Predictive Crosstalk Fixing Using XGBoost Regressor

Andy Li
Aniket Agrawal
Lalit Arora
Anmol Khatri
Soumik Mukharjee
Praveen Chinta

intel

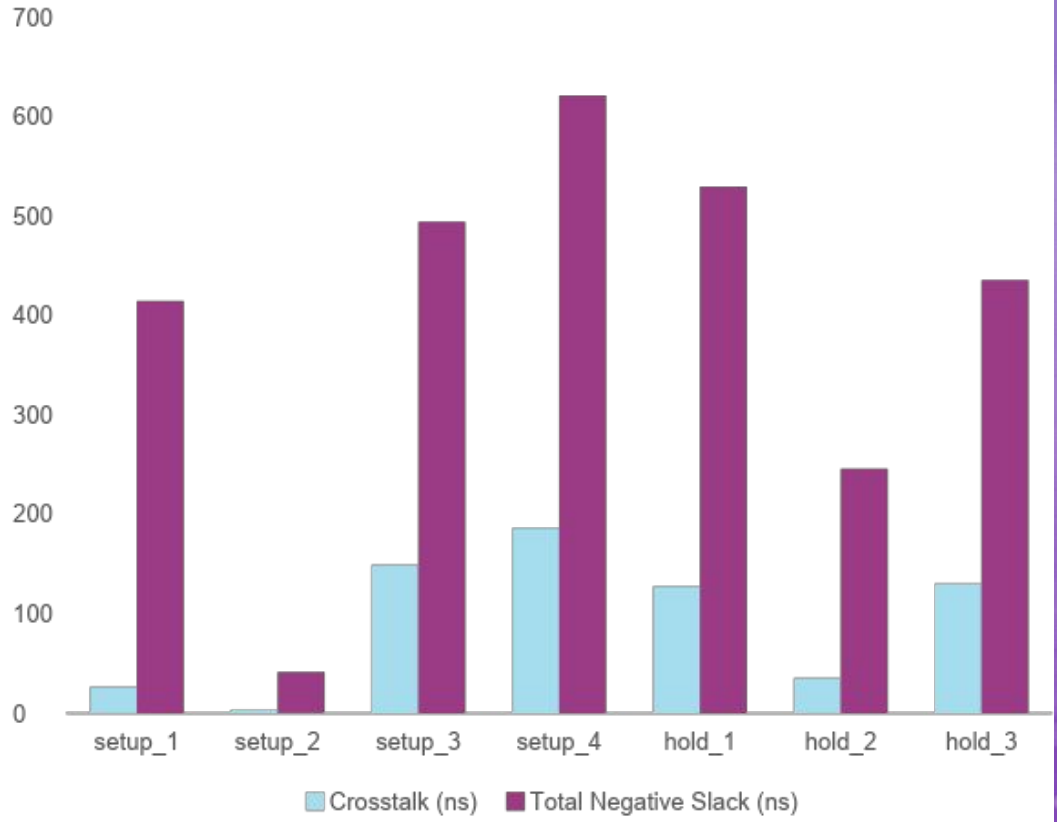
Index

- Motivation
- Proposed Solution
- Crosstalk prediction using XGBoost regressor
  - High level data flow
  - Primary and secondary features
  - Cascading of ML models

- Crosstalk fixing
- Evidence/results
- Summary and future plans

Motivation

- Crosstalk constitutes a major bottleneck for timing closure for deep submicron technologies.
- For some critical corners crosstalk delay delta can even constitute up to 30% of the total negative slack .
- Crosstalk impacts both setup and hold at the same time.
- Fixing crosstalk manually is a cumbersome and iterative process with extensive use of timing engine and update timing.
- An algorithm which can predict crosstalk without timing tool and with ballpark accuracy independent of timing engine is a huge motivation.



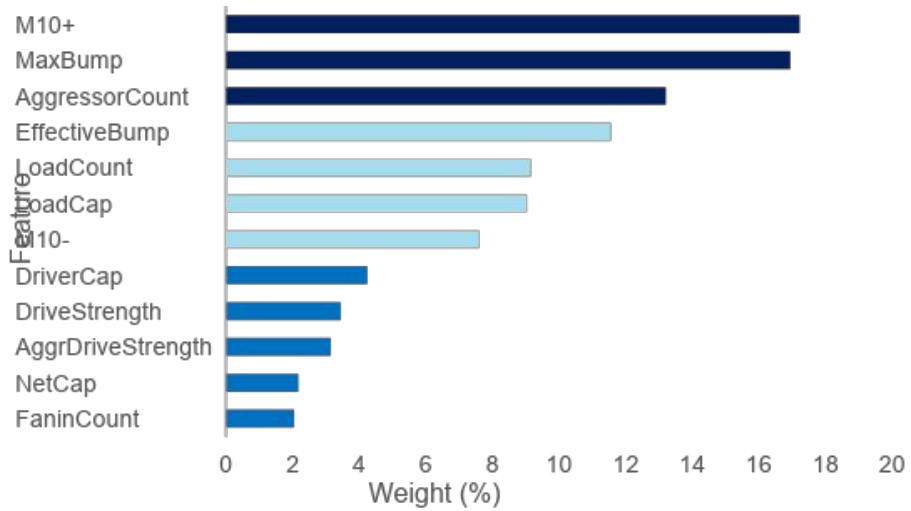
crosstalk severity in the design across multiple corners

