Maximizing Operator Impact

Leveraging Machine Learning to Improve Trending Analysis for the Magnetospheric Multiscale Mission

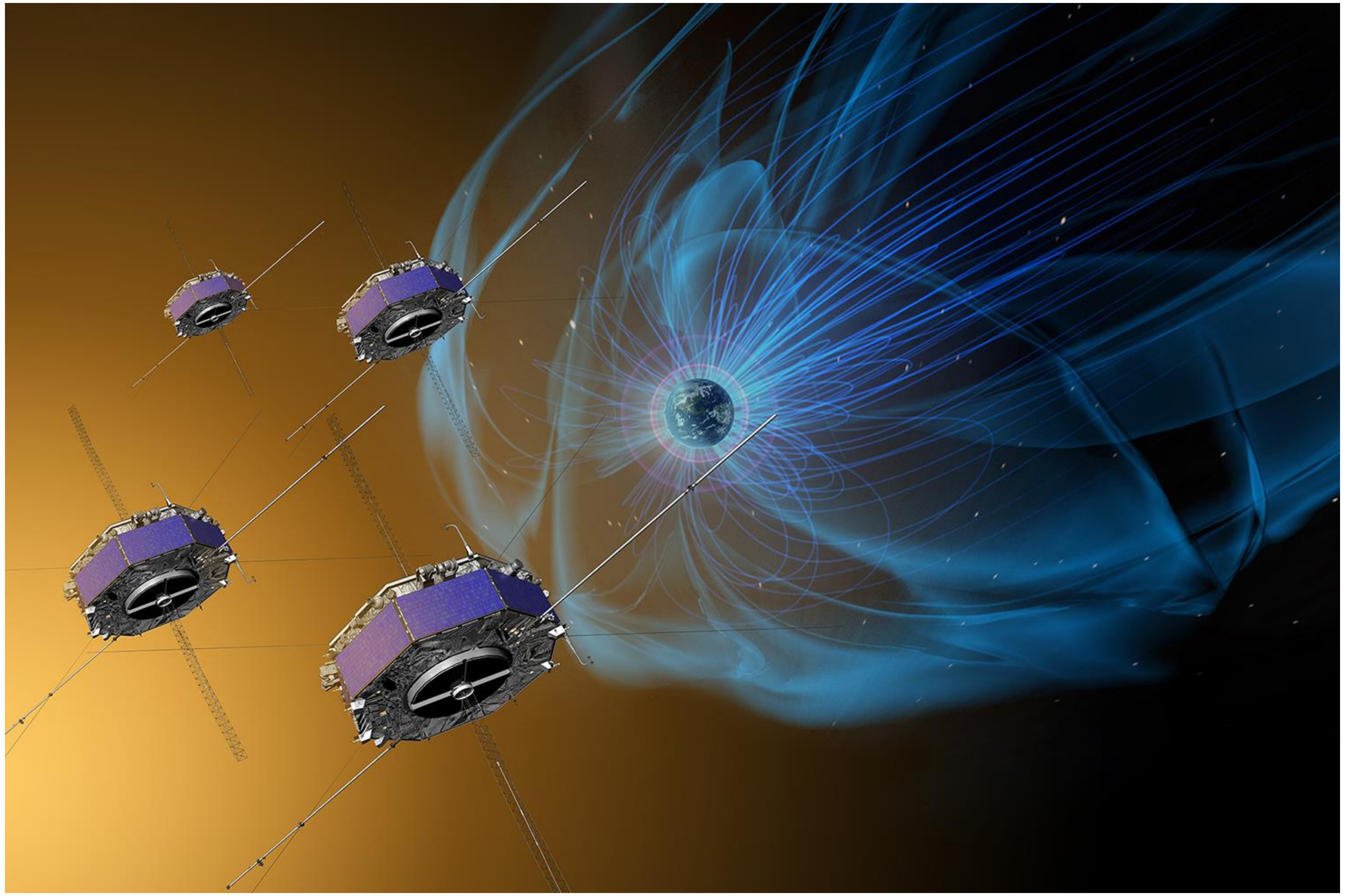
Shannon Bull

NASA's Goddard Space Flight Center

GSAW 2025
Track: Anomaly Detection
Tuesday, Feb. 25th, 2025
Los Angeles, California

