Library anomaly detection using Deep learning

Srinivas Bodapati, Vidya Sagar Reddy Gopala

Motivation

- Standard cell characterization data at multiple process, voltage and temperature (PVT) runs into Terabytes.
  - Validation becomes a big data problem

- Library collateral/data quality is essential to SoC design.
  - Collateral error(s) may cause design failure
  - Cost of fixing library data is expensive if errors found late in the design cycle
  - Negatively Impacts project schedules.

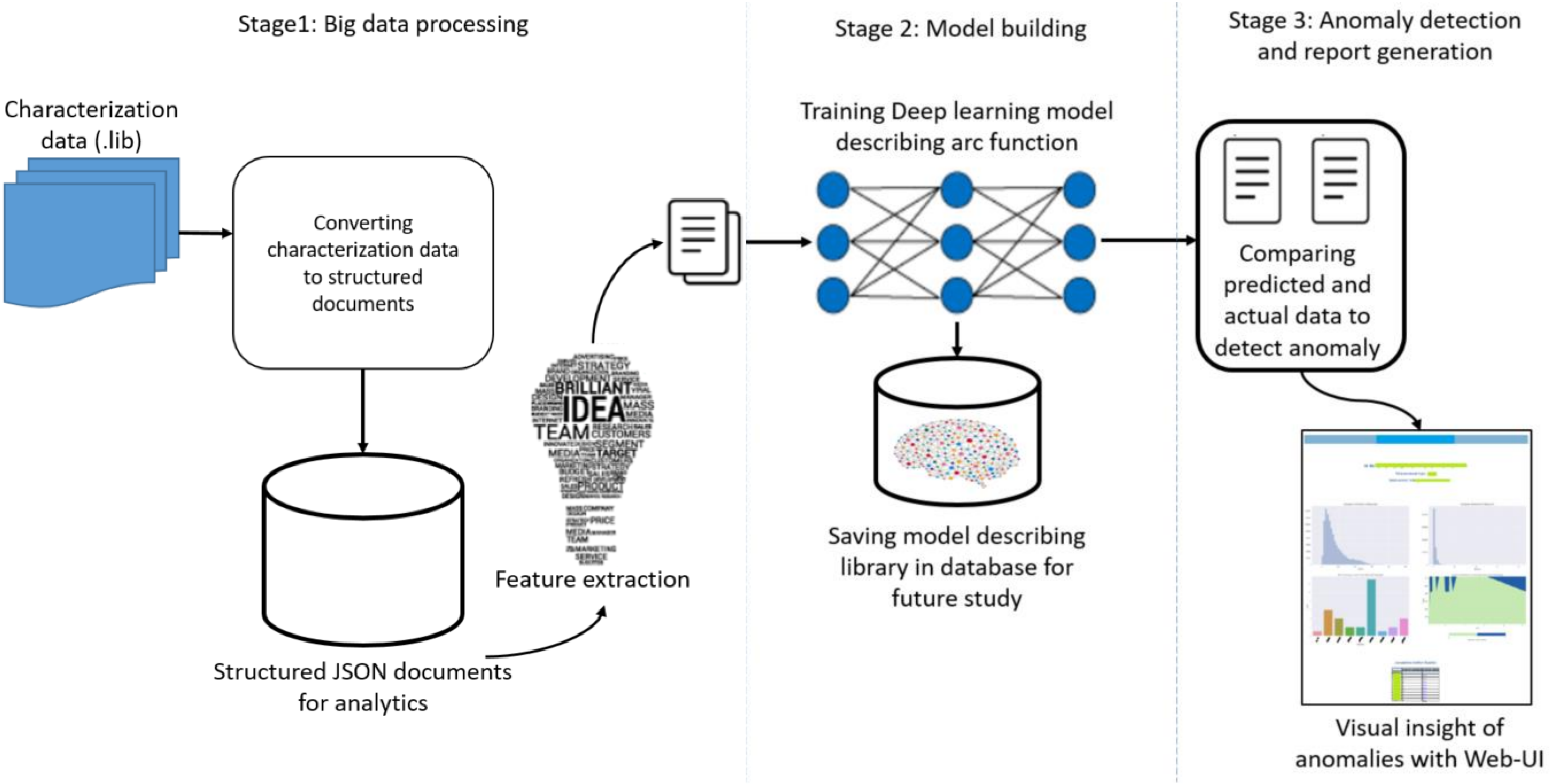
- Need scalable validation solution that is
  - Accurate, fast (quick turn around time (TAT)), enables easy debug

Proposed Solution

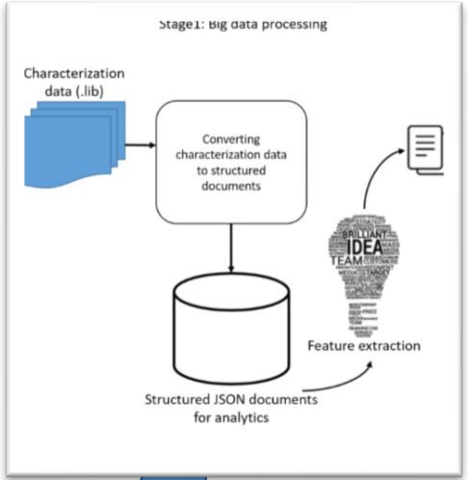
- Developed new deep learning based library validation infrastructure with Web-UI that is fast, accurate and enables easier debug of collateral.
  - It allows automatic root cause analysis and captures anomalies and trends.

