

Machine Learning Perspective in VLSI Computer-Aided Design at Different Abstraction Levels



Malti Bansal and Priya

Abstract In the past few decades, machine learning, a subset of artificial intelligence (AI), has emerged as a disruptive technology which is now being extensively used and has stretched across various domains. Among the numerous applications, one of the most significant advancements due to Machine Learning is in the field of Very Large Scale Integrated Circuits (VLSI). Further growth and improvements in this field are highly anticipated in the near future. The fabrication of thousands of transistors in VLSI is time consuming and complex which demanded the automation of design process, and hence, computer-aided design (CAD) tools and technologies have started to evolve. The incorporation of machine learning in VLSI involves the application of machine learning algorithms at different abstraction levels of VLSI CAD. In this paper, we summarize several machine learning algorithms that have been developed and are being widely used. We also have briefly discussed about how machine learning methods have transuded the layers of VLSI design process from register transfer level (RTL) assertion generation to static timing analysis (STA) with smart and efficient models and methodologies, further enhancing the quality of chip design with power, performance and area improvements and complexity and turnaround time reduction.

Keywords VLSI · Machine learning · SoC · Integrated circuits · Algorithms · Computer-aided design · SVM · Regression

1 Introduction to Machine Learning and VLSI CAD

Machine learning is an upsurging field which is transforming the world by incorporating the advancement in artificial intelligence (AI) and data science that deals with the systems which can learn from the experience or preexisting data and improve without being specifically and comprehensively programmed for the same. In a nutshell, machine learning enables the computers with human intelligence [1].

M. Bansal (✉) · Priya

Department of Electronics and Communication Engineering, Delhi Technological University (DTU), Delhi 110042, India

Machine learning centers around developing an effective and efficient learning algorithms that assists the machines to analyze the available data and train itself to accurately predict for the unknown data samples.

In 1950s, a computer program for playing checkers was created by Arthur Samuel who worked with IBM in the AI domain. In this program, Samuel used a scoring mechanism using the locations of the pieces on board and tried calculating the probability of each side winning. Using a minimax technique, the program selects its next move, and this technique eventually got developed into Minimax algorithm. A variety of mechanisms were also developed by Samuel to allow his program become better. His program remembered all the locations on the board that was valued for the reward function. His program had learned from the available data samples and predicted the reward function. Arthur Samuel was the first who coined the term machine learning [2]. Since then, machine learning is gradually evolving in the various domains like quantum computing, robotics, data mining, very large-scale integrated circuits (VLSI), automation, artificial intelligence, Internet of things (IoT), medical, finance, military, etc. As machine learning is mainly running on neural networks and logical and computational algorithms, it makes a system both smarter and beneficial. Due to its accuracy, reliability, efficiency and ability to improve, machine learning is excessively effective over the human or natural intelligence [3]. In this paper, primary emphasis has been done on how machine learning is being applied in IC chip design and automating the EDA tools and technologies.

Very large-scale integration (VLSI) deals with the process of integrating hundreds of thousands of transistors on a single chip which is called integrated chips (IC). Due to the integration of large number of transistors, the design of such integrated circuits has turned so complex and time consuming that it becomes a challenging task to achieve it manually by the design engineers. This created the need of a computer-aided design tools for the design, verification and optimization of IC circuits which would run various computers programs that automate the processes involved without much manual calculations. This is often called electronic design automation (EDA). The development of hardware description languages, tools and technologies is helpful in simplifying the design, verification and optimization at various abstraction levels of IC chip fabrication. A variety of EDA tools are used to cut the design cycle and help the engineers to simplify the design process.

Since with the advent of time, technology nodes are dropping in VLSI leading to the lesser chip area, better performance and low power digital circuits. The demand of power, performance and area (PPA) optimization has increased. This is one of the reasons that the complexity of IC is also increasing with time which means that better CAD tools and technologies are required. Machine learning helps VLSI CAD tools to improve and become more efficient. In the further sections, we have discussed about machine learning extending its reach in VLSI at various abstraction levels to solve the design problems dealing with large amount of data, improving on existing algorithms and optimizing the learning methods for automation.

2 The Basic Paradigm of Machine Learning

The basic model of machine learning comprises several steps starting with the collection of data samples from the environment/situations/experiences. This is the preexisting data samples that help the machine to learn a given task. Data is prepared after the collection by removing errors, missing values, repeated values, etc. Next step is the most important, i.e., choosing an algorithm model which differs with the different tasks. There are numerous learning algorithms in machine learning. Now, the model is trained iteratively followed by the evaluation in which the model is tested against unknown data sample. Further, tuning of the model parameter is done to increase the performance of the model proceeded with the final step of making the predictions in the real scenario. The performance in the tasks improves as the machines gain experience while executing the tasks and updating the model each time [4]. The basic model and its process have been represented in the block diagram (Fig. 1).

