Assuring Complex and Critical Artificially Intelligent Systems

"AI assuring AI"

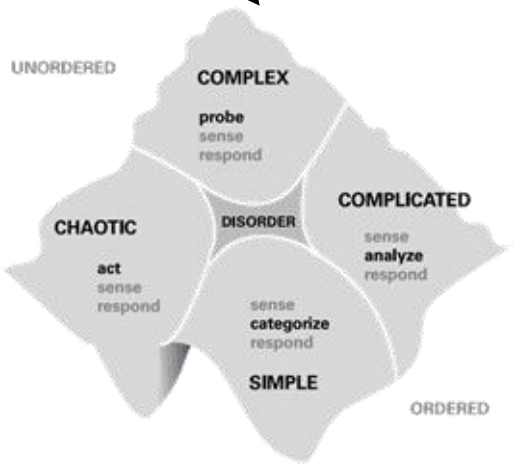
Randall McCutcheon
Royal Australian Air Force – Chief of Defence Force Fellow 2026

DATAWorks 2026
21 April 2026

UNSW AUSTRALIA
Canberra

Assuring Complex and Critical Artificially Intelligent Systems

- Performs as expected
- Free from Vulnerabilities
- Robust to Threats



Per Snowden and Boone's Cynefin model

Cost of error is high

Contains static (non-updating) Machine Learning component/s

Human made

UNSW AUSTRALIA Canberra

Overview

Motivation

Lit Review

Objectives and Scope

Experiment

Results and Analysis

Conclusions and Recommendations

Why

In simple or complicated systems, assurance can be provided up front

- Systems Engineering process – Verification and Validation
- The Configuration, Role, and Environment (and timeframe) feeds in to the boundary of acceptable performance

Systems incorporating AI are different

- Output is theoretically deterministic, practically stochastic
- Cannot define the boundary of acceptable performance
- AI is complex by its nature

Motivation

If we can't define the boundary of acceptable performance up front …

- Monitor the system to ensure it remains within the boundary of acceptable performance
- There could be 'instructive patterns', and AI is good at pattern matching

Research problem

- The purpose of the study was to explore the concept of machine learning based artificial intelligence providing assurance for an AI/ML system

Research question

- Can a machine-learning-based monitor be used as cognitive instrumentation to provide assurance for an artificial intelligence system?

What's out there?

Up front comprehensive testing to define the boundary of acceptable performance (Chandrasekaran et al, 2024)

System monitors

- What, How, Why, and their limitations
- Run Time Assurance – formally encoded (Petersen et al, 2020)
- Drift detection
- Anomaly / Novelty detection
- Out of Distribution detection
- AI/ML techniques

Testing monitors

- Test designs may need to be developed such that neural network input is specifically chosen to cause misbehaviour (Pullum et al, 2007)