Proposed Solution

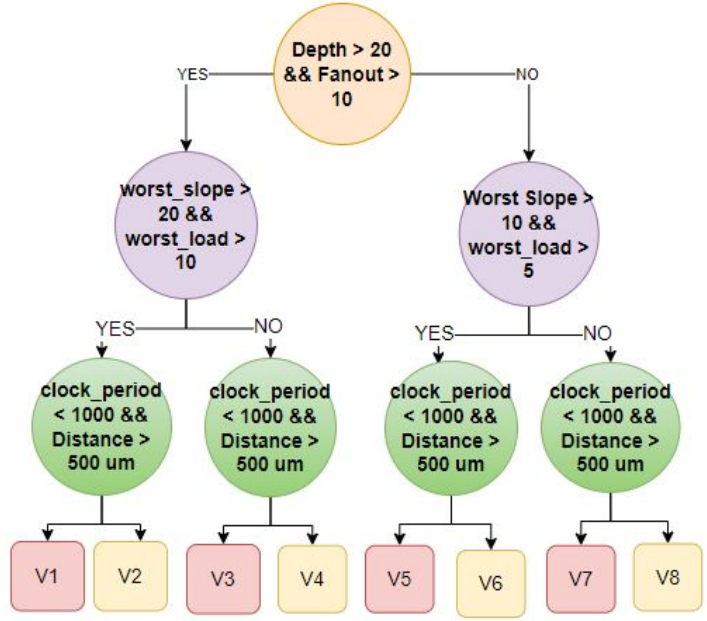
- An ML based predictive crosstalk fixing algorithm can identify and target bottleneck crosstalk issues with ballpark accuracy without intervention of timing engine.

- Supervised Learning is a branch of Machine learning where both the data components (feature set) and actual outputs are present.
- XGBoost is a supervised learning-based ML algorithm . It is based on decision trees with Dynamic boosting.
- Once trained XGBoost can predict the crosstalk with ~0.3 ps mean absolute error (MAE)

More info on XGBoost regressor :: https://www.geeksforgeeks.org/xgboost



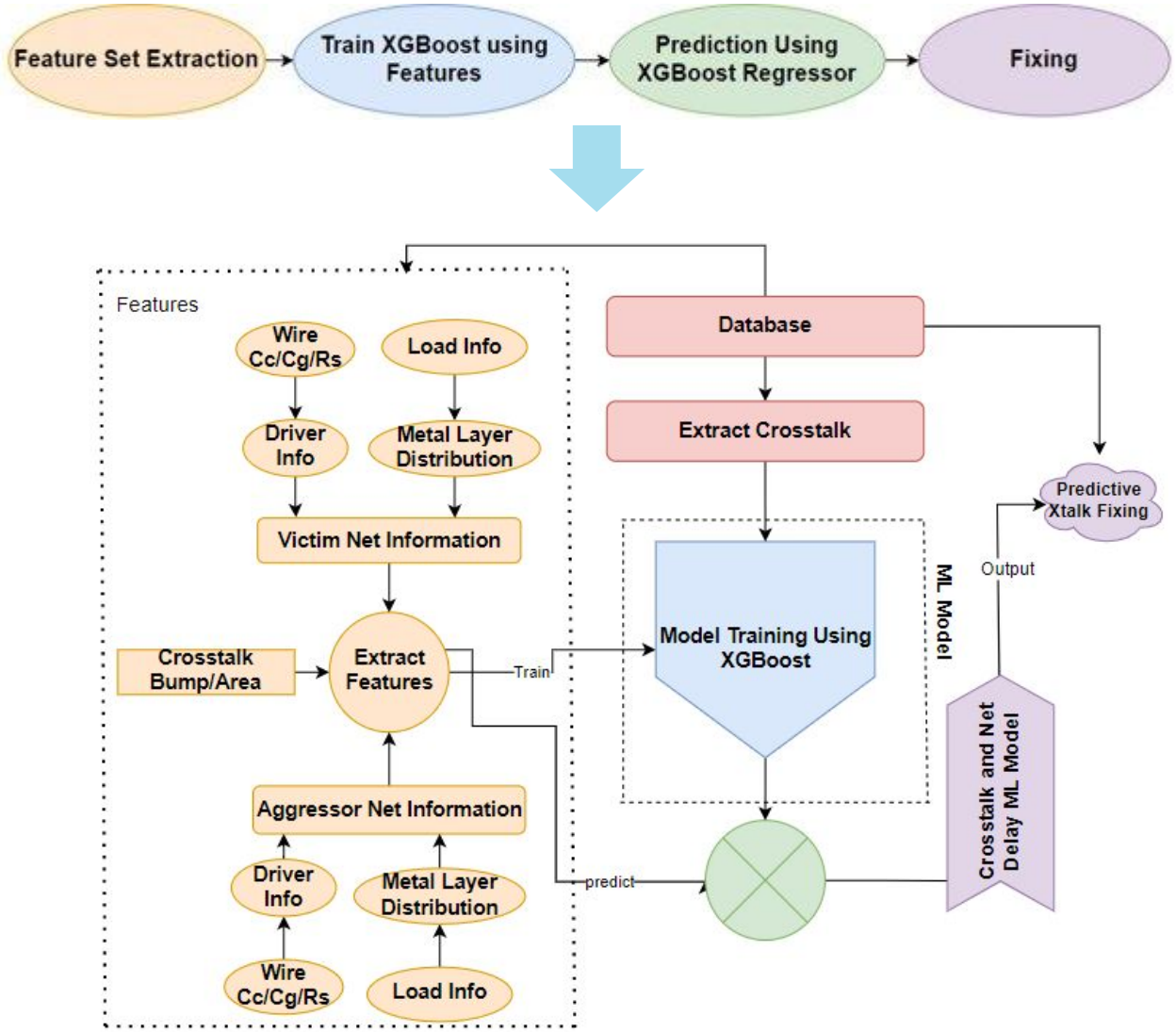
Feature importance in crosstalk prediction using XGBoost