- Enables detection of anomalies in library characterization
  - Strange/erroneous char data due to any reason
  - Invalid trends
  - Identifies the exact cell/timing-arc and char point that

Library Validation Architecture



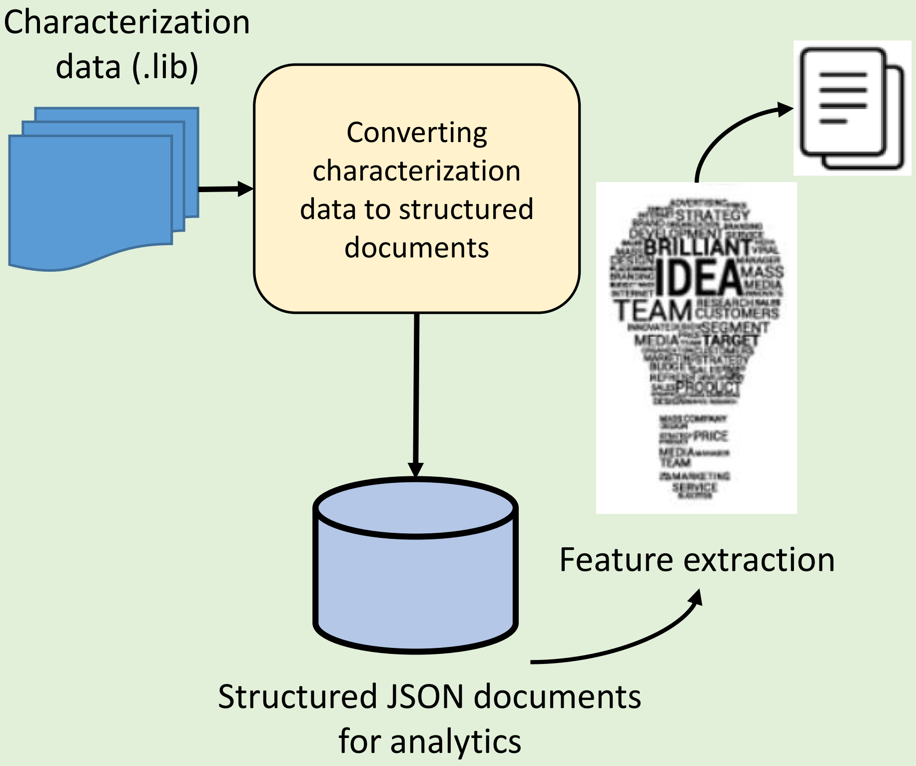
Stage 1: Big data processing of library data



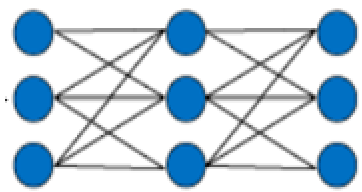
Stage 1

Terabytes of Library Characterization data is converted to Structured data with any distributed computing Frameworks.

- Parsed library data is converted into JSON documents for further analytics.
- For each measurement, features are extracted for model inputs.
- Extracted features include library, topology, cell and arc information.