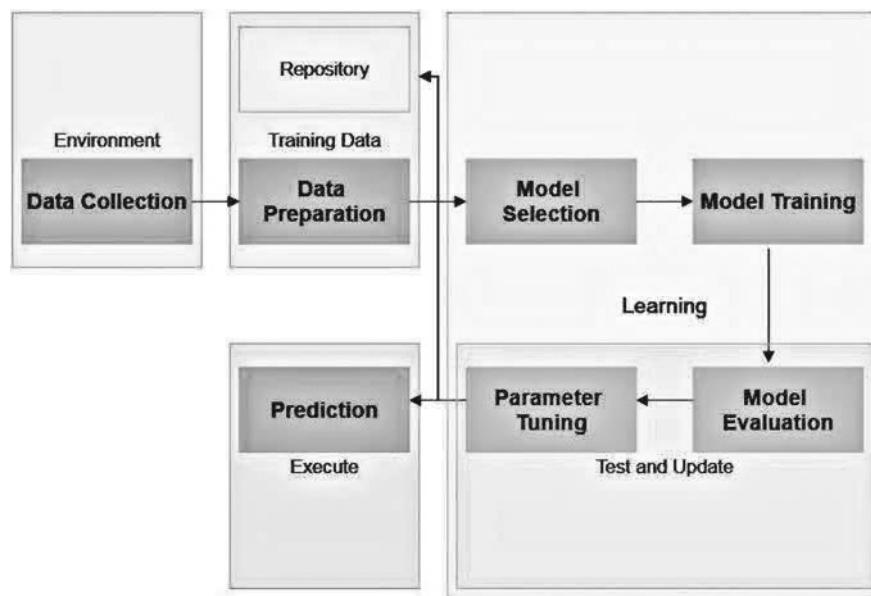


Fig. 1 Basic model of machine learning with process steps

3 Areas of Machine Learning

Machine learning has countless number of applications, and it offers solutions to many real-time issues. Some of the areas where machine learning is being progressively applied are mentioned below [3]:

- (i) Face Detection and Recognition, (ii) Visual Perception, (iii) Classification, (iv) Adaptive Systems, (v) Modeling, (vi) Speech and Image processing, (vii) Automation, (viii) Problem Solving, (ix) Genetics, (x) Anomalies Detection, (xi) Games, (xii) Internet of Things (IoT), (xiii) Quantum Computing, (xiv) Medical Diagnosis, (xv) VLSI, (xvi) Stock Market Trading, (xvii) Virtual Personal Assistant, (xviii) Online Fraud Detection, (xix) Speech Recognition, (xx) Traffic Prediction, (xxi) Email spam and Malware Filtering, (xxii) Service Personalization and (xxiii) Computer Vision.

4 Machine Learning Algorithms

The highest level of abstraction in machine learning methods is based on the source of data/information that directs into the learning [4]. It is broadly classified into three categories. These are: (i) unsupervised learning, (ii) supervised learning and (iii) semi-supervised.

In unsupervised learning, only input data is available, and some structure or label needs to be developed to distinguish between the input data samples. In supervised learning, input and corresponding output data samples are available along with the structure/labels. In semi-supervised learning, only some fractions of input data samples have corresponding output pairs, i.e., few of them are labeled or structured [5]. Figure 2 represents the basic classification of machine learning methods and the algorithms used for learning in different methods.

A brief summary of different machine learning algorithms has been represented in the form of table that briefly lists the uses, advantages and its drawbacks which can be useful in the appropriate selection of algorithm and is given in Table 1.

5 Drawbacks of Machine Learning

Although machine learning is very powerful and approved advancement in various domains, it still has its drawbacks which are discussed as follows.

During the training and learning process, a significant volume of data is used. And also the data that we use in the process should be of impartial consistency and high quality which might need the generation of more data, and hence, more time, space and power are required for better quality of results. The other drawback is that some authentic and dependable resources are required in case learning algorithms show time-consuming errors and complexity. It is very important to check that the