Decision Trees Simplified

Crosstalk Prediction Using XGBoost Regressor

High level Data Flow



Crosstalk prediction Using XGBoost

Composite Aggressor

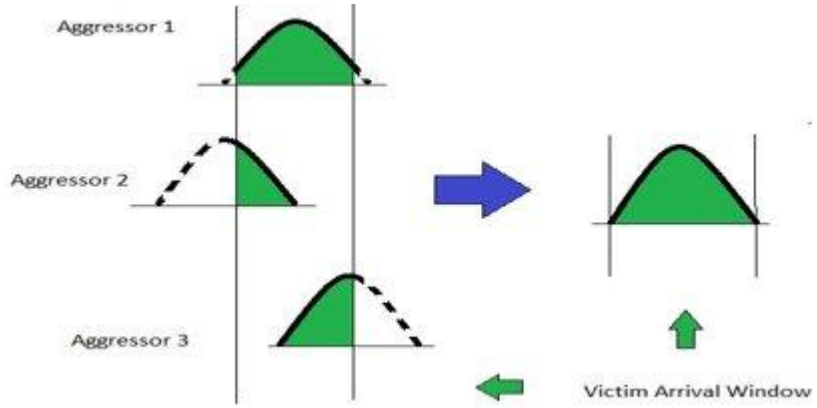
- Approximated equation of i^{th} aggressor is given by -

$$A_i = a_i t_i^2 + b_i t_i + C_i \; ; \; t_1 < t_i < t_2 \; ; \; i \in 1,2,3 \ldots N$$

where t_1, t_2 are the aggressor timing window bounds.

- Composite aggressor bump is calculated from victim timing window bounds t_{v1}, t_{v2} and composite aggressor area –

$$Composite\ Aggressor\ area = \sum_{i=1}^{N} \int_{t_{v1}}^{t_{v2}} A_i$$



Equivalent aggressor with composite effect of all the aggressors

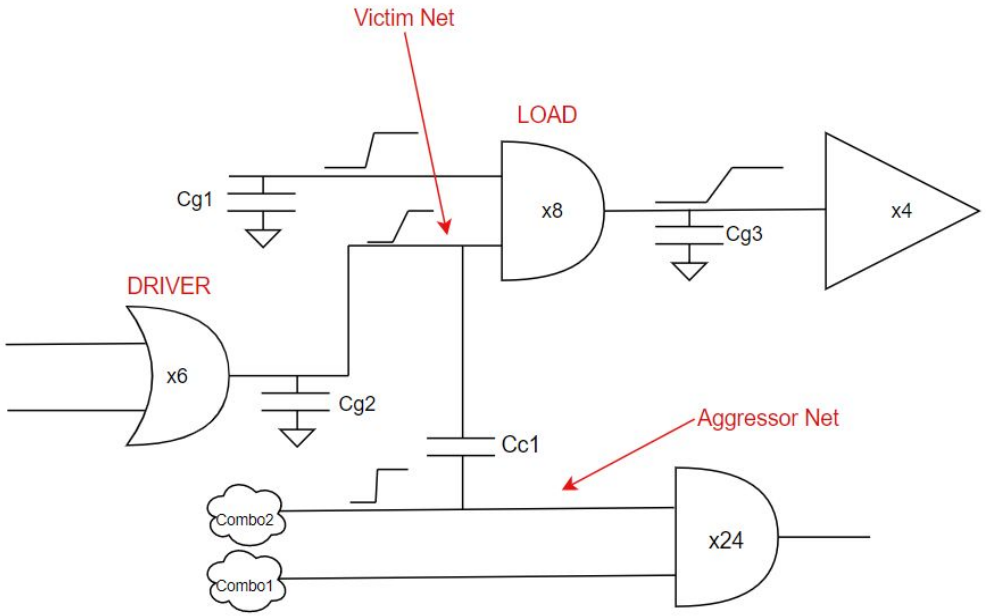
Crosstalk Prediction Using XGBoost Regressor

