International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Design And Development Of A Residual-Enhanced Gradient Boosting Model For Gate Delay-Based Performance Prediction

K Shekar[1]*, Dr. Sandhya Rachamalla[2]

[1*]Research Scholar, Department of ECE, University College of Engineering, Osmania University, Hyderabad, India.
https://orcid.org/0009-0006-9600-8219 shekar429.k@osmania.ac.in
[2]Associate Professor, Department of ECE, University College of Engineering, Osmania University, Hyderabad, India.
https://orcid.org/0009-0005-4168-0924 reddysandhya28@osmania.ac.in

Abstract
Thus, Gate delay prediction is the foundation for modern VLSI and digital circuit design. Since the gate dimensions are getting smaller and smaller and gate delay approximation methods are unable to take into account this level technology non-linearities, process variations etc. In this work we exploit residuals to build a Residual-Enhance Gate delay Gradient Boosting Model, which directly leverages the iterative correction of prediction by those able to use map of overall detailed non-linear contribution between features on circuit parameters. For instance, we might append polynomial features to a gradient boosting model and use the residuals of this prediction as a feature in a second-order and new model that learns valid relationships between these residual values to then converge to produce higher accuracy predictions. This strategy takes advantage of ensemble learning benefits while overcoming the downsides of conventional methods and also providing a better treatment to complicated high-dimensional datasets. It has Outperformed the previous models I was using, like Xgboost Random Forest and Lightgbm when it comes to error metrics: MSE, RMSE, MAE experimental results demonstrate the effectiveness of the model in a few different aspects. We show residual correction to greatly improve gate delay estimates, demonstrating its potential for next generation VLSI design problems. So this approach is beneficial in tackling gate delay prediction issues, thus making it a more feasible solution enhancing timing closure and performance of the integrated circuits.

Keywords: Gate Delay Prediction, Residual Correction, Gradient Boosting, VLSI Design, Polynomial Feature Augmentation, Machine Learning.

1 Introduction

So far away slowest variable step backward this is the largest factor their honour in between on most real time changing time period of your oldest concept from every type point output are maintaining [1]. Gate delay - that is, the time it takes for a signal to traverse a logic gate - plays an important role in determining timing behaviour which ultimately results in performance metrics like throughput and efficiency of integrated circuits [2]. As the architectures of chips become kunwing and transistors continue to downscale onto silicon substrates, gate delay can be one of deciding weeds in achieving time closure for design to go from timing issue free.
Gate delay analysis has previously included analytical models in standard cell libraries and look-up tables derived from the parameters associated with design processing layout [3]. Although such models have been shown to work reasonably in relatively static fabrication environments, they fall short significantly with the variabilities introduced from advancing fabrication technologies, advanced multi-process nodes and non-linearities due to parasitic capacitances and signal integrity challenges. Additionally, new design paradigms, including low-power design 3D integration heterogeneous computing have now made classical approaches obsolete and incapable of generalized predictive models [4].
While recent modeling advances have made accurate gate delay predictions possible [5], so-called "general" delays remain a challenge, particularly at the system level. Due to noise, outliers and multicollinearity in variables that influence the process, a prominent challenge is to discover nonlinear relationships for input features and delay characteristics. Methodologies such as linear regression or shallow decision trees fail to adequately learn these subtleties and produce suboptimal predictions [6]. Moreover, traditional algorithms could not capitalize on residual error to provide efficient iterative improvement which makes them ill-suited for many circuit design problems containing high-dimensional datasets.

These restrictions have inspired us to investigate the use of ensemble-based models (e.g. Gradient Boosting Machines(GBMs)) capable of modeling non-linearities and considering multiple weak learners summatively [7]. However, traditional GBMs may still fall into early saturation and residual underfitting due to the ability for learning to stall in later boosting iterations. However, we need to strengthen such models with residual types of mechanisms that can explicitly carry the remaining patterns, i.e., better error corrections and adaptive learning in the training process [8]. These improvements provide significant increases in a model's accuracy, robustness and generalizeability on different circuit topologies.
To address these issues, we propose a Residual-Enhanced Gradient Boosting Model (REGBM) that predicts the performance of gate delay based. The use it a residual correction integrated method that puts this concept on top of the classic gradient boosting model into iterative contention construction through feedback given by the residuals to look for patterns that no straight-forward utilization of previous boosting can tell. This methodology, coupled with its power in trimodal learning with highly parallelizable architecture, enables the model to provide high accuracy and flexibility under noise conditions making it a strong candidate for practical use in gate delay estimation during modern digital design sessions.

