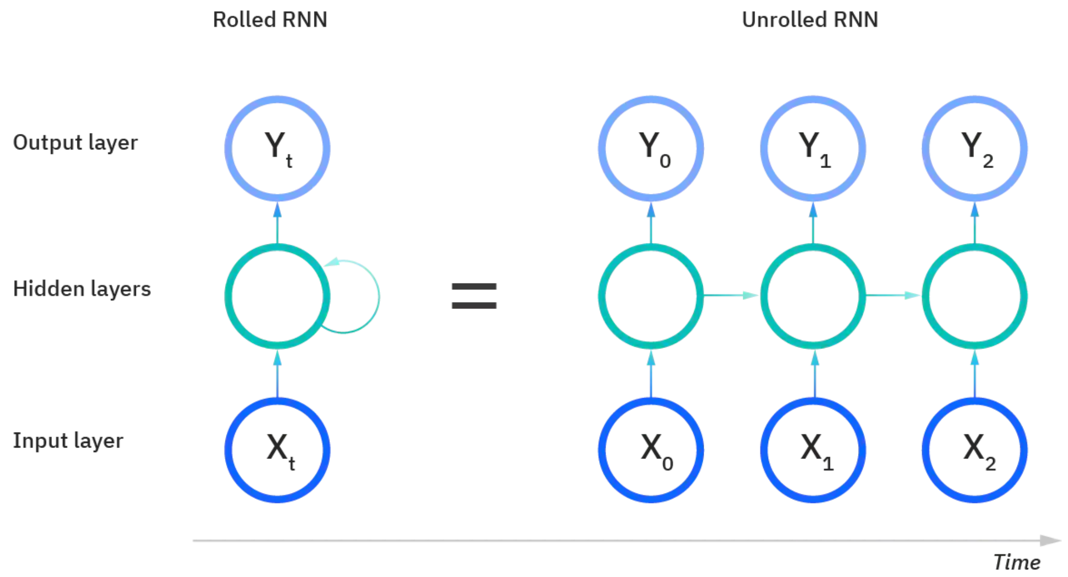


IBM RNN image [1]

South Carolina

RNN

- On forward propagation:
  - Input and hidden state passed in initially
  - Output prediction and hidden state passed back into RNN until desired iterations complete
- On backpropagation:
  - Backpropagates ‘through time’
  - Compounded loss with decay over state

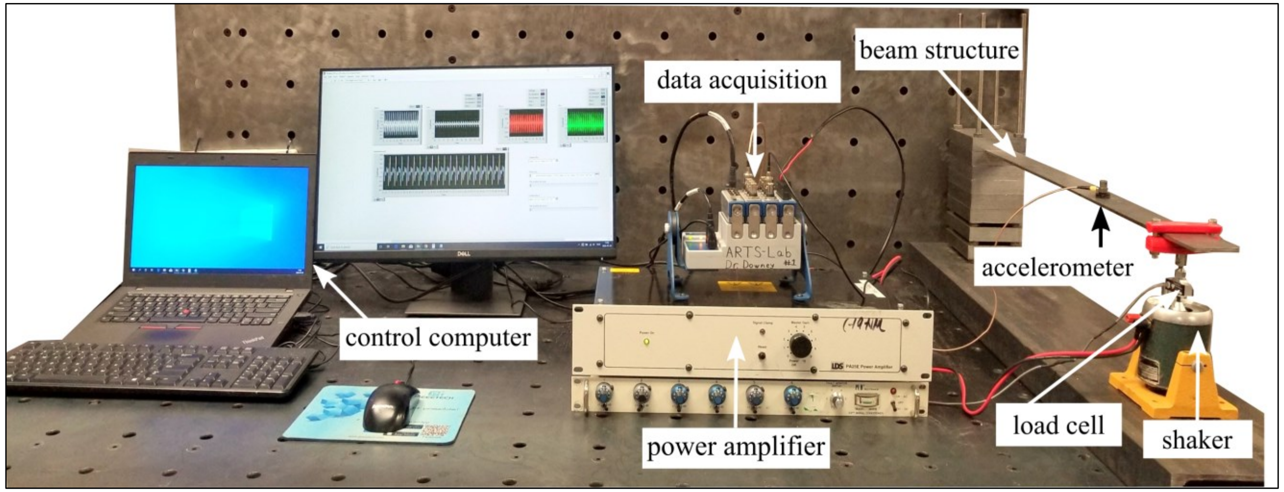


IBM RNN image [1]