Primary and Derived Features

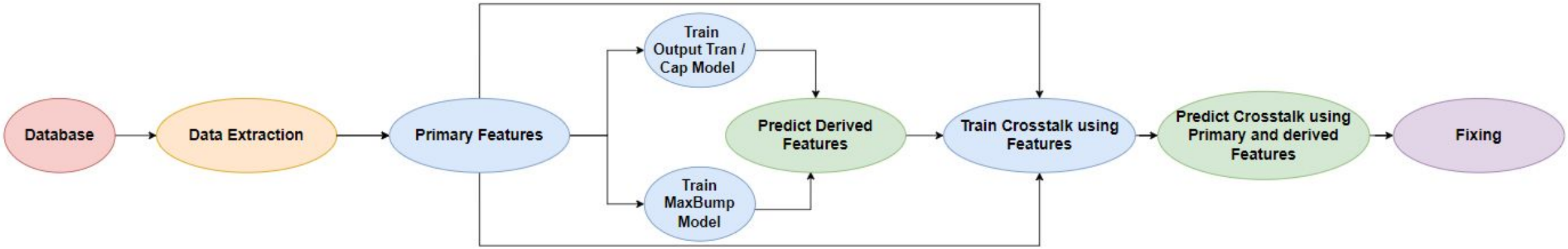
Features – Components that will help determine the crosstalk value.

- **Primary Features –** Independent features which are not impacted by timing engine updates. e.g., Driver input pin slew , net topology etc.
- **Derived Features –** Features that are dependent on timing engine update e.g., Driver output pin slew , max bump of an aggressor etc.

For best results we need a cascaded ML model which has an intermediate stage where all the derived features are also predicted using primary features.



