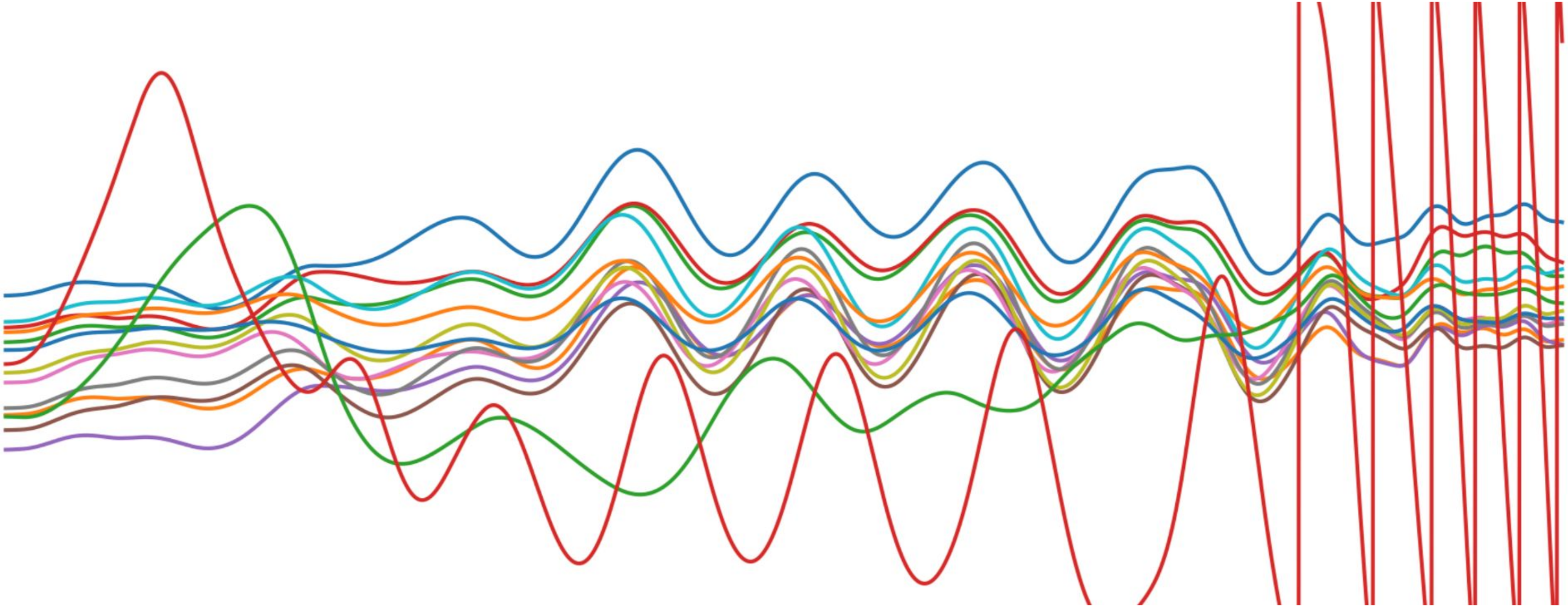


Using machine learning to predict the onset of blackouts

Data Study Group

Team Strathclyde

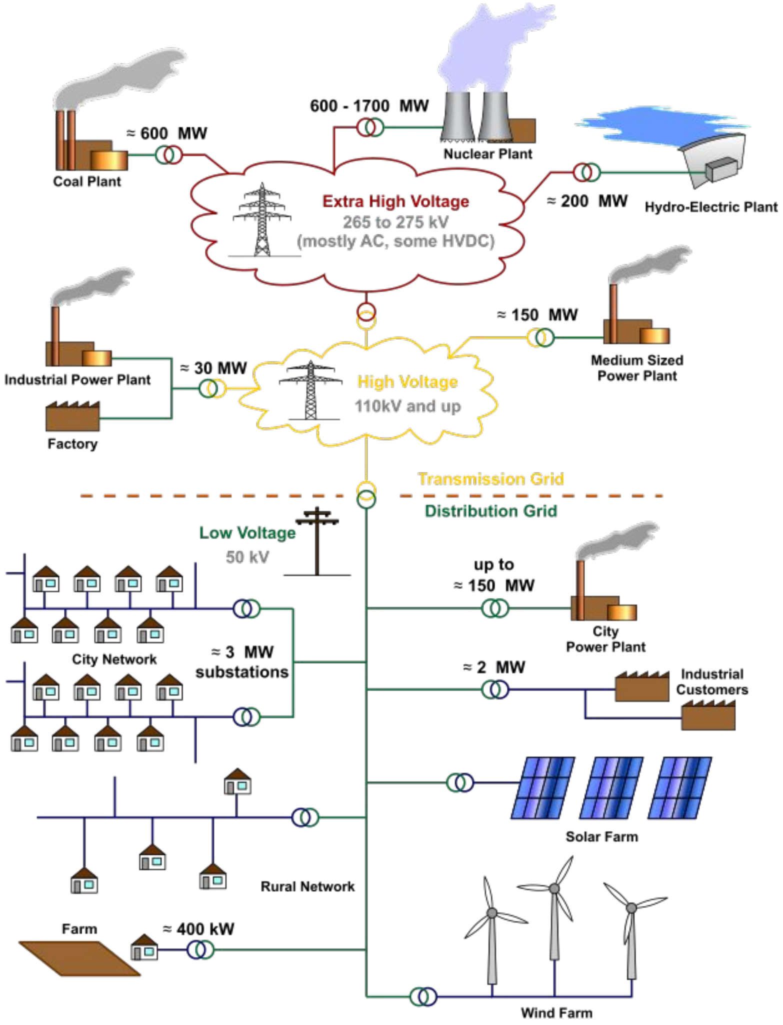
University of Strathclyde

The Alan Turing Institute

What is a power grid?

A power grid is an interconnected network for delivering electricity from producers to consumers.

- **Generators** that produce electric power
- **Buses** - electrical substations for stepping electrical voltage up for transmission, or down for distribution
- High voltage **transmission lines** that carry power from distant sources to demand-centers