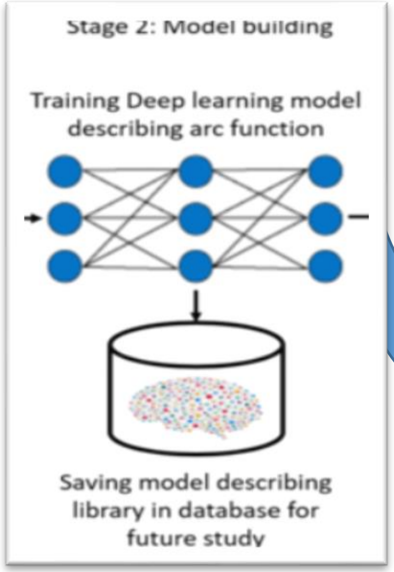


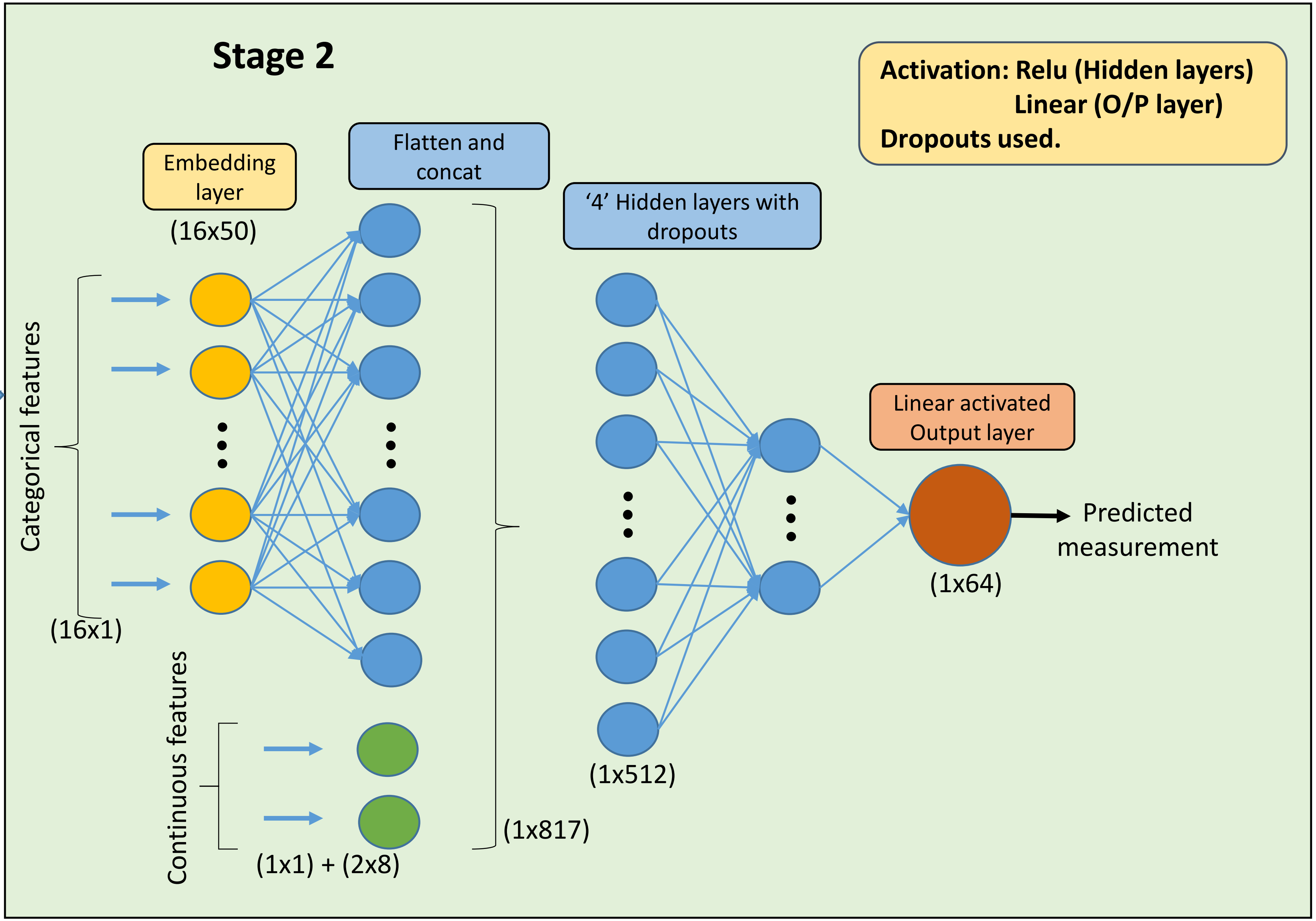
Neural network



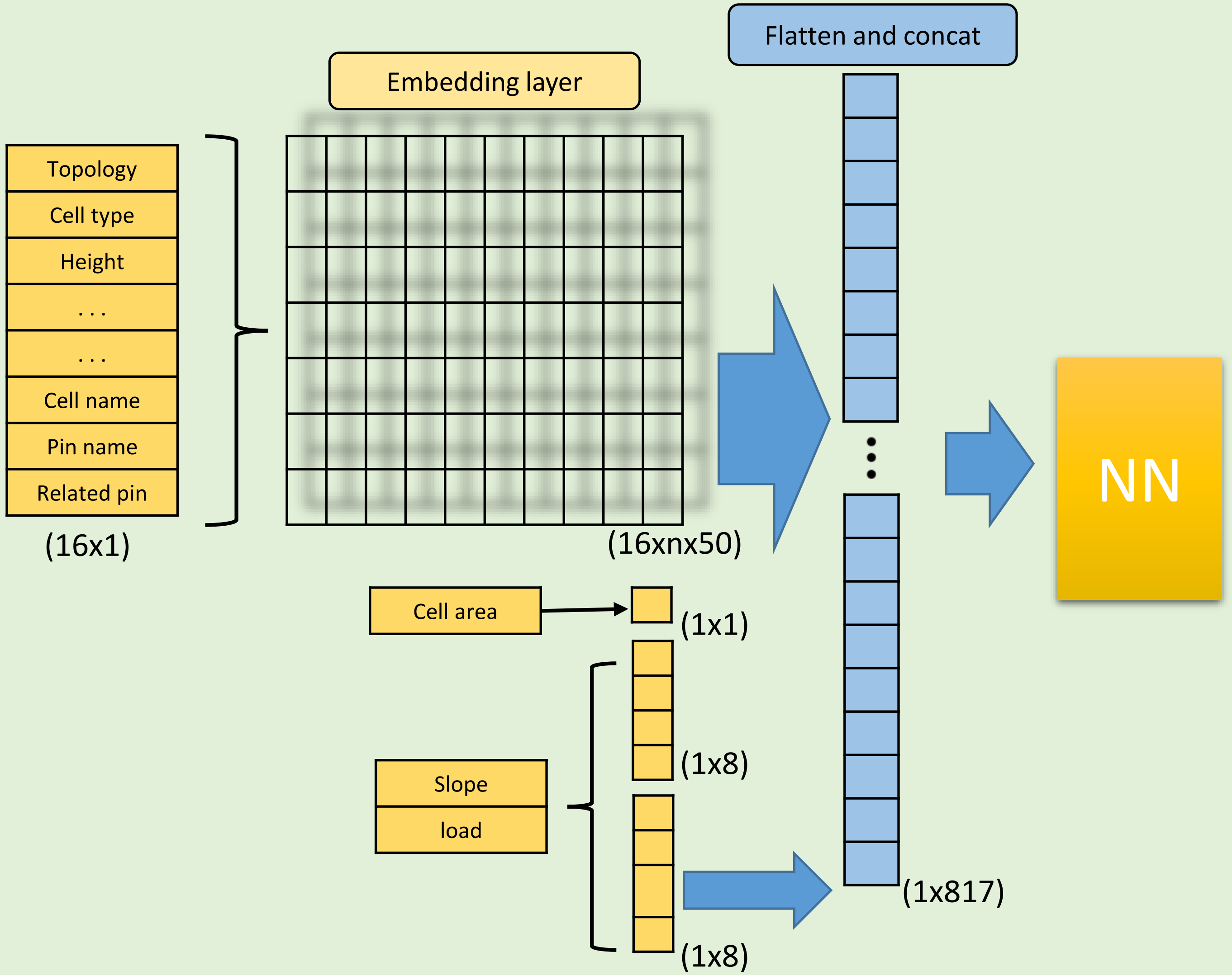
Data-frames of input features are fed to neural network to predict required measurements such as delay, slew, capacitance and more.