Effect of changing input tran on output pins

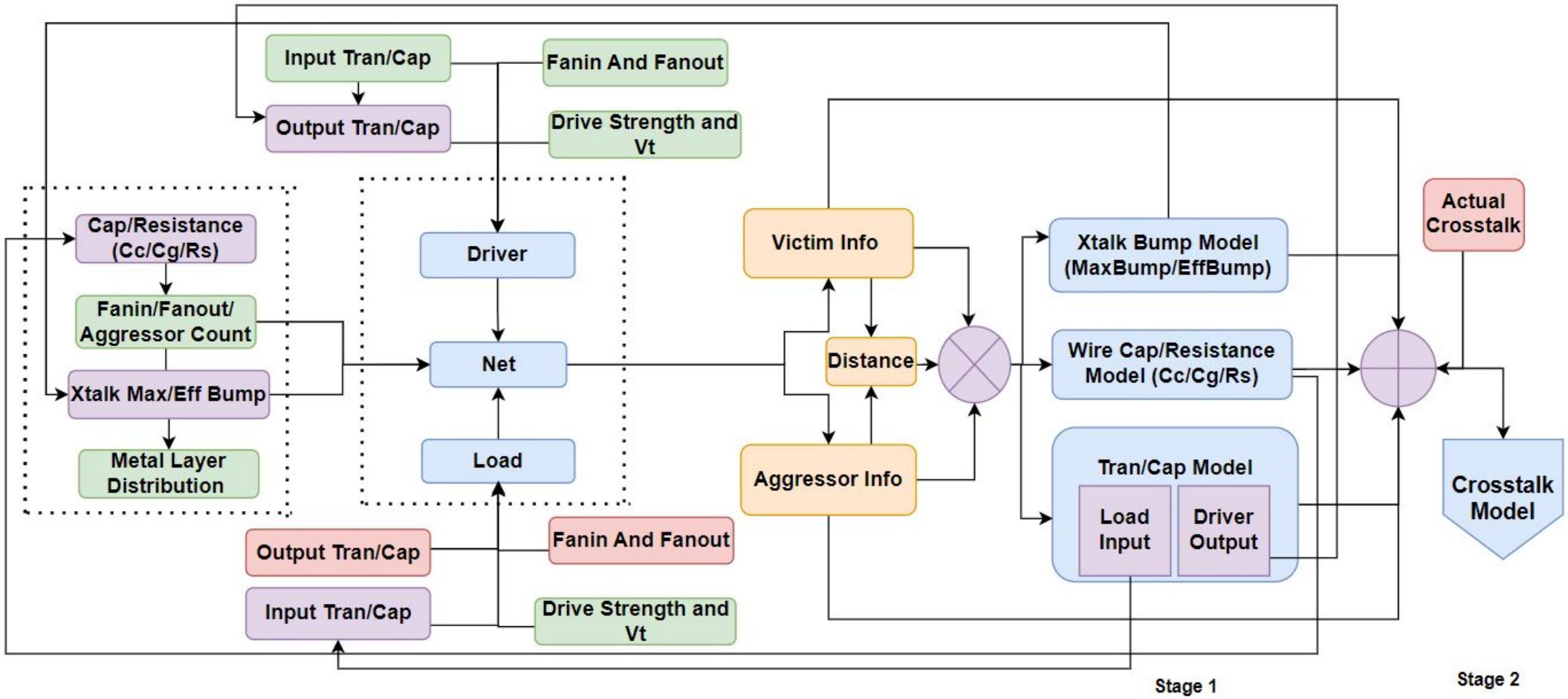


Dataflow for crosstalk fixing

Crosstalk Prediction Using XGBoost Regressor

Cascading of ML Models

- Stage 1 (intermediate stage) represents all the derived features that are needed to be modelled first in order to predict crosstalk in stage 2. For all prediction XGBoost Regressor is used after trying multiple Algorithms and comparing the Error.
- Diagram Shows only the Training Part of the Algorithm, Prediction part is covered in High level diagrams (refer previous slides)



Cascaded ML modelling to train crosstalk

Crosstalk Fixing 1/2

Steps

Crosstalk fixing iterations based on below priority chart

Priority	Step
P1	Victim upsize
P2	Aggressor downsize