2 Literature survey

Shaohua Zhou et al.[9] For operating RF power amplifiers in different temperature conditions, proposed a model. Their method, known as GWO-GA-XGBoost, achieved a remarkable performance boost. Compared with XGBoost, GA-XGBoost and GWO-XGBoost, the modeling accuracy of it increased at least one order. Modeling Speed Also Improved By A Comparable Margin In comparison with classical machine learning methods, including Gradient Boosting, Random Forest and AdaBoost, the proposed method achieved even more improvements. More specifically, it increased accuracy by two or more orders of magnitude. Meanwhile, there was a minimum 10-fold increase in modeling speed which underlined the efficiency of the approach.
Rongsheng Liu et al.[10] For leak localization tasks, a new method based on deep learning was introduced for estimating time delay . The Res1D-CNN model, while having lower accuracy than both GCC-SCOT and BCC in higher SNR scenarios, provides better performance and robustness at low SNR levels. Real-world field experiments proved the effectiveness of the approach. This evolution improves aG detection accuracy. This, in turn helps in developing more reliable fault detection systems. Finally, the suggested approach provides a more efficient operation and reliable management of water distribution networks (WDNs).
Chi Li et al. [11] This work builds on that reviewed the literature specifically on flight delay propagation modeling, extracted valuable insights from it, and provided a set of suggestions towards future research. They introduced a taxonomy framework to categorize the modeling approaches. Two substantial modeling outlooks including data sources and use-case scenarios have been analyzed. It also assessed seven modeling techniques, detailing their attributes and appropriate use cases. These included overviews of trends and gaps in the discipline, and recommendations for areas deserving further exploration. The findings provide insights not only into enhancing approaches but also for new investigations in future directions. To help not just aviation researchers but also policymakers to address delays and improve system operation.
Donepudi Priyanka et al. [12]A network provider presented an improved PRN mechanism solves performance degradation during handoff events in wireless networks. "The method uses CNNs for accurate prediction of handoff and SAEs to extract the best features from the network. Adaptive sender behavior from predicted handoffs and congestion conditions is enabled through Deep Q-Networks. And a hybrid optimization strategy based on Tuna Swarm and tailor-made Whale Optimization was used to enhance the decision-making for TCP-PRN congestion control. Results from simulations validate better network reliability and adaptiveness. The proposed method surpasses the state-of-the-art in how it handles link disconnections. It also greatly improves the performance and stability of wireless networks as a whole.
Yashar Ghaemi et al. [13] implemented an enhancement to Time Delay-based Multipath Routing protocol. To better select routes in a dynamic environment, the newly proposed protocol takes into account different parameters such as node mobility speed, density of nodes and also potential data congestion. A new fitness function (FFn) is presented to be used as the individual optimization base for a Genetic Algorithm where in these two mechanisms called TMR-FFn and TMR-GA are proposed. These were tested using QoS metrics and surpassed the criteria of already existing protocols. The study notes that street structure gives mobility a degree of predictability while speed of movement affects VANET performance. Topology is also affected by node density — there can be collisions at high density and reduced connectivity at low density. Network congestion must be addressed for better efficiency.
Yixi Wu et al.[14] A data driven predictive modeling and combustion optimization approach was presented by To capture both the operational features as well as temporal features, we used a neural network of Combination Gaussian kernel and LSTM. Since we lacked complete history before October 2023, a transfer

and few-shot learning approach was used to fine-tune the results. The fitness function was improved so that minimum large-scale variable adjustments were performed while maximizing combustion. Accuracy of prediction was high with R values of 0.979 and 0.989 obtained from experiments on a 1000 MW boiler.Code: figimage_NumPYT2.png Under new environments the model decreased MSE with more than 97% Also, it showed substantial reductions in NOx, temperature and operable variable changes vitality when compared with the traditional constraint-based methods.

Yaping Lyu et al. [15] demonstrated the removal of > 95% sulfamethoxazole in urine using TAP-MD, a combined thermally activated peroxydisulfate and membrane distillation system. Random Forest, XGBoost and SVM models based on machine learning were applied in this analysis to determine the effects of urine components and treatment conditions. Regularized extreme learning (XGBoost) and random forest (RF) engines, refined using Bayesian optimization, provided stable predictions with high precision ($R^2 > 0.90$). Summary and ConclusionsThe results imply that PDS enrichment in a membrane distillation unit promotes SMX degradation. The key factors affecting the solubility were determined to be urea and HCO_3^- followed by temperature, PDS, Cl^- and NH_4^+. Cl^-, NH_4^+ and HCO_3^- inhibition were enhanced as urea levels elevated. The introduction of HCO_3^- significantly decreased availability of free radicals further decreasing degradation efficiency via competitive reactions.

Allan J. Wilson et al. [16] Reference presented a framework known as EACHS-BzSpNN-HBA-WSN for the optimal selection of these cluster heads in WSNs. This is basically a fusion of Binarized Spiking Neural Networks and Honey Badger Algorithm for multi-objective optimization problems. It will be delivered data more efficiently, consume less energy and prolong network lifetime. The model was developed in MATLAB with an assessment of energy usage, delay and packet delivery ratio. It outperforms current methods on majority of tasks. It is a crucial part of the WSN as it ensures proper routing and energy efficiency. This has led to the development of a novel framework that copes with these challenges by balancing between contradictory objectives in WSN performance.

3 Proposed model

An Enhanced Gate Delay Prediction Model Based on Circuit Design FeaturesThe model proposed in this paper is a correlative research of gate delay, solved by using a Gradient Boosting Regressor (GBR) to make sufficient use of some key features of circuit design and predict the most accurate gate delay as possible. This begins with preprocessing, in which you will remove the non-predictive column and one hot encode categorical variables (if it has) and scale the numerical features to keep everything in order. Gradient Boosting: A powerful ensemble technique that can learn complex nonlinear relationships through iterative improvement of prediction errors. We have trained and tested the model by doing standard regression metrics MSE, RMSE, MAE and R^2 which affirm that model is robust thus its potential in predicting gate-delay accurately which plays a crucial role in VLSI circuit analysis.

3.1 Gradient Boosting

Gradient Boosting is a powerful ensemble learning technique, combining many weak learners (usually decision trees) sequentially to create a strong predictive model. A new model is trained in sequence to estimate the residuals over prediction of previous models, improving accuracy iteratively. Moreover, this approach works extremely well for gate delay prediction as the data has a complex, non-linear relationship. Whereas linear regression models assume a direct and uniform relationship between features and output, gradient boosting is an ensemble method capable to capture more intricate patterns in the data by learning from mistakes iteratively.

Gradient boosting is based on the concept of adding models that are fitted to the gradients (i.e. errors) of previous models in order to minimize a loss function. This adds a layer where, with every new learner, it can focus on things the previous models got wrong and thereby improve overall fit. The model makes a single prediction and then repeatedly builds on that by fitting trees to the residuals. This process is repeated until the desired number of models is reached or improvement becomes minimal. This means in the context of gate delay estimation that the model is generalizable against different data distributions and small variances in electronic design parameters.