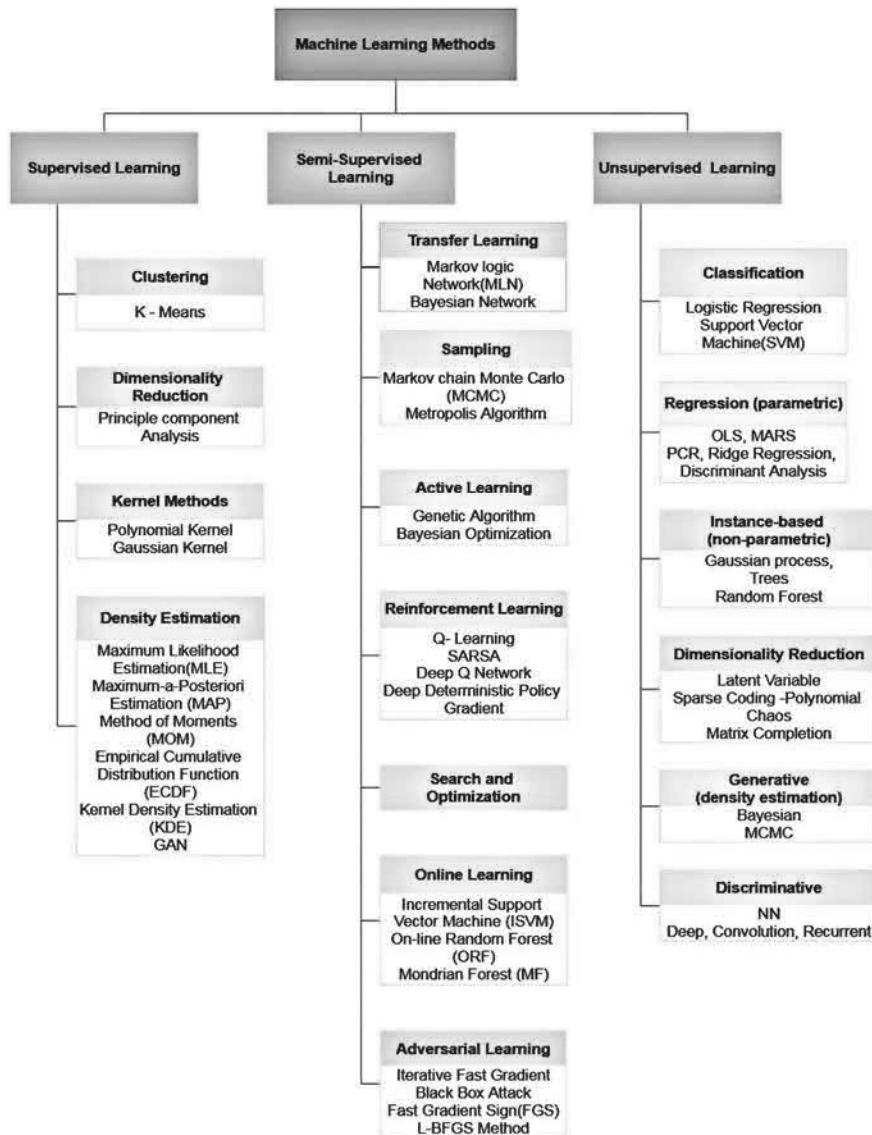


Fig. 2 Classification of machine learning algorithms

algorithms that have been assisted in the process are producing the desired output or not because to get the desired output, we need an accurate learning algorithm with high performance. Selection and availability of such an accurate algorithm is also a challenge. Machine learning still needs a lot of improvements in algorithms and the software that performs the analysis on datasets. Moreover, due to large volume of data, error susceptibility is also high which needs to be taken care of while using a particular

Table 1 Machine learning algorithms and their uses, advantages and its drawbacks

S. No.	Machine learning algorithms					
	Name of algorithm	Author	References	Uses	Advantages	Drawbacks
1.	Gradient Descent	Ray et al.	[6]	To minimize cost function	Efficient, stable error gradient	Never converges for too high and too low learning rate
2.	Linear Regression	Ray et al.	[6]	Models continuous variables, prediction Data analysis process	Easier to understand Easy to avoid over fitting	Not a good fit for nonlinear relationships, cannot handle complex pattern, over simplifies real-word issues
3.	Multi-variate Regression Analysis	Ray et al.	[6]	Used on number of independent variable and single dependent variable	Deeper insight to relationship between variables Models complex real-time issues Realistic and practical	Complex, high knowledge is required for modeling, sample size needs to be high, difficult to analyze
4.	Logistic Regression	Ray et al., Sethi et al., Bhandari et al. and Kohli et al.	[6, 7]	Used on classification problem	Simple to implement Ease of regularization Efficient in computation and training, no scaling required, reliable	Unable to solve nonlinear problem, prone to over fitting, does not work well unless all the variables are identified
5.	Decision Tree	Ray et al.	[6]	Used on regression and classification problem	Suitable for regression Classification problem, easy to interpret and handle, capability to fill missing values, high performance due to efficiency of tree traversal	Unstable, difficult to control size of tree, it may be prone to sampling error, and it gives a locally optimal solution—not optimal solution. Prone to over-fitting

(continued)

Table 1 (continued)