P3	Break the net + insert buffer
P4	Metal layer upgrade
P5	Spacing the nets *
P6	Shielding the nets *

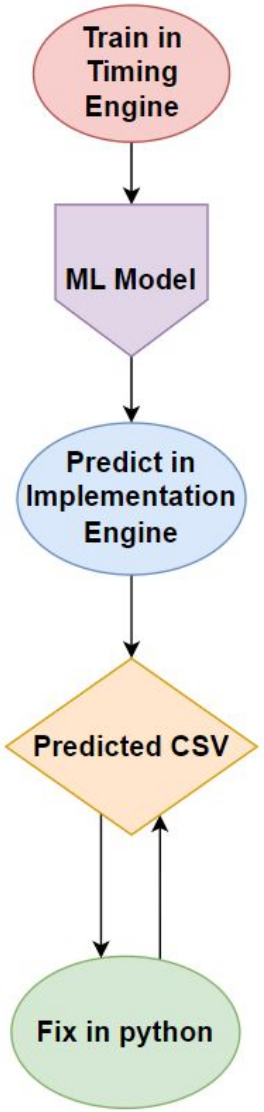
Priority of steps in Crosstalk fixing

* Future Updates

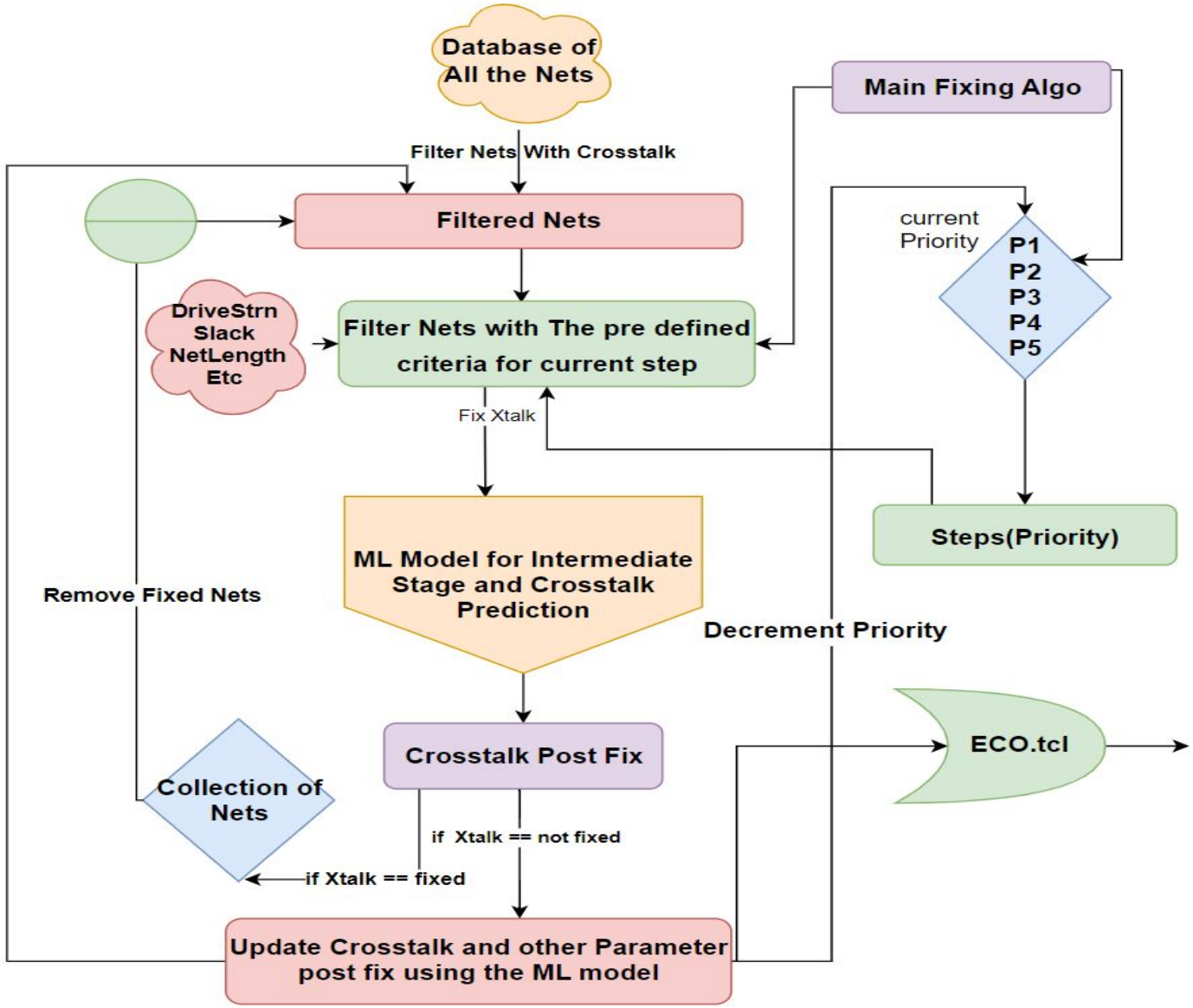
- Initial scrutinization will be performed on crosstalk impacted nets and will undergo some lower priority fix method which may benefit more. E.g., Buffering a long net to fix crosstalk instead of upsizing driver.
- Initial Filtering will be based on timing slack, net's segment length, availability of empty tracks in promotion layers, crosstalk magnitude and victim/aggressor driver strengths.
- Every fix iteration will undergo crosstalk prediction using ML model after modifying the input features.