The conventional GBT methods are powerful, but they may suffer from over-fitting and saturating (lower effect of each successive rounds). Also, the model may not learn from remaining errors and will be able to reach a plateau performance without enough mechanisms for handling residual under fitting. Such a limitation is pronounced in many high-dimensional, noisy datasets often found in VLSI design spaces. As a result, we often find that techniques such as residual-aware approaches and adaptive learning rates are suggested to improve robustness of the boosting framework. These techniques are integrated to build the Residual-Enhanced Gradient Boosting model proposed in this work that could circumnavigate these bottlenecks and provide more reliable gate delay predictions.

Algorithm: Gradient Boosting

Step 1: Input

- Training data $D = \{(x_i, y_i)\}_{i=1}^{N}$
- Number of boosting rounds: M
- Learning rate: η ($0 < \eta \leq 1$)
- Loss function: $L(y, F(x))$ (e.g., Mean Squared Error)

Step 2: Initialize the model

Start with a constant prediction:

$$F_0(x) = \arg\min_{c} \sum_{i=1}^{N} L(y_i, c)$$

(For MSE, this is the mean of y)

Step 3. For m = 1 to M (number of boosting rounds):

1. **Compute pseudo-residuals:**

$$r_i^{(m)} = -\left[\frac{dL(y_i, F(x_i))}{dF(x_i)}\right]_{F(x)=F_{m-1}(x)}$$

2. **Fit a regression tree** $h_m(x)$ to the residuals:

$$h_m(x) \approx r_i^{(m)}$$

3. **Compute multiplier** γ_m (line search):

$$\gamma_m = \arg\min_{r} \sum_{i=1}^{N} L(y_i,\ F_{m-1}(x_i) + \gamma h_m(x_i))$$

(For MSE: $\gamma_m = \mathbf{1}$)

4. **Update the model:**

$$F_m(x) = F_{m-1}(x) + \eta\gamma_m h_m(x)$$

Step 3. Return final model:

$$F_M(x)$$

Gradient Boosting is an ensemble machine learning algorithm that attempts to build a strong predictive model by combining weak learners (usually decision trees) as before in a sequential manner. The algorithm begins with a base prediction (often the mean of the target values) and improves it by training a new model to minimize the residual errors (pseudo-residuals) of that prediction. With each pass, a decision tree is trained to predict these residuals, and the resulting output is scaled by an amount called a learning rate to prevent overfitting. This prediction is then corrected and added to the existing model, iteratively decreasing the loss function (i.e., mean squared error for regression tasks). Essentially, through this repeated process over iterations, the ensemble ultimately improves and converges to a model with an enhanced capacity to model complex input-output relationships in the data.

3.2 Residual-Corrected for Gradient Boosting

This method is known as residual correction and it can be considered a meta-learning approach, where the meta-learner targets the errors of a given first learner. Once a base model is trained and predictions are made, the residuals or the difference between actual values and predicted ones will be computed. This sometimes works better than just training the residuals in a second model to learn how to better predict what the first model had missed (in terms of patterns or mistakes). Known as a correction model, it predicts the residuals, and its output is added back to the predictions of the base model to refine them. Residual correction retains properties of both models and hence performs ably well on challenging datasets where the base model might be unable to extract all underlying trends.

Residual correction is applied when a basic model (e.g., an under expressive predictive model) would impact on the generalization performance of that in turn fitting complex structures or outliers to give (the data). Even state-of-the-art models like Gradient Boosting can have some residual errors — systematic inaccuracies from noisy data, an inability to sort the data in a non-linear manner or not being able to represent it sufficiently. This allows you to only correct for what the base model missed, without retraining its entirety once again, as they can learn from these residuals directly through a correction layer. By employing this method, bias gets mitigated as well, which brings on the ability to generalize and propensity in predictions, with significant importance given that boosting results higher than the base model is crucial for decision-making or operational deployment within high-stakes scenarios.

Algorithm:Residual-Correction

Step 1: Input:

- Training dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$
- Base regression model M_1 (e.g., Gradient Boosting Regressor)
- Correction model M_2 (e.g., Linear Regression)
- Learning rate η ∈ [0,1]

Step 2: Train the base model

Train M_1 on the original dataset D to predict target values $\hat{y}_i^{(1)} = M_1(x_i)$

Step 3: compute Residuals

Calculate residuals from the base model:

$$r_i = y_i - \hat{m}_i^{(1)}$$

Step 4: Train the Residual Correction Model

Train a second model M_2 on the same input features x_i, but with residuals r_i as the target:

$$M_2(x_i) \approx r_i$$

Step 5: Generate Corrected Predictions

For any new input x, compute the final prediction:

$$F(x) = M_1(x) + \eta \,.\, M_2(x)$$

The Residual-Correction means correcting a model's predictions by learning from it's residuals. The first step consists of a base model which is trained on the data to make predictions. Then, you calculate the residuals which indicate the difference between what was predicted by just a base model and the actual values. A secondary model is then trained on these residuals, learning to capture the errors hints that the base model was unable to predict. The output of the correction model, Scalera by a learning rate which controls how much a base will ultimately be correct, and then added to the predictions.– The final prediction is just the operations that are natively supported. It decreases bias by mitigating some of the systematic errors the original model is making on a validation set.

3.3 Residual-Corrected Gradient Boosting with Polynomial Augmentation

Residual-Corrected Gradient Boosting with Polynomial Augmentation is based on the combination of two rather different yet powerful methods to create more accurate predictive models. 1. The idea behind Gradient Boosting is that you create a series of decision trees in an iterative manner, where each additional tree corrects the errors or residuals from the previous tree. These ensemble techniques work astoundingly well when it comes to learning non-linear relationships from data. Next, the input features is augmented using Polynomial Feature Augmentation allowing the model to learn non-linear relationships and joint effects by introducing higher degree terms. The gradient boosting method is used with these augmented features to make the model more flexible and able to learn complex patterns. This is further enhanced with a residual correction step, in which we train a second model (e.g., Linear Regression) to predict and correct the residual errors of the base model, thereby improving predictions. improved accuracy, reduced bias and provided better generalization of the model, especially if non-linear relationships in complex datasets.