The Magnetospheric Multiscale (MMS) mission

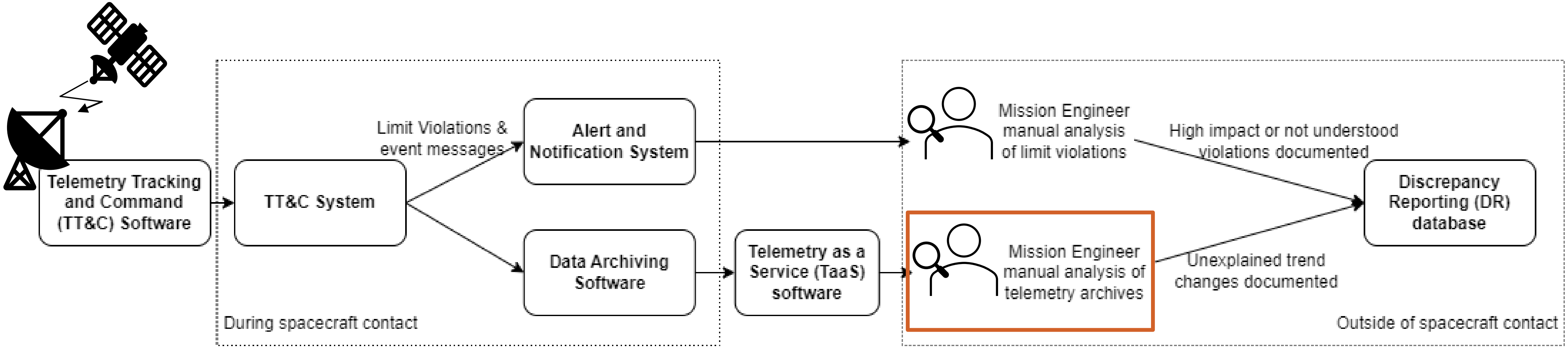
- Launched in 2015
- Consists of 4 identical spacecraft flying in a tetrahedral formation
- Explores magnetic reconnection - connection and disconnection of the Sun and Earth's magnetic fields
- Its data products are used to improve the models that predict space weather, benefiting both terrestrial and space exploration activities
- Currently in second extended mission phase
- GSFC manages MMS mission operations



The mission's four identical spacecraft fly in a pyramid-shape to measure magnetic field lines and charged particles in three-dimensions (Source: nasa.gov)

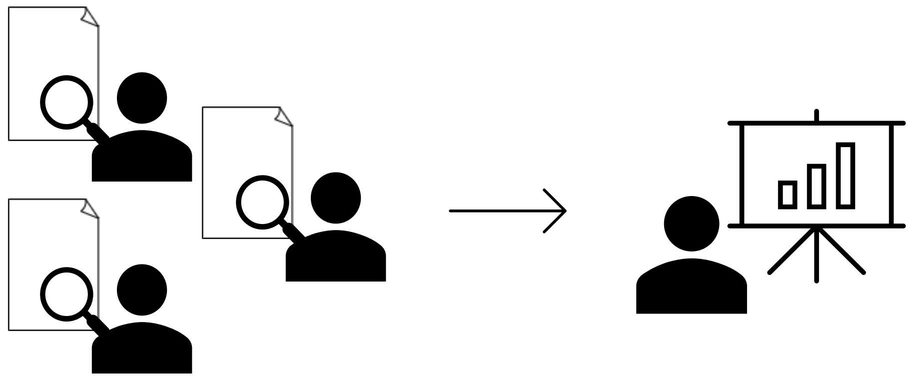
MMS Mission Operations

- Telemetry analysis consistently takes place throughout operations:
  - During spacecraft contact
  - Back orbit analysis
  - All other analysis outside of contact periods
- Telemetry analysis is necessary to detect and diagnose mission anomalies
  - There are currently thresholding techniques to automatically detect possible anomalies

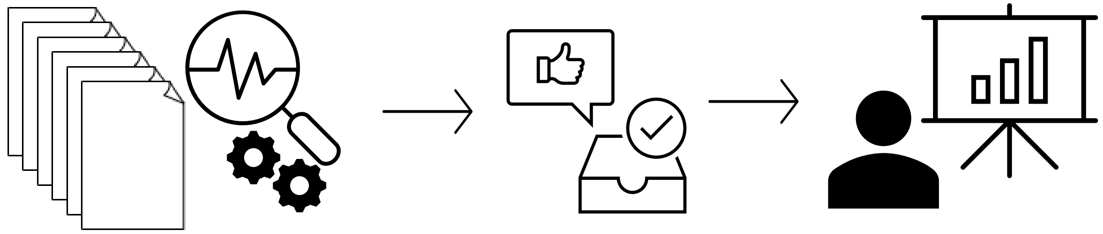


Objectives

- Objective: Use Machine Learning (ML) techniques on MMS mission telemetry data to develop a suite of algorithms capable of predicting and detecting mission anomalies
  - Go beyond existing rules-based automation
  - Provide operators with new insights into telemetry data patterns
  - Reduce the time it takes to identify anomalies



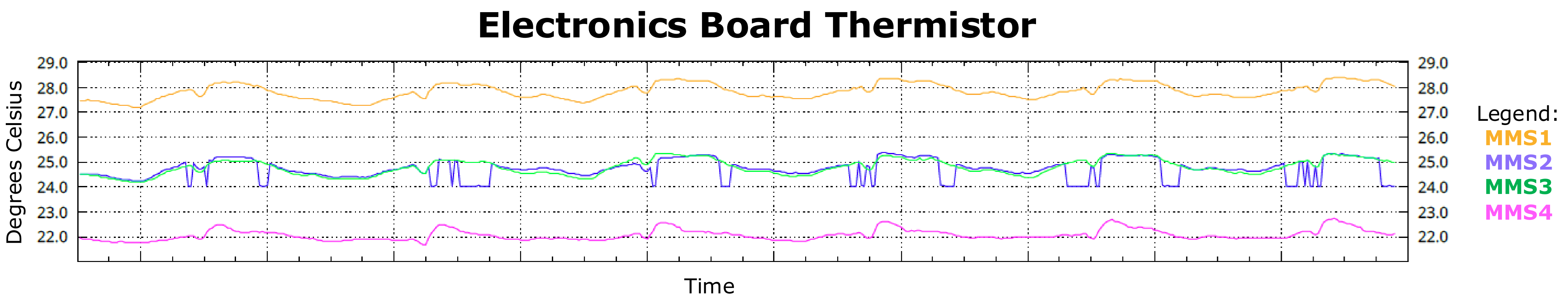
Currently: Mission Engineers manually review all subsystem data for trend performance.