Stage 2: Deep Neural Net Architecture





Features and embedding

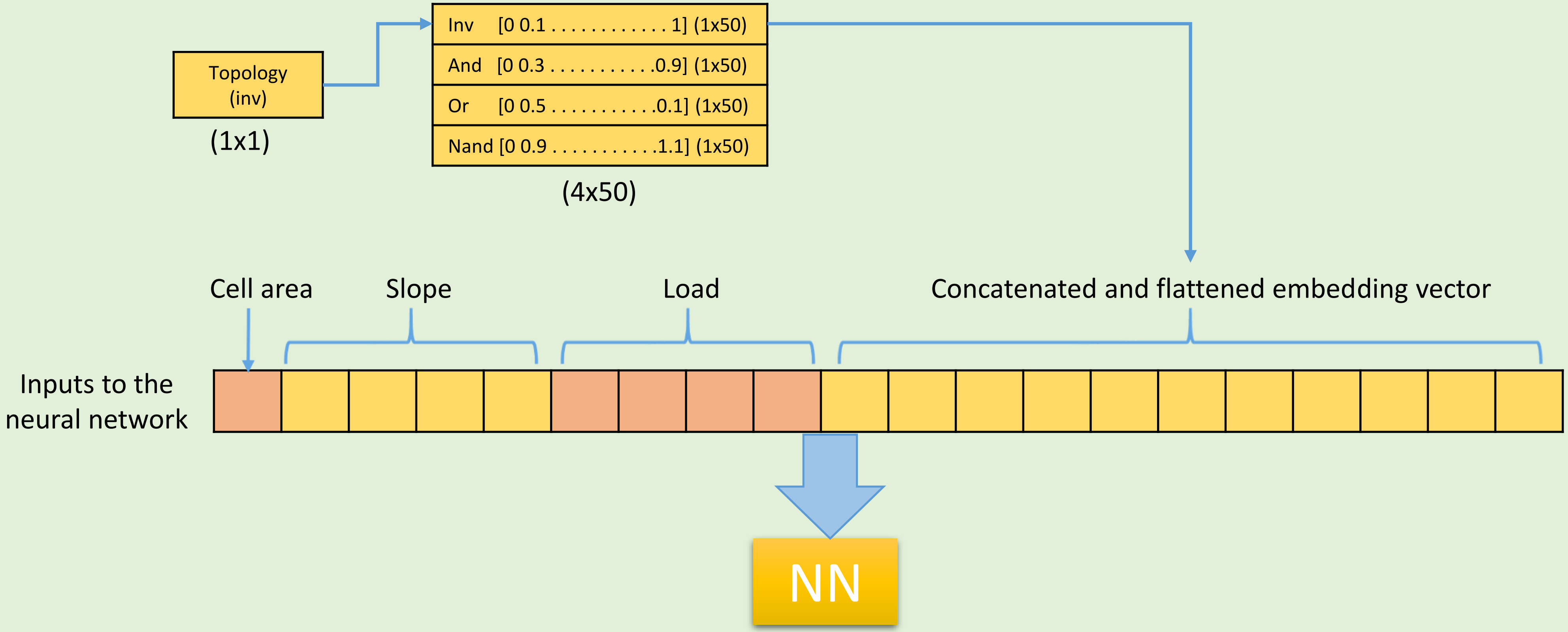


- Embedding layer greatly improves the modelling.
  - Maps categorical variables to vector space.
  - Vector is adjusted with the training.
- Categorical inputs are encoded, padded and fed to embedding layer.
- Continuous features are concatenated with flattened output of embedding.
- Embedding layer is followed by hidden & output layers.

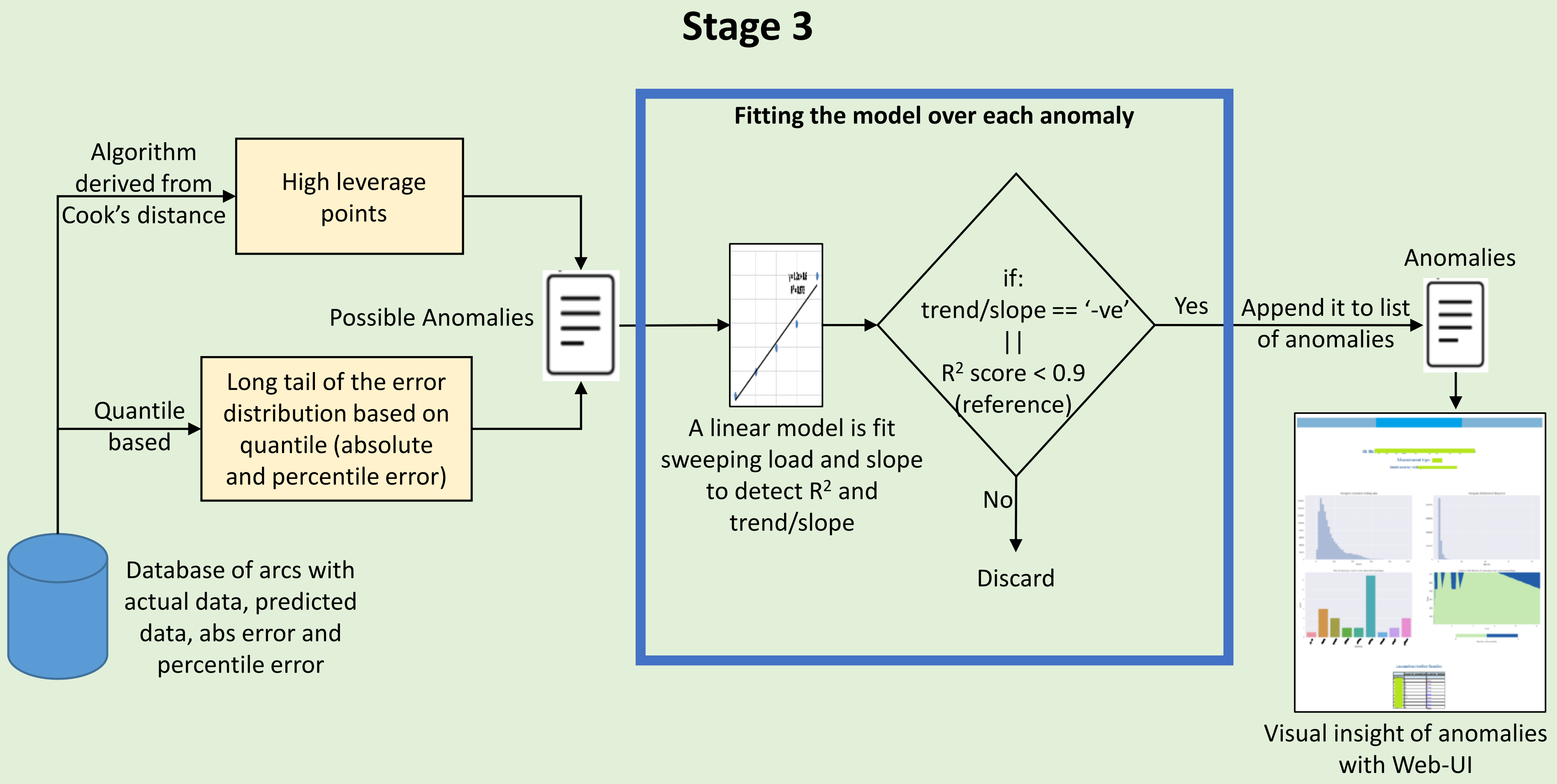
- Here in the block,
  - **'n'** **:** number of unique categories (size of vocab)
  - **'50':** Standard size of vector good enough for multi-dimensional grouping.