- **Distribution lines** that connect individual customers



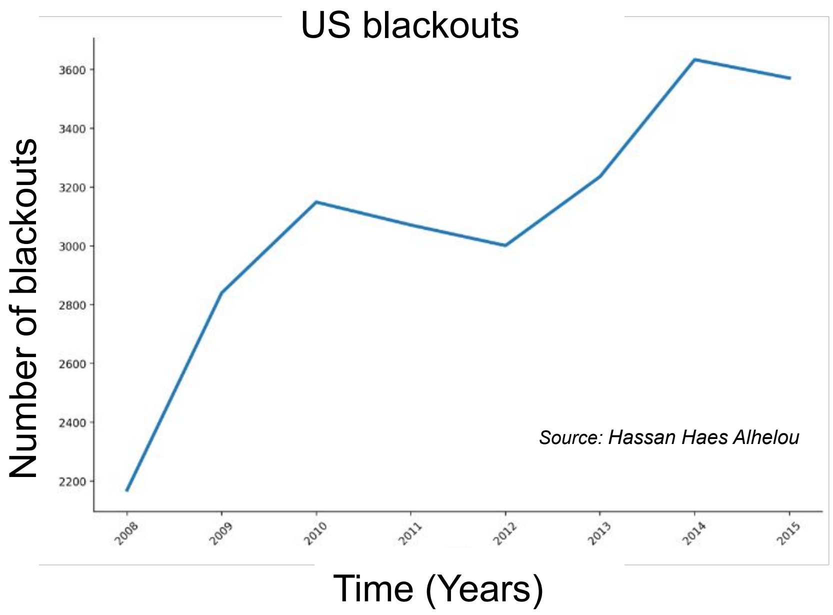
The problem

- Power grids are vital parts of modern infrastructure and electrical blackouts have catastrophic consequences.

- Blackouts are becoming more frequent due to added grid complexity and an increase in extreme weather events linked to climate change *.

- High renewable penetration from wind and solar increases the vulnerability of the grid to blackout events following a disturbance (fallen tree, lightning strike, car crash etc.)

