Rohit_24-VLS-11_m.tech.pdf

Rashtriya Raksha University

Document Details

Submission ID

trn:oid:::30110:142451263

Submission Date

Jun 10, 2026, 5:11 PM GMT+5:30

Download Date

Jun 10, 2026, 5:22 PM GMT+5:30

File Name

Rohit_24-VLS-11_m.tech.pdf

File Size

1.5 MB

25 Pages

4,820 Words

26,926 Characters

8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text

Match Groups

41 Not Cited or Quoted 8%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

3%	Internet sources
2%	Publications
6%	Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

41 Not Cited or Quoted 8%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

3%	Internet sources
2%	Publications
6%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	**www.frontlinejournals.org**	<1%
2	Submitted works	**University of Newcastle upon Tyne on 2026-03-05**	<1%
3	Submitted works	**University of Leeds on 2026-05-09**	<1%
4	Internet	**www.nature.com**	<1%
5	Submitted works	**Chesterfield College on 2018-04-17**	<1%
6	Submitted works	**IUBH - Internationale Hochschule Bad Honnef-Bonn on 2024-04-07**	<1%
7	Internet	**dspace.daffodilvarsity.edu.bd:8080**	<1%
8	Submitted works	**Prairie View A&M University on 2017-07-27**	<1%
9	Publication	**Lecture Notes in Computer Science, 2016.**	<1%
10	Internet	**escholarship.org**	<1%

11	Internet	
e-archivo.uc3m.es		<1%
12	Publication	
Arvind Dagur, Sohit Agarwal, Dhirendra Kumar Shukla, Shabir Ali, Sandhya Sharm...		<1%
13	Submitted works	
Universiti Teknologi Malaysia on 2010-08-22		<1%
14	Submitted works	
University of Southern California on 2013-09-20		<1%
15	Internet	
learning-gate.com		<1%
16	Internet	
patents.justia.com		<1%
17	Submitted works	
University of Lancaster on 2025-12-31		<1%
18	Internet	
www.mdpi.com		<1%
19	Publication	
B.S. Carlson, Suh-Juch Lee. "Delay optimization of digital CMOS VLSI circuits by tr...		<1%
20	Submitted works	
CSU, San Jose State University on 2025-05-03		<1%
21	Publication	
Huizi Li. "Piano Automatic Computer Composition by Deep Learning and Blockch...		<1%
22	Submitted works	
Illinois Institute of Technology on 2025-02-19		<1%
23	Submitted works	
Jawaharlal Nehru Technological University on 2011-01-28		<1%
24	Submitted works	
Queensland University of Technology on 2025-05-29		<1%

25	Submitted works	
South Bank University on 2018-05-08		<1%
26	Submitted works	
University of Wollongong on 2020-04-15		<1%
27	Internet	
investigacion.unirioja.es		<1%
28	Publication	
Zhang, Shida. "Reliability Characterization of Mixed-Signal Circuits: Integrated Vo...		<1%

CHAPTER-1

INTRODUCTION

1. Background

With the rapid development in very large-scale integration technology, the complexity in modern digital circuits hass also increased significantly [1]. Modern ICs contain transistors in Billions of number, and at such huge number the timing characteristics of each logic gates together determine whether a chip meets its target clock frequency. In case of CMOS logic circuits, One of the most important characteristics are Delays, because the delay of a circuit is directly related to the speed and reliability of the circuit [2].

2. Propagation Delay and Its Significance

Propagation delay means that the time interval between the moment an input signal crosses a defined voltage threshold and the instant the corresponding output signal crosses the same threshold. The reference point used for both the input and the output transitions is 50% voltage level of the full supply swing. There are two delay components which are typically measured: the high-to-low propagation delay (tpHL), which corresponds to a falling Transition of output , and the low-to-high propagation delay (tpLH), which corresponds to a rising Transition of output. The mean of these two quantities Gives the overall propagation delay of the gate.

During the physical design stage , accurate Knowledge of propagation delay is necessary because setup and hold time violations at flip-flop inputs directly can cause Timing failures. As clock frequencies push beyond the multi-gigahertz range, even sub-picosecond estimation errors can lead to timing closure problems which needs expensive design re-spins.

3. Factors Affecting CMOS Gate Delay

The propagation delays in the case of CMOS logic gates are affected by the output capacitive load and input slew rate . The output load capacitance, which comprises the gate capacitance of Load connected at output, interconnect parasitic capacitance determines the charge that must be delivered or removed during a switching event. A larger load takes longer to get charged or discharged, thereby increasing the delay.

The input slew rate that is, How much time does the voltage transition takes at the input of gate also plays a crucial role. The pull-up and pull-down transistor networks spend more time in their active regions due

to slower transitions, causing high short-circuit current and a delayed crossing of the output threshold value. Beyond load and slew, supply voltage variations changes the overdrive available to the transistors, and shift in temperature changes carrier mobility and threshold voltage, both of which changes the gate delay .

1.4 Limitations of Traditional Delay Estimation

If traditional analytical or empirical models are used to compute delays in the circuit, then this may not be accurate because these traditional models may miss to capture the nonlinear relationship between the circuit parameters and the delays in the circuit. While highly accurate SPICE-level simulation, solves multiple mathematical equations at every time step and therefore demands significant computational resources and consumes lot of time.

1.5 Deep Learning for Delay Modeling

To mitigate the above limitations, the deep learning-based models have shown promise in modeling complex nonlinear relationships in complex circuits. Deep neural networks are very effective in predicting the variations in the delay for different load and slew rates based on the simulation data provided. It is evident that the deep learning-based models can efficiently handle the complex relationship between the parameters and the circuit topology. Therefore, the integration of the concept of deep learning in the field of CMOS delay is very effective.Several families of deep learning models are well suited to this regression task. Straightforward nonlinear mapping through stacked dense layers offered by Fully connected deep neural networks (DNNs). And when the input parameters are arranged as structured vectors Convolutional neural networks (CNNs) can exploit local feature interactions. Long Short-Term Memory (LSTM) networks, originally designed for sequential data, can also capture ordering dependencies among circuit parameters. Transformer architectures, which is a self-attention one, learns pairwise relationships among all the input features simultaneously. A hybrid approach that combines the attention mechanism of the Transformer with the dense regression layers of a DNN can potentially use the strengths of both architectures.