In the future: Mission Engineers leverage model outputs to review subsystem data.

Methods: Our Data

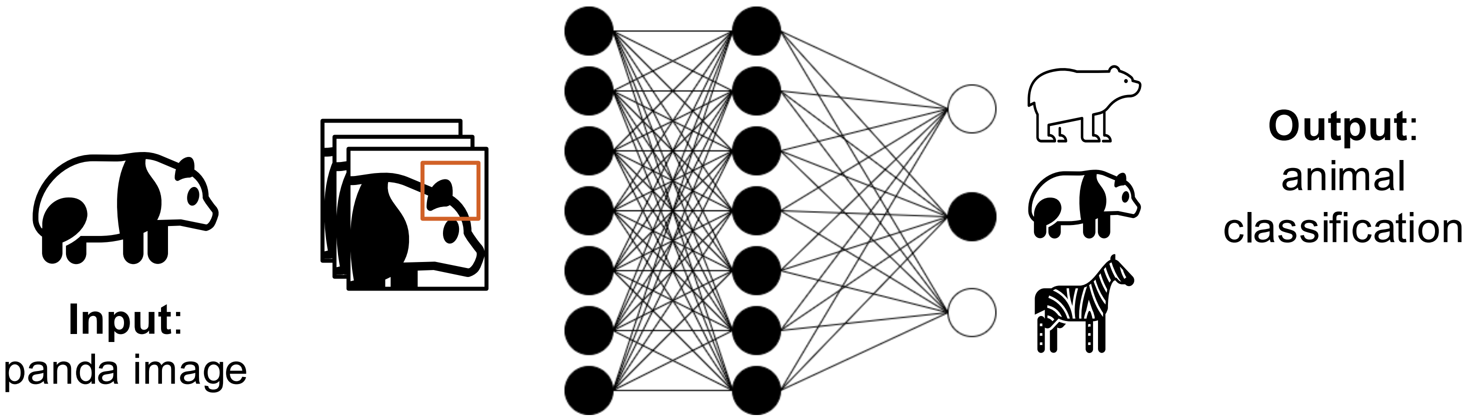
- Training Data: telemetry archives from TaaS software
  - Focus on mnemonics from one subsystem for one of the four spacecraft at a time
- Testing Data: previously documented mission anomalies



The graph above contains data from an anomalous trend that did not trigger automatic limit violations

Methods: Our Model

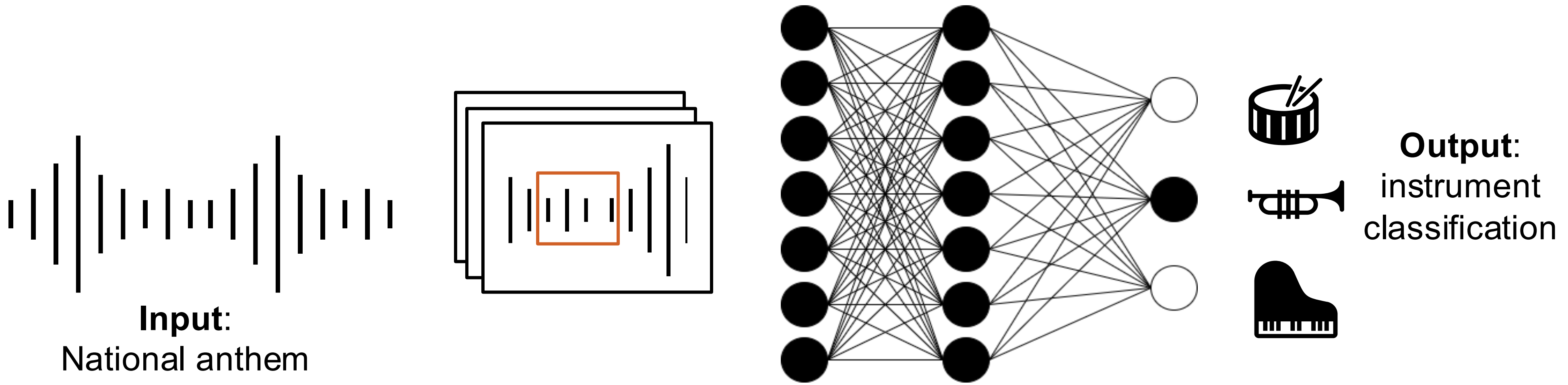
We use a specialized version of a Convolutional Neural Network (CNN) called a Temporal Convolutional Network (TCN). Classic CNNs use filters to extract features. The combination of these features can help in classification problems.



In the above example, 2-Dimensional filters may look at a body part and capture the shape, color, or texture of a particular body part. Feeding a modified image should still result in proper classification

Methods: Our Model

We use a specialized version of a Convolutional Neural Network (CNN) called a Temporal Convolutional Network (TCN). Classic CNNs use filters to extract features. The combination of these features can help in classification problems.