S. No.	Machine learning algorithms					
	Name of algorithm	Author	References	Uses	Advantages	Drawbacks
6.	Support Vector Machine	Ray et al., Choudhary et al. and Gianey et al.	[6, 9]	Used on regression and classification problem	Handles both semi-structured and structured data, can handle complex function, less probability of over fitting, scales up the high-dimensional data, does not get stuck in local optima	Low performance in large datasets, difficult to find appropriate kernel function Does not work in noisy dataset. No probability estimates. Difficult to understand
7.	Bayesian Learning	Ray et al.	[6]	To handle incomplete datasets	Prevents over-fitting, no removal of contradictions required	Prior selection is not easy Distribution can be influenced by prior, wrong predictions possible, complex computation
8.	Naïve Bayes	Ray et al., Sethi et al., Bhandari et al. and Kohli et al.	[6, 7]	Used on binary and multi-class classification problems	Easy to implement, gives good performance, less training data required, scales linearly with predictors and data samples, handles continuous, discrete data. insensitive to irrelevant features	Model often outperforms, too simple, cannot be applied directly, requires retraining, stops scaling when data points are high, more runtime memory required, complex computation for more variables

(continued)

Table 1 (continued)

S. No.	Machine learning algorithms					
	Name of algorithm	Author	References	Uses	Advantages	Drawbacks
9.	K-Nearest Neighbor	Ray et al., Choudhary et al. and Gianey et al.	[6, 9]	Used on classification problems	Simple and easy to implement, cheap and flexible classification, suitable for multi-modal classes	Expensive, computation is distant and intense, less accuracy, no generalization, data large sets

dataset and learning algorithms. Drawbacks can also be related to a specific machine learning algorithm such as nonlinearity, sampling errors, overfitting, noisy datasets, incomprehensible datasets, low performance, complex and expensive computation and insufficient runtime memory.

6 Application of Machine Learning in VLSI CAD Abstraction Levels

There has been substantial advancements in the semiconductors industry over the past five decades mainly due to expeditious technological progress applied in the field of integrated circuits (IC) to improve its performance, reduce the area, power requirements and cost [9]. This technological advancement allowed the incorporation of nearly billions of transistors on a single chip commonly categorized as very large-scale integration (VLSI). Chip designing in VLSI involves number of processing steps at different abstraction levels, concisely called VLSI design flow.

Each abstraction level in VLSI design flow has its customized EDA tool that perfectly covers all the aspects relevant to a given task in the analysis and designing of chips, often referred as computer-aided design (CAD). Figure 3 represents the steps involved in VLSI design flow along with the abstraction levels.

As the chip is designed from system specification to final layout, VLSI CAD requires the addressing of design problems at each level of abstraction. The machine learning methods that we have listed in the previous section have numerous approaches that can be applied in solving the challenges and problems faced in each step of VLSI design process. Since the main concern behind VLSI CAD was to reduce the increasing complexity at each abstraction level by developing automated simulation or generation/synthesis tools [5], VLSI CAD basically needs methods to model and test the input datasets, and machine learning algorithms serve the same purpose. Many machine learning methods have the capability to deal with the design problems and complexity of VLSI CAD. In further sections, we enumerated the applications of machine learning in various steps of VLSI designing.