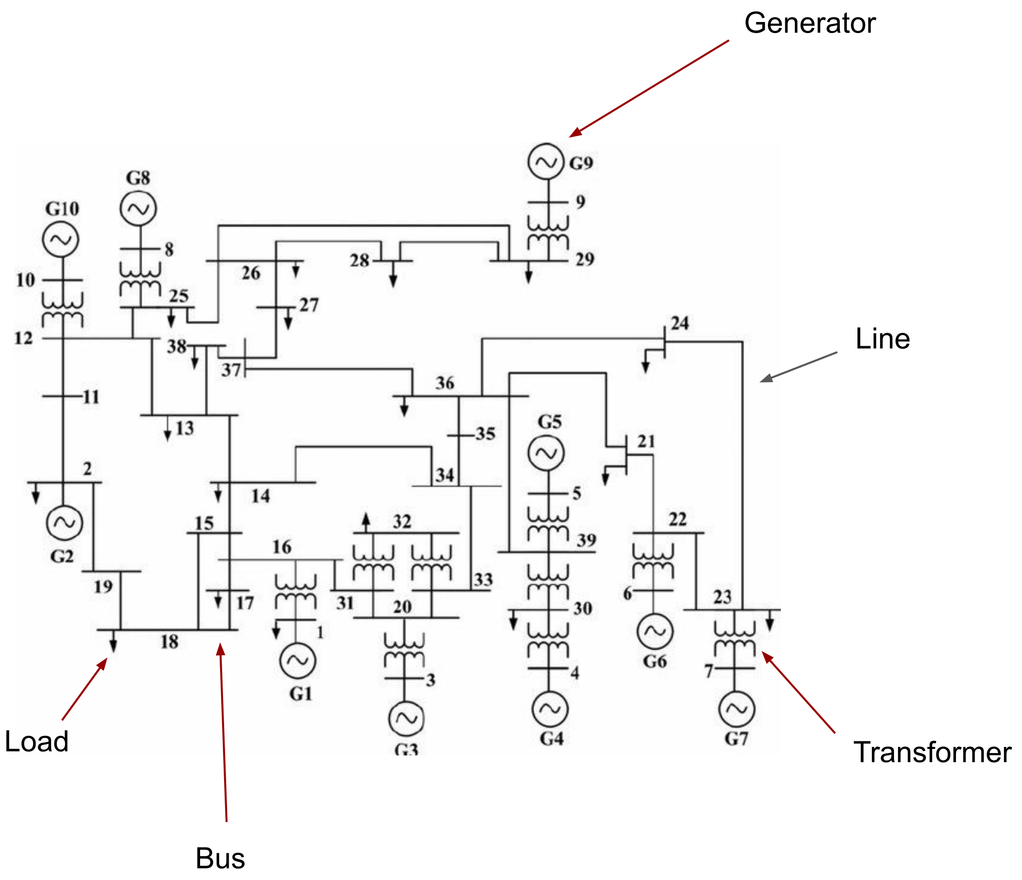
- Power grids are connected networks, hence disturbances often lead to sequential / cascading failures, causing widespread blackouts.



* (50% of global blackouts from 2011 to 2019 were related to weather)

Data simulation

- ~45000 mathematical simulations of a power grid
- Each simulation starts with a disturbance on a single line in the system. Our task is to predict the subsequent cascading failures in the system which vary in magnitude
- Our dataset contains 250 features which pertain to the main elements in the grid (generators, lines, busses and loads)
- Given the extreme non-linearity and complexity of modern power grids, it is impossible to analytically calculate dynamic behaviour in real time, hence, machine learning is highly appropriate in this use case



Exploratory data analysis

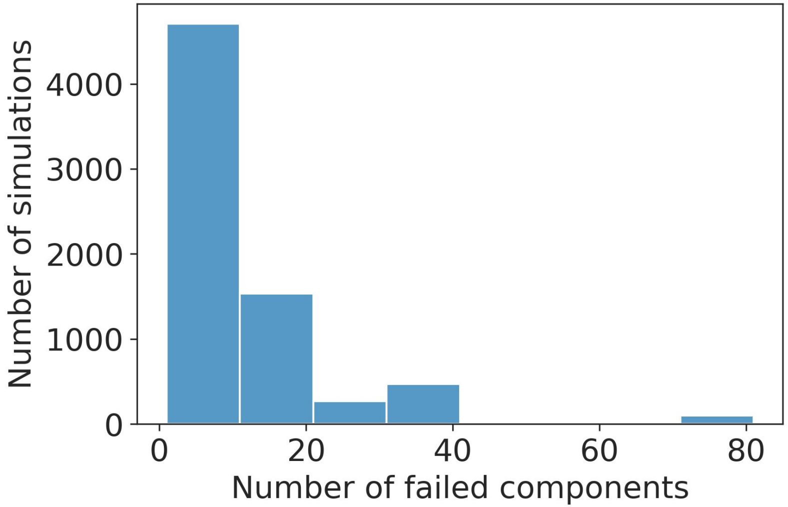
- The dataset is imbalanced with only 7131 simulations with failures.
- Some of the simulations are discarded due to errors and algorithm convergence issues