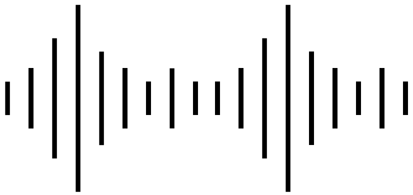


When considering music. 1-Dimensional CNN filters may qualify the pitch, volume, brightness of a sound. Modifying (slowing down, reversing) the audio file should not impact classification.

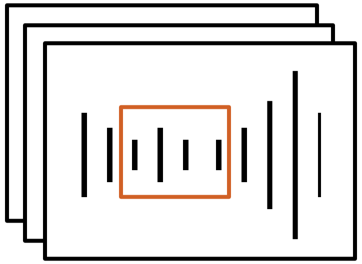
Methods: Our Model

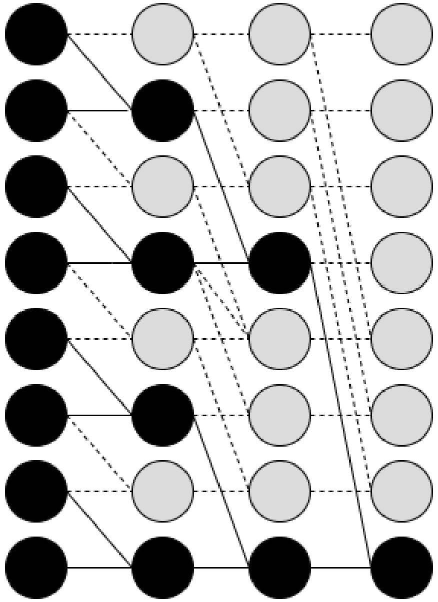
A TCN is used with sequential data, when order matters.

- Causal (temporal) convolutions preserve temporal order
- Dilated convolutions capture long term dependencies



Input:
National anthem





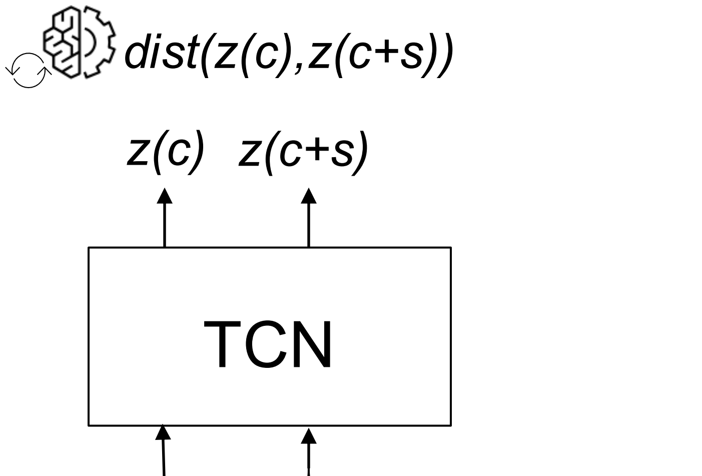
YES

Output:
Is this song the National anthem?

The TCN's modified architecture maintains the temporal integrity and captures short and long-term features in the data. Slowing down, reversing, or rearranging the audio will change the classification.

Methods: Training Our Model

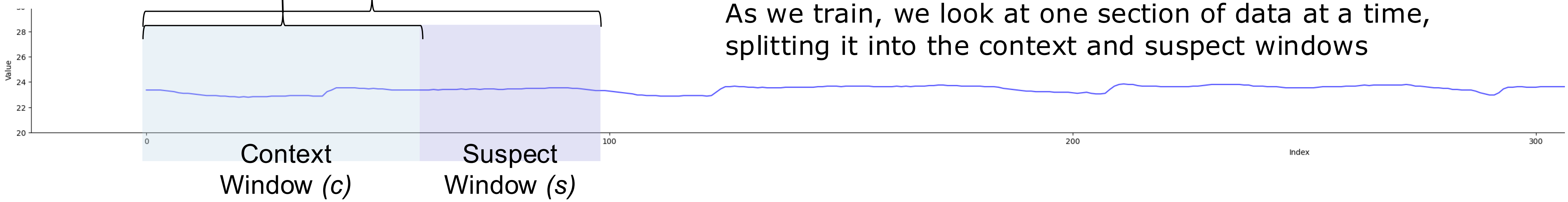
Inspired by Carmona et. al's publication "Neural Contextual Anomaly Detection (NCAD) for Time Series" we use a TCN to detect anomalous trends in telemetry data.



The distance between these vectors is computed. The model aims to produce a distance near 0 when the suspect window is nominal.