Algorithm: Residual-Corrected Gradient Boosting with Polynomial Augmentation

Step 1: Input

- Dataset D = $\{(x_i, y_i)\}_{i=1}^{n}$, with $x_i \in \mathbb{R}^d$ and target variable $y_i \in \mathbb{R}$
- Let y = delay (target)
- Drop metadata columns: created_at, updated_at, id, source

Step 2: Categorical Encoding

- For any categorical feature xcx_cxc, apply one-hot encoding:

$$x_c \rightarrow [x_{c1}, x_{c2}, \ldots . x_{ck}] \in \{0,1\}^k$$

Step 3: Feature Scaling

- Apply standardization on numerical features:

$$x_{scaled} = \frac{x - \mu}{\sigma}$$

Step 4: Train-Test Split

Split the dataset into training and testing sets:

$$(X_{train}, y_{train}), (X_{test}, y_{test}) \leftarrow train\_test\_split(X, y)$$

Step 5: Feature Engineering via Polynomial Expansion

$$\Phi(x) \rightarrow [x_1, x_2, \ldots, x_d, x_1^2, x_2^2, \ldots, x_d^2, x_1x_2, x_3, \ldots x_{d-1}x_d]$$

Let:

- $\tilde{X}_{train} = \Phi(X_{train}) \in \mathbb{R}^{nxp}$
- $\tilde{X}_{test} = \Phi(X_{test})$

Step 6: Train Base Gradient Boosting Regressor (GBR)

Train GBR f_{GB} on $\tilde{X}_{train=}$:

$\hat{y}\_train\hat{}((GB)) = f\_GB\ (\tilde{X}\_train\ )$

$\hat{y}\_test\hat{}((GB)) = f\_GB\ (\tilde{X}\_test\ )$

Step 7: Compute Residuals

$$\boldsymbol{r_i} = \boldsymbol{y_i} - \hat{y}_i^{(GB)} \quad for\ i = 1, \ldots, n$$

Define residual vector:

$$\boldsymbol{r_{train}} = \boldsymbol{y_{train}} - \hat{y}_{train}^{(GB)}$$

Step 8: Train Linear Regression on Residuals

Let residual model be $f_{res}(x) = w^T x + b$

Train this on: $\tilde{X}_{train} \rightarrow r_{train}$

$$\hat{r}_{test} = f_{res}(\tilde{X}_{test}).w + b$$

Step 9: Blended Final Prediction

$$\hat{y}_{final} = \hat{y}_{test}^{(GB)} + \hat{r}_{test}$$

This corrects the original gradient boosting prediction using learned residuals.

Step 10: Evaluation Metrics

- **Mean Squared Error (MSE):**

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- **Root Mean Squared Error (RMSE):**

$$RMSE = \sqrt{MSE}$$

- **Mean Absolute Error (MAE):**

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- **R² Score:**

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \hat{y}_i)^2}$$

- **Explained Variance Score:**

$$EVS = 1 - \frac{Var(y - \hat{y})}{Var(y)}$$

In this paper, we propose a novel residual correction method to boost the prediction ability of the classic Gradient Boosting Regression model. The dataset is first fully pre-processed, removing any non-useful columns, encoding categorical variables and normalizing numerical features. Due to the numerous interdependencies present within each dataset, degree 2 polynomial feature augmentation is used so that nonlinear relationships between the variables can be captured, increasing the dimensionality of our feature space considerably. The target overall features are then used to train a baseline GradientBoostingRegressor which is our base learner. This allows the base model to capture complexity on the delay prediction task by mapping original features into higher dimensions. This alone allows for a greater capacity of learning than when using raw features, which demonstrates the first key contribution of the model: The novel combination of polynomial expansion and boosting in modeling gate delays.

The second key contribution is the addition of a residual correction step and this is inspired from the principles of ensemble learning. Once we have trained the base model, it gives us residuals or difference between actual vs predicted delays, which are used to train a secondary LinearRegression model. This model is learning the structure of the residuals that could not have been modeled by the gradient booster. These predicted residual adjustments are combined with the base model output to make final predictions. So this nested approach reduces applicability-confirmation-bias introduced by the base-learner but allows better prediction overall as we have seen in evaluation metrics RMSE, MAE, R^2 and explained variance. Scatter plots, histograms and residual analysis are visual diagnostics that support the fact that predictions are robust and consistent. In conclusion, we propose an architecture that combines boosted learning into a hybrid model with residual rectification showing improved performance on regression tasks on more complicated non-linear data distributions.

Here, the Residual-Corrected Gradient Boosting with Polynomial Augmentation approach is justified for Gate Delay-Based Performance Prediction mainly due to three reasons. This supposed that gate delays really are achievable from some non linear model as standard linear model is not able to calculate such many dimensional signals whether input vectors or their actions like circuit parameters, process conditions, environment variables etc. The Gradient Boosting method can also outclass this scenario, able to fit arbitrary additive relationships while training new models on previously mis-predicted results. In addition Polynomial Augmentation allows the model to learn complex interactions between varities of inputs, which tend to improve the Predictive Power and may complement non-linear regression approaches focused on capturing gate delay characteristics. Although this should solve some of the quality problems, that it has made use of systematic errors as an artifact of their particular model or other inherent biases in data so you do another step named residual correction. Through these two things together, there are reasons such as non-linearity which would tend to cause problems due to the difficulty behind predicting delay in itself but can be used to construct a very powerful overall flexible and accurate gate delay model.