With the rapid development in very large-scale integration technology, the complexity of modern digital circuits is also increasing significantly [1]. In the case of CMOS combinational logic circuits, the delays are one of the most important characteristics, and this is because the delays in the circuit are directly related to the speed and reliability of the circuit [2]. It is observed that the propagation delays in the case of CMOS logic gates are significantly affected by dynamic factors, and this includes the capacitive load and input slew rate [3]. If the delays in the circuit are estimated using traditional analytical or empirical models, then this may not be accurate because the conventional models may not be able to capture the nonlinear relationship between the circuit parameters and the delays in the circuit [4].

To overcome the above limitations, the recent advancements in the field of deep learning-based models have shown promise in modeling complex nonlinear relationships in electronic circuits. Deep neural networks are effective in predicting the variations in the delay time for different loading and slew rates based on the simulation data provided [5]. It is evident that the complex relationship between the parameters and the circuit topology can be efficiently handled by the deep learning-based models. Moreover, the frameworks allow efficient optimization in the circuit design based on the parameters identified as critical in the delay time. Therefore, the integration of the concept of deep learning in the field of CMOS delay modeling has shown promise in improving the performance and analysis in the field
.

CHAPTER 2

1

LITERATURE REVIEW

The application of machine learning and deep learning techniques to VLSI circuit analysis has attracted growing attention in recent years. This chapter reviews the key contributions that form the background and motivation for the present work.

2.1 Machine Learning for High-Speed Circuit Design

7

Melikyan (2024) provided a comprehensive treatment of how machine learning algorithms can be applied to the design and optimization of high-speed ICs operating at data rates between 128 and 512 Gbps [6]. The book covers regression-based approaches for predicting signal integrity metrics and demonstrates that ML-driven surrogate models can replace time-consuming electromagnetic simulations during the early design phase. The work underscores the broader trend of embedding data-driven methods into traditionally simulation-heavy EDA workflows.

2.2 Timing Modeling for Complex Logic Gates

Beniwal et al. (2025) introduced LiMo, a framework that couples Boolean satisfiability (SAT) solvers with machine learning algorithms to construct timing models for complex logic gates under multiple input switching (MIS) conditions [7]. By enumerating the relevant switching scenarios through SAT-based analysis and training ML regressors on the resulting delay data, LiMo captures timing behavior that conventional single-input-switching models overlook. Their results showed notable accuracy improvements for gates with high fan-in, where MIS effects become dominant.

Vagenas et al. (2026) extended the MIS modeling paradigm by developing a neural-network-based predictor that simultaneously estimates timing, dynamic power, and glitch characteristics of CMOS gates under arbitrary input switching patterns [8]. Their unified model eliminates the need for separate characterization runs for each performance metric, offering a significant reduction in library characterization effort while maintaining prediction errors below 5% across a range of standard cell topologies.

2.3 Deep Learning for Circuit Optimization

Xue et al. (2025) proposed DOMAC, a differentiable optimization framework for designing high-speed multipliers and multiply-accumulate (MAC) units [9]. By formulating the design decisions as continuous parameters within a differentiable computation graph, DOMAC enables gradient-based optimization of both the circuit structure and its timing characteristics. The framework demonstrated that deep-learningguided search can discover multiplier architectures with lower delay-area products than those produced by conventional synthesis tools.

2.4 Reliability-Aware ML Optimization

Akhtar et al. (2025) addressed the reliability dimension of circuit design by developing RelOps, a machine-learning-based methodology for optimizing standard cell reliability across process, voltage, and temperature (PVT) variations in FinFET digital circuits [10]. Their approach jointly minimizes the impact of aging mechanisms such as bias temperature instability (BTI) and hot carrier injection (HCI) while preserving power and delay targets. The study demonstrated that ML-guided sizing can extend circuit lifetime without incurring the area overhead associated with traditional guard-banding strategies.

2.5 DNN-Based Delay and Power Prediction

Elwehili and Aissa (2025) employed a deep learning architecture to predict the oscillation frequency and power consumption of CMOS voltage-controlled oscillators (VCOs) as a function of component sizes [11]. By training a DNN on SPICE-generated data covering a wide range of transistor width and passive element values, they showed that the model could guide component sizing toward target specifications with prediction errors under 2%. Their work highlights the applicability of DNN-based surrogate modeling to analog and mixed-signal circuit blocks in addition to digital gates.

6. CNN and LSTM Applications in CMOS Circuits

Palani et al. (2023) applied deep-learning-based compression and classification models to CMOS image sensor data, demonstrating that CNNs can extract meaningful features from raw sensor outputs for efficient downstream processing [12]. Although their application domain differs from delay prediction, the feature

8 extraction capability of convolutional layers is directly relevant to the CNN-based regression model developed in this thesis.

Tang et al. (2026) reported on a CMOS-compatible lithium niobate domain-wall entropy engine and employed LSTM networks to characterize the temporal dynamics of the random number generation process [13]. Their use of LSTM to capture time-dependent device behavior parallels the approach adopted in this thesis, where LSTM layers are used to model sequential dependencies among circuit parameters that influence propagation delay.

7. Transformer Architectures in Circuit Analysis

Ding et al. (2026) analyzed and designed transformer-based CMOS D-band wideband frequency doublers, employing transformer-coupled topologies to achieve broadband operation [14]. While their use of the term "transformer" refers to passive on-chip inductors rather than the attention-based neural network, the mathematical modeling challenges they faced — predicting nonlinear circuit behavior across a wide frequency range — mirror the regression problems addressed by attention-based deep learning models in the present work.