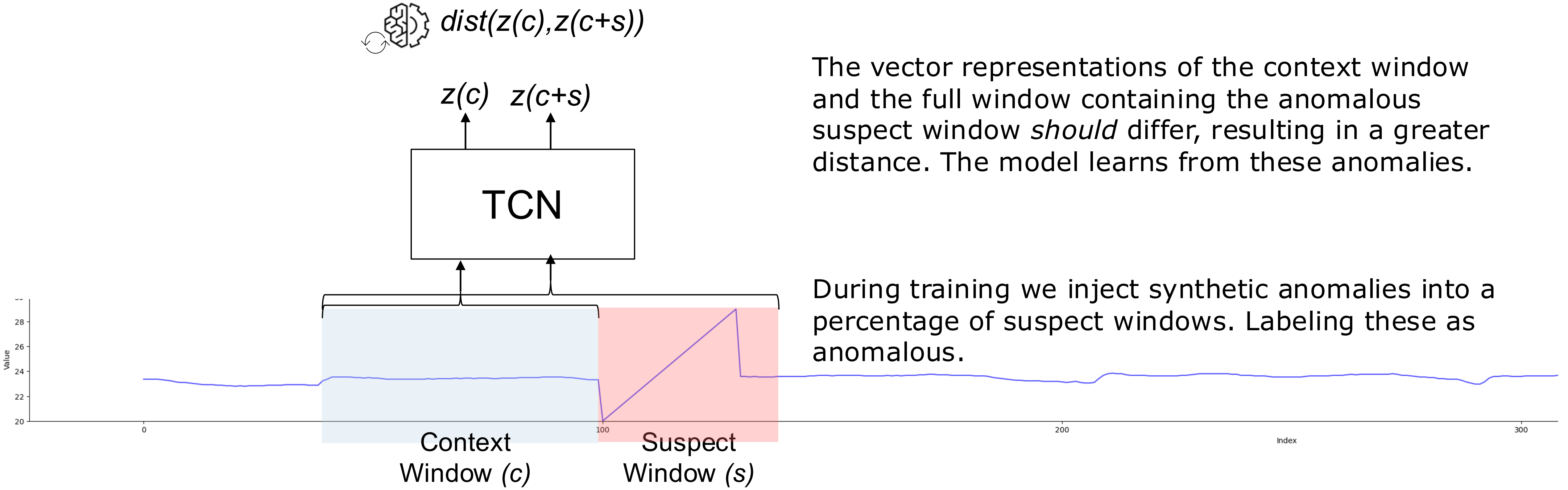
The context window *(c)* and the full window (*c*+s) are each passed through our TCN which produces new vector representations of the input data

As we train, we look at one section of data at a time, splitting it into the context and suspect windows



Methods: Training Our Model

Inspired by Carmona et. al's publication "Neural Contextual Anomaly Detection (NCAD) for Time Series" we use a TCN to detect anomalous trends in telemetry data.

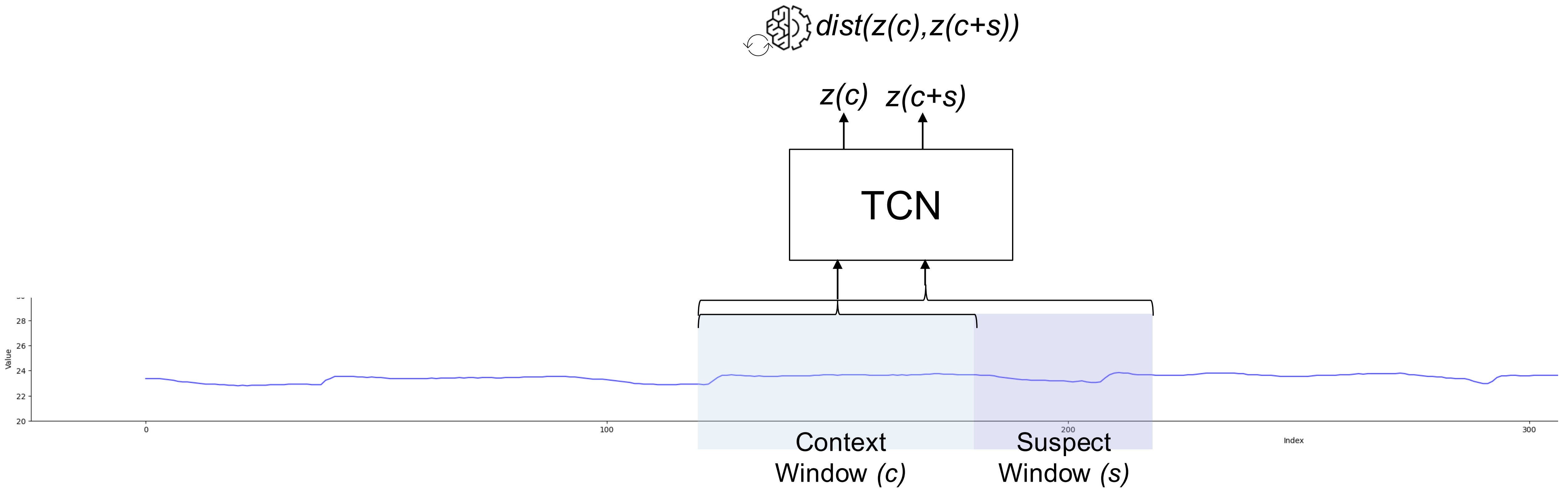


The vector representations of the context window and the full window containing the anomalous suspect window *should* differ, resulting in a greater distance. The model learns from these anomalies.

During training we inject synthetic anomalies into a percentage of suspect windows. Labeling these as anomalous.

Methods: Training Our Model

Inspired by Carmona et. al's publication "Neural Contextual Anomaly Detection (NCAD) for Time Series" we use a TCN to detect anomalous trends in telemetry data.