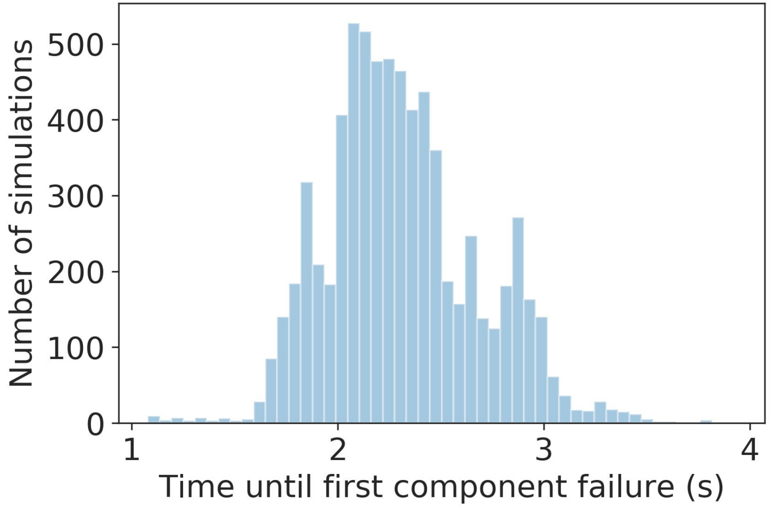
Flag	Event	Number of instances
0	No component failures	35934
1	One or more component failures	7131
3	Errors in simulation	999

Table 1: Summary of failure events for all simulations

Exploratory data analysis

- In cases where cascading failures occur, the majority of these cases contain less than 10 failures. These labels are sparse, typically each simulation contains thousands of time steps
- The average time of the first cascading time is approximately 2.5 seconds. Further justification for machine learning in this application over analytical solutions, fast prediction time in real-world settings is critical

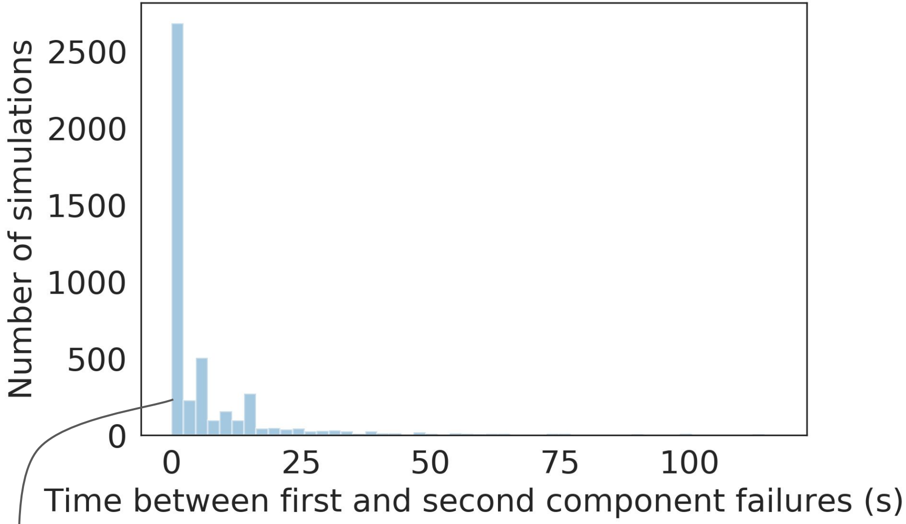


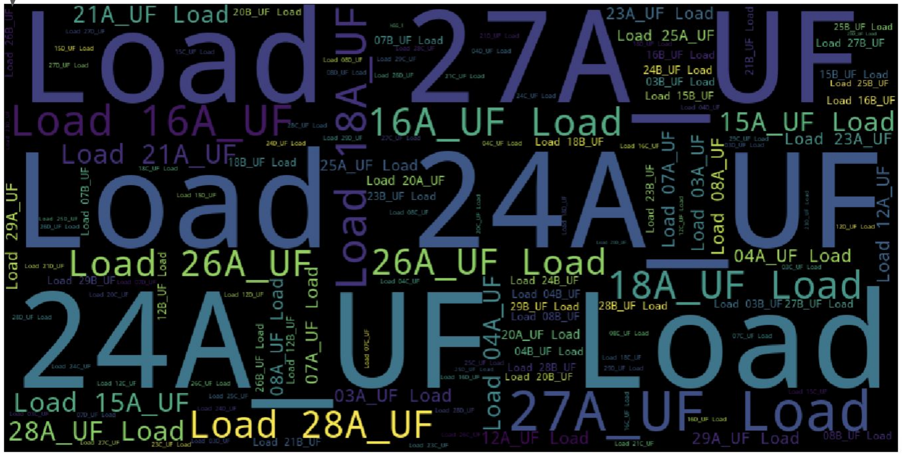


Exploratory data analysis

- In cases where cascading occurs, the time between the first and second failure is fast (in the order of milliseconds)