Embedding layer explained

- Here for example, let us consider 1 categorical input (topology) with 4 unique categories and vector size of 50 for each unique category. Let us assume **inv**, **and**, **or** & **nand** as unique types in category called topology.
- A embedding layer of size 1X4X50 is build and trained to group categories closer on multi-dimensional space.

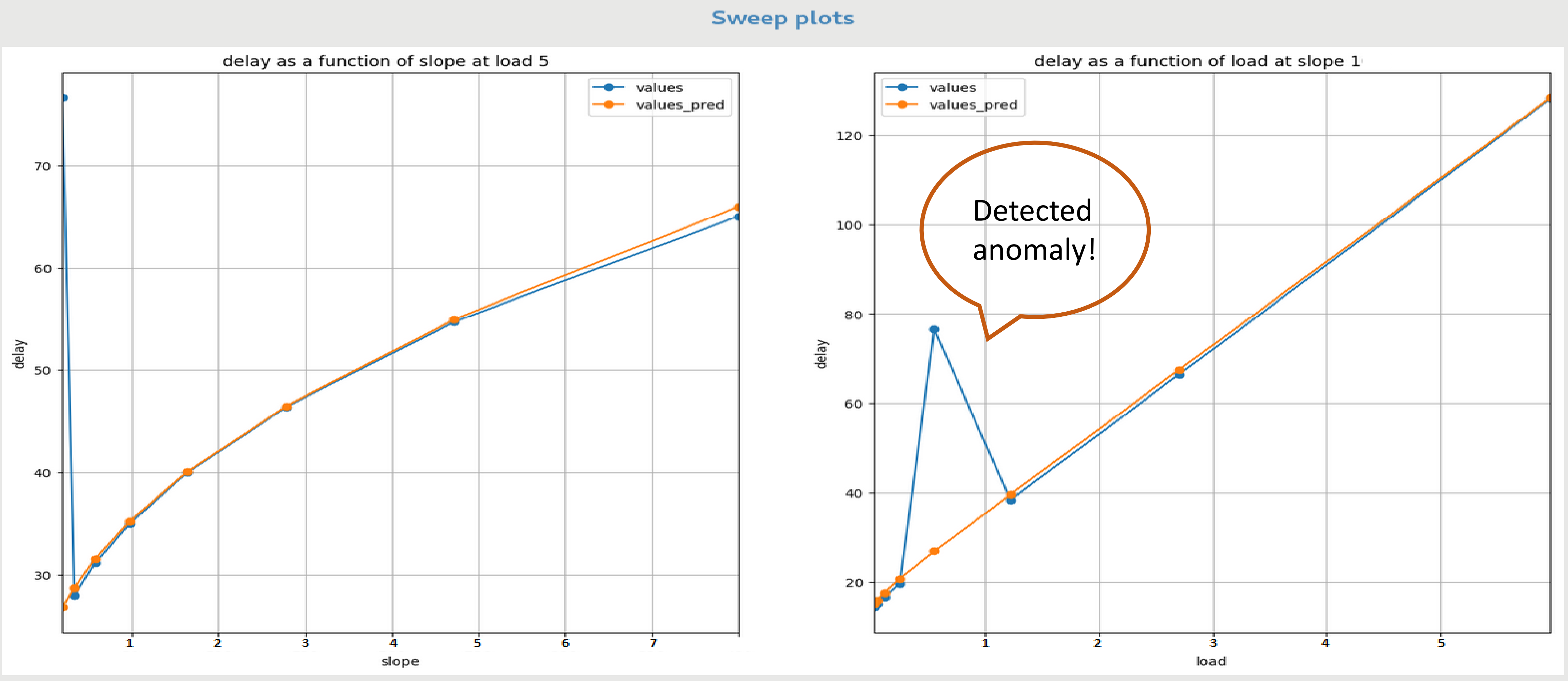