8. Summary of Literature and Research Gaps

The reviewed literature establishes that machine learning and deep learning techniques are increasingly being adopted for VLSI timing prediction, power estimation, and design optimization. However, several gaps remain that motivate the present study:

1. Most existing works focus on a single deep learning architecture; a systematic comparison of DNN, CNN, LSTM, and Transformer models on the same delay prediction dataset is lacking.
2. The potential of hybrid architectures that combine the attention mechanism of Transformers with the dense regression capability of DNNs has not been explored for CMOS delay modeling.
3. Few studies jointly address delay prediction and delay optimization within a unified deeplearning-based framework.
4. The combined effect of load capacitance, input slew, supply voltage, and temperature on gate delay has not been thoroughly studied using modern deep learning architectures.

This thesis addresses these gaps by developing, training, and comparing five distinct deep learning models on a comprehensive CMOS delay dataset and proposing a Hybrid Transformer–DNN architecture that achieves superior prediction accuracy

sented a framework called DOMAC, which is a differentiable optimization framework for designing multipliers and MAC units using deep learning algorithms [9].

CHAPTER 3

METHODOLOGY

This chapter describes the overall approach to develop the Deep Learning based model to predict propagation delay. Each step of the process is covered which includes the selection of circuits, simulations, then dataset creation, data preprocessing, feature engineering, baseline model development, design of Deep learning architecture, training the process and the metrics which is used to evaluate the performance.

3.1 Circuit Selection

This research on fundamental CMOS logic. These gates are selected for the following reasons:

1. **Basic Building Blocks**: The NAND gate is the foundation of all the digital circuits. Any Boolean function can be realized using NAND.

2. **Fundamental Electrical Characteristics**: This gate capture the essential switching characteristics of CMOS circuits which includes the rise/fall time, propagation delay which are common to all the CMOS logic designs.

3. **Practical Relevance:** This gate are widely used to make complex digital system like adders, MUX, decoders, FF and also the memory units. At this level the accurate delay estimation is very crucial for the timing analysis of large circuits.

4. **Diverse Topologies:** The NAND (series NMOS with parallel PMOS represent main transistor configuration in CMOS logic, that offer a comprehensive basis for building and evaluating the prediction model.

14 A CMOS NAND gate has 2 PMOS in parallel and 2 NMOS in series. When A and B are high, pull down conducts otherwise pull up.

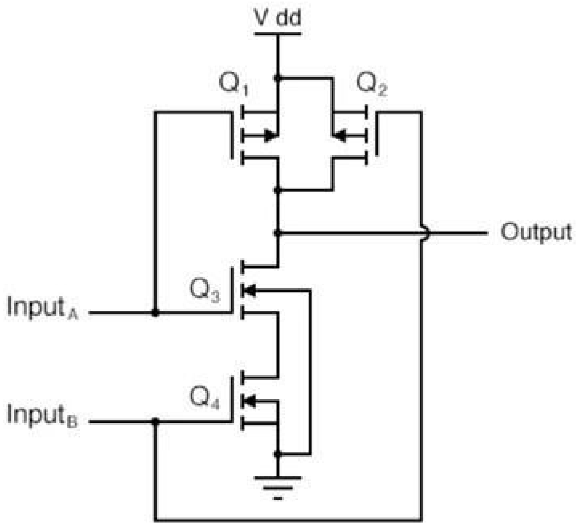


Fig. 3.1: CMOS NAND Gate

3.2 Simulation Setup

Transient simulation is done by applying input pulse and output is analyzed.

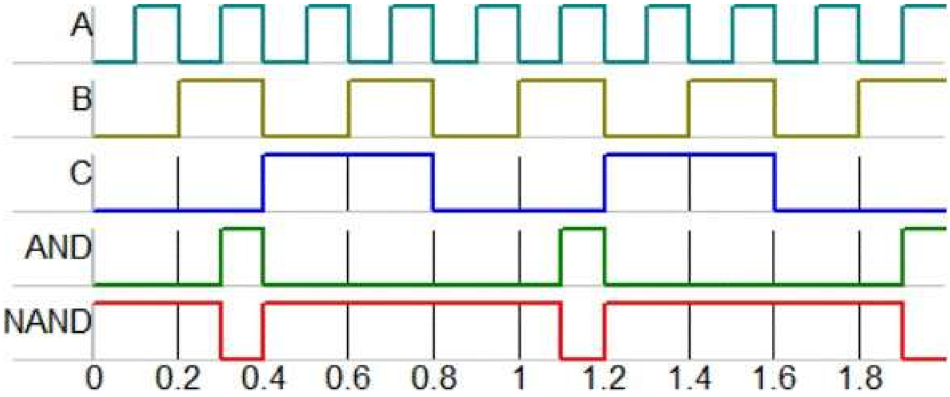


Fig. 3.2: Pulse Input Waveforms Used for Transient Simulation

This Figure describes the pulse input signals applied to the inputs A and B for transient simulation.

3.3 Parameter Sweep

Parameter sweep analysis is done to see how delay is influenced by operating conditions and to see the overall performance of CMOS logic circuits. Parameter sweep simulations runs multiple simulations, systematically changing circuit parameters. This approach helps in understanding the sensitivity of the circuit under different scenarios and gives a detailed characterization of the design.

In the first phase of the analysisLoad variation is involved by sweeping the output load capacitance over different values. Load capacitance is directly impacting the charging and discharging of the output node. With increase in capacitance, the circuit takes more time to switch, leading to increase in propagation delay. Reviewing the delay characteristics under different lconditions of load, helps us understand the drive strength and capability of performance of the logic gates.

In the second stage, slew variation is involved where we vary the rise/fall times of the input are varied systematically. Faster input transitions resulrs in reduced delat and improving the switching performance, slower transitions may increase the delay and signal quality may degrade. The sensitivity of the gate delay to different input conditions can be evaluated more accurately.

In the final stage, voltage variation is analyzed by sweeping the Vdd across different ranges.Supply voltage influences switching speed, delay and power dissipation. Lower Vdd reduces the power consumption but can also slow down the circuit operation. Higher Vdd improvea the switching speed but increases the power usage. We can get a better understanding of the circuit behavior by evaluating it under different voltage conditions.