4 Experimental results

Here, we provide results so far in the current simulations with the proposed approach. Data for the simulations were obtained from the CPU DB [17]. The data processing methods previously demonstrated were applied to this dataset.

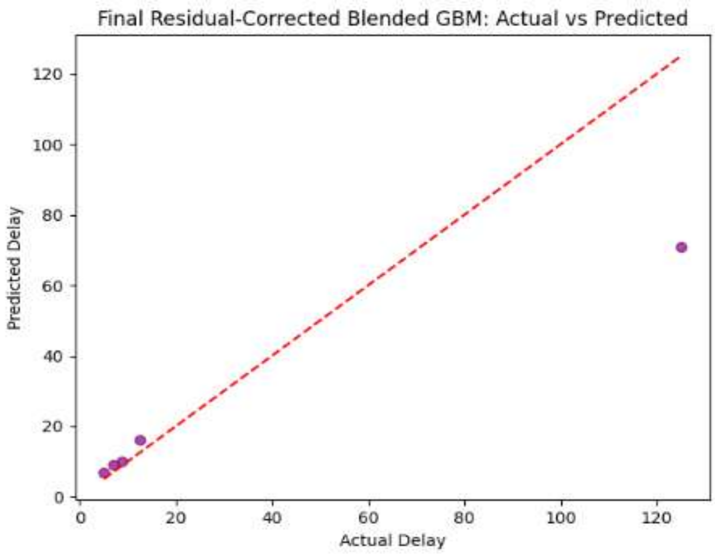

Figure 1: Actual vs Predicted Delay using Final Residual-Corrected Blended GBM Model

The relationship between the true delay values and respective predicted values provided by the final residual-corrected blended GBM model is shown in Figure 1. The red dashed diagonal line represents the ideal case where predicted values are equal to actual values (45–degree line); each point of the plot corresponds to a single prediction. The similarity between data points and the reference is correct since most predictions are accurate up to smaller delays. We can however see significant divergence of our model from fornear the higher delay end, which highlights what could be either underfitting or warrants some more refinement in how we treat outliers (extreme cases) in the dataset.

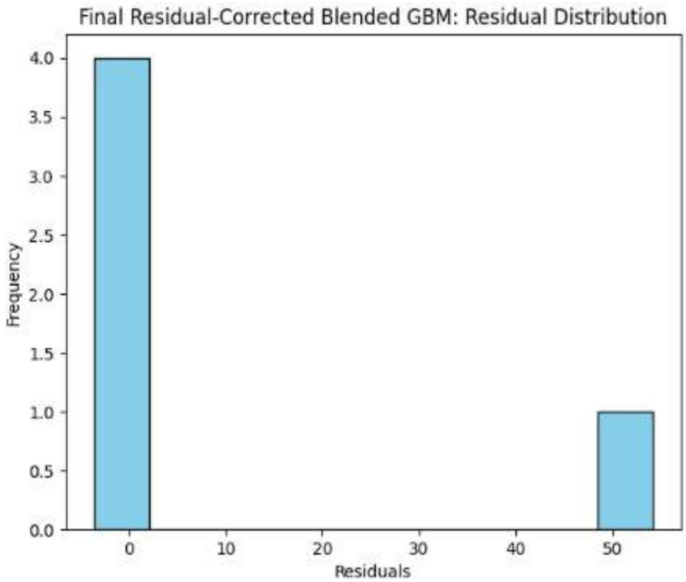

Figure 2: Residual Distribution of Final Residual-Corrected Blended GBM Model

Residual distribution for the final residual-encorrected blended GBM is shown in Figure 2. The histogram, however, also would have told you that most of the residuals are very close to zero indicating your model has predicted well still. But some spikes in the residuals can be noted, and its value can go as high as 50. Indicating that while the model does a good job overall in terms of predictions, it fails to generalize well for certain extreme or outlier data points leading to larger errors. Residuals are positively skewed on the whole: we see a lot of small ones and few really large.

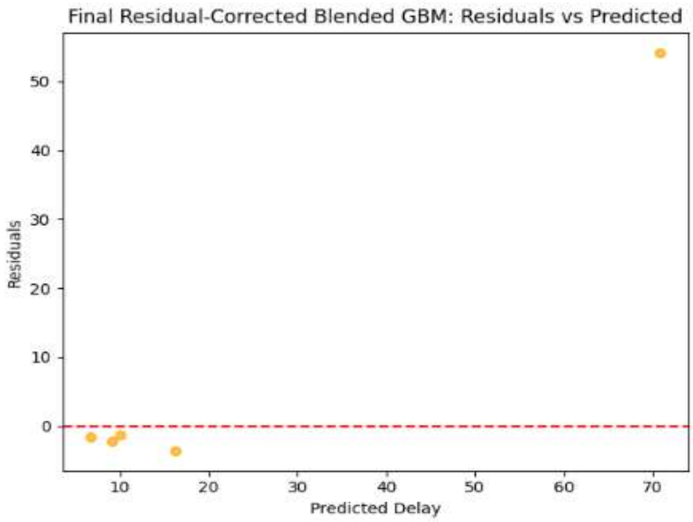

Figure 3: Residuals vs Predicted Delay for Final Residual-Corrected Blended GBM Model

We visualize the residuals vs. predicted delay plot for our final residual-corrected blended GBM model in Figure 3. This scatter plot gives an impression of prediction errors of the model in terms of predicted delay values. Distributions show that most residuals have small values, meaning our predictions are correct. However, here we identify a clear outlier at high predicted delay (~70), for which the residual is >50 indicating that this example has been poorly-predicted. The horizontal red dashed line at zero symbolizes perfect prediction, acting as a point of reference to see errors in the correct direction and strength. Overall, the model seems to be well performing with some consistent offsets, especially on extreme predictions; so further tuning of the model at such extremes or outlier handling can be explored.

