Using LSTM Recurrent Neural Networks to Predict Excess Vibration Events in Aircraft Engines

AbdElRahman ElSaid[*], Brandon Wild[†], James Higgins[†],
Travis Desell[*]

Department of Computer Science[*], *Department of Aviation*[†]
University of North Dakota

UNIVERSITY OF NORTH DAKOTA

Motivation: General Aviation Safety

General aviation comprises 63% of all civil aviation activity in the United States; covering operation of all non-scheduled and non-military aircraft [2, 4].

While general aviation is a valuable and lucrative industry, it has the highest accident rates within civil aviation [3].

For many years, the general aviation accident and fatality rates have hovered around 7 and 1.3 per 100,000 flight hours, respectively [1].

1. B. Elias. Securing general aviation. DIANE Publishing, 2009.
2. K. I. Shetty. Current and historical trends in general aviation in the United States. PhD thesis, Massachusetts Institute of Technology Cambridge, MA 02139 USA, 2012.
3. National Transportation Safety Board (NTSB), 2012. https://www.ntsb.gov/safety/mwl5_2012.html
4. Aircraft Owners and Pilots Association(AOPA), January 2014. http://www.aopa.org/About-AOPA/Statistical-Reference-Guide/General-Aviation-Safety-Record-Current-and-Historic.aspx

Motivation: The National General Aviation Flight Database

The National General Aviation Flight Information Database (NGAFID) has been developed at the University of North Dakota as a central repository for general aviation flight data. It consists of per-second flight data recorder (FDR) data from three fleets of aircraft.

As of June 2016, the database stores FDR readings from over 300,00 flights, consisting of over 550,000 flight hours with more being added daily. It currently stores over 1.2 billion per-second records of flight data (~2TB). The NGAFID provides an invaluable source of information about general aviation flights, as most of these flights are from aviation students, where there is a wider variance in flight parameters than what may normally be expected within data from professionally piloted flights.

Motivation: The National General Aviation Flight Database

Time series flight data for this work was gathered from the NGAFID, and this has been made available publicly for other interested researchers:

http://people.cs.und.edu/~tdesell/ngafid_releases.php

Motivation: Flight Data Prediction

Having the ability to predict flight parameters based on multiple other parameters as input is a first step towards developing sensors which can intelligently detect anomalous behavior or predict accident precursor behavior. Bringing machine learning strategies into flight data analysis and accident prediction has great potential for preventing future accidents in a proactive manner.

Further, these same strategies can be used to predict and prevent hardware failures or suggest pre-emptive maintenance, reducing costs for airlines.

UND UNIVERSITY OF NORTH DAKOTA

Motivation: Flight Data Prediction

Various parameters contribute to engine vibration:

- engine design
- size
- service life span
- aircraft type
- placement
- weather
- pilot action
- etc.

Motivation: Flight Data Prediction

Much work has been done to generate physical models to predict vibration, however these are tied to all these parameters which may not be readily available.

The goal is to create a system which can generically predict vibration using FDR data.

Long-Short-Term-Memory Recurrent Neural Networks

"Learning to store information over extended period of time intervals via recurrent backpropagation takes a very long time, mostly due to insufficient, decaying error back flow."

- S. Hochrieter & J. Schmidhuber [5]

5. Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

Long-Short-Term-Memory Recurrent Neural Networks

Typical flight data involves between 10-100s of flight parameters gathered potentially multiple times per second and potentially asynchronously. An average flight in the NGAFID with 1 hz sampling has ~5800 per second records.

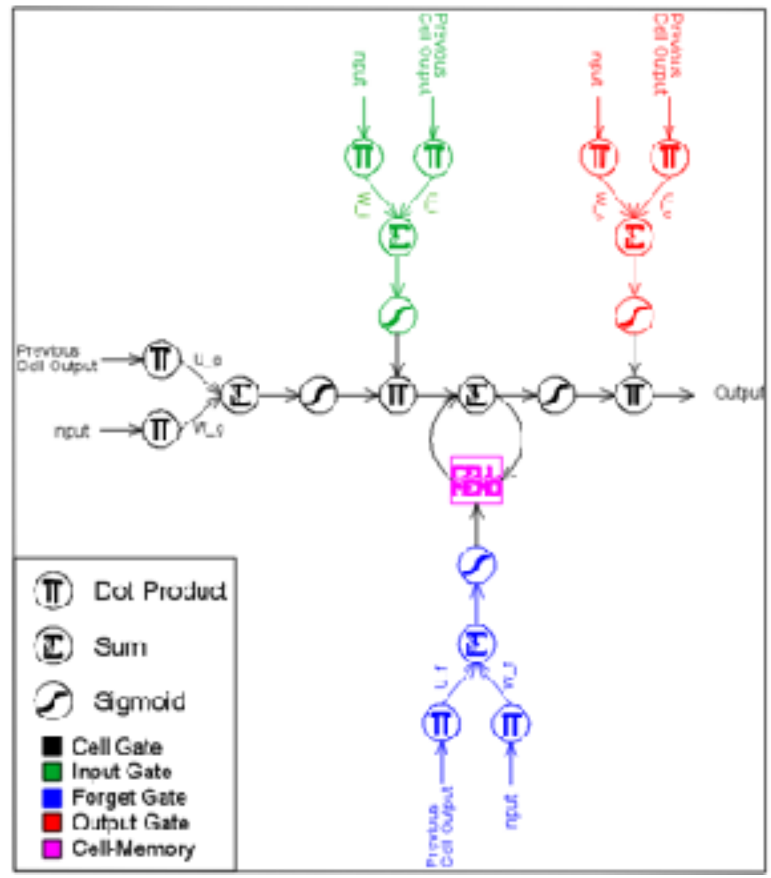
LSTM RNNs provide a solution to training what would otherwise be extremely deep RNNs.