All these parameter sweeps are done using the batch simulation techniques, where we run multiple simulation automatically. Minimal manual effort is required and it provides efficient generation of large amounts of data that is characterized.

3.4 Delay Extraction

The propagation delay measurement methodology employs the standard 50% input voltage to 50% output voltage transition method. The delay is calculated as the time difference between when the input signal reaches 50% of its voltage level and when the output signal reaches 50% of its voltage transition level. This method is used for consistent measurement of both the rise delay time (tPLH) and fall delay time (tPHL) in CMOS logic circuits.

Once we get the delay values, they are then exported into a csv format to process further and analyze it. The generated dataset is then utilized to do the characterization studies, for performance evaluation and ML based prediction of propagation delay in the CMOS circuits.

3.5 Dataset Creation

It begins with collecting the results generated from the experiment. These simulation outputs are stored in csv format and then imported into a python environment using the Pandas library. This allows the data to be organized and then it can be further processed for analysis.

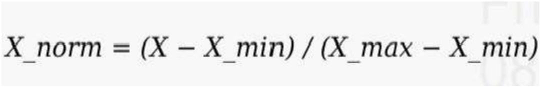
After the data is imported, input features that influence the timing behavior are checked. These features will include gate type, load capacitance, slew rate, Vdd, temp. Every parameter plays a very significant role to see the switching characteristics and determine the overall performance of CMOS circuit. The delay values corresponding to each input conditions generates a dataset, that is further used for the analysis, characterization and training the learning odels for delay prediction.

3.6 Data Preprocessing

Data preprocessing generates the dataset for the DL model traning. First the dataset is cleaned by removing the duplicates or missing entries using different functions. It is done because it improves the data quality and will ensure that accuracy is maintained and model performance is not affected.

To ensure data quality and reliability, dataset undergoes through preprocessing before training:

1. Invalid Data Removal: Data points are identified and removed from unstable SPICE simulations (e.g., convergence failures, unrealistic delay values).
2. Outlier Filtering: Outliers are identified and removed using the Interquartile Range (IQR) method. Data points lying below Q1 − 1.5×IQR or above Q3 + 1.5×IQR are taken as outliers and are excluded from the dataset.
3. Feature Normalization: Normalizing of all input features are done using Min-Max. This ensures equal contribution of all the features to the model training and prevents features with large values from dominating the process. The normalized value of a feature is calculated using the following equation:



$$X\_norm = (X - X\_min) / (X\_max - X\_min)$$

3.7 Model Development

In the present study, different deep learning models have been developed to predict the propagation delay in CMOS combinational logic circuits for different load and input slew conditions.

⑱ Model 1: Deep Neural Network (DNN)

A Deep Neural Network predicts CMOS propagation delay using circuit parameters through dense layers with ReLU activation to model nonlinear timing behavior. Input feature vector can be represented as eqn(1).

$$X = [G , C_L , S , V_{DD}, T]$$

⑪ This vector contains the circuit parameters used to predict delay . Hidden Layer Transformation can be expressed as

$$h_{(l)} = f(W_{(l)}\, h_{(l)} + b_{(l)})$$

⑮ ReLU activation function can be expressed as $f(x)=\max(0,x)$

output layer (Delay prediction)can be expressed as

$$td = W_o\, h_{(l)} + b_o$$

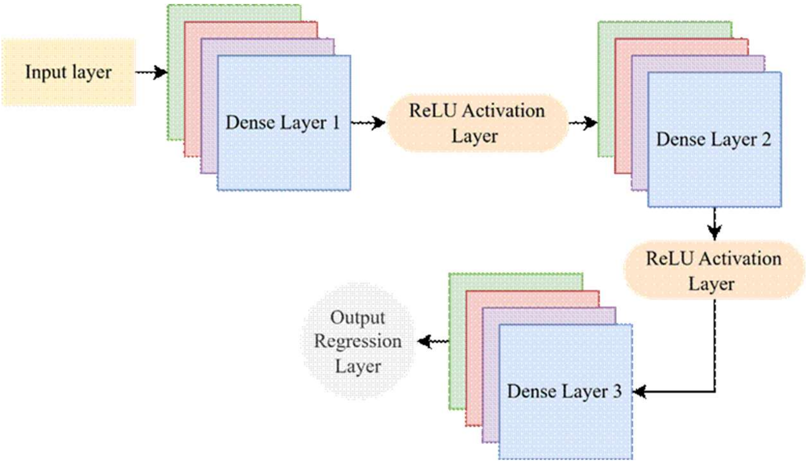


Fig. 3.4: Architecture of DNN

⑧ Model 2: CNN-Based Regression Model

The CNN-based regression model captures the interaction between circuit parameters and their effects on the delay. The feature vector, consisting of load, slew, voltage, and gate types, is rearranged and fed into a ⑦ Conv1D layer to learn the correlation between features. Next, the MaxPooling layer is used to reduce the features, followed by dense layers for regression to compute the propagation delay.