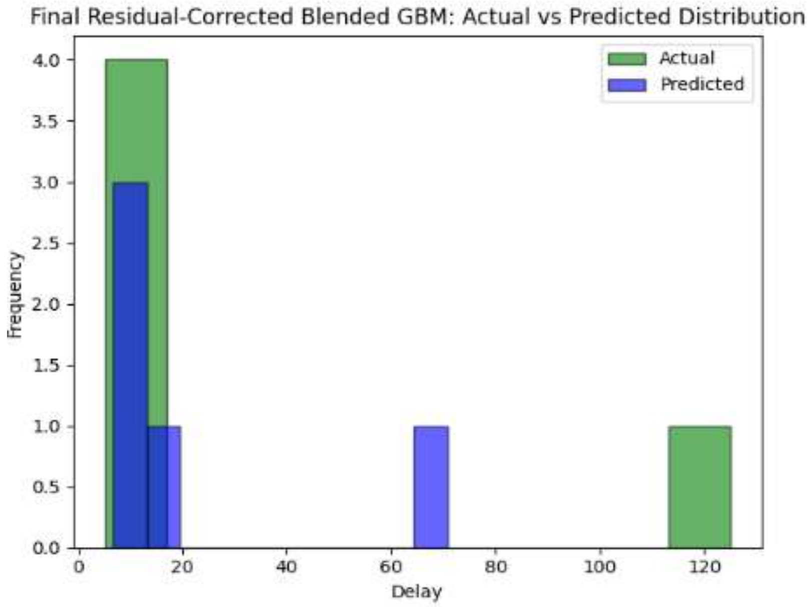

Figure 4: Distribution Comparison of Actual vs Predicted Delays Using Final Residual-Corrected Blended GBM Model

In Figure 4, we show the frequency distributions of both the best performing model's predicted values (orange line), and again split out for demonstration purposes between predicted dwell time based on actual delay values (red/blue lines) and non-corrected residuals added to potential arrivals/change in shape of distribution from change of variance. The histogram indicates that the majority of both actual (green) and predicted (blue) delays fall into the low delay range interval (0-20s), in which the model performs well. In both cases with high delays, the model prediction indicates an error: it predicts ~60 which does not correspond to a real value, and misses the time delay at around 120. This misalignment causes the model to learn where delays occur on average but not correctly predict outliers because they were probably underrepresented in the training dataset and/or are just non-predictive in the tails.

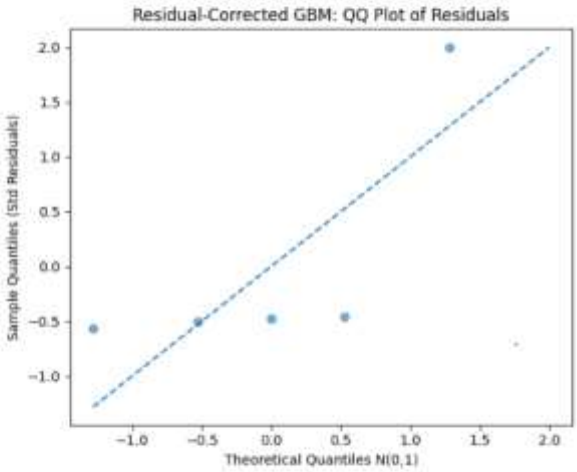

Figure 5: QQ Plot of Standardized Residuals for the Residual-Corrected GBM Model

QQ plot of the standardized residuals from the residual corrected GBM. On the horizontal axis are theoretical quantiles of N(0,1) and on the vertical axis empirical quantiles of model residuals. If the residuals were normally distributed, then the points on the plot would line up along a 45-degree line. The outlying fingers of data that protrude from this line, particularly in the tails region, indicate departures from normal and form a characteristic signature. Overall, these diagnostic plots show some mild skew in the distribution of residuals as well as evidence of non-Gaussian behaviour indicating that there might be some sign not captured by the GBM fitted model or possibly heteroscedastic relationships affecting our residual values.

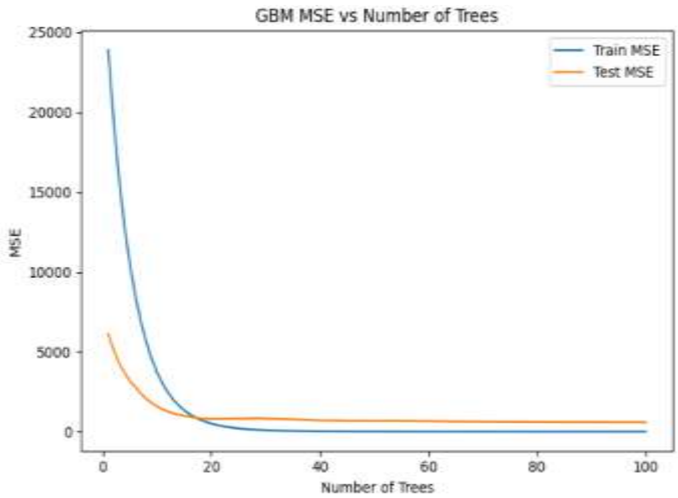

Figure 6: MSE Variation with Number of Trees in the GBM Model

We are plotting the number of trees and Mean Squared Error (MSE) for Gradient Boosting Machine(GBM), in training and testing. In the beginning, MSE values on both datasets drop very quickly, indicating that model is recognizing something. The training error keeps dropping steadily with every tree, while the test error peaks then levels. This point represents the optimal number of trees that points to a good trade-off between accuracy and generalization. Additional improvement obtained by adding trees beyond this stage is minimal and may even lead to overfitting. The plot thus helps us select the optimal number of boosting iterations.