Crosstalk Fixing 2/2

Flowchart

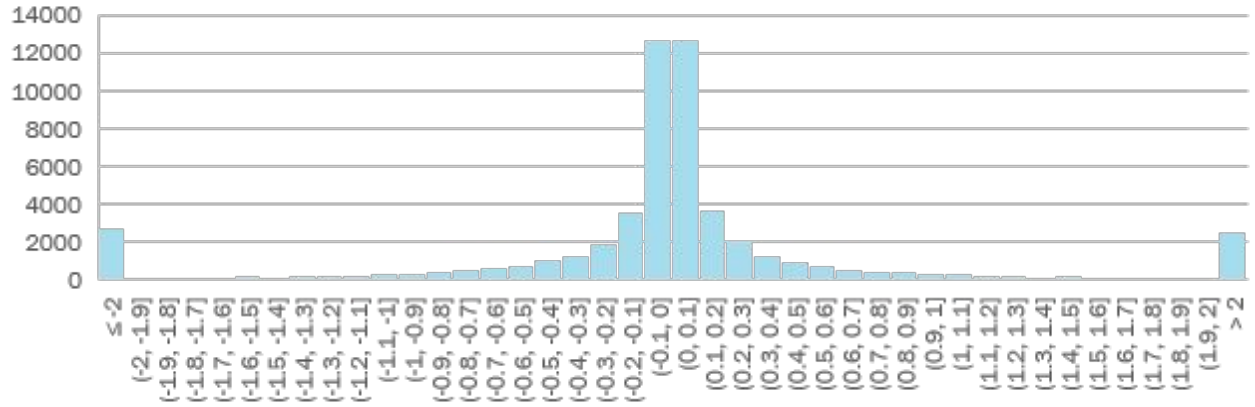


High Level flow for Crosstalk fixing



Crosstalk fixing using machine learning

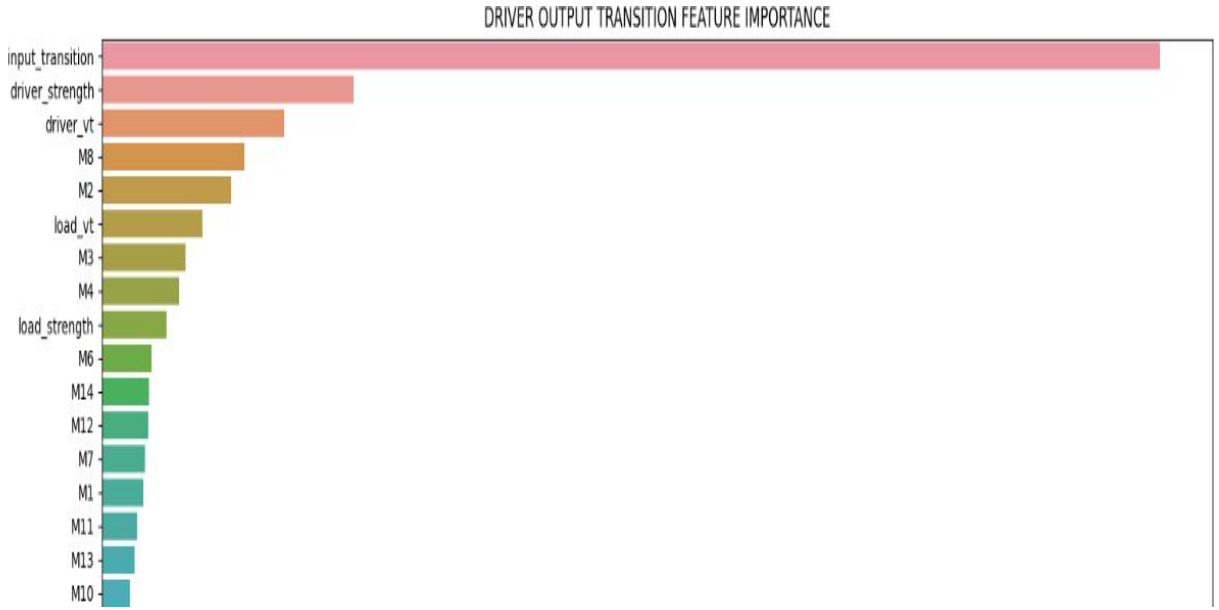
Evidence 1/2 :: intermediate stage



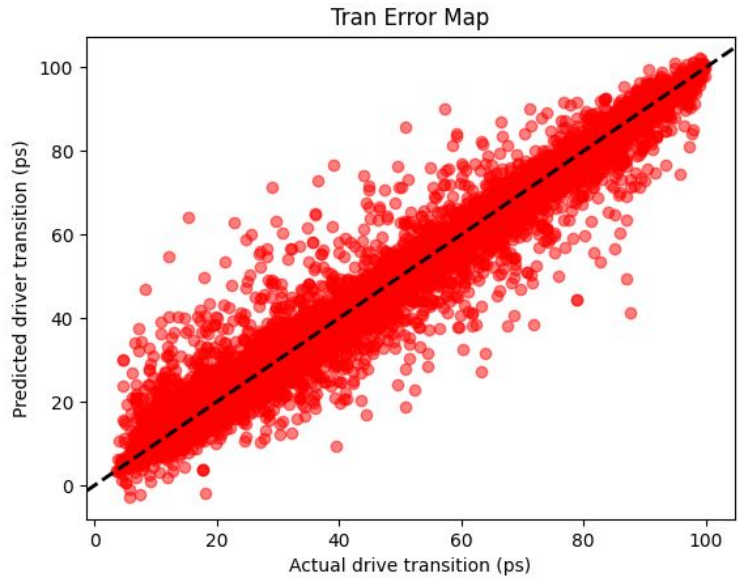
Delta driver output tran = actual tran – predicted tran

Train/Predict	MAE	MAE %
Driver Output Tran	0.3 ps	0.6 %
Load Input cap	0.12 fF	1.5 %
Load Input Tran	0.43 ps	0.9 %
Effective Bump	0.004 ps	2.3 %

Mean absolute error in various intermediate models



Driver output tran feature Importance



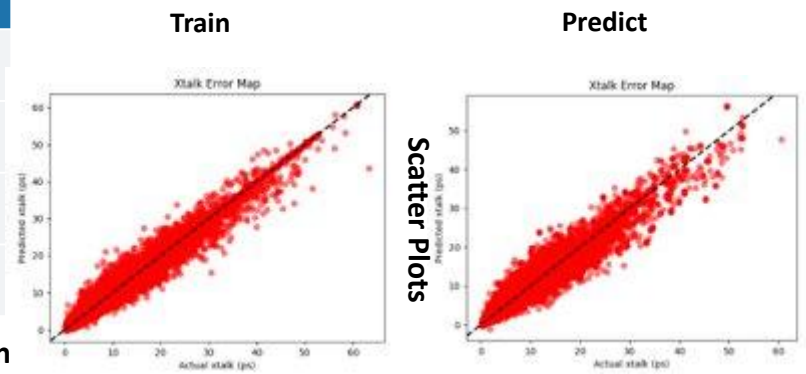
Driver output tran Scatter Plot

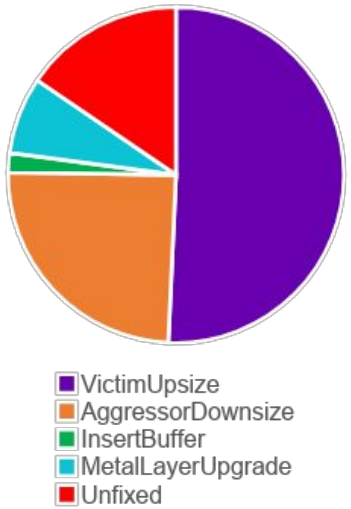
Evidence 2/2 :: crosstalk prediction and fixing

- ✔ For training/prediction dataset is taken from multiple SoCs ranging from 10K to 25K nets.
- ✔ Mean absolute error (**MAE**) in experiments conducted is within 0.2 ps (train) and 0.3 ps (predict)
- ✔ Algorithm used to fix the crosstalk (using ML model) shows ~5-20% improvement in timing.
- ✔ Predictive fix can bring down the runtime from 3-4 hours to ~10 minutes.