Stage 3: Anomaly detection and report generation

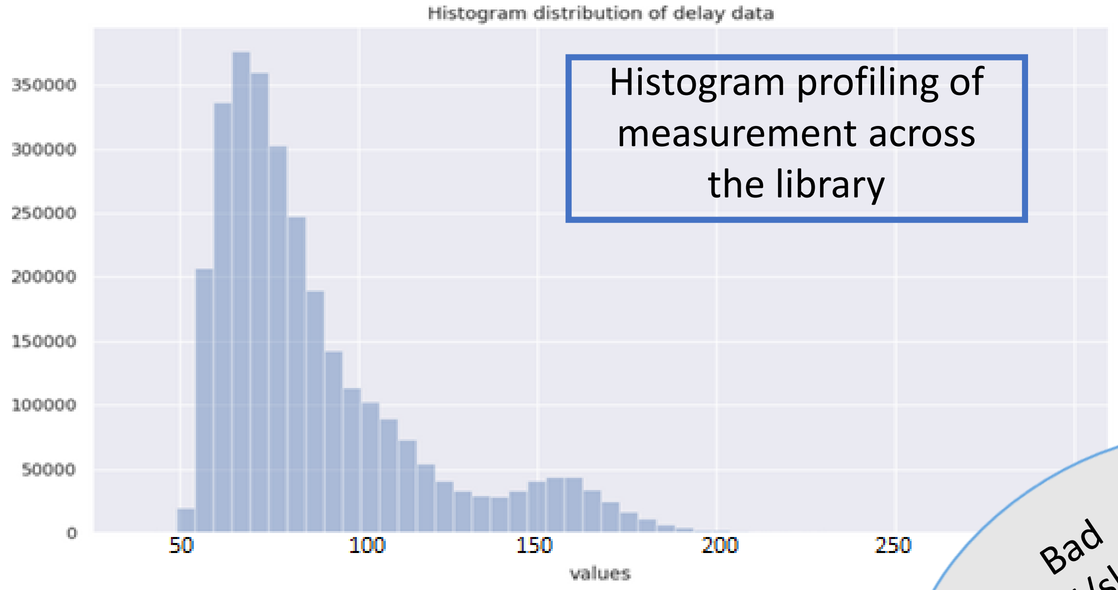


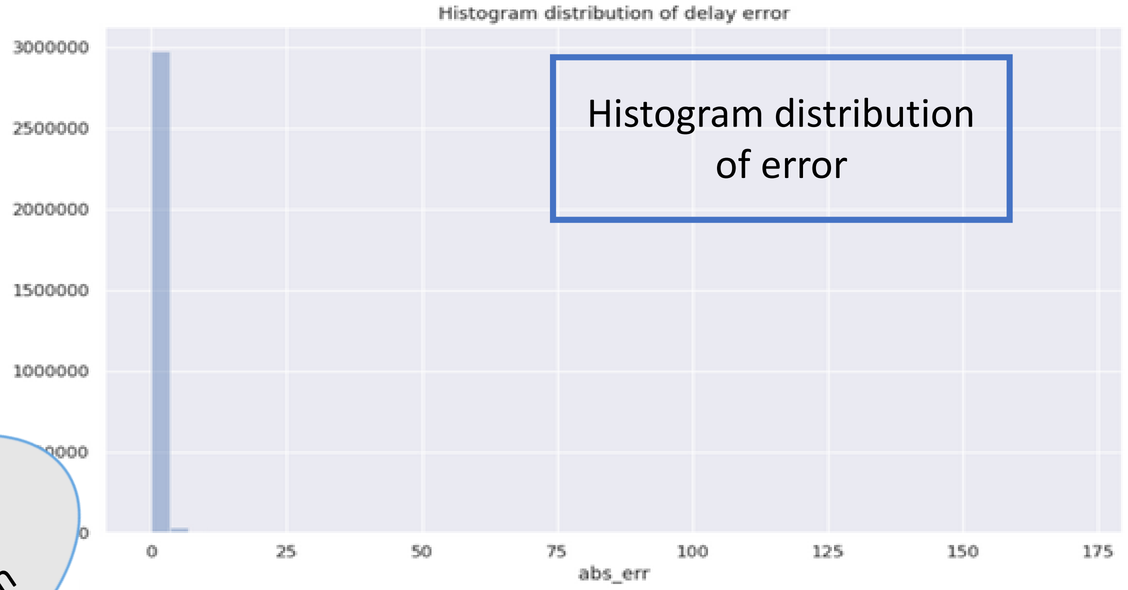
- Web-UI is developed to visualize anomalies and navigate through characterization database to debug the anomaly easily.
- Anomaly characterization is difficult as there are very few anomalies, so we need to filter true outliers after prediction stage.