Feature Representation for this model can be given as,

$$X = [x1, x2, x3, x4, x5]$$

Convolution Operation can be expressed as,

$$z_i = \sum_{k=0}^{K-1} \quad w_k x_{i+k} + b$$

Activation Function can be given as,

$$a_i = max\ (0, zi)$$

Max Pooling function can be expressed as,

$$p_j = max\ (aj \times m, \ldots, aj \times m + m - 1)$$

Dense Layer Regression cab be mathematically given as,

$$\hat{t}d = Wdp + bd$$

Loss Function for CNN can be given as ,

$$L = \frac{1}{N} \sum_{i=1}^{N} \quad (t_d - \hat{t}_d)^2$$

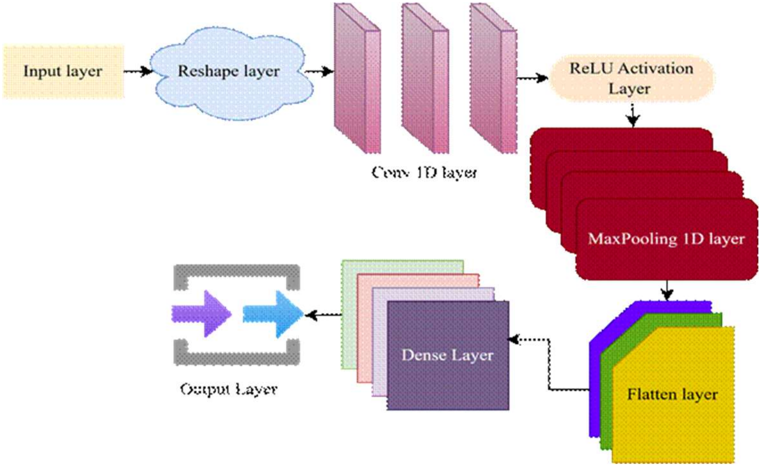


Fig. 3.5: Architecture of CNN-based Regression Model

Model 3: LSTM Model

The LSTM model processes circuit parameters as sequences to capture dependencies affecting CMOS delay. Sequence features are learned through LSTM layers and mapped by a dense layer to accurately predict propagation delay.

Sequence Representation for LSTM model can be given as,

$$X=\{x1,x2,...,xt\}$$

Input Gate can be expressed as,

$$i_t = \sigma(Wixt + Uiht-1 + bi)$$

Forget Gate can be expressed as,

$$f_t= \sigma(W_f x_t + U_f h_{t-1} + b_f) \text{ Cell}$$

Candidate can be mathematically given as,

$$\tilde{C}_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$$

Cell State can be updated as ,

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t$$

Output Gate can be expressed as,

$$o_t = \sigma(W_0 x_t + U_0 h_{t-1} + b_0)$$

Hidden State can be expressed as,

$$h_t = o_t \, tanh \, (_t)$$

The final Delay can be Predicted using eqn,

$$t\hat{}d = W_d h_t + b_d$$

Overall Model Objective can be given in eqn. (20),

$$t_d = F(G,CL,S,VDD,T)$$

The deep learning models learn the nonlinear relationship and it can be given in eqn,

$$\hat{t}_d = f_\theta(X)$$

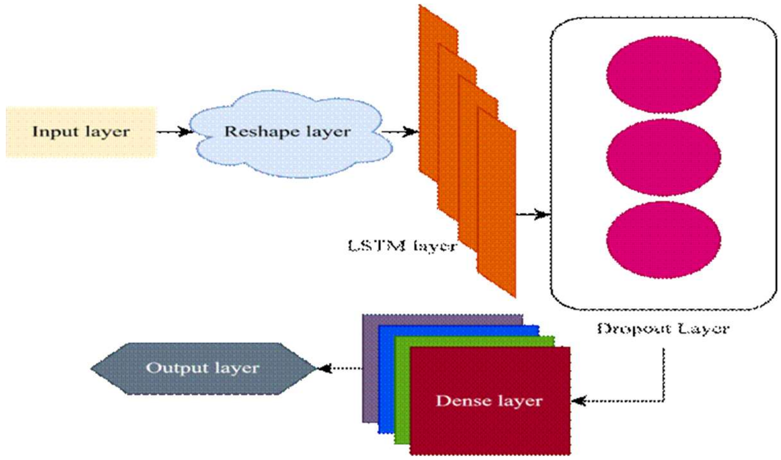


Fig. 3.6: Architecture of LSTM Model

Model 4: Transformer-Based Model

The Transformer model predicts the propagation delay of CMOS circuits by learning the relationship between different parameters. Transformer architecture uses an attention mechanism to know the importance of each input feature, unlike the traditional neural networks. Model can capture complex interactions.

The main thing about this model is the mulri-head attention mechanism, that allows the network to review how different input features are dependent. It focuses on multiple feature relationships in parallel. The features thar are extracted are then passed throught feed-forward neural network layers. These layers filter the info and improves the accuracy of the model.

At last, a regression output layer is there which is used to generate the delay value. Since, it can captute complex features, this model can provide accurate delay predictions for CMOS Acombinational logic circuits.

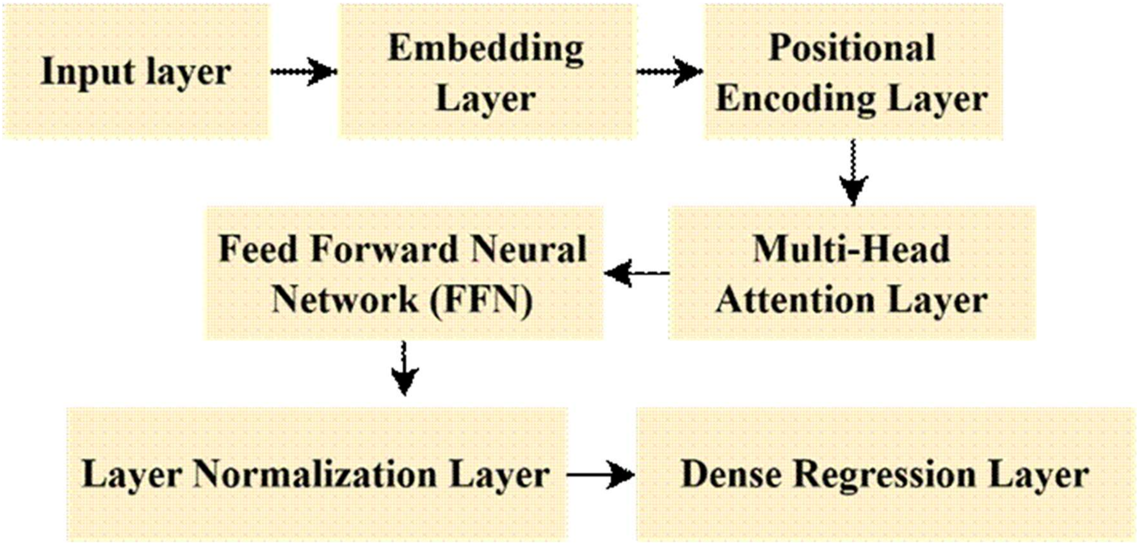


Fig. 3.7: Blocks in Transformer-Based Model

Model 5: Hybrid Transformer-DNN Model

A hybrid Transformer–DNN model predicts propagation delay by combining both Transformer networks and deep neural networks. Transformer to capture complex relationship and DNN for accuracy. Here, the input circuit parameters are first converted into embedded. The embedded features are further passed through a transformer block. The features that get extracted by the transformer are then forwarded to multiple dense layers of DNN. These layers are connected and learn from the non linear relationships between extracted features and propagation delay values.This combination improves the model's ability so that it can capture the complex behavior more effectively.