LSTM Cell Design

The following gates control flow through a LSTM neuron:

1. the input gate, which controls how much information will flow from the inputs of the cell
2. the forget gate, which controls how much information will flow from the cell-memory
3. the output gate, which controls how much information will flow out of the cell.

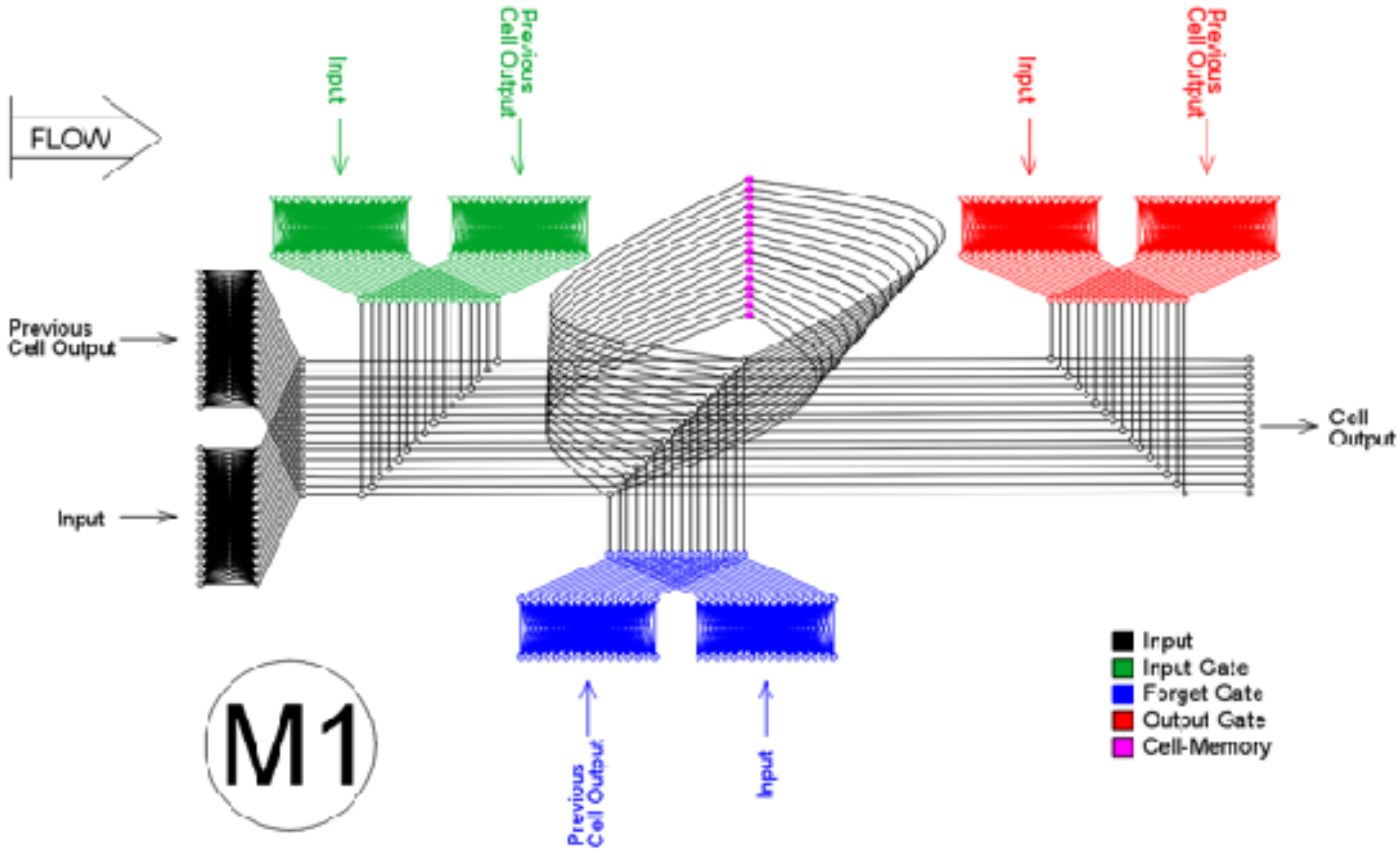
This design allows the network to learn not only about the target values, but also about how to tune its controls to reach the target values