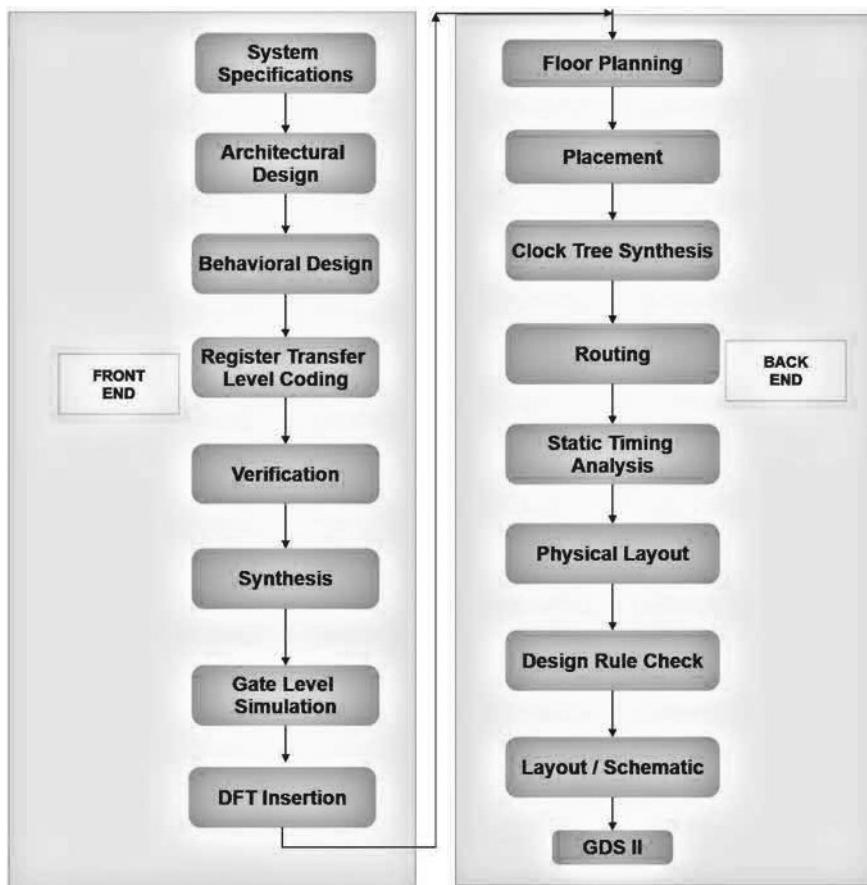


Fig. 3 Process in VLSI design flow

6.1 Machine Learning in Automatic Generation of Assertion in RTL Design

In hardware VLSI design cycle, simulation-based design verification is an important but exhaustive step which takes most of the resources and time. As the design starts getting complex with the integration of more number of hardware components, it becomes challenging for the designer to verify if the design hardware implementation meets the specification or not [5]. The behavior of the design can be monitored using assertions during simulation with the help of properties or temporal logics. Assertions are the instructions that are added in the RTL code to find errors in the design code which even stimulus-based testing cannot find out. Various HDLs have different types of assertions to incorporate in the design code which deals with a specific type of bugs. It is a very challenging task for designers to write effective and specific assertions

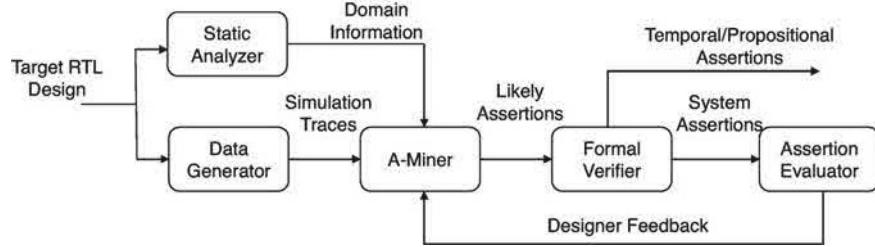


Fig. 4 GoldMine methodology [5]

manually [10]. So, multiple proposals have come up to automatically generate the assertions with the help of machine learning algorithms. The assertions generated by machine are non-uniform and discursive. One of the example is GoldMine, a tool that automatically generates assertions using data science for the analysis and verification of design code [11]. Figure 4 represents the GoldMine methodology in terms of a block diagram.

Data generator is used to simulate the register transfer level (RTL) code. A-Miner helps in mining the data obtained from simulation. Static analyzer feeds the domain information from Verilog design code to A-Miner. If the assertions produced by A-Miner are not correct, the verification fails, and assertion evaluator sends a designer feedback to A-Miner to improve the different sets of assertions. This is how the problems with learned assertions are resolved, and it eventually improves the quality of assertions [10]. Basically, A-Miner uses decision tree algorithm to improve the assertion generation process due to its simplicity, preciseness and scalability [12]. Another alternative machine algorithm that can be used is best-gain decision forest (BGDF) algorithm, coverage guided associated mining and PRISM algorithm [5].

6.2 Machine Learning in Chip Testing

The major task in chip testing is to decide if a chip should be approved or not which is done by calculating different chip parameters and affixing some predefined criteria. It is presumed that the chip with predefined criteria gets approved satisfying the system design specification [5]. These predefined criteria are operational frequency, chip area, power consumption, timing constraints, performance robustness, etc. These criteria directly affect the manufacturing yield and quality of chip.

Statistical optimization is required to solve the issues related to chip testing. It enhances the manufacturing yield by meeting all the area, timing and power design constraints. On-chip process variation (OCV) is also taken into account while calculating the chip slack which is a parameter that needs to be modeled using random variables [13]. There is additional slack during testing called test margin which is

computed using a model for the optimization of chip from the perspective of timing constraints [5].

6.3 Machine Learning for Physical Design Routing

Routing is one of the major steps in back-end of the VLSI design flow that is followed after the placement of pins, logic cells and preplaced cells. It refers to the process of wiring and connecting the pins, logic cells or preplaced cell according to the gate-level netlist produced at the end of front-end of the process [14]. It is a challenge to route billions of flops/logic cells on a chip along with maintaining the design rule checks and constraints.