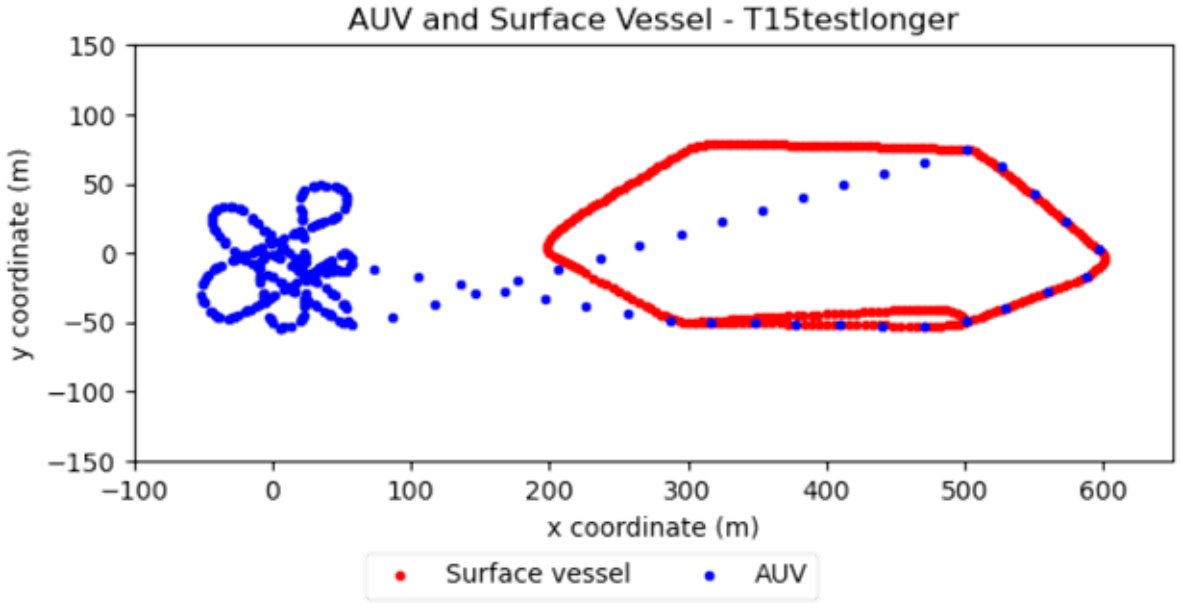
Objectives and Scope

Use a baseline AI system

- McCutcheon et al (to be published)
- AUV tracking
- Neural Kalman Filter

Apply an ‘Assurance AI’ component as Cognitive Instrumentation

Simplicity a key objective



Mean Absolute Error:
Measured Data = 59m
Converted Measurement Kalman Filter = 43m
ML Method (Long Short Term Memory) = 34m

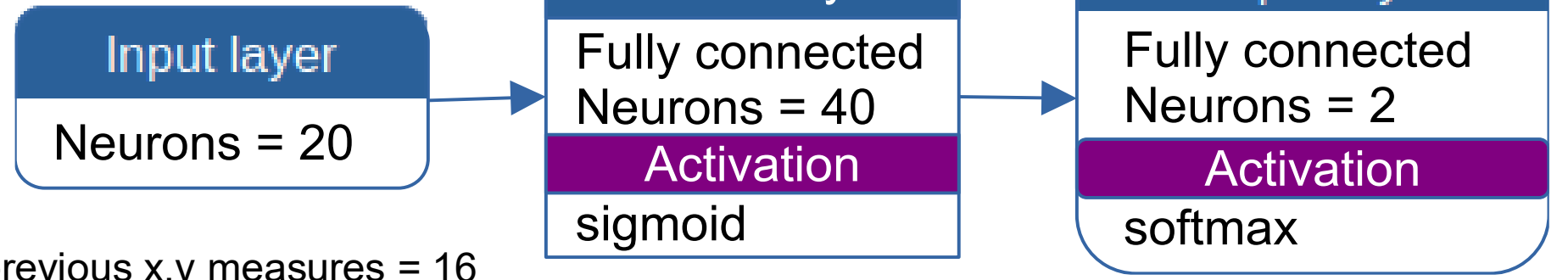
Experiment

Assurance AI (AAI) model
- Multi-layer perceptron
  - 2 Categories
  - 10 Categories
  - Autoencoder
- Training:
  - Input-output pairs as features
  - Ground-truth performance as target
    - Split evenly between categories

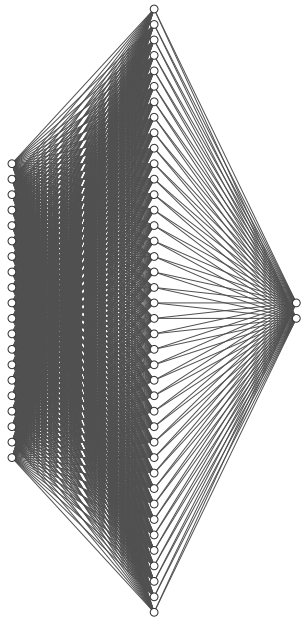
Test included manually injected errors
- Single ‘zero’
- Multiple ‘zeros’
- Single outlier

UNSW AUSTRALIA Canberra

2 Category Architecture



8 previous x,y measures = 16
1 current x,y measure = 2
1 predicted x,y position = 2
20