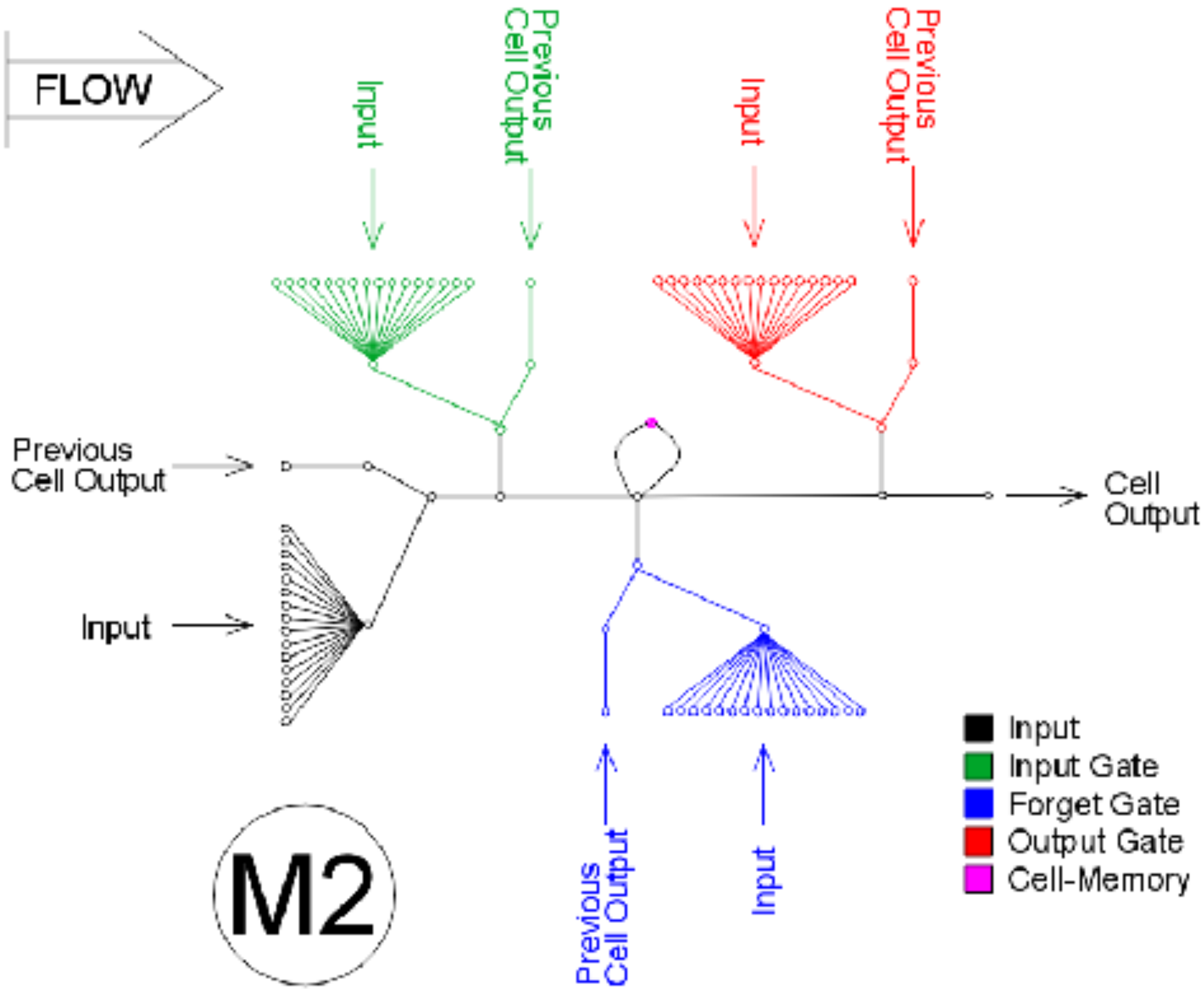
LSTM RNN Architectures

LSTM neurons were arranged into three different architectures and trained to predict the vibration paramter 5, 10 and 20 seconds into the future.

First Layer(s) LSTM Cells (M1)

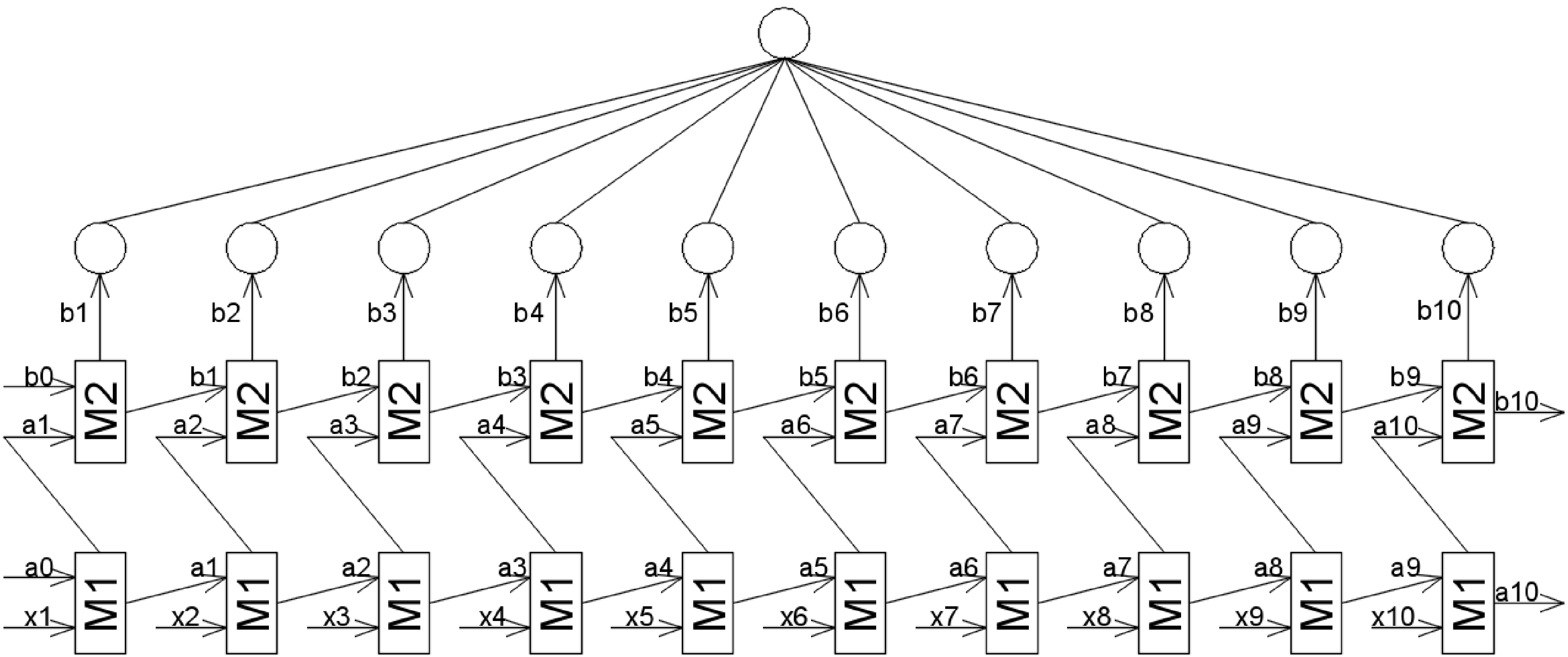


Second Layer LSTM Cells (M2)