South Carolina

Cantilever Beam Experimental Setup

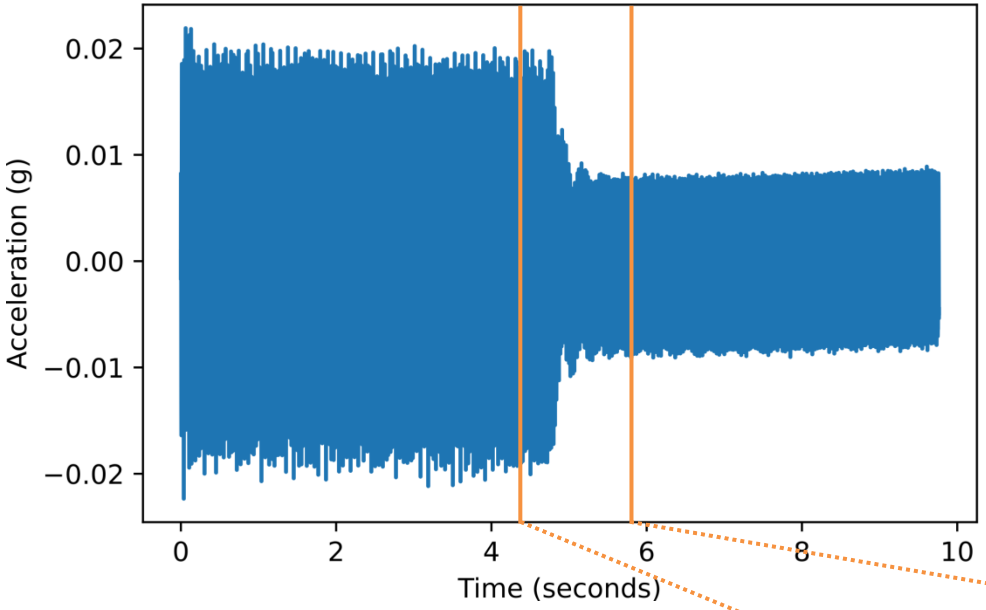
- Cantilever Beam[2]
  - Steel
  - 759x 50.66 x 5.14 mm
- Piezoelectric accelerometer [2]
  - 0.5-9,000 Hz