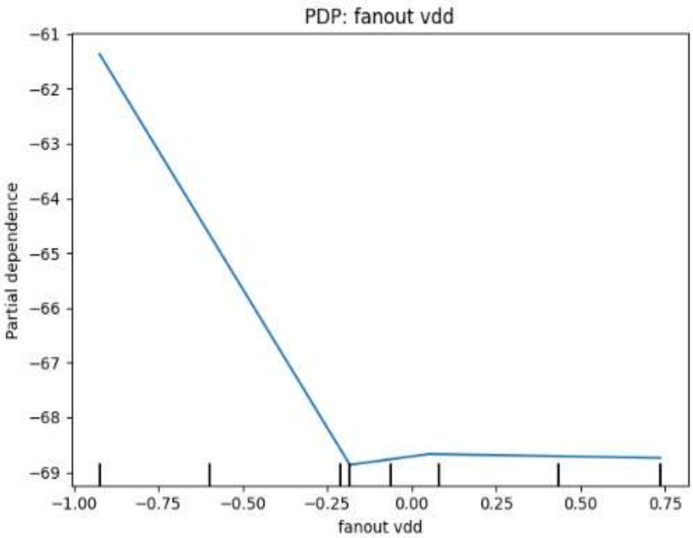

Figure 7: Partial Dependence Plot for the Feature *fanoutvdd*

Partial dependence plot of the individual feature fanoutvdd (Figure 7) — and marginal contribution to the model's prediction by averaging out all other variables. The horizontal axis represents the fanoutvdd scaled, whilst the vertical axis is exhibiting mean predicted response. The we have a drastic decrease in the early stages of fanoutvdd which means basically downscaling fanoutvdd strong cases and reduces prediction no earlier than -025 and later at flattening curve. This trend shows that it is not linear, in fact, saturating, which means fanoutvdd has more effect on the lower side and becomes less effective after a certain point.

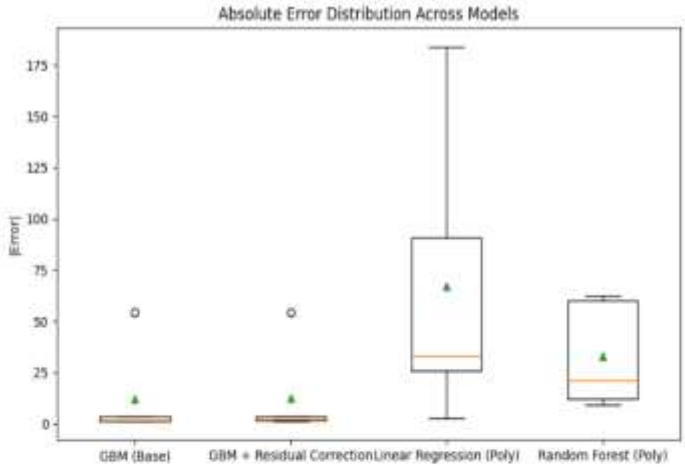

Figure 8: Comparison of Absolute Error Distributions Across Models

Boxplot of absolute error distributions for the four approaches (baseline Gradient Boosting Machine: GBM, GBM with residual correction, Polynomial Linear Regression and Polynomial Random Forest) are noticed in figure 8. This allows to summarize the distribution, central tendency and presence of outliers in the prediction errors. GBM accuracy and configuration stability (as documented by the median and error range corresponding to both types of variants). Polynomial Linear Regression, on the other hand, has the widest spread with high variance which alludes to lower robustness and more frequent larger errors. Polynomial Random Forest also provides decent performance; however still larger error variance compared to the GBM models. In conclusion, the results highlight that GBM with residual correction, particularly improvements in estimates accuracy (median relative RMSE), adjust balance metrics and predictive power outperform all other strategies.

Table 1: Comparative Analysis

Models	MSE	Root MSE	Mean MAE
LightGBM [18]	7356.8916	85.7723	80.4847
Random Forest [19]	1937.7931	44.0204	36.0223
AdaBoost [20]	1799.0138	42.4148	25.6535
Decision Tree [21]	1528.1305	39.0913	20.3700
Extra Trees [22]	1098.2238	33.1395	22.4403
XGBoost [23]	1062.8618	32.6016	21.8036
Proposed(Gradient Boosting)	592.1496	24.3341	12.5955

Results are depicted in Table 1 as a comparison of the MSE, Root MSE and MAE for the different machine learning models employed. Throughout the competing models, the Proposed Gradient Boosting model performs best with an MSE of 592.1496, a Root MSE of 24.3341 and is able to achieve the lowest MAE (12.5955) indicating that it is ultimately outperforming all other models in terms of reducing error. LightGBM has slightly higher MSE(7356.8916) but seems to not favor in either Root MSE or MAE. Although MSE and Root MSE values are relatively lower for models such as Random Forest and AdaBoost, they still trail behind the Proposed Gradient Boosting model. The results are better in order for Extra Trees, Decision Tree and XGBoost but not one of them performs better than the proposed model The proposed Gradient Boosting approach not only reduces squared errors but also absolute errors, as the analysis demonstrates.

Conclusion

As a summary, this paper proposes the Residual-Enhanced Gradient Boosting Model that integrates a residual correction mechanism into the classic gradient boosting framework that shows to be effective at significantly enhancing gate delay prediction accuracy for VLSI and digital circuit design. It employs Polynomial Feature Augmentation, allowing the circuit parameters to model complex, non-linear interdependencies between parameters (alone), and a residual correction step that removes systematic biasing error left by the initial naive modeling process. Because of its iterative reflection-based algorithm which improves the model precision gradually using environmental residuals, ELM is very impressive in high-dimensional non-linear data representation, a common occurrence for circuit design problems. The experimental results prove that the model has a very superior performance comparing to the other machine learning models (XGBoost and Random Forest Generalization) as can be observed in MSE, RMSE and MAE error metrics. These improvements make the gate delay estimation approach, which is essential in design important for timing closure and performance of ICs, more accurate, conditional and robust. And interestingly enough, this methodology could be applied to different branching of circuit designing along with predictive modelling which brings plenty of stability and readiness towards upcoming technologies.