At the end then the output regression layer predicts the delay pf CMOS circuit. DNN based feature improves prediction and gives better performance.

The architecture contains mainly below components:

- Input Feature layer
- Embedding layer
- Transformer attention block
- Feature concatenation layer
- Dense hidden layers
- ReLU activation layer

- Output regression layer

The objective of the model is to predict continuous delay values of CMOS circuits using supervised regression learning.

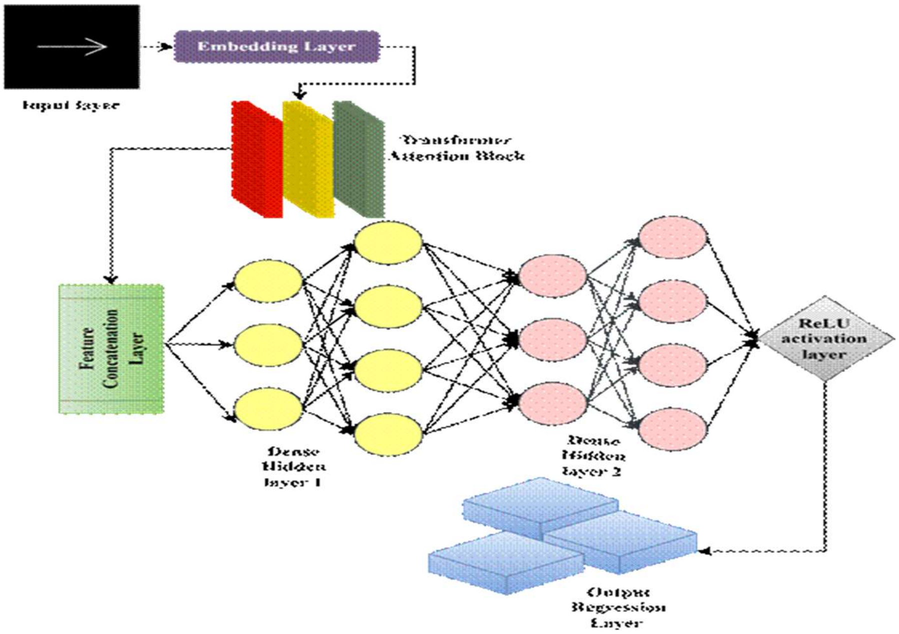


Fig. 3.8: Transformer–DNN Hybrid Regression Architecture for CMOS Delay Prediction

5.1 Advantages of the proposed architecture

- Improved feature dependemcy learning: The transformer captures long-range interactions among the timing parameters.
- Higher prediction accuracy: Hybrid learning improves delay estimation precision.
- Better generalization: The model performs very well across different process conditions.
- Reduced manual modeling: There is no need for manual analytical delay equations.
- Scalability: The architecture adapts to large-scale VLSI datasets.

Applications of the proposed model:

- Static Timing analysis
- ASIC design optimization
- FPGA timing estimation
- Low power VLSI systems

- Physical design automation
- AI-assisted EDA frameworks

5.2 Conclusion

The transformer-DNN hybrid regression architecture provides an efficient solution for CMOS delay prediction. The transformer block captures complex feature parameters using self attention mechanisms, while the dense neural network performs accurate non linear regression. The integration of this learning and deep regression improves timing prediction and accuracy as compared to traditional methods.

CHAPTER-4

RESULTS

4.1Training Parameters and Configuration

Parameter	**Value**
Loss Function	Mean Squared Error (MSE)
Optimizer	Adam / AdamW
Batch Size	32
Epochs	100

During training , the dataset is used to optimize the performance of DL model for accurate delay prediction. The MSE(LOSS FUNCTION) is used to evaluate performance of the prediction , whereas the Adam or Adamw optimizer is used to optimize the network weights with a learning rate of 0.001.Traning is performed in batches of 32 for 100 epochs.

4.2 Performance Comparison of All Models

Model	**MAE (ps)**	**RMSE (ps)**	**R² Score**
DNN Model [11]	4.21	5.38	0.962
CNN Regression Model [12]	3.74	4.92	0.971
LSTM Model [13]	3.58	4.71	0.975
Transformer Model [14]	3.12	4.15	0.981
Proposed Hybrid Model	**2.64**	**3.52**	**0.988**

Model evaluation assesses prediction accuracy of deep learning models for CMOS propagation delay. Predicted delays from model.predict() are compared with simulation values. Using Scikit-learn , metrices such as MAE , RMSE , R^2 are compared.These metrices quantify prediction errors and model fit , enabling performance comparison to identify the most accurate delay prediction model.

The DNN model achieved an MAE of 4.21 ps and an RMSE of 5.38 ps with an R2 score of 0.962. Though DNN model is most successful in learning non linear relationship among CMOS dependencies. CNN regression model improved the prediction performance with an MAE of 3.74ps and RMSE of 4.92ps. This improvement is mainly due to CNN's ability to extract local feature patterns. CNN models are generally less effective in learning global depe ndencies among timing features.

Among all the models, our proposed model achieves the best performance with the lowest MAE of 2.64ps and RMSE of 3.52ps, with highest R2 score of 0.988. We can see that the performance is improved, which shows the effectiveness of combining transformer based contextual learning with deep neural network regression capabilities.