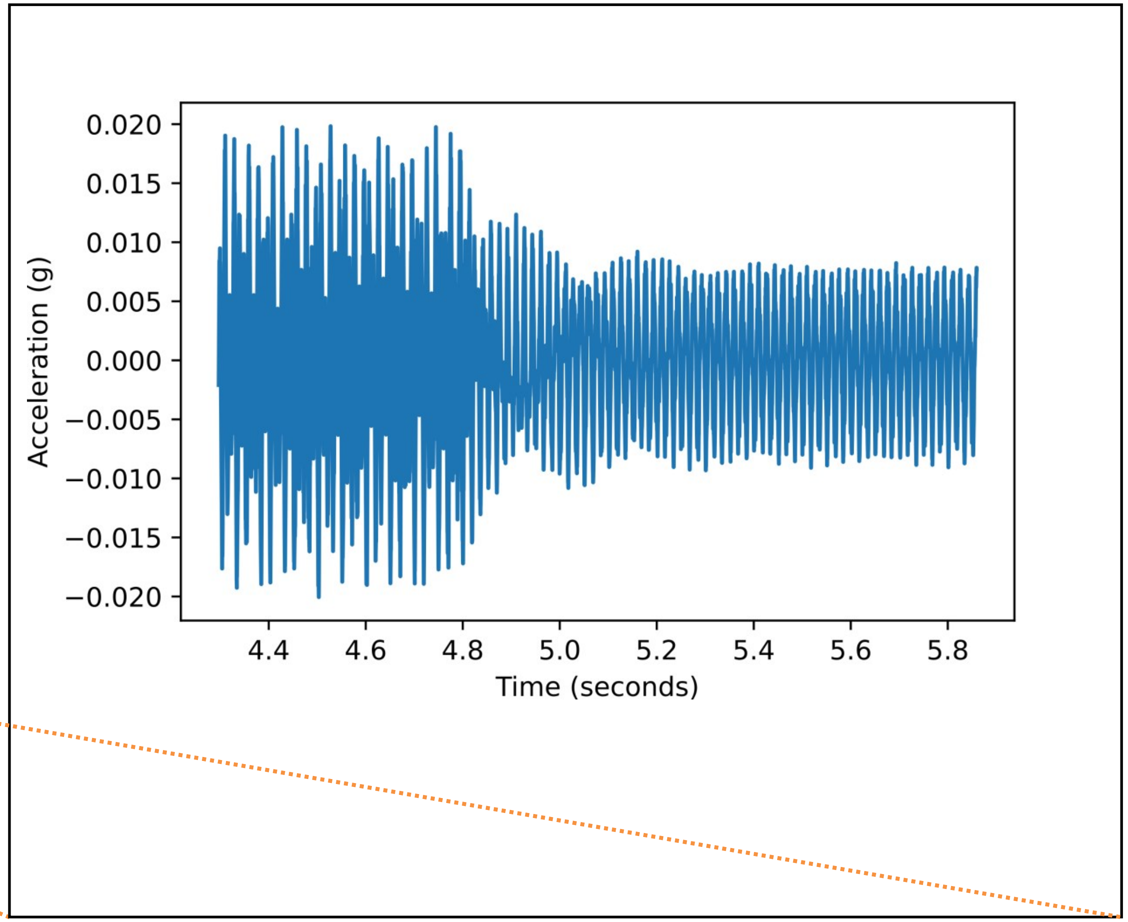


Experiment setup [2]

- Sensitivity: 100 mV/gElectromagnetic shaker applies sine force to beam [2]
- Nonstationarity introduced during the experiment [2]
- More info here: https://github.com/High-Rate-SHM-Working-Group/Dataset-4-Univariate-signal-with-non-stationarity [2]

The non-stationarity

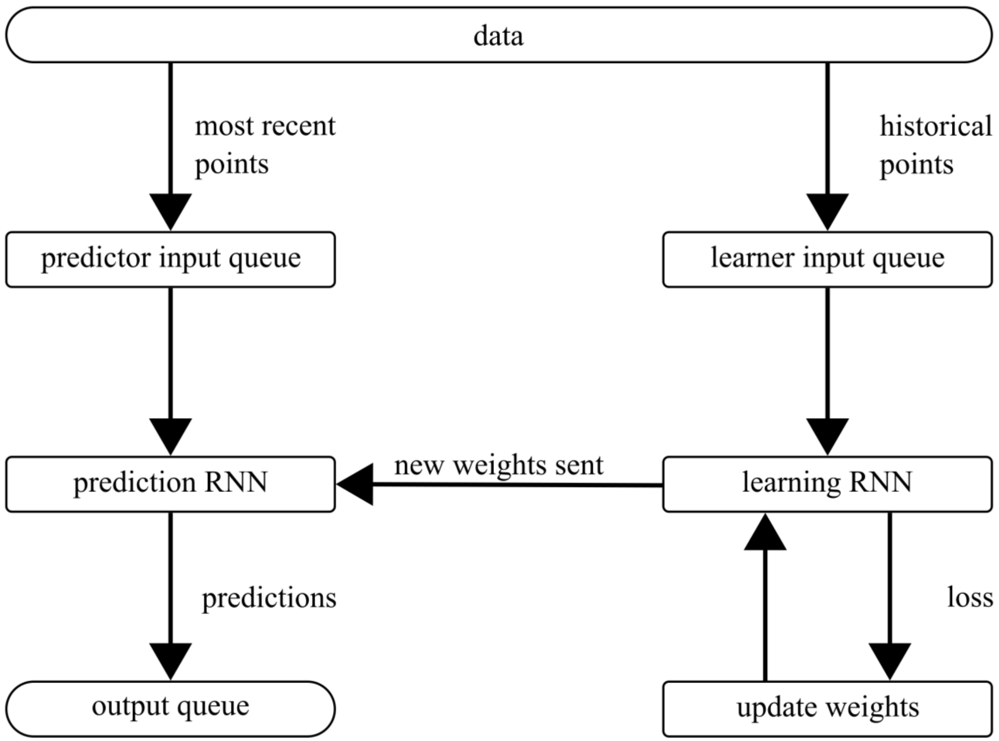




The beam excitation elicits a two regimes of frequency response separated by some “non-stationarity” demarcating the two regimes