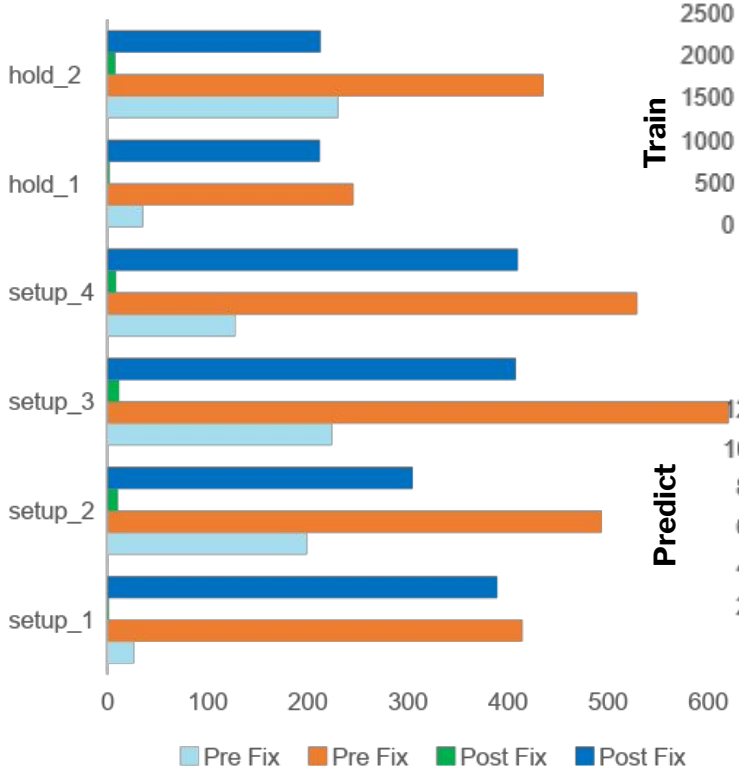
Train/Predict	MAE	MAE %
Train Dataset1	0.19 ps	6.85 %
Train Dataset2	0.14 ps	5.05 %
Predict dataset1	0.38 ps	7.17 %
Predict dataset2	0.27 ps	8.79 %

Table of Mean absolute error in crosstalk prediction

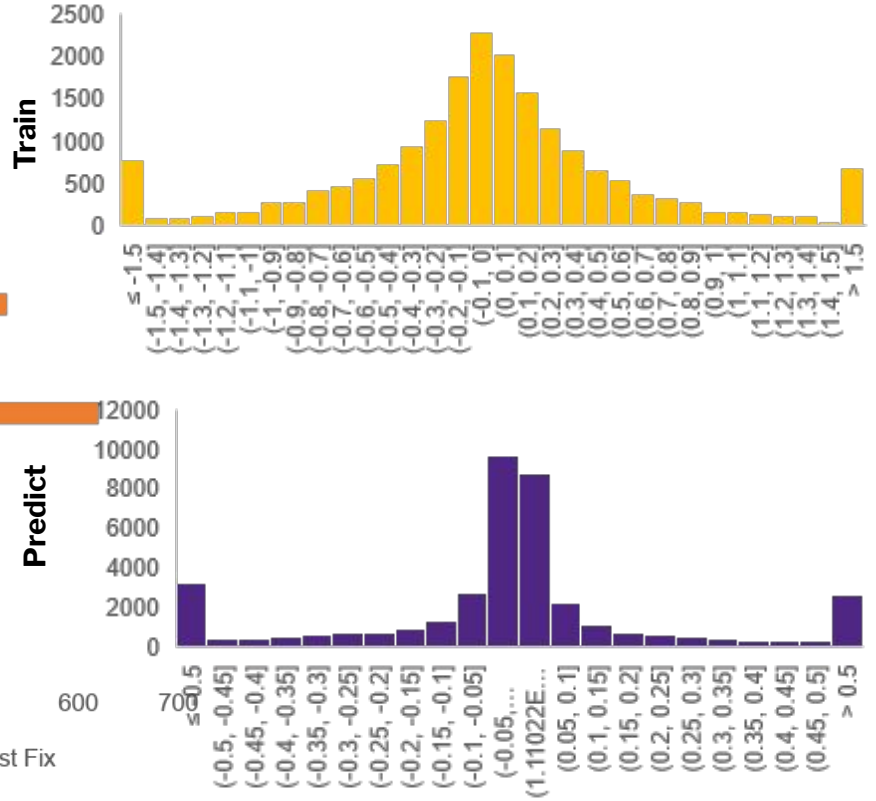




Distribution of steps taken to Fix crosstalk



Timing recovery through crosstalk fixing



Delta = Predicted Crosstalk – Actual Crosstalk

Summary and Future Plans

Key Finding / Benefits -

- Training and prediction error is less than 8 % which is under the acceptable limit .
- Reduces almost 5-20 % of timing slack (depending upon various corners) with zero manual effort .
- Reducing number of iterations in convergence of timing (both setup and hold) .

Future Plans -

- Addition of below steps in fixing algorithm .
  1. Spacing of nets.
  2. Shielding of nets.
- Reinforcement learning based modelling of “crosstalk fixing with auto selection of optimal steps”