The reduction in MAE and RMSE indicates that the proposed model minimizes prediction errors significantly compared to exixting models.

Overall results show that proposed model outperforms conventional deep learning models in terms of prediction accuracy, robustness, and regression efficiency.

The results show that all deep learning models have good accuracy in the prediction of the delay in the CMOS process. The best results are obtained using the Hybrid Transformer-DNN model with the least errors and the highest R^2 value.

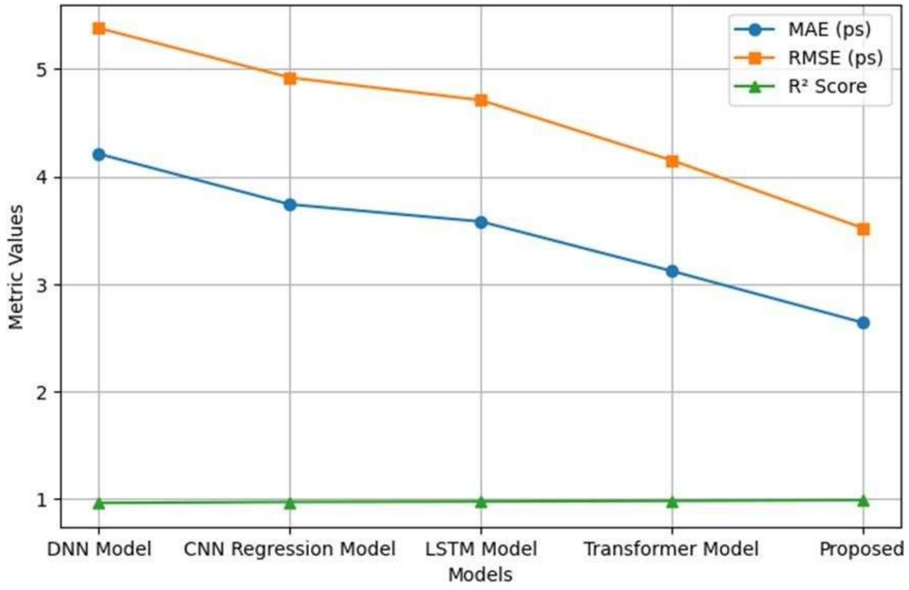


Fig 4.1 Performance comparison of Delay Prediction models

6

Fig 4.1 depicts the performance comparison of various deep learning models for the prediction of CMOS delay using MAE, RMSE , R^2 values . In the figure , it is observed that the lowest error values are achieved by the proposed model , known as Hybrid model , along with the Highest value of R^2

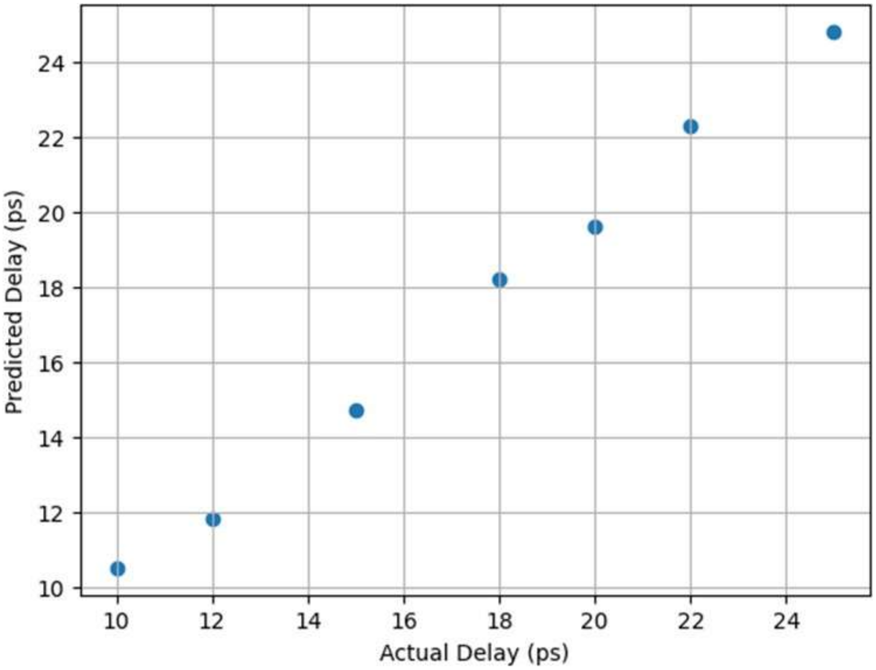


Fig 4.2 Actual vs Predicted Delay

The actual values range from 10–25 ps, and the corresponding calculated values range from 10.5–24.8 ps, showing that the model is able to learn the relationship between the values.