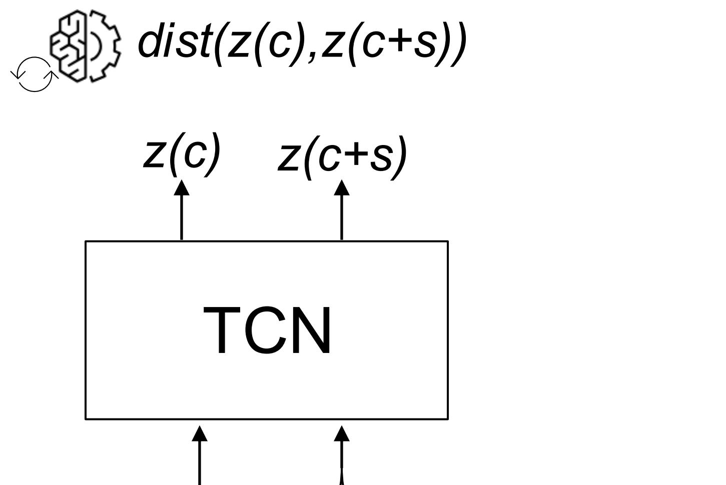


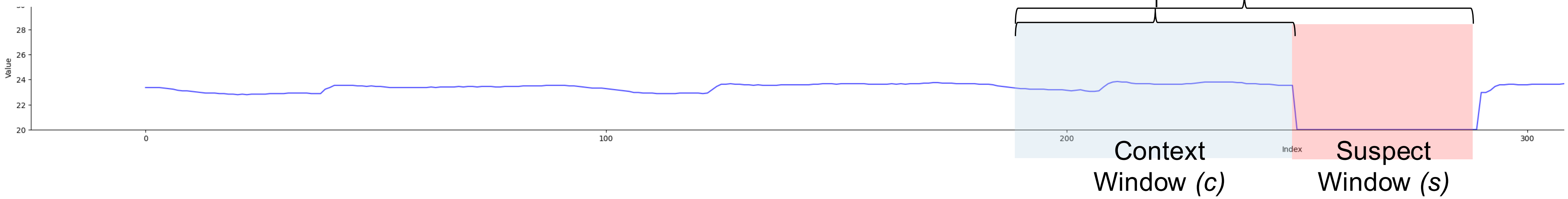
Methods: Training Our Model

Inspired by Carmona et. al’s publication “Neural Contextual Anomaly Detection (NCAD) for Time Series” we use a TCN to detect anomalous trends in telemetry data.

The injected synthetic anomalies include known behavior that has caused previous anomalies:

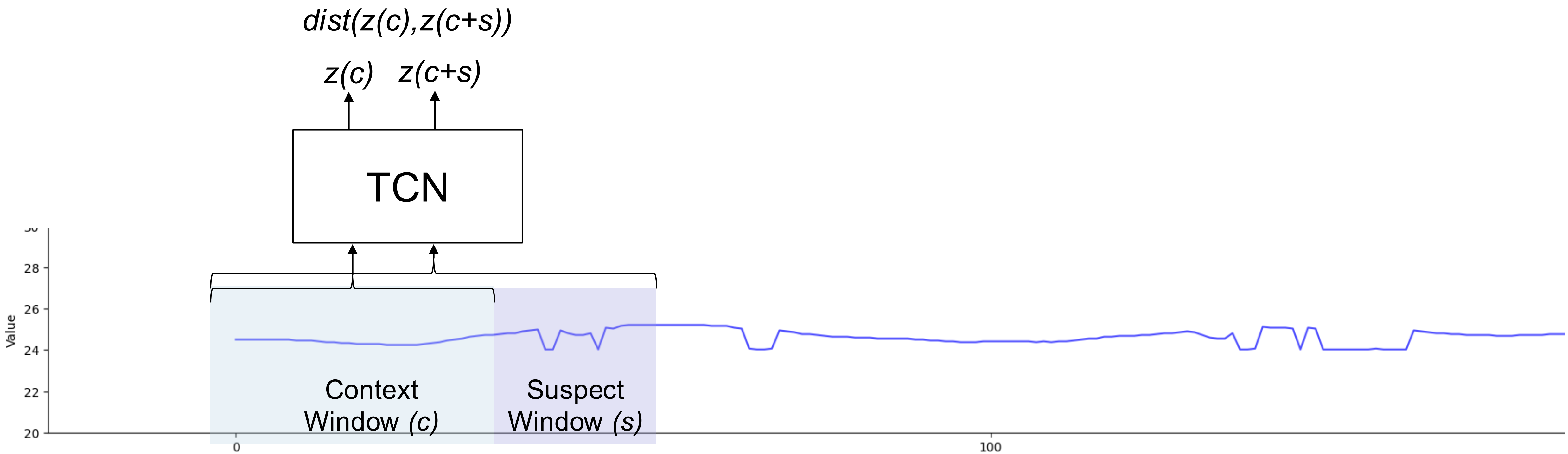
- Spikes
- Flatlines
- Other irregular patterns