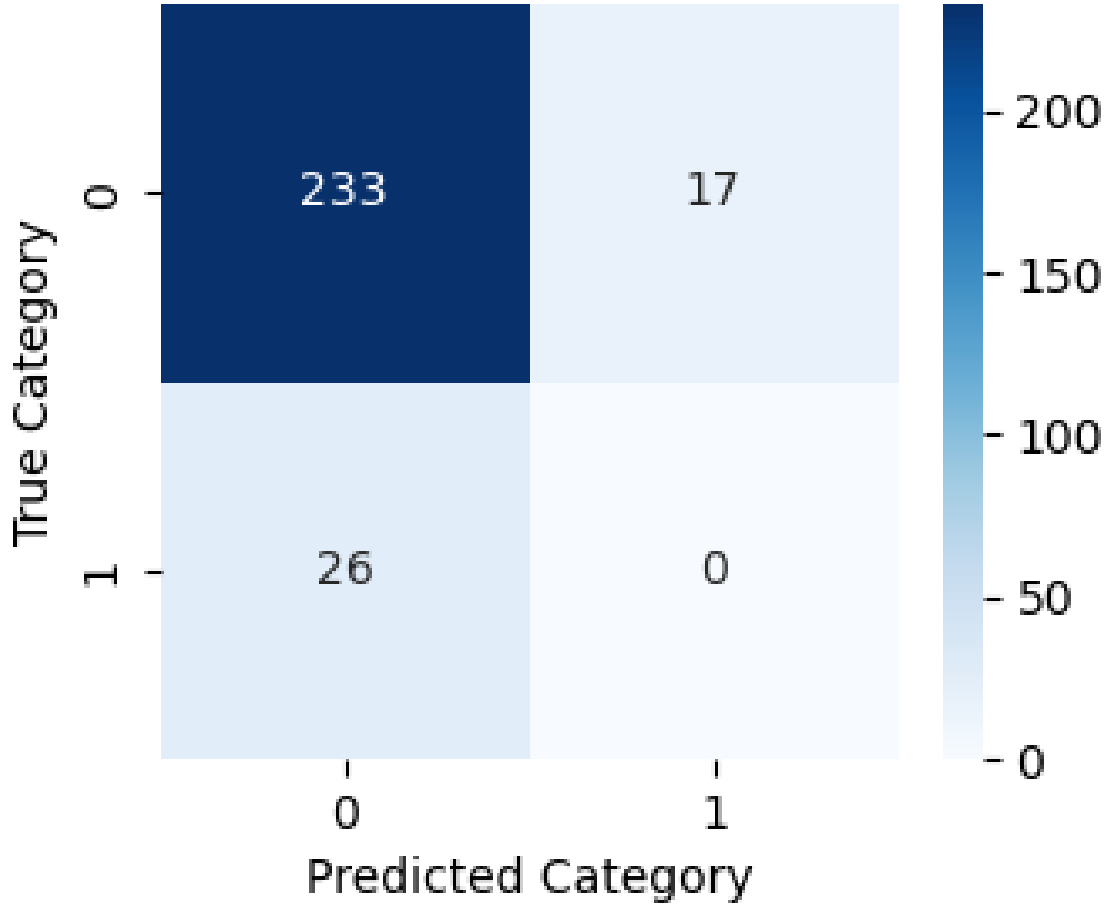


Results - 2 Category

Accuracy = 0.844

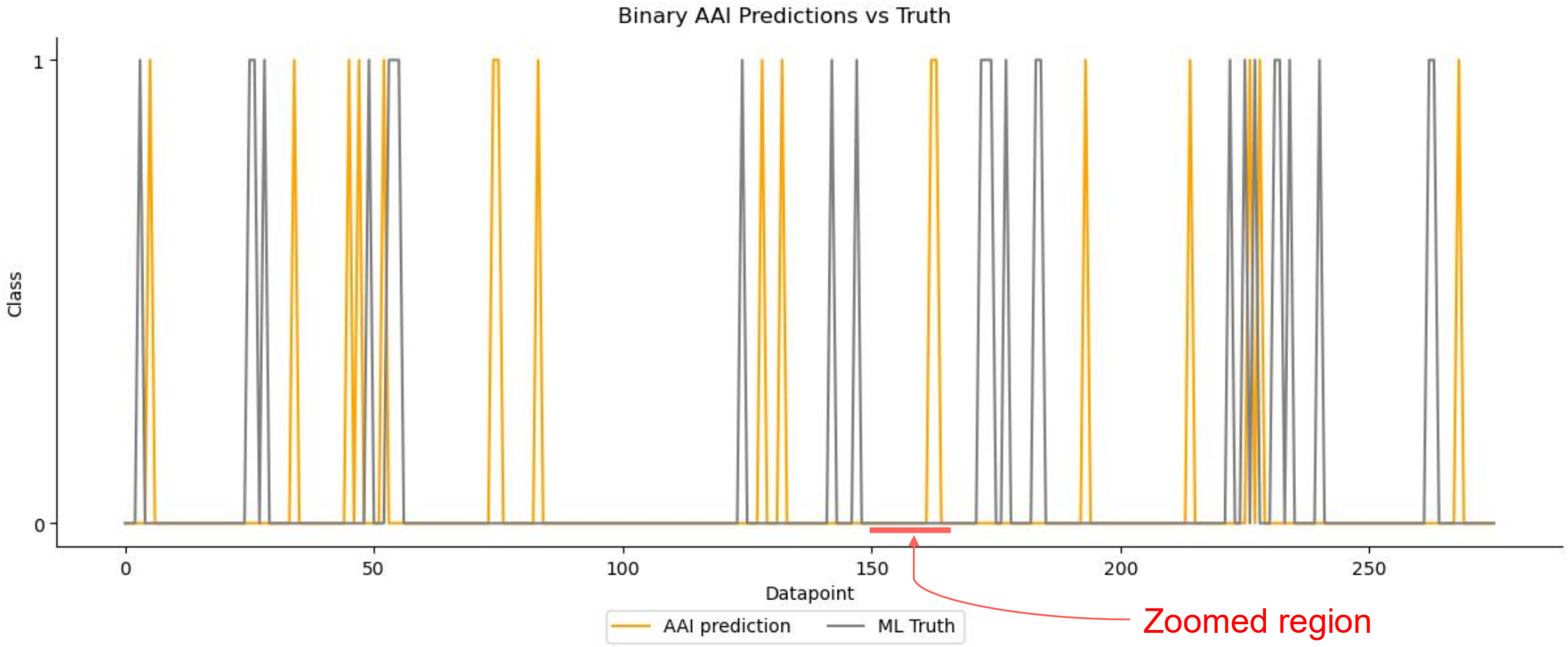
- Balanced accuracy = 0.466
- False Positive Rate = 0.068