PaiRNN Design

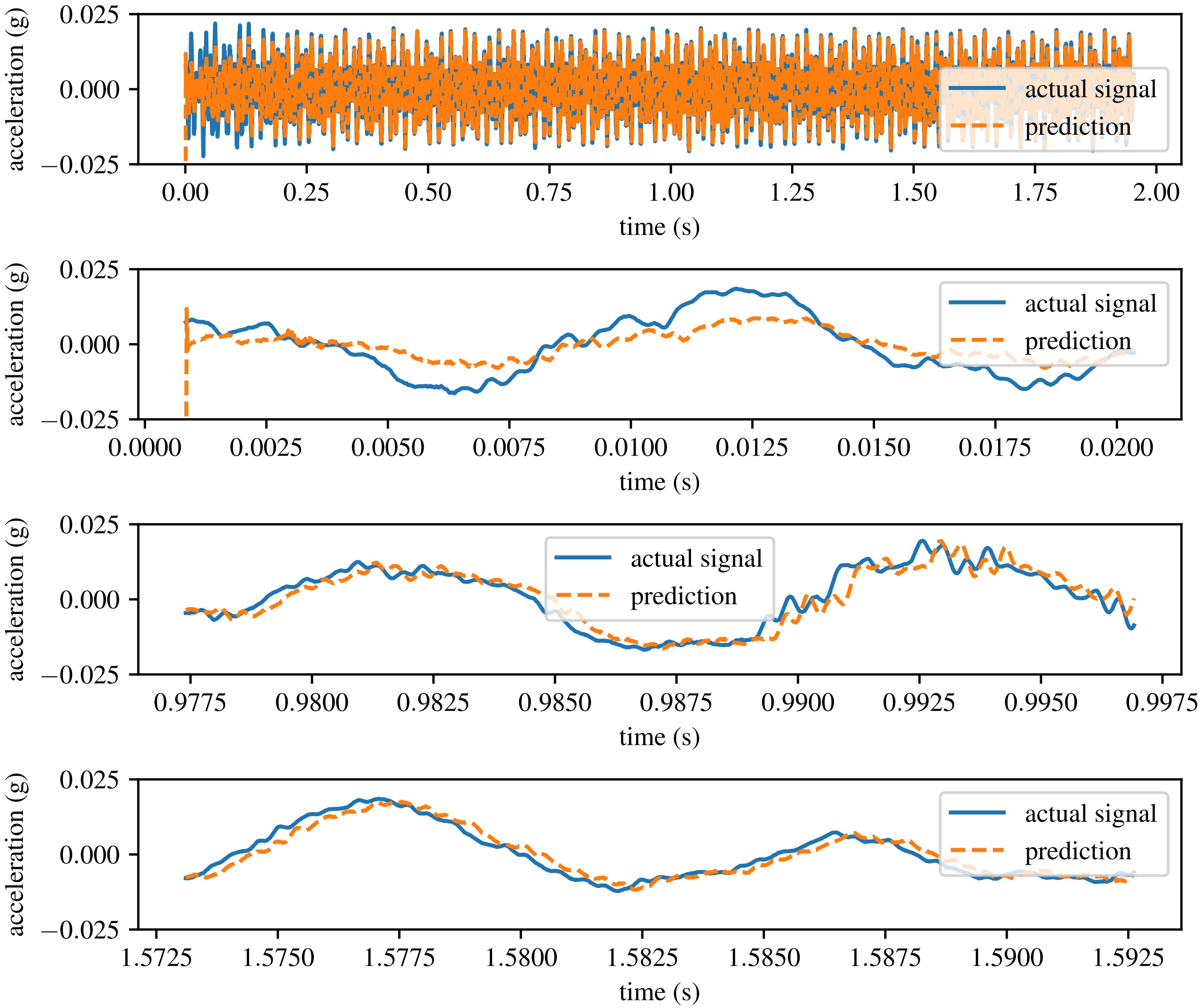
- Live learning model architecture consisting of dual instantiation of an RNN model:
  - Predictor – Makes inferences on future given current data; given weights derived from Learner
  - Learner – Makes inferences on current data given historical data; adjusts weights w.r.t. observed loss
  - Each shares 9 weights total; 8 input + 1 hidden
- Queues – (FIFO) Containers to receive data from stream for model inferencing



South Carolina

PaiRNN Performance

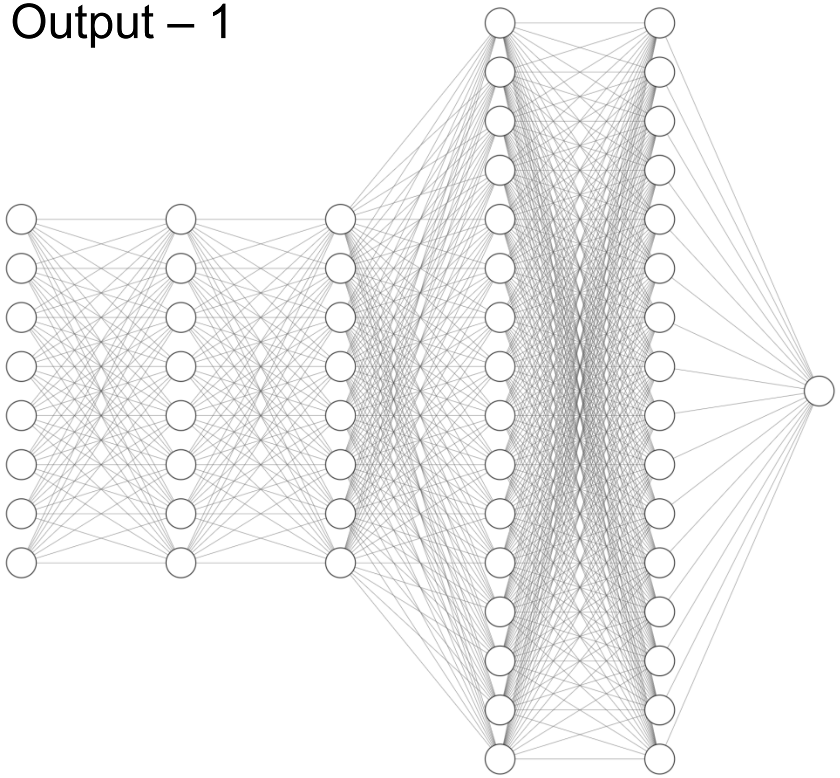
- Compared prediction vs observed signal at various time slices
- Converges within 1 second
- Adjusts well albeit imperfectly to signal post-nonstationarity event
- Each loop takes 500 microseconds (upper-bound on average)