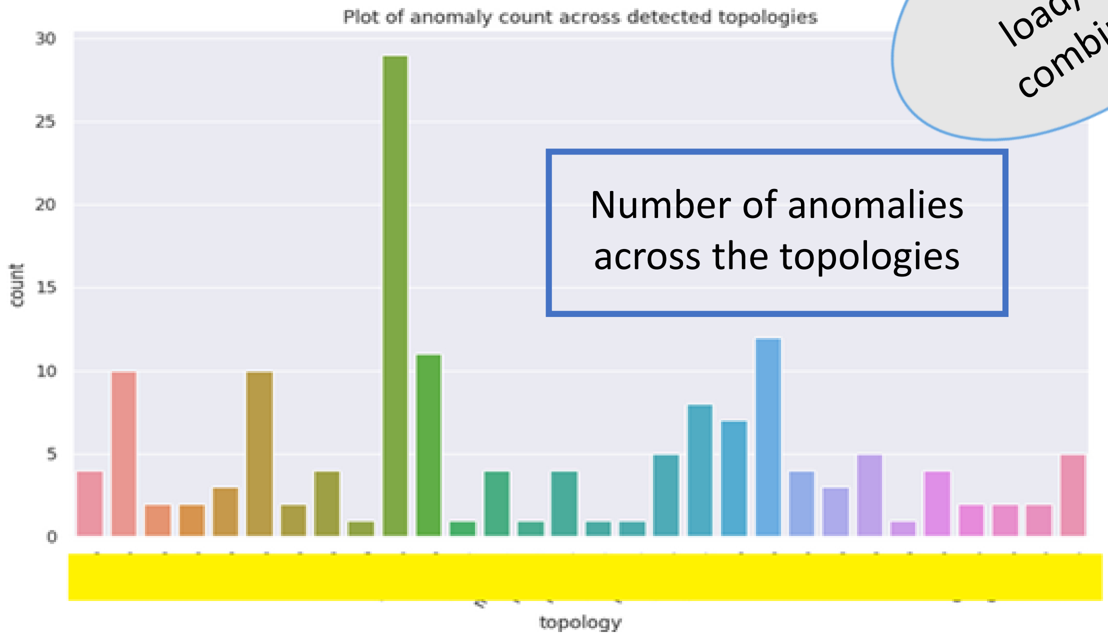
Results

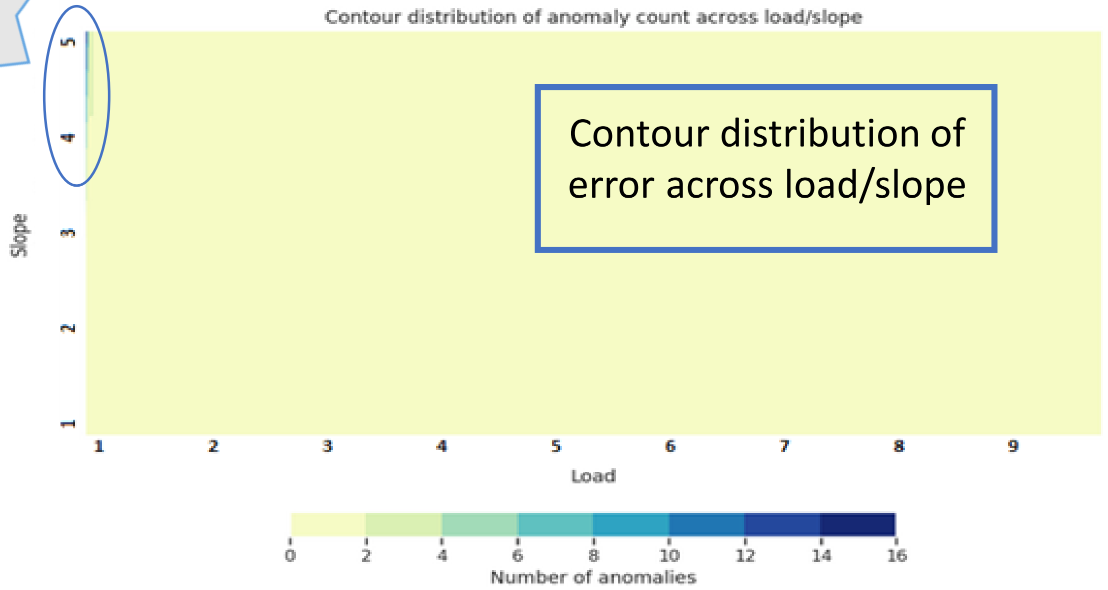

Sweep plots
delay as a function of slope at load 5
values
values_pred
delay
slope
70
60
50
40
30
1
2
3
4
5
6
7
delay as a function of load at slope 1
values
values_pred
delay
load
120
100
80
60
40
20
1
2
3
4
5
Detected anomaly!

Results (Contd..)









Summary

- Presented a deep learning based advanced validation tool to capture trends over entire library.
  - Statistical anomaly methods used in detecting anomalies.
  - More robust than standard threshold based outliers.
- Detects anomalies which do not follow the trend, source of anomaly could be
  - Simulation failures/errors, lsf issues
  - Incorrect configuration/setup
- Adaptable trend based validation not susceptible to technology changes like transistor models, interconnect models etc.
  - Traditional methods require thresholds that may change with technology changes.
- Identifies true anomalies and reduces false positives compared to traditional threshold based methods.
  - Traditional methods require aggressive thresholds else they generate millions of violations.
- Identifies cells/arcs with bad trends and navigates designer to exact source of the anomalies.
- Deep learning validation QA captures the trend and correlation across the library and cells. Wherein, traditional methods only captures local trends of the arc.