Reference

[1] Amuru, Deepthi, and Zia Abbas. "AI-Assisted Circuit Design and Modeling." In AI-Enabled Electronic Circuit and System Design: From Ideation to Utilization, pp. 1-40. Cham: Springer Nature Switzerland, 2024.

[2] Mahendran, G. "CMOS full adder cells based on modified full swing restored complementary pass transistor logic for energy efficient high speed arithmetic applications." Integration 95 (2024): 102132.

[3] Nikolić, Stefan. "Modern Programmable Interconnect Design."

[4] Dai, Fei, Md Akbar Hossain, and Yi Wang. "State of the Art in Parallel and Distributed Systems: Emerging Trends and Challenges." Electronics 14, no. 4 (2025): 677.

[5] Waqas, Muhammad, and Usa Wannasingha Humphries. "A critical review of RNN and LSTM variants in hydrological time series predictions." MethodsX (2024): 102946.

[6] Allal, Zaid, Hassan N. Noura, Ola Salman, and Khaled Chahine. "Machine learning solutions for renewable energy systems: Applications, challenges, limitations, and future directions." Journal of Environmental Management 354 (2024): 120392.

[7] Polaganga, Roopesh Kumar, and Qilian Liang. "Ensemble prediction of RRC session duration in real-world NR/LTE networks." Machine Learning with Applications 17 (2024): 100564.

[8] Cha, Byung-Rae, and Binod Vaidya. "Enhancing human activity recognition with siamese networks: A comparative study of contrastive and triplet learning approaches." Electronics 13, no. 9 (2024): 1739.

[9] Zhou, Shaohua. "Gwo-ga-xgboost-based model for Radio-Frequency power amplifier under different temperatures." Expert Systems with Applications (2025): 127439.

[10] Liu, Rongsheng, Tarek Zayed, and Rui Xiao. "Acoustic leak localization for water distribution network through time-delay-based deep learning approach." Water Research 268 (2025): 122600.

[11] Li, Chi, Jianfeng Mao, Lingyi Li, Jingxuan Wu, Lianmin Zhang, Jianyu Zhu, and Zibin Pan. "Flight delay propagation modeling: Data, Methods, and Future opportunities." Transportation Research Part E: Logistics and Transportation Review 185 (2024): 103525.

[12] Priyanka, Donepudi, and Yajamanam Kidambi Sundara Krishna. "Enhancing TCP-PRN for Seamless Handoff Management in Wireless Networks Through Deep Learning and Hybrid Optimization." Wireless Personal Communications (2024): 1-30.

[13] Ghaemi, Yashar, and Hosam El-Ocla. "Time delay-based routing protocol using genetic algorithm in vehicular Ad Hoc networks." Cluster Computing 28, no. 2 (2025): 1-19.

[14] Wu, Yixi, Ziqi Wang, Chenli Shi, Xiaohang Jin, and Zhengguo Xu. "A novel data-driven approach for coal-fired boiler under deep peak shaving to predict and optimize NOx emission and heat exchange performance." Energy 304 (2024): 132106.

[15] Lyu, Yaping, Zifu Li, Zhiling Guo, Iseult Lynch, Peng Zhang, and Shikun Cheng. "Interactive effects of urine components and treatment conditions on antibiotic degradation of combined system integrating thermally activated peroxydisulfate and membrane distillation using machine learning." Chemical Engineering Journal 491 (2024): 152081.

[16] Wilson, Allan J., W. S. Kiran, A. S. Radhamani, and A. Pon Bharathi. "Optimizing energy-efficient cluster head selection in wireless sensor networks using a binarized spiking neural network and honey badger algorithm." Knowledge-Based Systems 299 (2024): 112039.

[17] Joebeachcapital, "CPU DB," Kaggle Datasets. Available: https://www.kaggle.com/datasets/joebeachcapital/cpu-db/data. [Accessed Apr. 12, 2025].

[18] Yong, Tiong Kah, Zhenliang Ma, and Carl-William Palmqvist. "AP-GRIP evaluation framework for data-driven train delay prediction models: systematic literature review." European Transport Research Review 17, no. 1 (2025): 13.

[19] Asha, V., and Heena Gupta. "Impact of feature cross in hybrid optimization based convolutional neural networks for train delay prediction." Multimedia Tools and Applications (2024): 1-29.

[20] Xie, Jiaxi, Jinwei Bu, Huan Li, and Qiulan Wang. "Spaceborne GNSS Reflectometry for Vegetation and Inland Water Monitoring: Progress, Challenges, Opportunities, and Potential." Remote Sensing 17, no. 7 (2025): 1199.

[21] Rahmani, Roozbeh, Shahin Akbarpour, Ali Farzan, Babak Anari, and Saeid Taghavi Afshord. "A new intelligent hybrid feature extraction model for automating cancer diagnosis: a focus on breast cancer." The Journal of Supercomputing 81, no. 5 (2025): 651.

[22] Bhure, Chaitanya, Geraldine Shirley Nicholas, and Fareena Saqib. "Machine Learning Applications and Attacks Using Side Channel Analysis." In AI-Enabled Electronic Circuit and System Design: From Ideation to Utilization, pp. 387-426. Cham: Springer Nature Switzerland, 2024.

[23] Chtioui, Sondoss, Sebti Mouelhi, Sébastien Saudrais, Toufik Azib, Marc Ille, Melanie Morel, and Frederic Oru. "XGBoost in Public Transportation for multi-attribute data: delay prediction in railway systems in real-time." IEEE Access (2024).