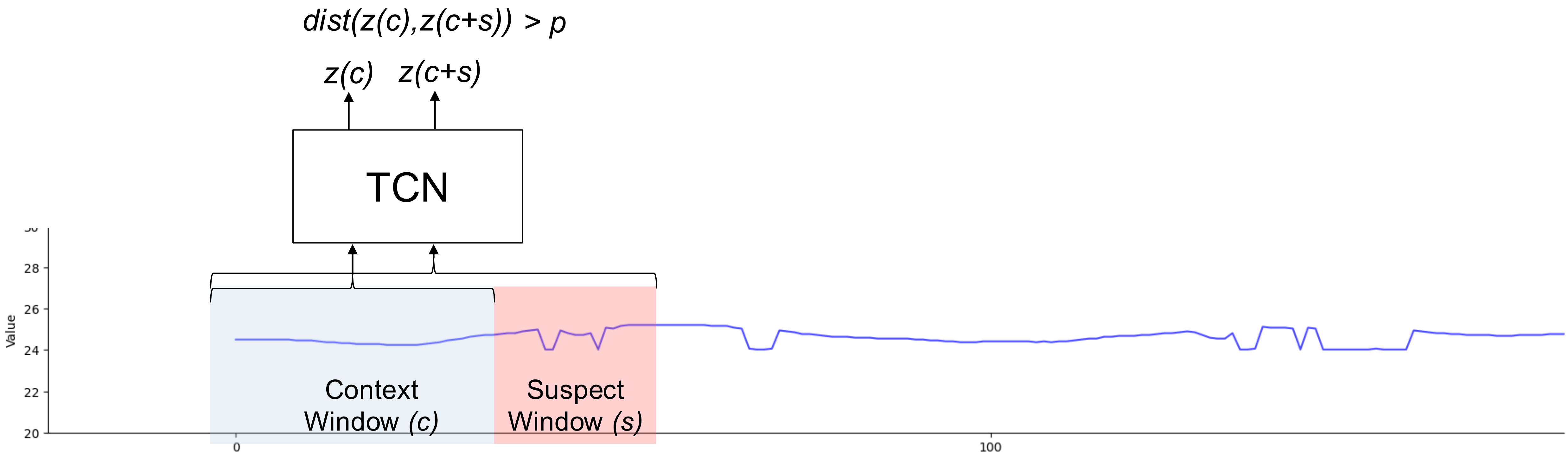
Methods: Testing Our Model

- We use the trained TCN model on new test data, sliding the context and suspect windows down one time step at a time
- The computed distance acts as our anomaly score. We threshold our scores using the median absolute deviation



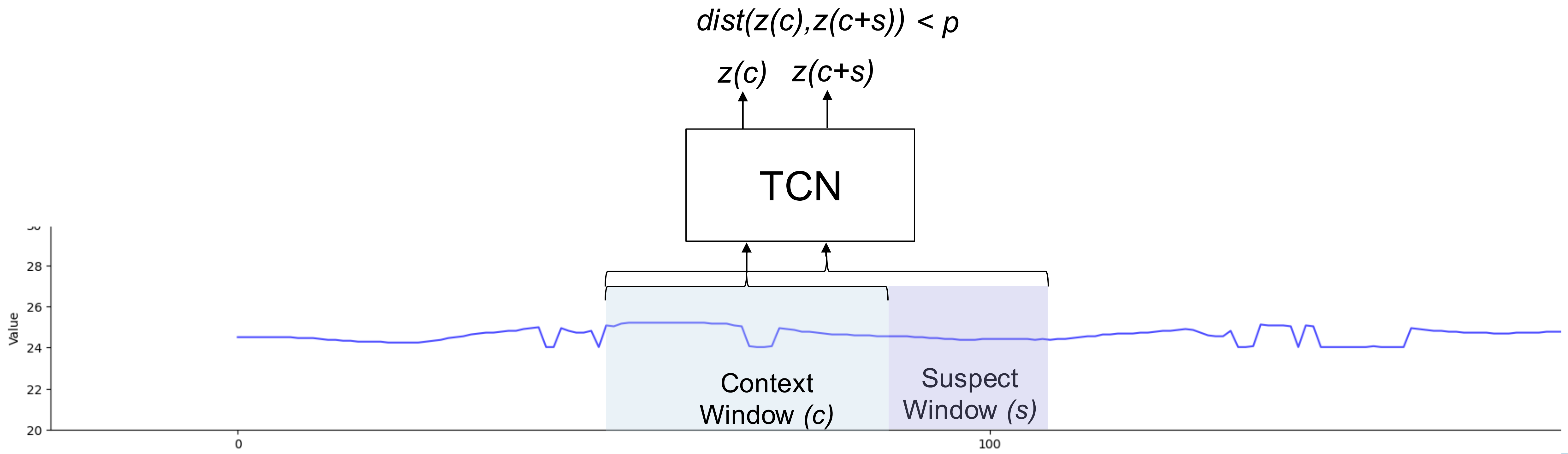
Methods: Testing Our Model

- We use the trained TCN model on new test data, sliding the context and suspect windows down one time step at a time
- The computed distance acts as our anomaly score. We threshold our scores using the median absolute deviation
- If the distance is greater than our threshold *(p)*, we flag the suspect window as an anomaly