Apparent response to manually injected errors



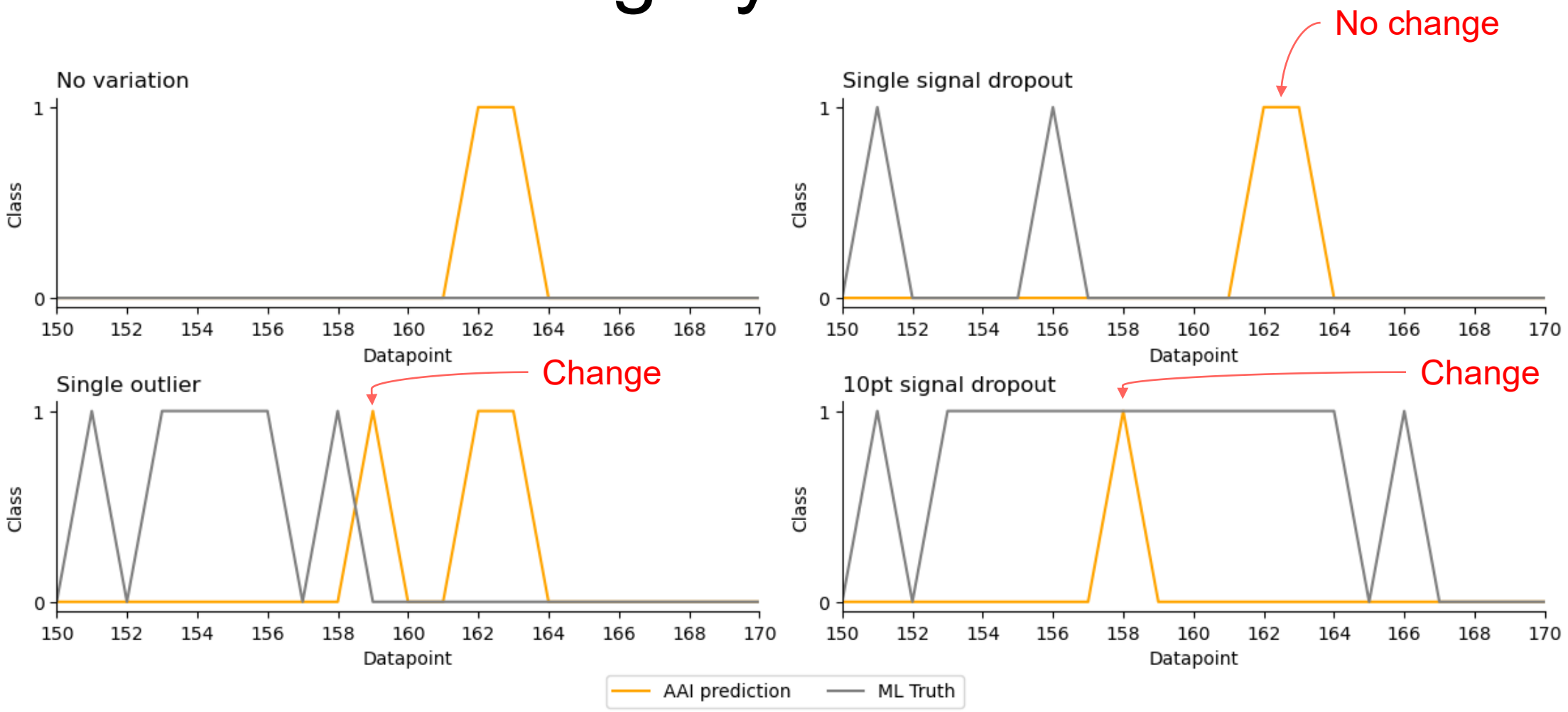
UNSW AUSTRALIA Canberra

Results – 2 Category



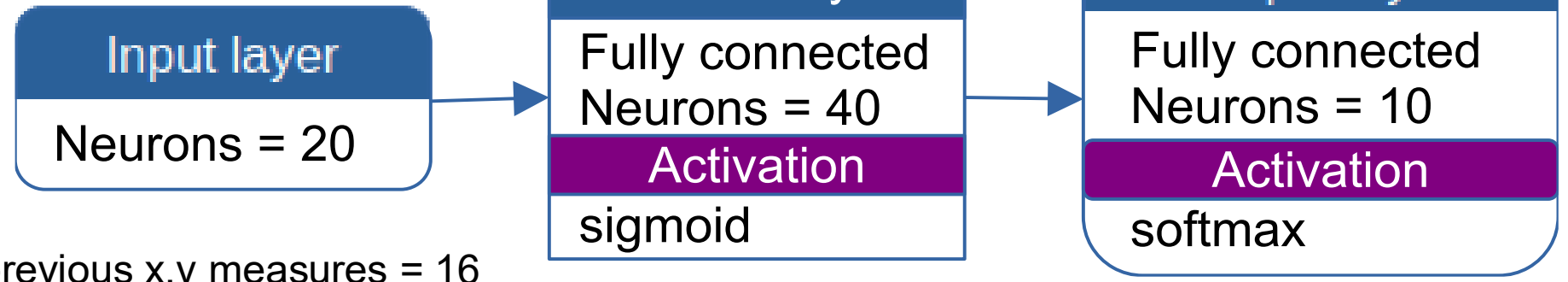
UNSW AUSTRALIA Canberra

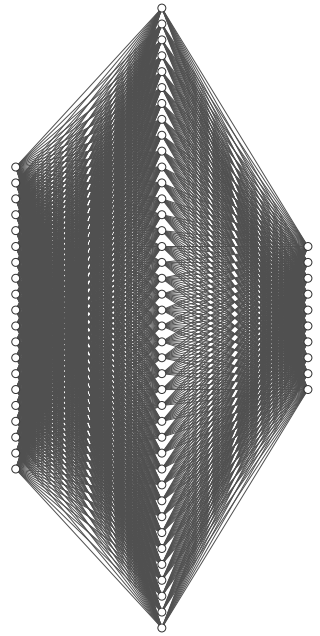