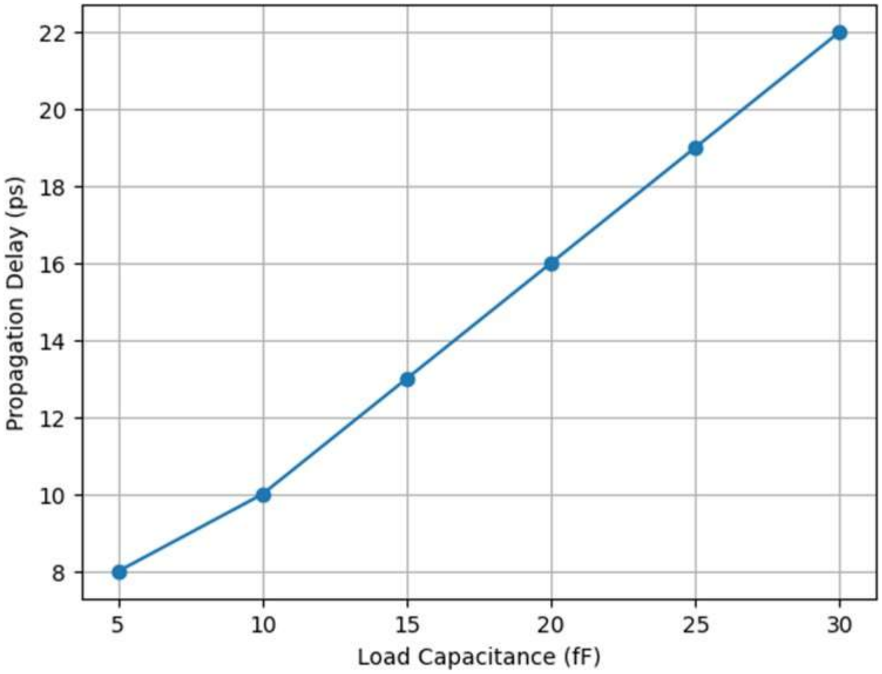


Fig 4.3 Delay vs Load Capacitance

2 3 As the load increases from 5 fF to 30 fF, the propagation delay increases from 8 ps to 22 ps, showing that the more the load, the more time is required for the charging or discharging of the output node by the CMOS gate.

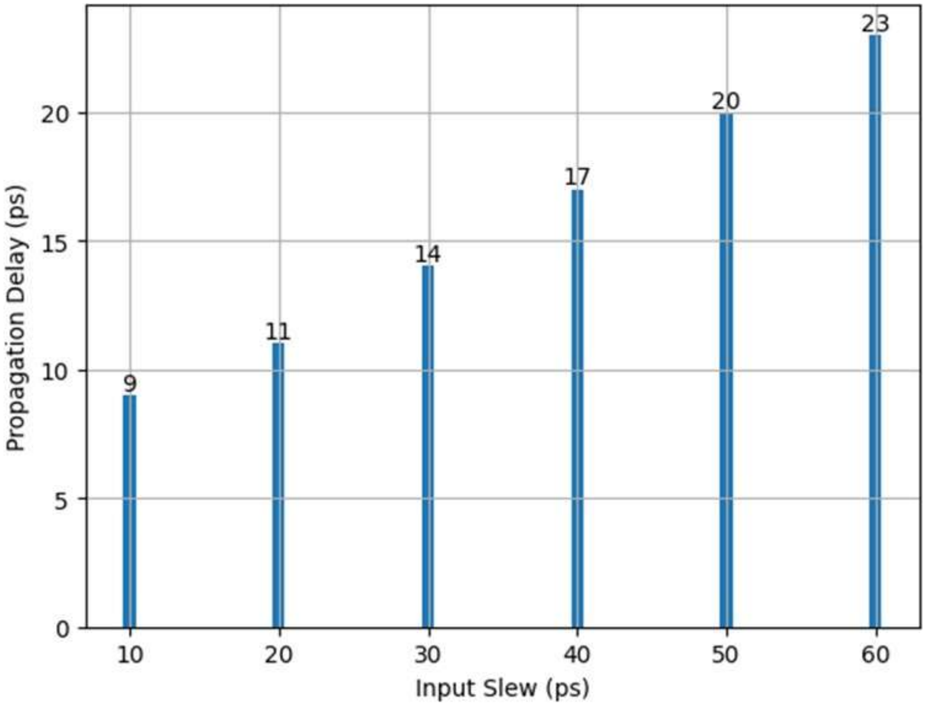


Fig 4.4 Delay vs Input Slew

As the input slew is increased from 10 ps to 60 ps, the propagation delay is increased from 9 ps to 23 ps. Propagation delay increases gradually as the input slew increases.

The observed trend shows us that a nearly linear relationship netween input slew and propagation delay. This behavior we can see because slower input transitions reduce the switching speed of CMOS transistors, thus increasing the overall charging and discharging time of the load capacitance.

4.3 Delay Optimization

Delay optimization uses the trained model to evaluate candidate circuit parameter combinations and selects the configuration with the minimum predicted propagation delay as the optimal solution. Let the input parameter vector can be denoted as eqn.

$$X = [G, CL, S, VDD, T]$$

The trained deep learning model predicts delay and it can be given in eqn.

$$\hat{t}\, d = f\theta(X)$$

Where, $\hat{t}\, d$ is the predicted propagation delay, $f\theta$ is the trained model function and θ is the learned model parameters.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

1. Conclusion

This thesis has shown that a deep learning-based delay modeling for CMOS combinational logic circuits under load and input slew variations can provide with accuracy.

- The results demonstrates that the model that is proposed can significantly increase the prediction of accuracy as compared to deep learning based models.
- As compared to the DNN model, the proposed model achieves improvements of 37.3% in MAE, 34.6% in RMSE, and 2.7% in R2.
- As compared to the CNN model, the proposed model shows improvement is prediction performance by 29.4% in MAE, 28.5% in RMSE, and 1.8% in R2.

- The results also show improvement of 26.3%in MAE, 25.3% in RMSE, and 1.8% in R2 , in comparision with the CNN model.

These results confirm the effectiveness of the proposed hybrid model for accurate delay prediction and efficient CMOS timing optimization.

2. Future Scope

The work presented in this thesis provides several opportunities for future research:

1. **Extension to Advanced Technology Nodes:** The proposed framework can be further used to explore for more advanced technologies like 14nm, 7nm and 5nm. In the advanced technologies, short channel behavior and process variations make the delay characteristics more non linear, in this scenario DL approaches may offer better predictions.

2. **Inclusion of Additional Gate Topologies:** This work focuses on CMOS NAND gate but the same process can be extended to more complex combinational logic cells like XOR, AOI, OAI and more multi stage circuts which are used in standard cell libraries.

3. **Multi-Output Prediction:** The future models can be extended to predict different circuit parameters together like delay, dynamic power, leakage power, slew
4. **Process Variation Modeling:** This model is capable of estimating delay variations caused by manufacturing process. Such models are good without relying much on monte carlo simulations.

5. **Integration with EDA Tools:** The trained model can be integrated in EDA tools that can enable faster and real time delay estimation.

6. **Transfer Learning Across Technology Nodes:** Transfer learning techniques can be used to reuse the model trained on one technology node for another node. This can significantly lower the simulation effort which is required for characterization.