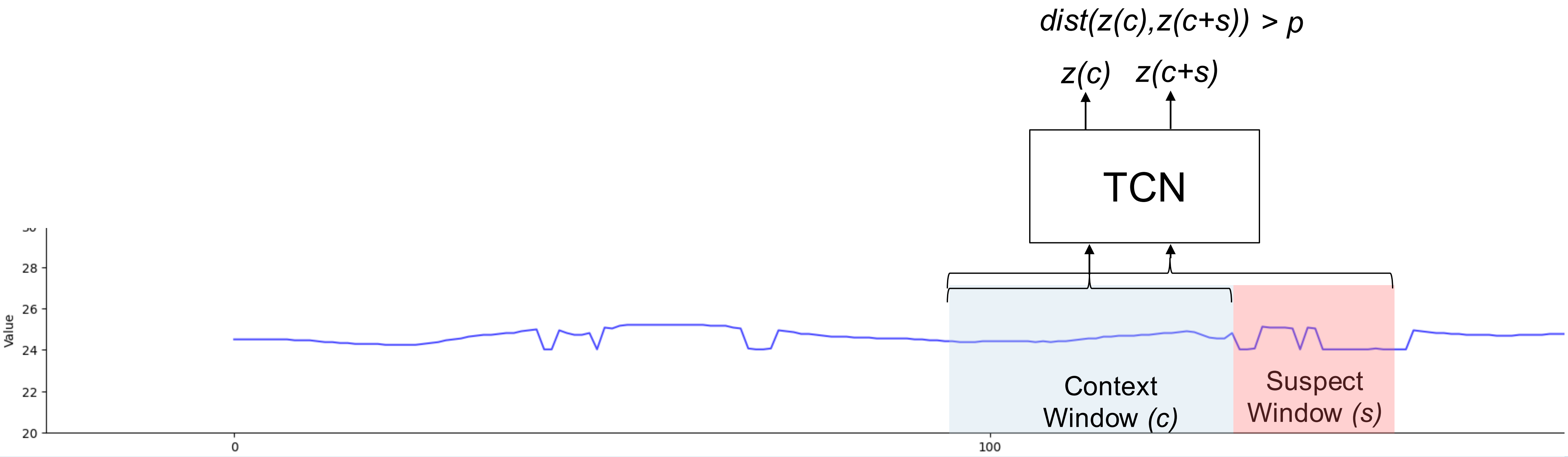
Methods: Testing Our Model

- We use the trained TCN model on new test data, sliding the context and suspect windows down one time step at a time
- The computed distance acts as our anomaly score. We threshold our scores using the median absolute deviation
- If the distance is greater than our threshold *(p)*, we flag the suspect window as an anomaly



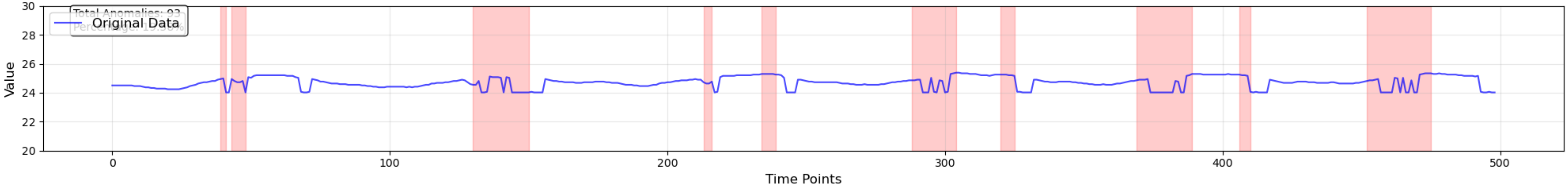
Methods: Testing Our Model

- We use the trained TCN model on new test data, sliding the context and suspect windows down one time step at a time
- The computed distance acts as our anomaly score. We threshold our scores using the median absolute deviation
- If the distance is greater than our threshold *(p)*, we flag the suspect window as an anomaly



Results

The results when applying our TCN across the entirety of the thermistor anomaly are visualized below. The red bars indicate a flagged anomaly.



Our results are encouraging, and further tuning of the model will improve accuracy.

Future Work

- Experiment with variations of the model architecture
- Apply model to new datasets
- Fine tune models with domain experts
  - Address false positives
  - Account for known events that impact data trends: (eclipse periods and maneuvers)
- Integrate models with existing GSFC telemetry processing tools like TaaS
- Apply a ‘library of models’ technique, giving operators more flexibility and insight

Questions? Comments?

Thank you ☺

Special thank you to the following major contributors:

Andre Williams, Software Engineer, NASA GSFC, andreali.williams.@nasa.gov

William Stanton, Systems Engineer, NASA GSFC, william.m.stanton@nasa.gov

James Clapsadle, MMS Mission Director, NASA GSFC, james.e.clapsadle@nasa.gov

Christopher Riley, MMS Mission Manager, KBR WYLE SERVICES LLC, christopher.m.riley@nasa.gov

Acronyms

Acronym	Definition
ANS	Alert and Notification System
CNN	Convolutional Neural Network
DR	Discrepancy Reporting
GSFC	Goddard Space Flight Center
ML	Machine Learning
MMS	Magnetospheric Multiscale
NCAD	Neural Contextual Anomaly Detection
TaaS	Telemetry as a Service
TCN	Temporal Convolution Network
TT&C	Telemetry Tracking and Command System

References

- Carmona C., Aubet F., Flunkert V., Gasthaus J. Neural Contextual Anomaly Detection for Time Series, URL https://arxiv.org/abs/2107.07702