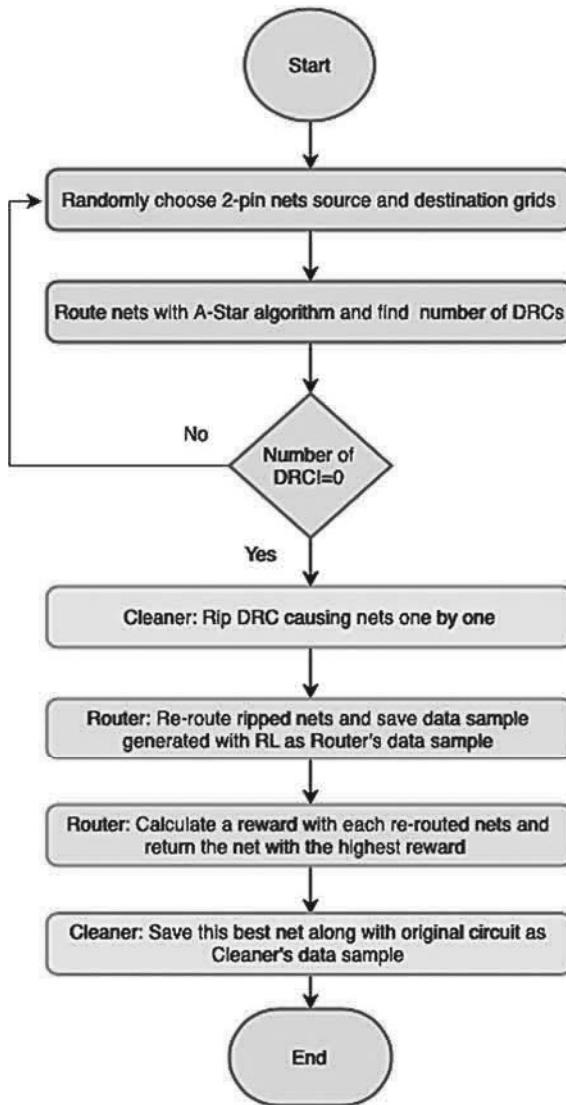
Since there is an insufficiency of large datasets, physical design routing process faces major issues in the learning using supervised machine learning algorithms [15]. In order to curb the issue of limited dataset, semi-supervised machine learning algorithm can be used. Reinforcement learning also termed as RL approach is one of the semi-supervised learning algorithms which learns from its own experience and hence does not require large datasets [14]. A routing technique based on RL Approach is Alpha RL Router which utilizes the Min–Max game methodology and physical design routing algorithm inspired by an Alpha Go Zero framework which was developed by Google in which software had learned to play the game of Go without any human intervention [16]. The basic methodology of routing issues worked on an assumption where router and cleaner are the two players collaborated as a game. An algorithm such as A-star is employed by the router which helps in performing the initial routing without taking into the consideration of violations in the design rule. Cleaner identifies the breaches in the design rules and chooses the best net that fixes those breaches/violations and rips the old nets and sends them one by one to the router to reroute the nets. Router in the next turn routes the ripped net with the best possible net identified by the cleaner. This process is repeated until all the nets on the chip are routed without any design rule violations. Cleaner focuses on the maximization of its reward score with each step of fixing the design rule violation. It is issued by the router depending upon the quality of prediction made by the cleaner [14]. Figure 5 shows the condensed methodology of router and cleaner.

The optimization in router prediction was done with the help of MCTS algorithm connected in feedback with the neural network (NNET) to predict the next move based on the result obtained from the comparison of probabilities obtained from MCTS and NNET.

6.4 Machine Learning in Physical Design Floor Planning

Due to technological advancements, new additional features and functionalities are being incorporated due to which SoC designs are getting extremely complex since

Fig. 5 Alpha-PD-router reinforcement learning-based methodology [14]



they require large number of standard cells, macros and IPs on a single chip. Moreover, it requires high computational speed to which machine learning and neural networks give an excellent assistance. But, there are few issues faced in automating the SoC designs such as the presence of target specific macros, i.e., hard macros, complex connection between different functional blocks and timing constraints associated with blocks or cells for proper functionality of entire chip [17]. Floor planning is the foremost step in physical design flow where most of issues can be dissolved and can be prevented from reoccurring at the later stages where the cost of testing and

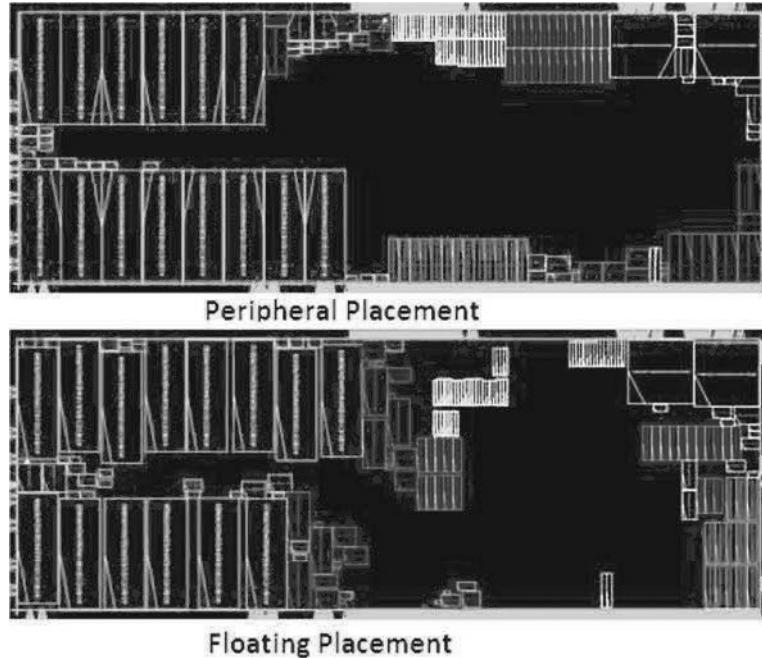


Fig. 6 Different styles of macro placement [17]