Results - 2 Category



UNSW AUSTRALIA Canberra

10 Category Architecture





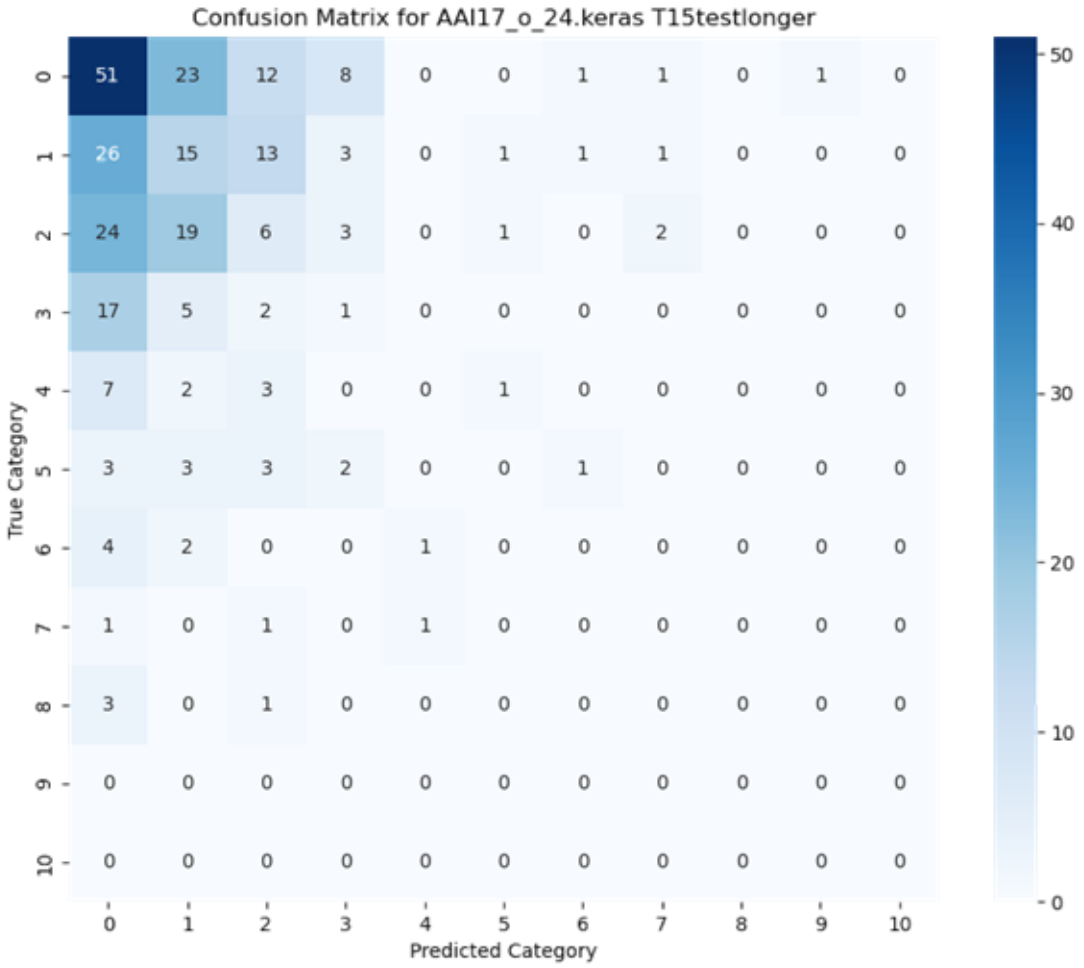
8 previous x,y measures = 16
1 current x,y measure = 2
1 predicted x,y position = 2
20