Model Architecture

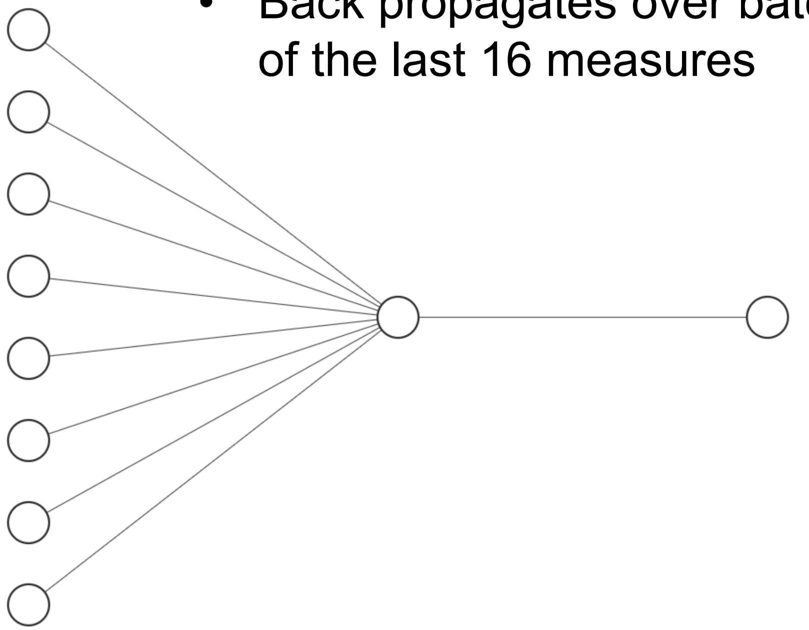
MLP

- Input – 8
  - The 8 prior measurements
- Hidden – 8,8,16,16
- Output – 1



PaiRNN

- Input – 8
  - The 8 prior measurements
- Hidden – 1
- Output – 1
- Back propagates over batches of the last 16 measures