removing bugs would be more. This step determines the position of the preplaced and logic cells on the core of the chip, and further, it is followed by the routing and connections of those cells. To perform the placement efficiently, there are several machine learning technologies that have been proposed such as large-scale high-quality macro placement, GPU and mixed-sized global placement [17] (Fig. 6).

Automatic placement techniques segregate the design into sub-blocks depending upon different characteristics, properties and requirements. These sub-blocks have different types of layout styles like peripheral placement and floating placement. These layout styles are chosen by the machine as per the need which is learnt over the experience and available datasets. Power planning is usually done by peripheral placement, and low congestion routing is done by floating placement [17].

6.5 Machine Learning in Static Timing Analysis (STA)

Static timing analysis contends with the timing performance of the design where it checks the timing constraint of each path and ensures the proper functionality of the design. If any of the paths does not meet the timing constraints, it is said to be a timing violation which eventually affects the performance of the design. The

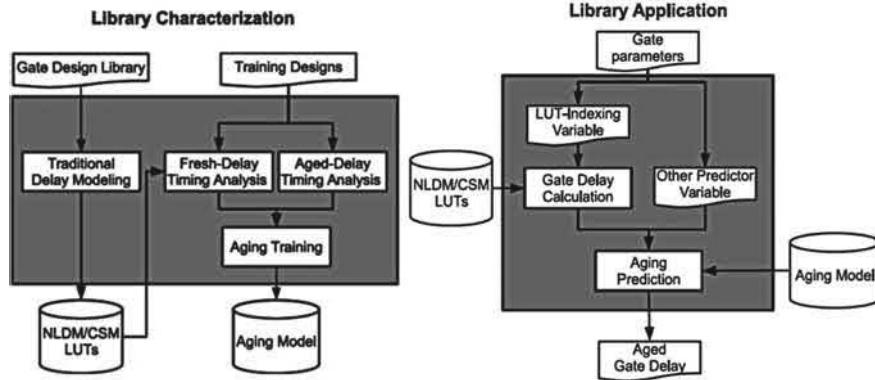


Fig. 7 Characterization and application of libraries in LSTA [18]

challenge associated with STA is high-dimensional correlation. A machine learning-based timing characterization is a potential solution to the above challenge [18]. Support vector machine (SVM) and artificial neural network (ANN) are few of the regression-based machine learning algorithms that can be exploited to correct the path-based timing violations to improve the accuracy [19].

For accurate STA, two main library parameters such as load capacitance and slew are considered to be two dimensions that need to be modeled. These two dimensions get destroyed over a period of time due to dynamic variations like negative bias temperature instability and hot carrier injection also called aging effect. This effect also affects the threshold voltage which ultimately changes timing requirements in STA. Learning-based STA (LSTA) flow can be briefly described in three steps (i) Recognition of challenges related to high dimensions in STA. (ii) Proposing a learning-based model that specifically deals with existing challenges. (iii) Experimenting and evaluating the algorithms on setup libraries included in STA. Figure 7 represents the characterization process of setup libraries in STA during training and application of modified libraries in timing step.

A predefined and preexisting two-dimensional LUT is used as aged cell delays along with a set of training design samples as fresh cell delays in the characterization process of setup libraries. These delays with other parameters included are used as the final training samples for machine learning algorithms. The models learn from these training samples. Finally, this learned model is consolidated to STA library. This modified STA library is applied in the same way, and preexisting ones were applied [18].

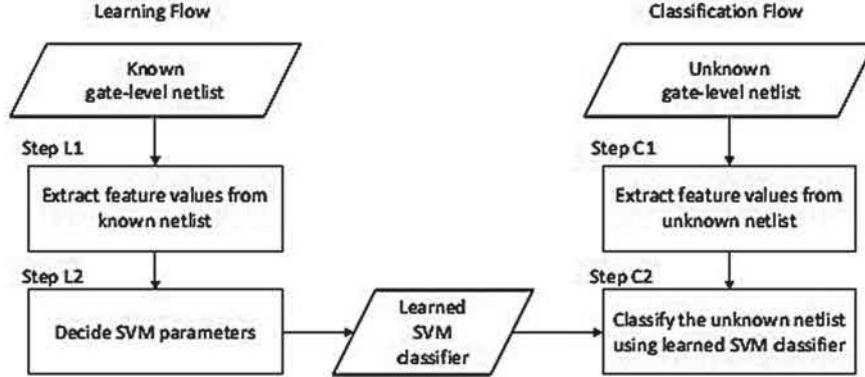


Fig. 8 Flow of learning and classification in gate-level netlist [20]