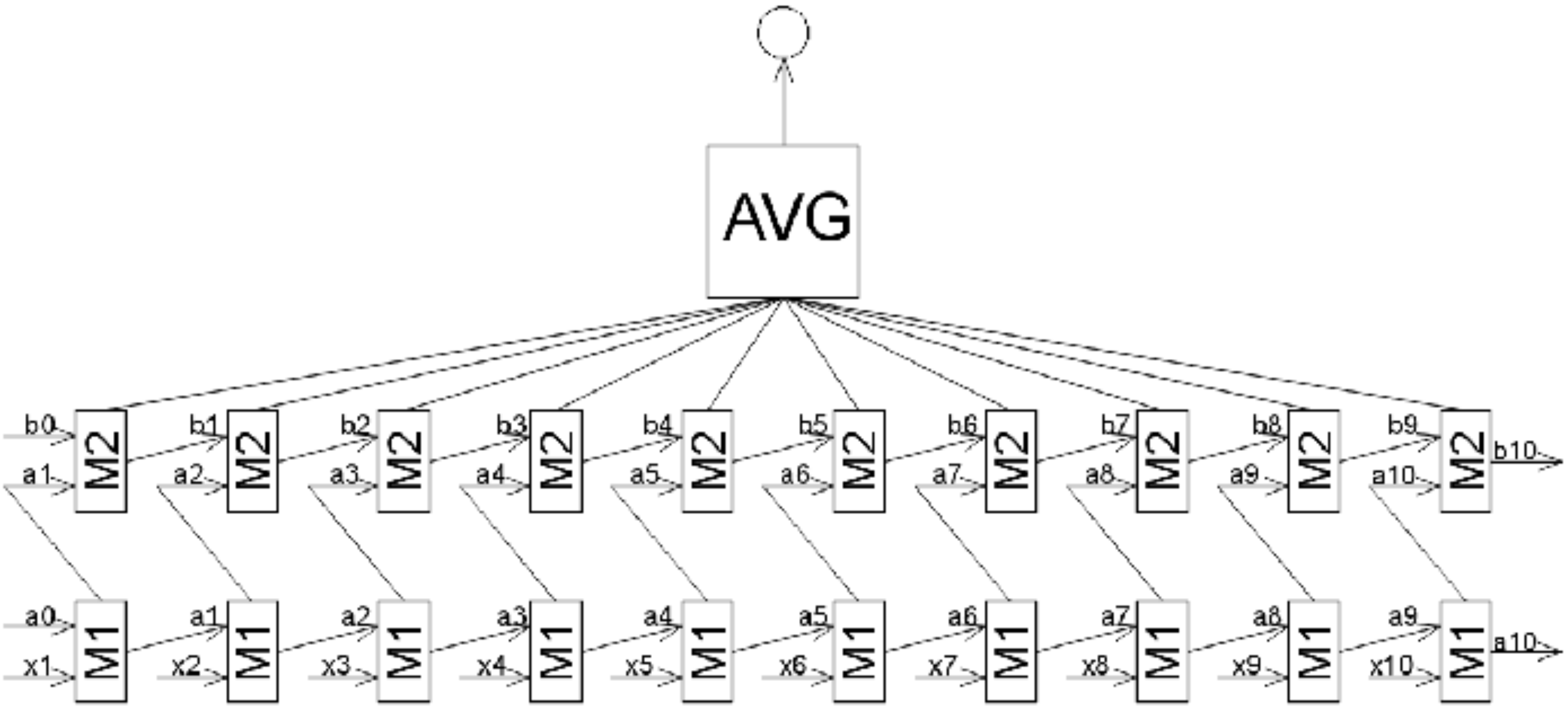


UNIVERSITY OF NORTH DAKOTA

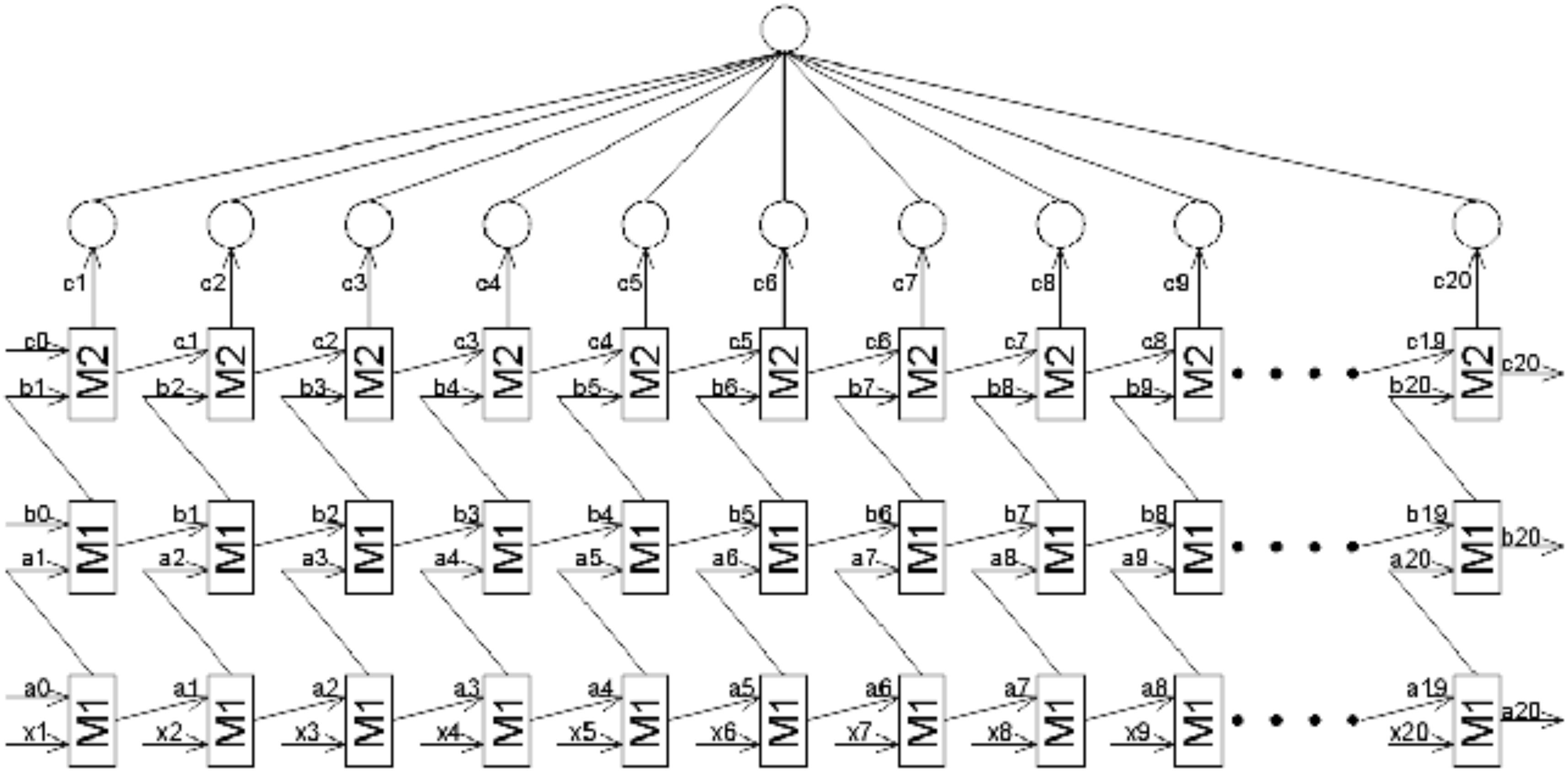
Architecture I - 21,170 weights



Architecture II - 21,160 weights



Architecture III - 83,290 weights



Implementation

Python's Theano Library was used to implement the neural networks.

Main benefits included Theano's ability to compute error gradients (as opposed to manually deriving these) and an efficient implementation.

Experimental Data

The following parameters were used as inputs to the RNNs (normalized between 0 and 1):

1. Altitude
2. Angle of Attack
3. Bleed Pressure
4. Turbine Inlet Temperature
5. Mach Number
6. Primary Rotor/Shaft Rotation Speed
7. Secondary Rotor/Shaft Rotation Speed
8. Engine Oil Pressure
9. Engine Oil Quantity
10. Engine Oil Temperature
11. Aircraft Roll
12. Total Air Temperature
13. Wind Direction
14. Wind Speed
15. Engine Vibration

Training and Testing Data

Training set:

28 flights

41,431 seconds of data

Testing set:

57 flights

38,126 seconds of data

UND UNIVERSITY OF NORTH DAKOTA

Activation Function

Sigmoid function performed significantly better than ArcTan, which resulted in distorted results.

Training Metrics

$$Error = \frac{0.5 \times \sum(Actual\,Vib - Predicted\,Vib)^2}{Testing\ Seconds}$$

$$Error = \frac{\sum[ABS(Actual\,Vib - Predicted\,Vib)]}{Testing\ Seconds}$$

Both mean squared error (MSE, top) and mean absolute error (MAE, bottom) were used to evaluate the RNNs.

MSE was used for training as it provided a smoother search space than MAE.

UND UNIVERSITY OF NORTH DAKOTA

RNN Training

RUN TIME (HOURS)

	05	10	20
Architecture I	9	8.98	8.85
Architecture II	8.44	8.41	8.4
Architecture III	21.6	19.7	18.5

The RNNs were trained for 575 epochs on a 3.5 GHz 12 core Mac Pro.

Training Results (MSE)

	Error at 5 seconds	**Error at 10 seconds**	**Error at 20 seconds**
Architecture I	0.000398	0.000972	0.001843
Architecture II	0.001516	0.001962	0.002870
Architecture III	0.000409	0.000979	0.001717

Testing Results (MSE and MAE)