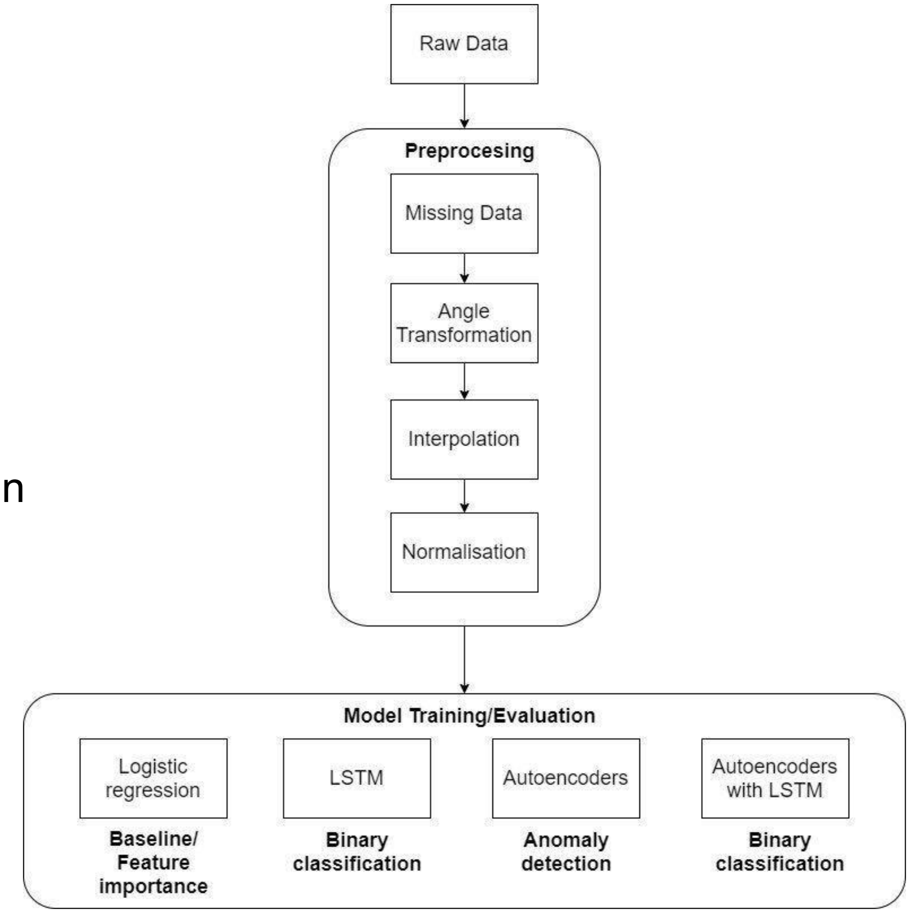
- Cascading failures represent a multi-class classification problem. This class distribution is highly imbalanced, and some classes are significantly more common than others. The majority of failures are related to the loads in the system





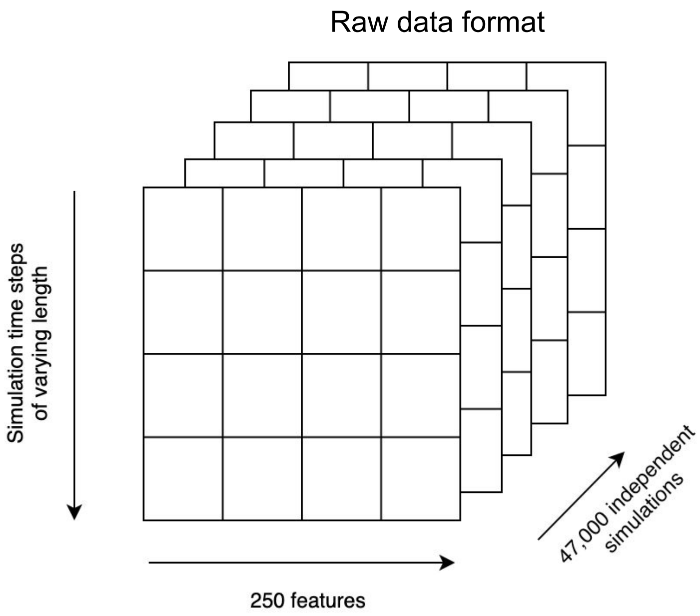
Methodology

- Exploratory data analysis and method selection
- Preprocessing
- Training / validation / test set generation
- Model training
- Evaluation