Results – 10 Category

Accuracy = 0.264

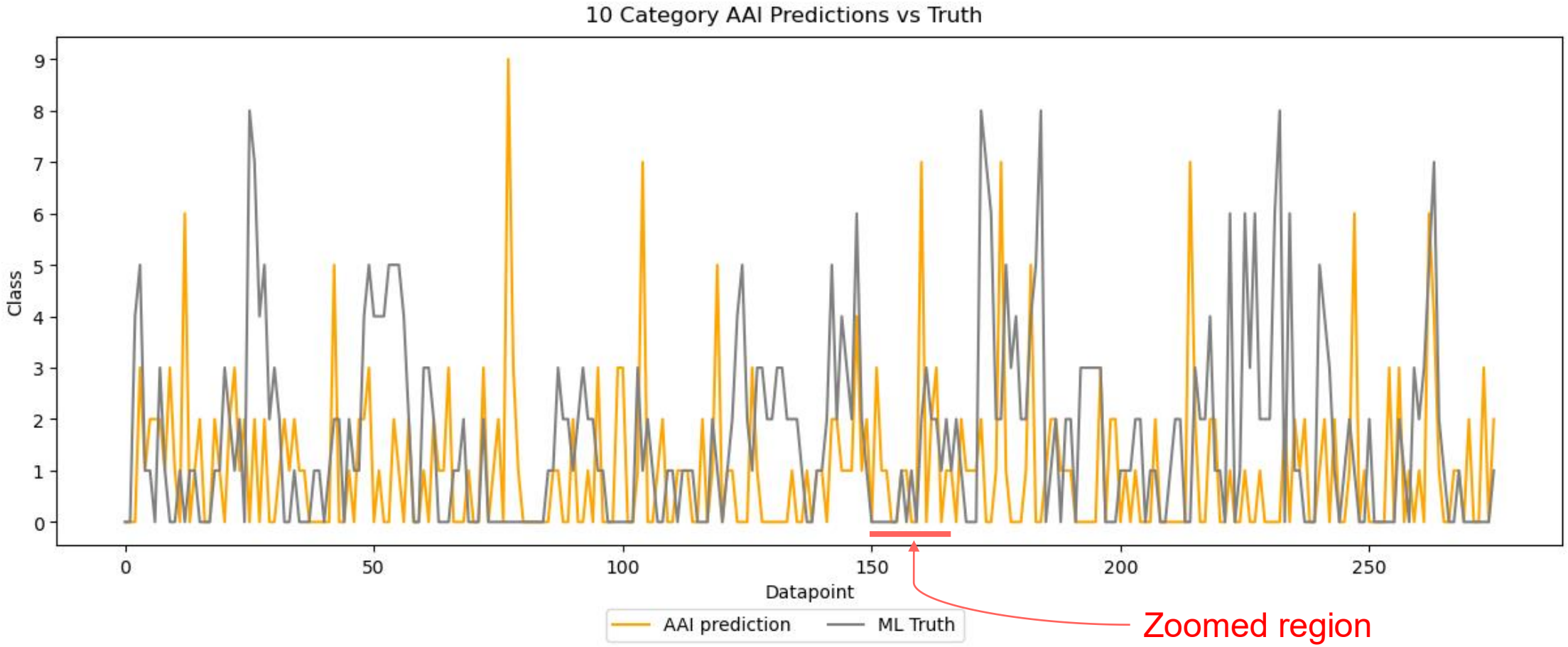
- Balanced accuracy = 0.0925
- Within 1 category = 0.583
- Within 2 categories = 0.764