Mean Squared Error

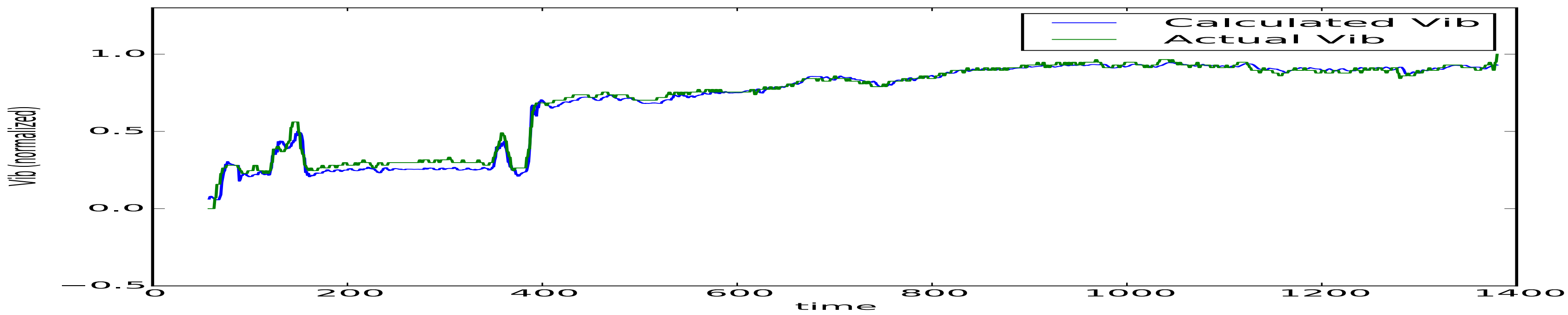
	Error at 5 seconds	Error at 10 seconds	Error at 20 seconds
Architecture I	0.001165	0.002926	0.010427
Architecture II	0.009708	0.009056	0.012560
Architecture III	0.002386	0.004780	0.041417

Mean Absolute Error

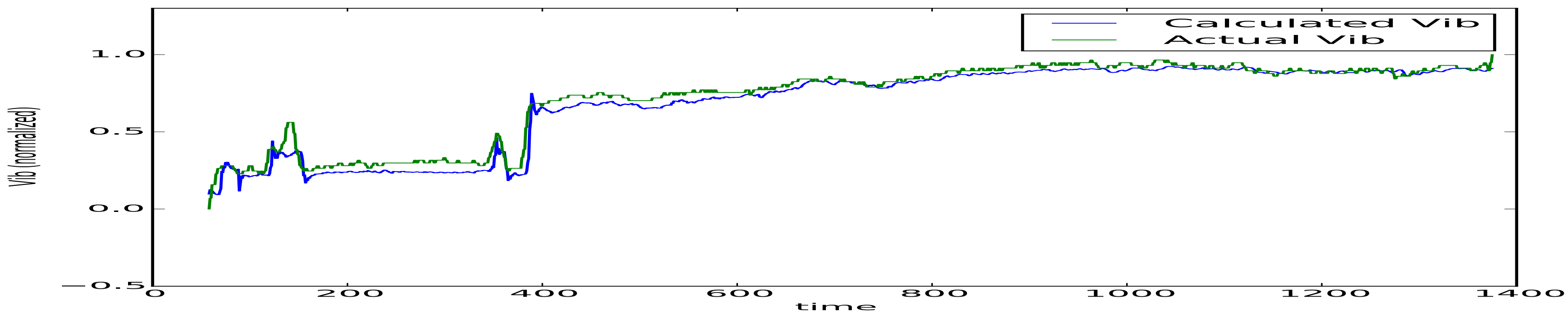
	Error at 5 seconds	Error at 10 seconds	Error at 20 seconds
Architecture I	0.033048	0.055124	0.101991
Architecture II	0.097588	0.096054	0.112320
Architecture III	0.048056	0.070360	0.202609