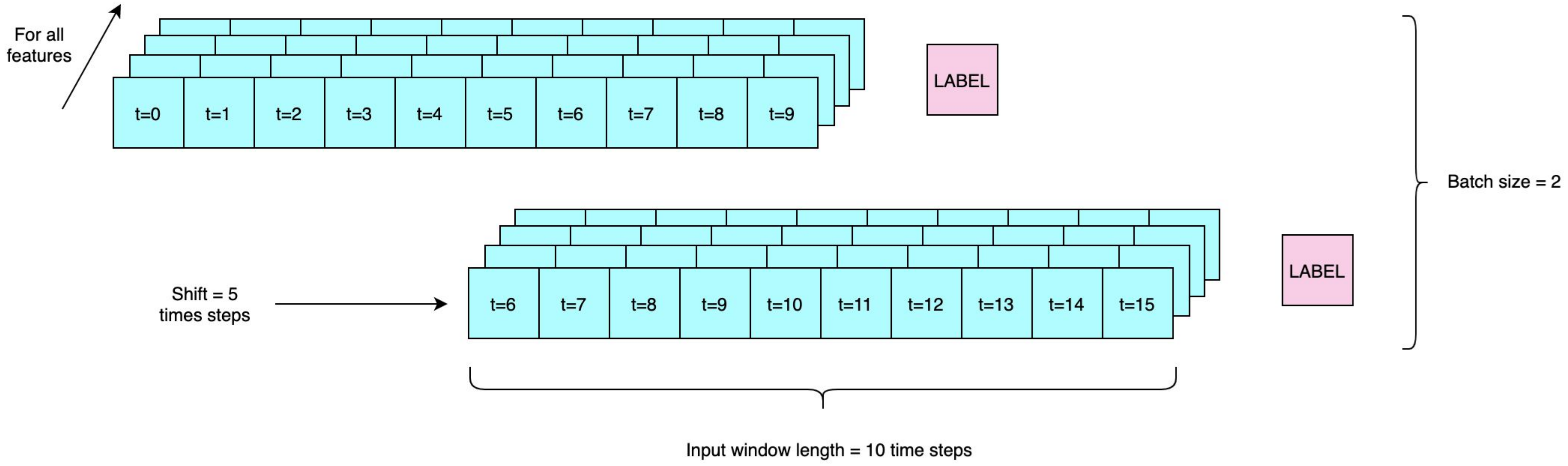
Data processing

1. Ensure all feature dimensions are the same for all simulations
2. Normalise each feature w.r.t statistics of all simulations (holding out a test set)
3. Interpolate time series to ensure even spacing
4. Remove data before the initial system disturbance
5. Window time series data
6. Generate target labels w.r.t each training window
7. Given size of dataset, previous steps were completed in a loop and windows saved to pickle format
8. Build model input pipeline



Data windowing

- Example of two data windows from a single time series for binary classification (P(window) = Failure)
- In our case, we choose input window length of 100, shift value of 25, and batch size of 32



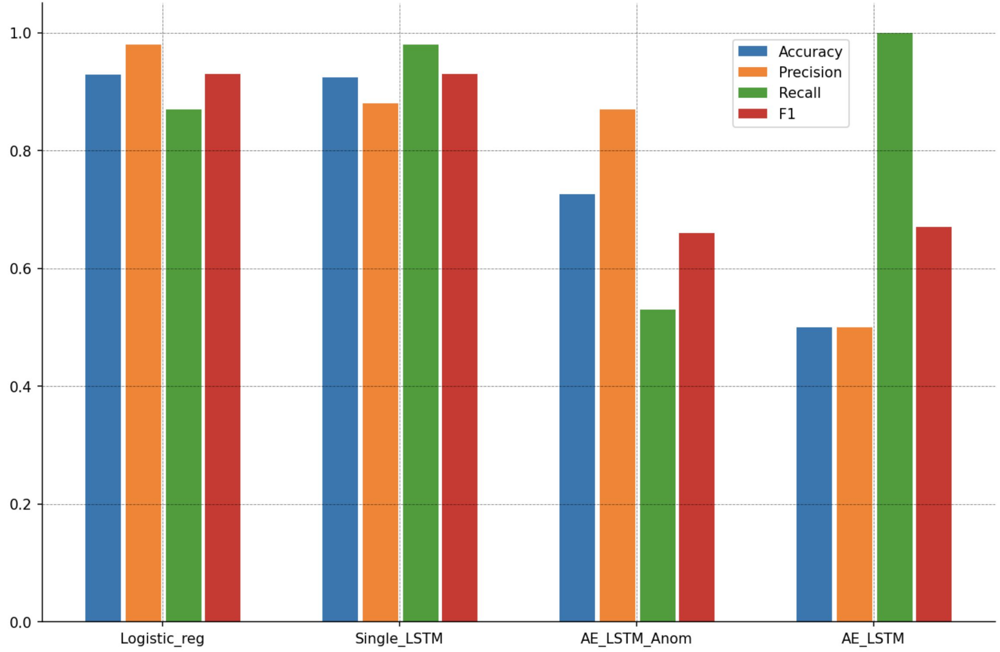
Model architectures

- Baseline - Logistic Regression
- Single layer LSTM model
- Autoencoder - unsupervised anomaly detection
- Encoder-decoder LSTM model