Apparent response to manually injected errors



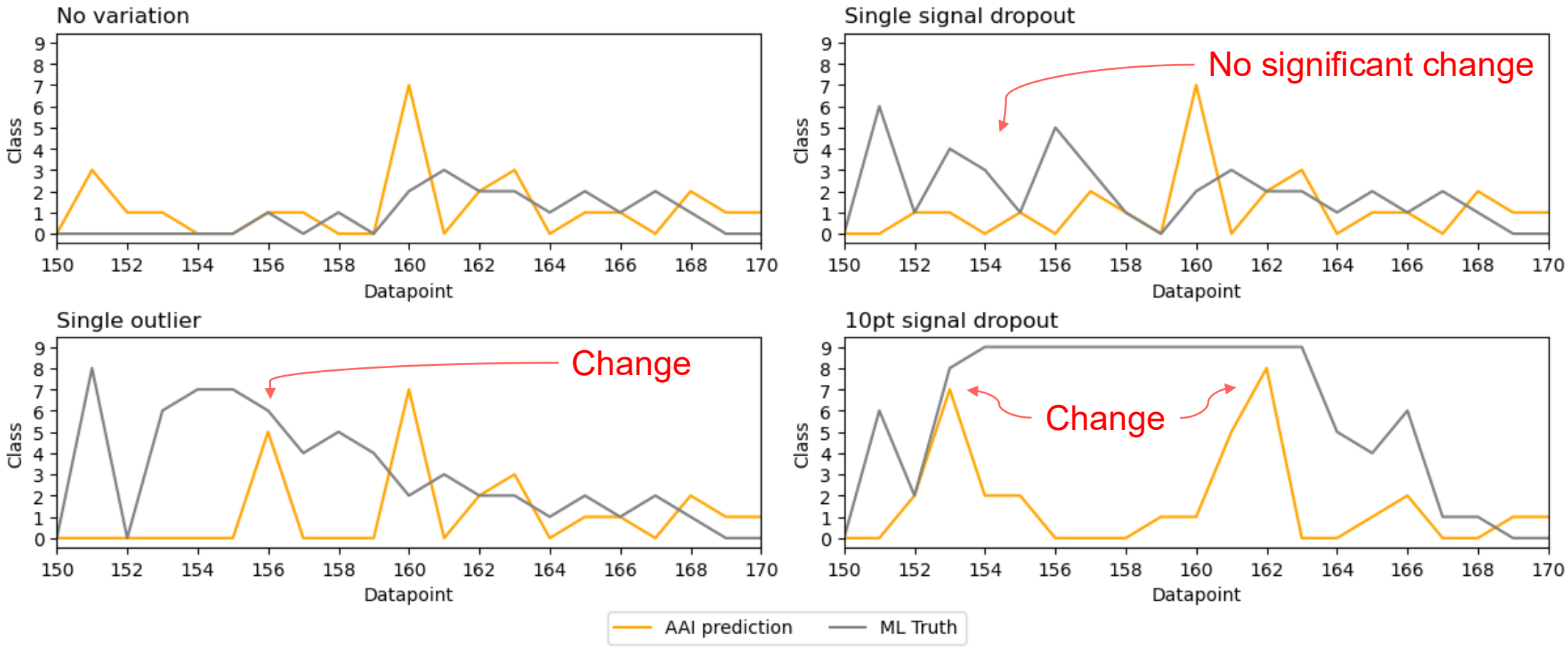
UNSW AUSTRALIA Canberra

Results – 10 Category



UNSW AUSTRALIA Canberra

Results – 10 Category



UNSW AUSTRALIA Canberra

Autoencoder Architecture

Input layer

Neurons = 20

Encoder

Fully connected
Neurons = 6

Activation

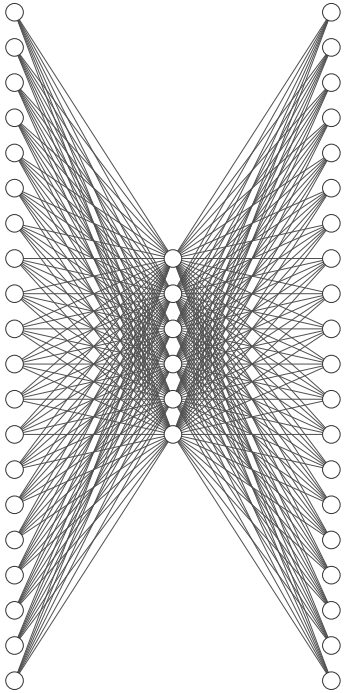
relu

Decoder

Fully connected
Neurons = 20

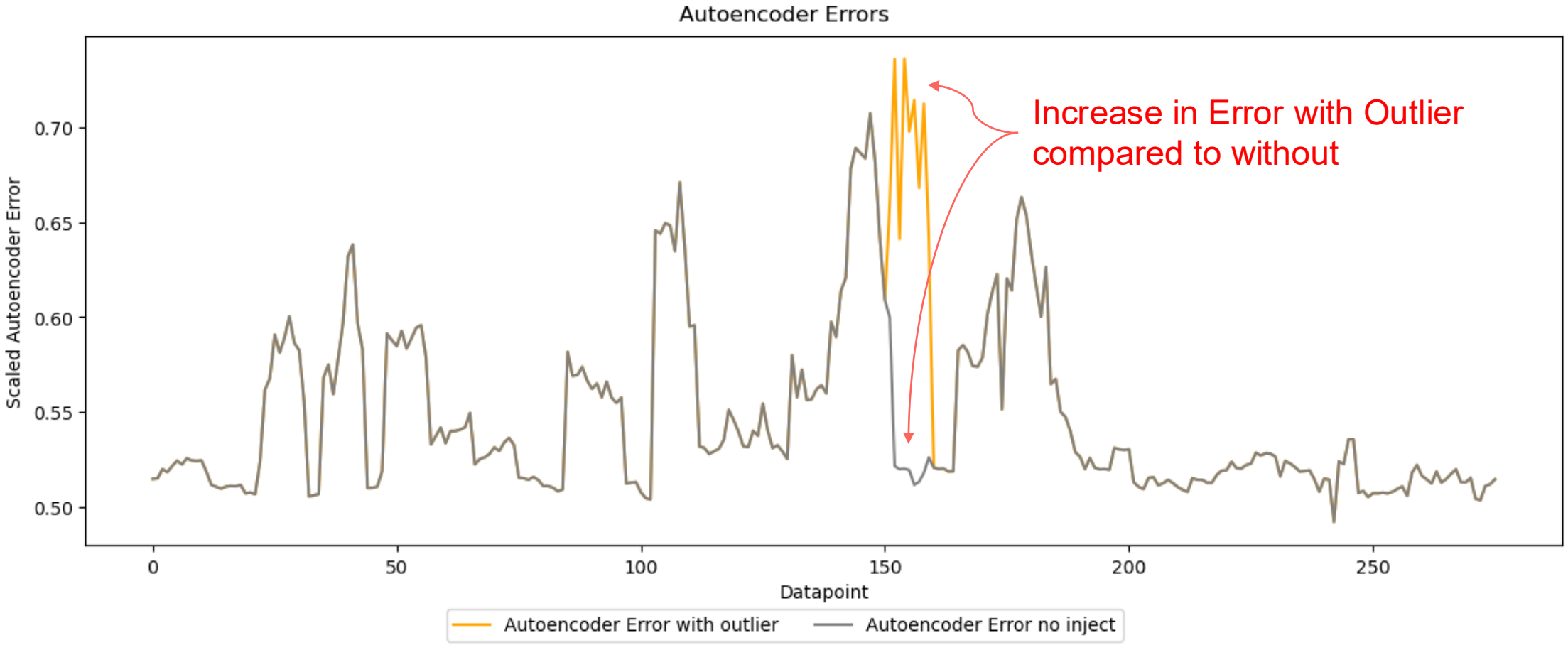
Activation

sigmoid



8 previous x,y measures = 16
1 current x,y measure = 2
1 predicted x,y position = 2
20

Results - Autoencoder



Analysis

Limited field utility in its current form

- Accuracies (2 Cat / 10 Cat)
- FPR (2 Cat)

AAIs did show sensitivity to manually injected errors

- Potential for this type of cognitive instrumentation to identify performance degradation, vulnerabilities, or threats

Conclusions and Recommendations

This research suggests ML-based cognitive instrumentation can provide an additional layer of assurance which traditional, reactive, formally encoded Run Time Assurance cannot achieve.

Future research could explore additional methods of ML based cognitive instrumentation, to increase accuracy and reduce false-positive rates