ieering
and Computing

Comparison

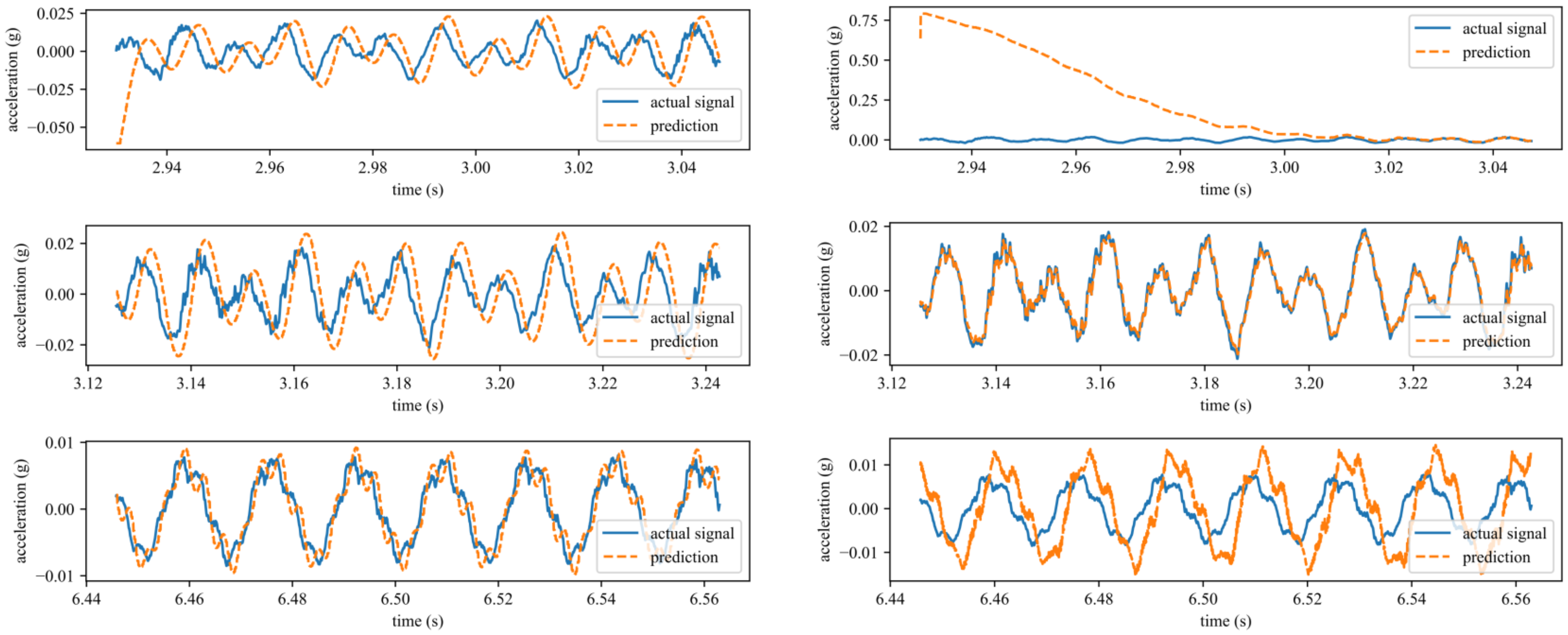
MLP

PaiRNN

Beginning of training

Stationary signal before non-stationarity

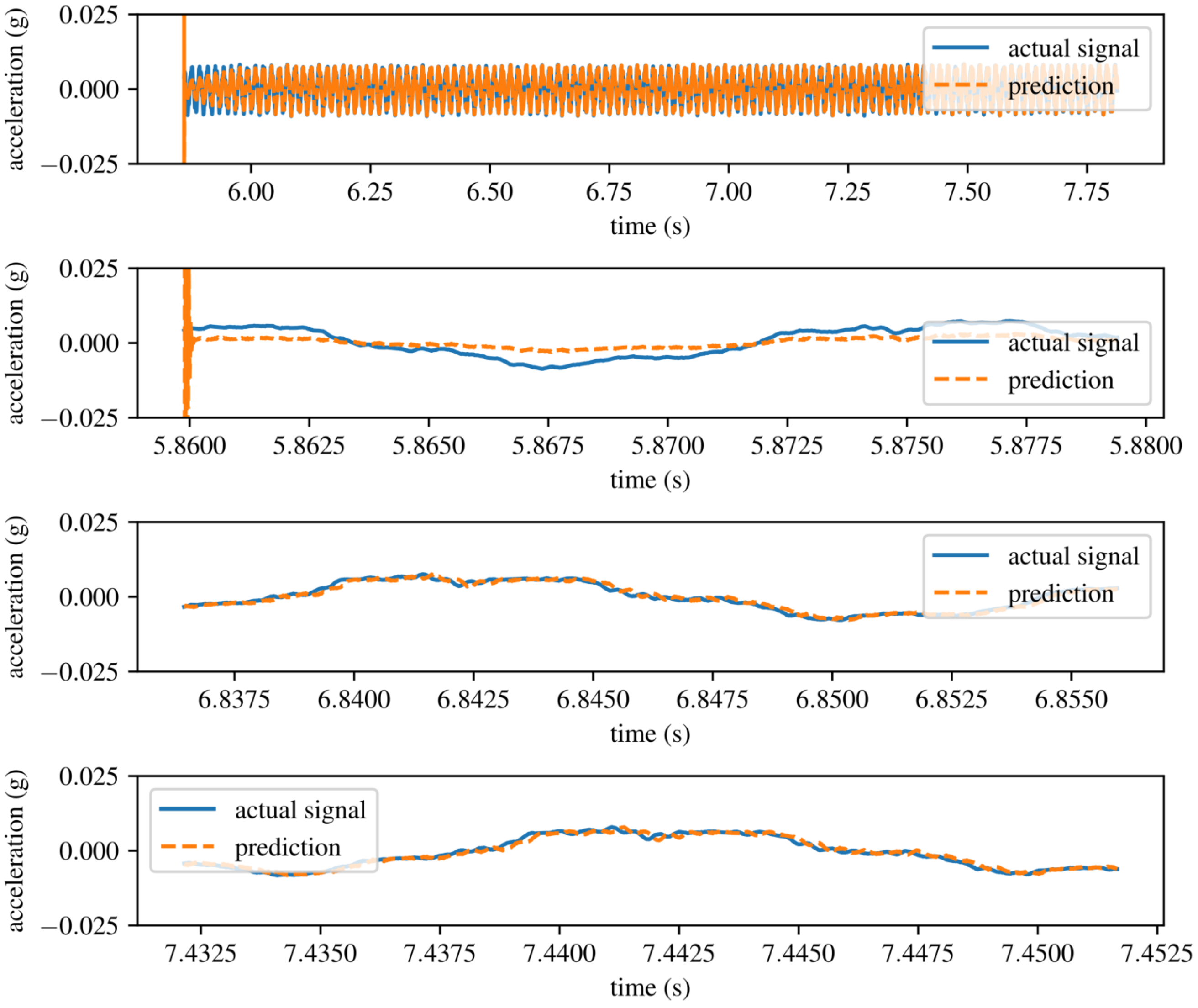
After non-stationarity



UofSC Engineering and Computing

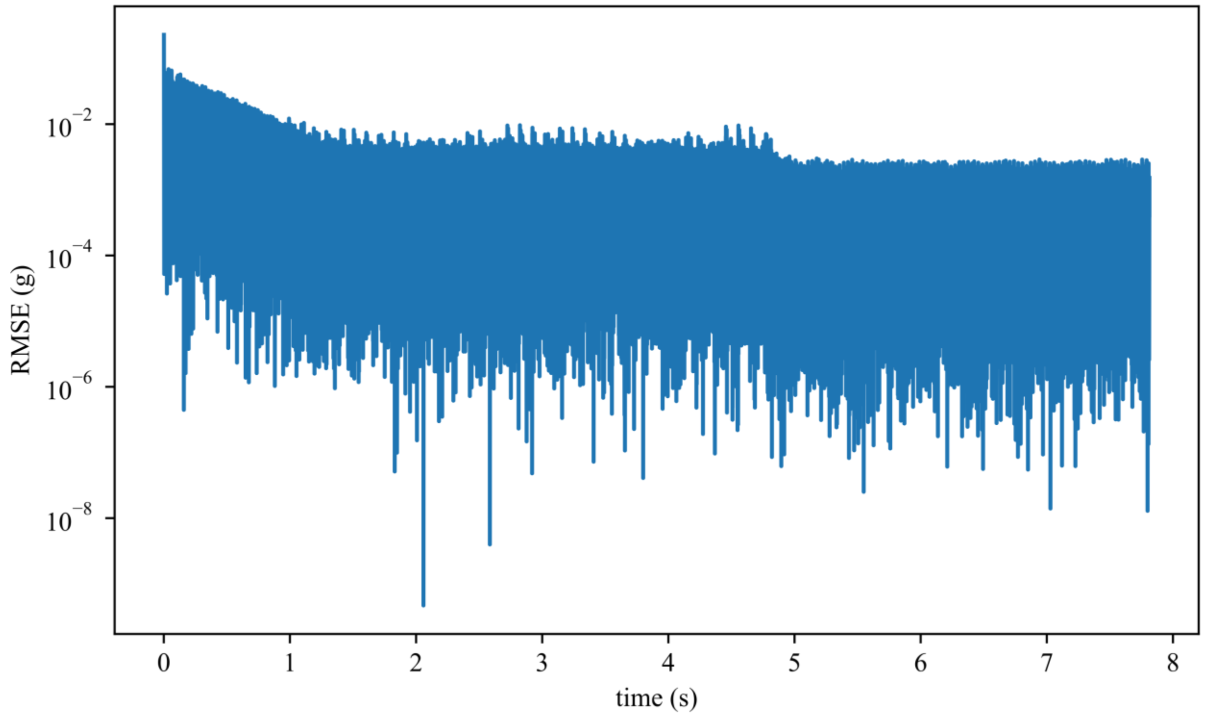
Post Nonstationarity Event

- PairRNN trained on stationary signal after nonstationarity event



Predictor Error

- Chart of predictor error over online session
- Model loss shows exponential decrease until 1.5 seconds followed by plateau
- Model shows another slight drop around the time of the nonstationarity event (though this is likely due to the overall drop in amplitude rather than improved learning)



Signal Metrics

Signal to Noise Ratio (SNR):

$$SNR_{dB} = 10 \log_{10}\left(\frac{P_{signal}}{P_{noise}}\right)$$

Time Response Assurance Criterion (TRAC):
Score between [0,1] the similarity between time traces by comparing the numerical error and time delay of each estimation. A TRAC score of 1 means perfect timing alignment, while a score of 0 means no temporal correlation between signals.

$$TRAC = \frac{\left[S_{ref}^{T} \cdot S_{gen}\right]^2}{\left[S_{ref}^{T} \cdot S_{ref}\right]\left[S_{gen}^{T} \cdot S_{gen}\right]}$$