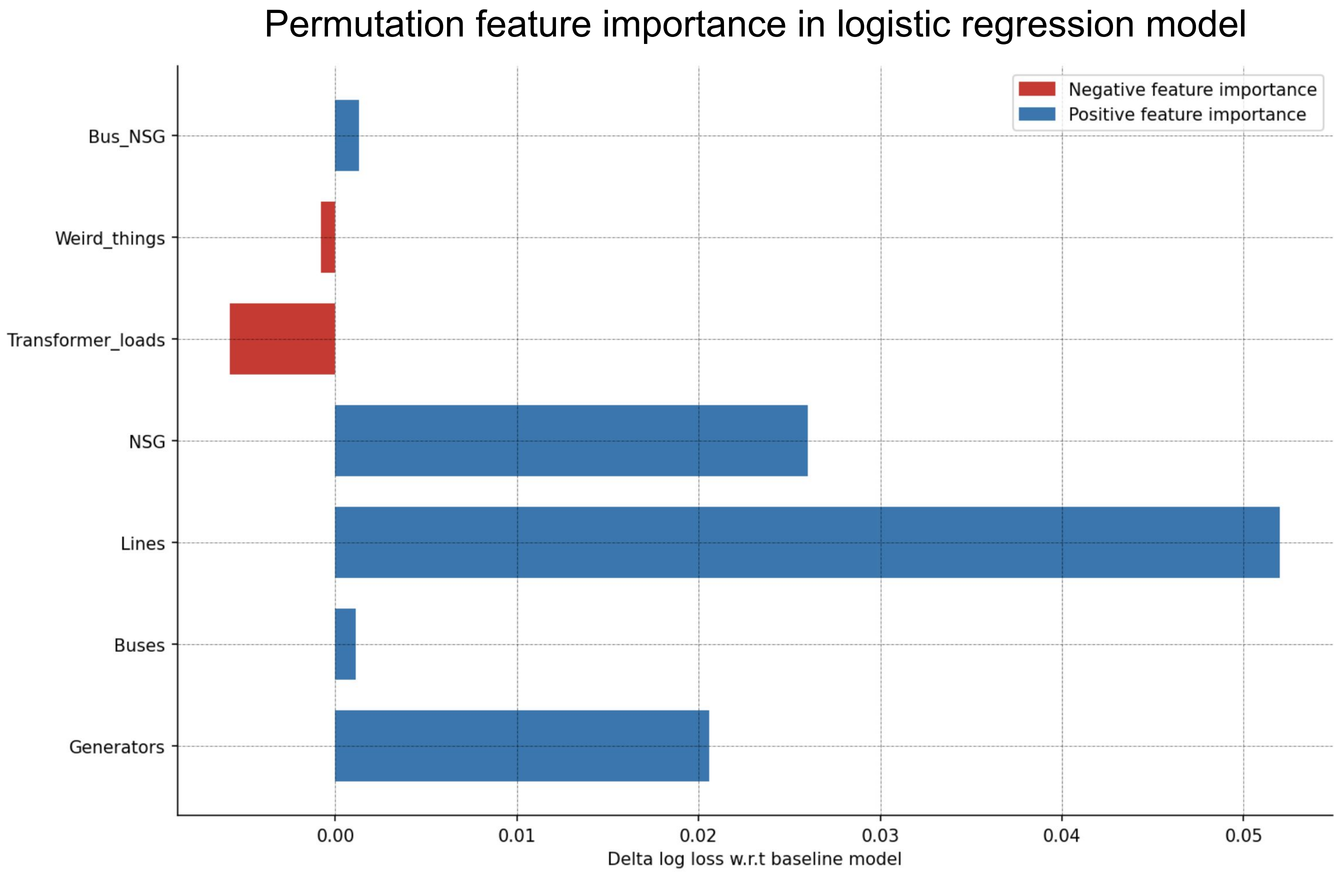
Complexity

Results and performance

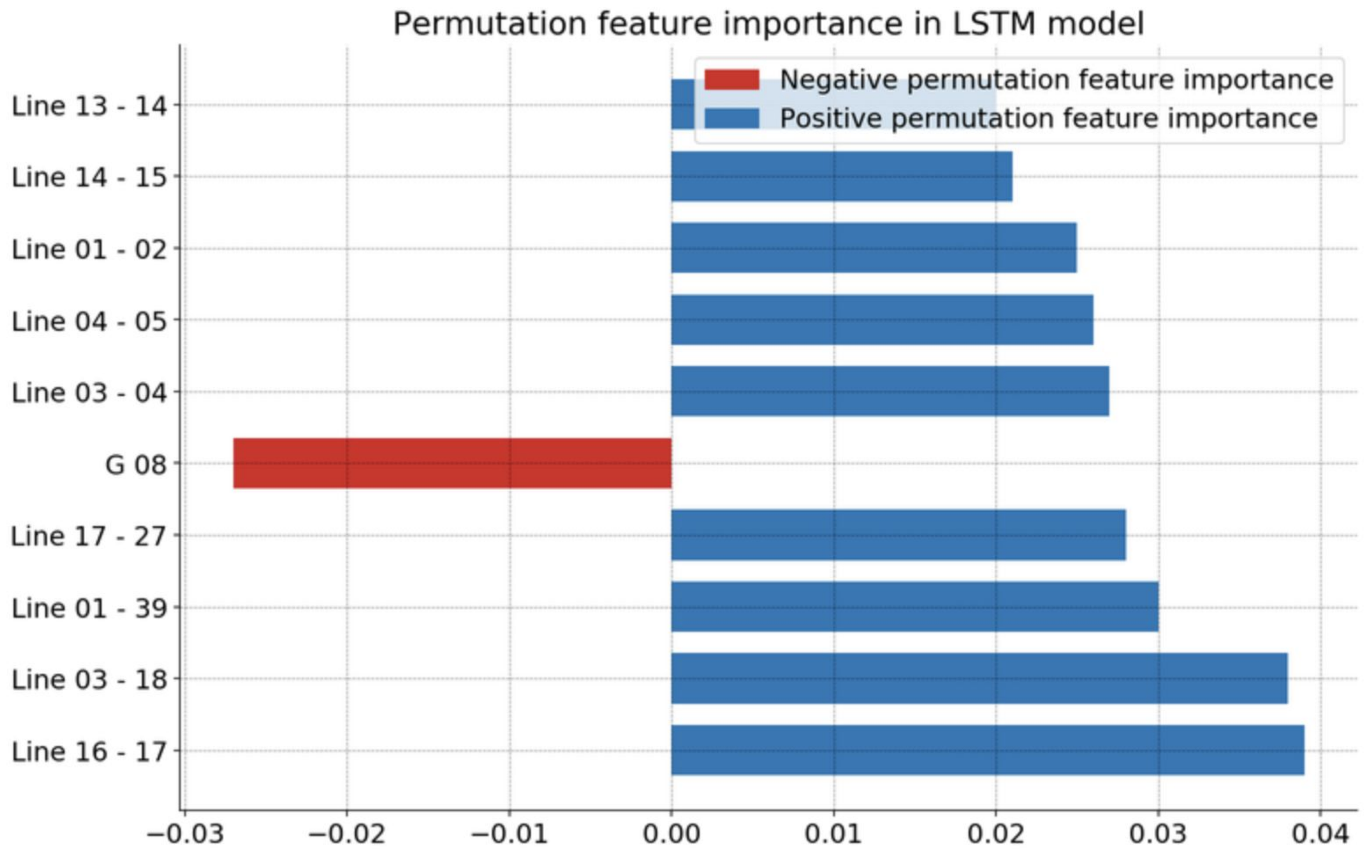
- Both single layer LSTM and logistic regression models achieve impressive results with an F1 score of 0.93
- We can successfully predict the first failure at latest 0.5 seconds before it happens
- We are able to achieve impressive results even without using the fail/no fail labels by learning what 'normal' looks like



Feature importance - grouped by category



Feature importance - by individual feature



Challenges

- The dataset is highly imbalanced with few simulations with failures (single component and multi-component / cascading failures).

- In the event of a cascading failure, the components fail almost instantaneously after the first failure. Hence, it is much more important to predict the first failure rather than predicting the exact sequence of the cascade.

- Due to the large size of the dataset, the methods used in our work are limited to ones that allow batch processing and incremental training, such as neural networks. This limited the explainability of the final solution presented.

Future work

- Which failures are more catastrophic?

- How will the models perform using less data (how early can we predict failure)?

- Can we predict which components will fail?
  - Graph neural networks
  - Feed forward neural networks

- Dimensionality reduction