Future research could explore different ML models in different application areas

Select Bibliography

[1] J. Chandrasekaran, T. Cody, N. McCarthy, E. Lanus, L. Freeman, and K. Alexander, “Testing Machine Learning: Best Practices for the Life Cycle,” *Naval Engineers Journal*, vol. 136, pp. 249–263, Mar. 2024.

[2] L. L. Pullum, B. J. Taylor, and M. A. Darrah, “Areas of Consideration for Adaptive Systems,” in *Guidance for the Verification and Validation of Neural Networks*, 1st ed., in Emerging Technologies. , Hoboken, NJ, USA: Wiley, 2015, pp. 5–37.

[3] L. Freeman, “Test and Evaluation for Artificial Intelligence,” *Insight (International Council on Systems Engineering)*, vol. 23, no. 1, pp. 27–30, Mar. 2020, doi: https://doi.org/10.1002/inst.12281.

[4] D. J. Snowden and M. E. Boone, “A Leader’s Framework for Decision Making,” *Harvard Business Review*, vol. 85(11), no. 11, pp. 68–76, Nov. 2007.

[5] N. Nayal, M. Yavuz, J. F. Henriques, and F. Güney, “RbA: Segmenting Unknown Regions Rejected by All,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2023. doi: 10.1109/ICCV51070.2023.00072.