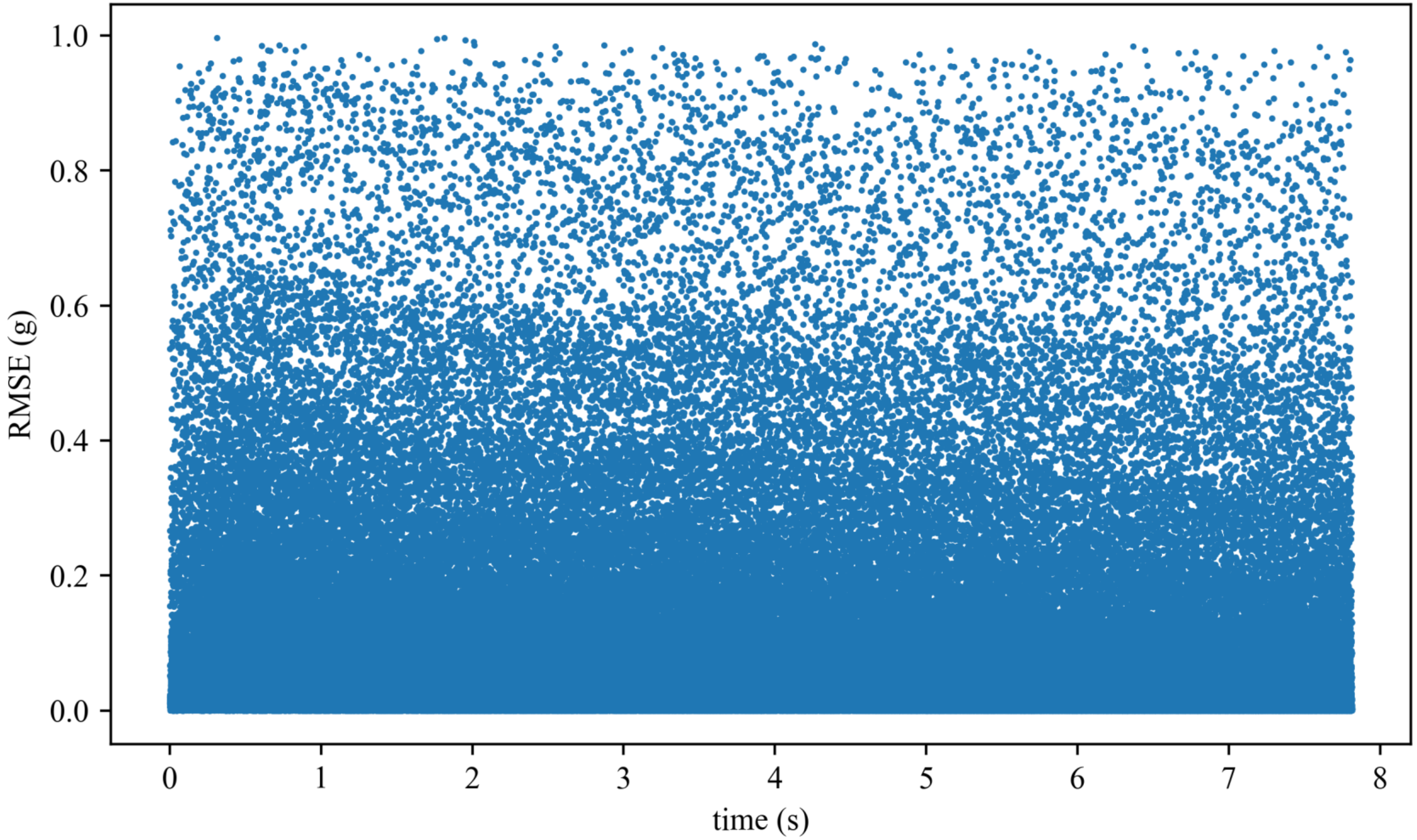
time frame	SNR_{dB}	TRAC
pre-nonstationarity	59.139	0.99913
pre-nonstationarity	54.567	0.99998

UofSC South Carolina

Funding

Learner Loss

- Chart of learner loss throughout online training
- Learner has repeated spikes of loss, but most loss values are below 0.6g



South Carolina

Conclusion

The paper demonstrates that RNNs are capable of concurrently inferencing and learning in an online setting.

South Carolina

References

- langthim, marcel. (2016). *Car collision test*. pixabay. Retrieved January 4, 2022, from https://pixabay.com/photos/crash-test-collision-1620592/
- Puja Chowdhury, Austin Downey, Jason D. Bakos and Philip Conrad, "Dataset-4-univariate-signal-with-nonstationarity," Apr. 2021. [Online]. Available: https://github.com/High-Rate-SHM-Working-Group/Dataset-4-Univariate-signal-with-non-stationarity
- https://www.ibm.com/cloud/learn/recurrent-neural-networks

South Carolina