Architecture I Predictions

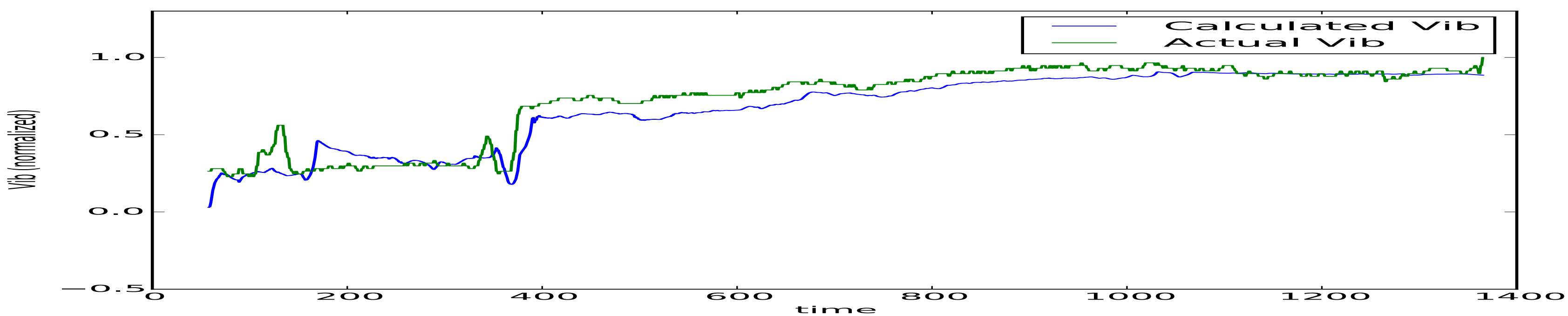
5s



10s

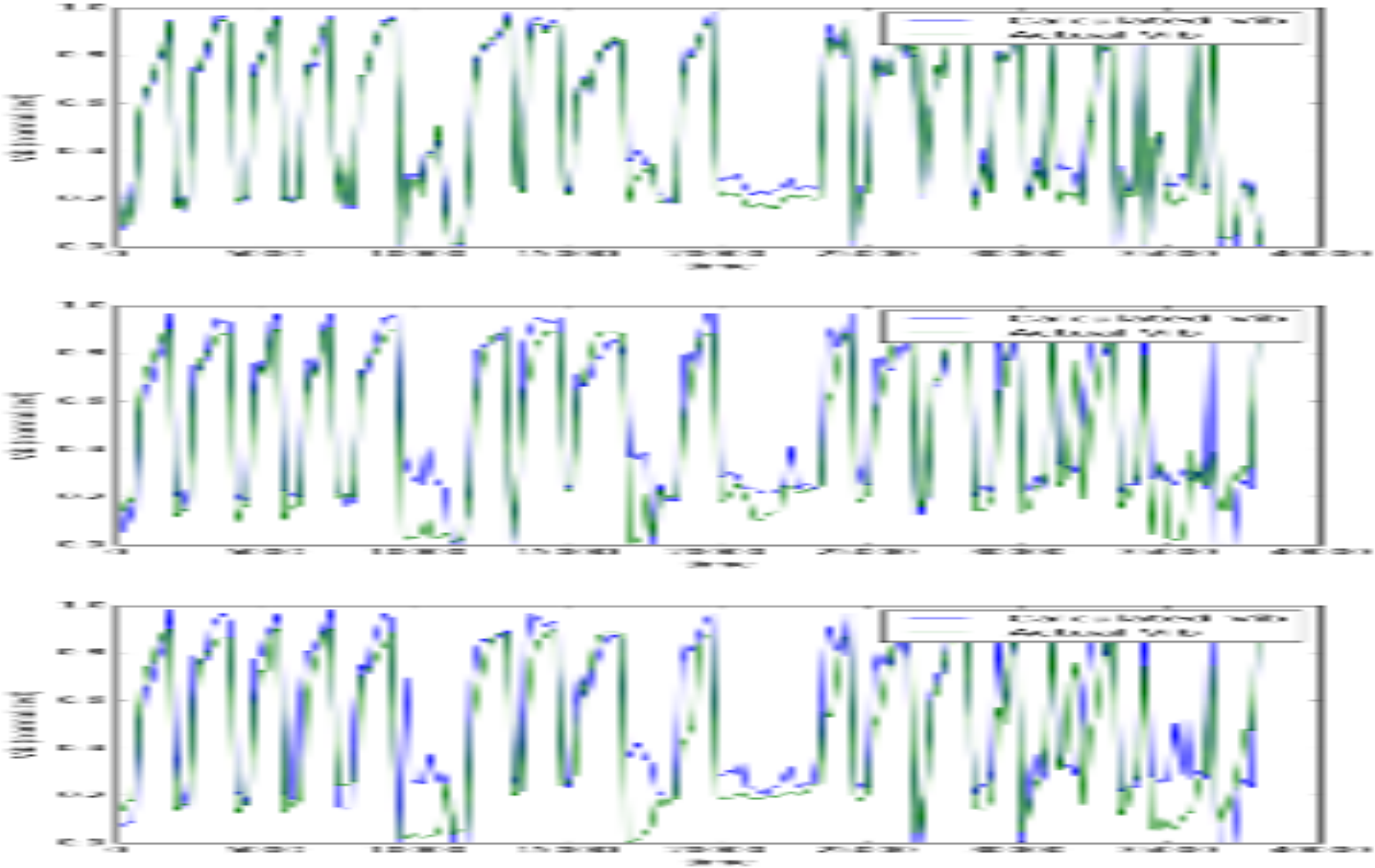


20s

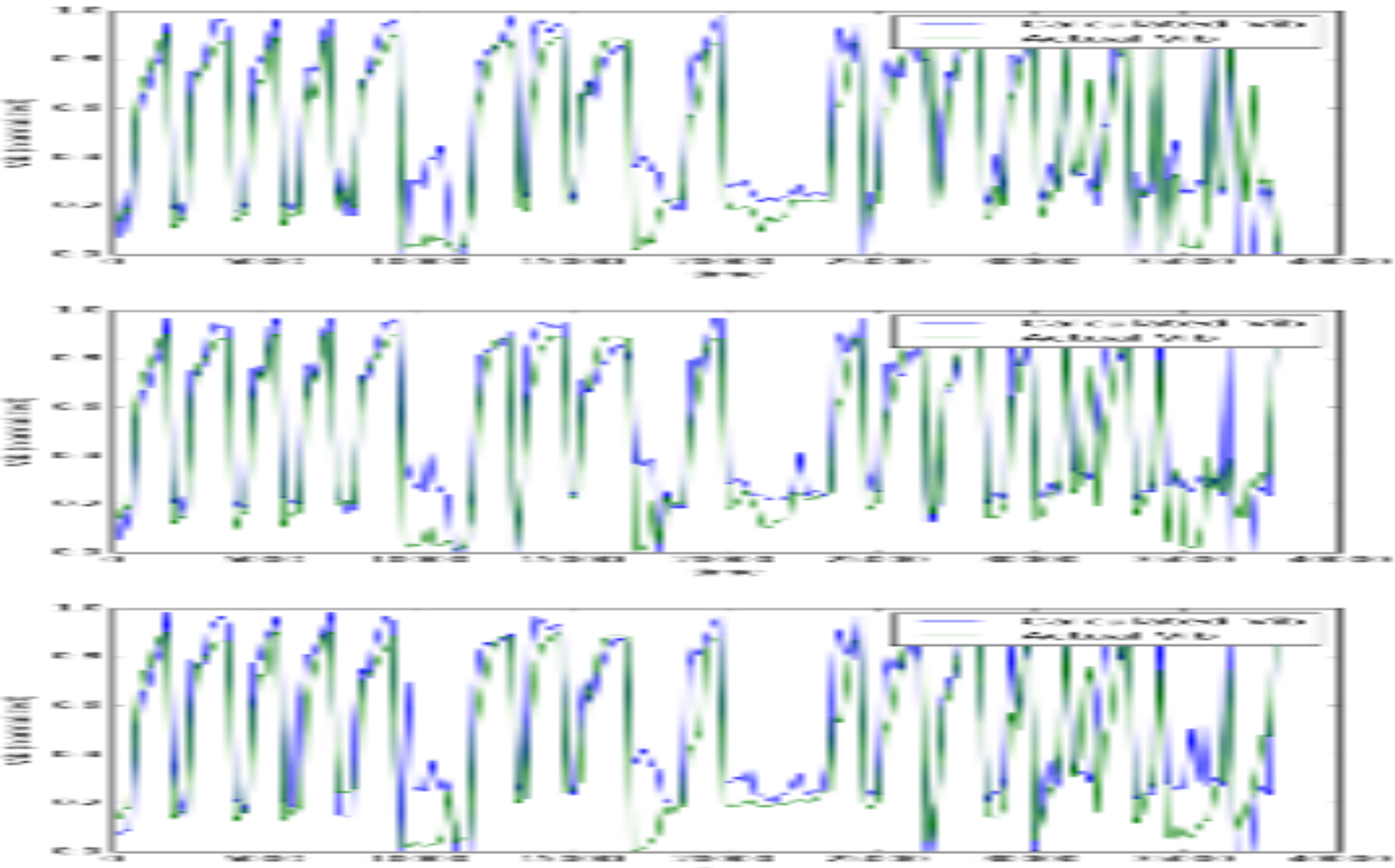


UNIVERSITY OF NORTH DAKOTA

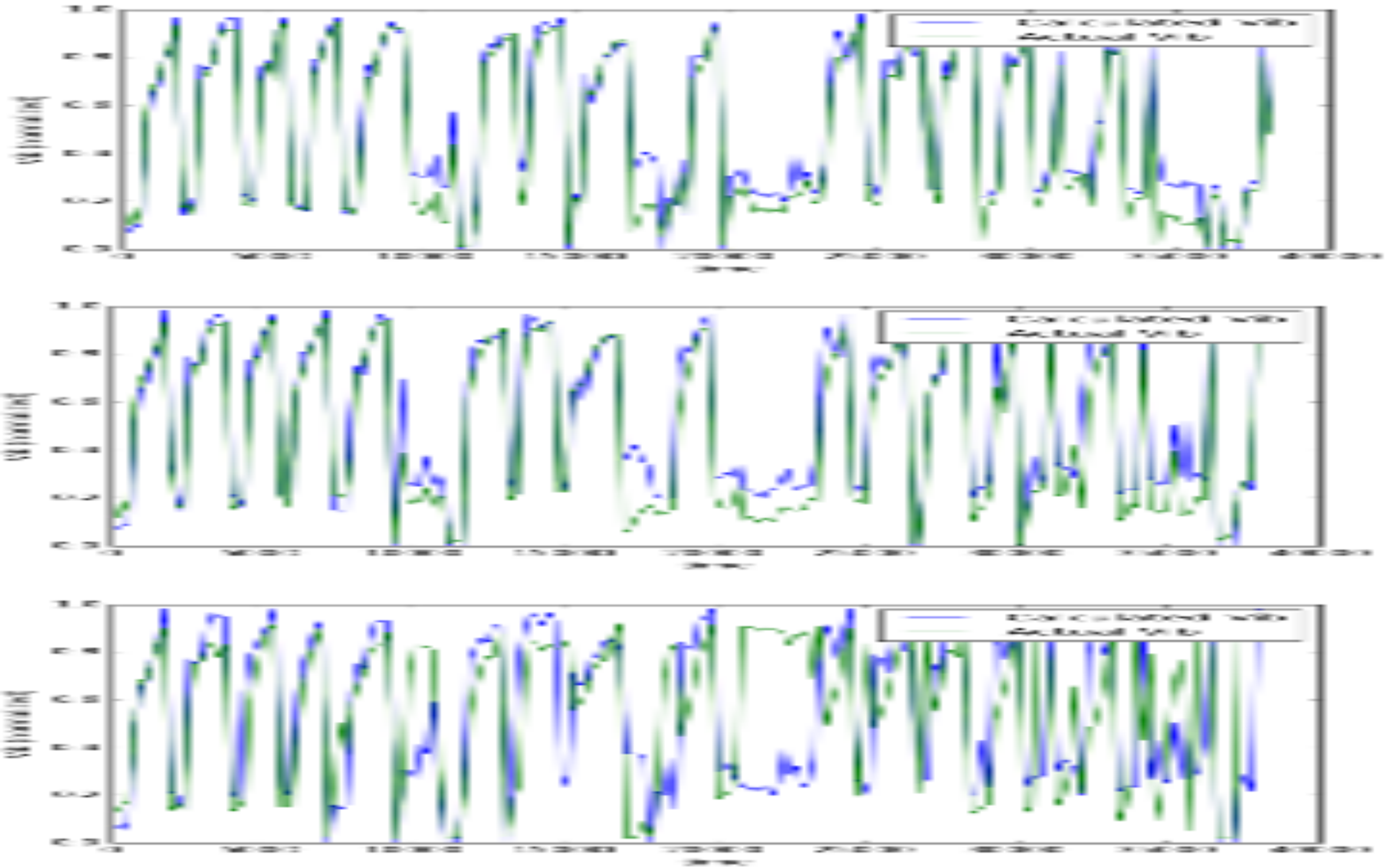
Testing Results: Architecture I predicting 5, 10, 20 sec



Testing Results: Architecture II predicting 5, 10, 20 sec



Testing Results: Architecture III predicting 5, 10, 20 sec



Conclusions

Architecture I provided the best predictions:

3.3% MAE for 5 seconds

5.51% MAE for 10 seconds

10.19% error for 20 seconds

Architecture III could potentially be trained longer.

RNNs did not train well on GPUs - needs future examination.

UND UNIVERSITY OF NORTH DAKOTA

QUESTIONS?