6.6 Machine Learning in Gate-Level Netlist

After the RTL synthesis, the design is implemented in the form of logic gates/cells. Gate-level netlist gives the information about the connectivity of gates and complexity of the circuit design. Designers frequently utilize third-party IC vendors' products which are not attested and dependable since their IC products may contain malwares, also termed as Trojans [20]. Trojans are serious concern as they may lead an IC to fail and may disclose sensitive information. Hence, early detection of Trojans particularly during gate-level netlist is of paramount importance. Machine learning algorithm can be implemented for efficient dynamic and static hardware Trojan detection [21]. Figure 8 shows the flow of learning at gate-level netlist.

Support vector machine (SVM)-based classification can be used as machine learning algorithm for the Trojan detection. This classification model classifies the gate-level nets as normal nets and nets with Trojans as 0/1 class. The foremost step while building up the model is to extract the features of Trojan nets based on the samples of nets that are affected with Trojans. In the next step, five features of Trojan nets are considered as five-dimensional vector, and many such vector samples are collected using SVM to rigorously train the classifier. The learned SVM classifier is tested and trained with set of unknown netlists until it starts predicting automatically the correct result with high accuracy.

6.7 Machine Learning in EDA and IC Design

Integrated circuit designs and electronic device automation are supported and offered advanced and high value tools and techniques by machine learning.

Figure 9 shows how machine learning and artificial intelligence provide design tools and technologies with newer and improved optimization objectives in the fields

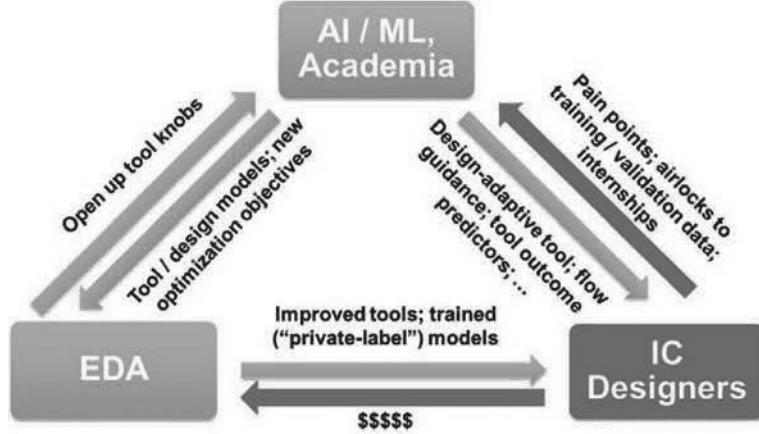


Fig. 9 Machine learning potential in EDA, IC design [22]

of EDA and also foster design adaptive tools, flow guidance, resource management, outcome predictor in the field of IC design.

7 Conclusion and Future Scope

This paper gives an insight on the machine learning methods such as supervised, semi-supervised and unsupervised learning. It also briefly discusses the most frequently and widely used machine learning algorithms that can be specifically used in various domains and focuses on the uses, advantages and drawbacks of these machine learning algorithms. This paper includes a brief description of the most recent research about the machine learning in VLSI computer-aided design. It covers the models and methodologies of machine learning algorithms being applied in the various steps of VLSI Design process such as RTL assertion generation, chip testing, static timing analysis, physical design routing and floor planning, and we learn across the paper that how machine learning automates the complex design, verification and implementation of integrated circuits without much of the human interference. It not only improves the quality of processes but also makes it efficient, accurate and reduces the human efforts.

Machine learning techniques offer great opportunities to EDA and IC design in the near future by opening up the possibilities of improved design convergence and higher levels of optimization. ML models are also expected to implement faster tools and accurate estimations in the respective applications [23, 24]. We aim to make an implementation of these algorithms and perform sets and simulations particularly in one of the steps of VLSI design flow.

References

1. M. Xue, C. Zhu, A study and application on machine learning of artificial intelligence, in *2009 International Joint Conference on Artificial Intelligence*, Hainan Island (2009), pp. 272–274. <https://doi.org/10.1109/IJCAI.2009.55>
2. A. Samuel, Some studies in machine learning using the game of checkers. *IBM J. Res. Dev.* **3**(3), 210–229 (1959). [CiteSeerX 10.1.1.368.2254](#)
3. M. Bansal, Priya, Application layer protocols for Internet of Healthcare Things (IoHT), in *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, Coimbatore (2020), pp. 369–376. <https://doi.org/10.1109/ICISC47916.2020.9171092>
4. M. Bansal, Priya, Performance comparison of MQTT and CoAP protocols in different simulation environments, in *Inventive Communication and Computational Technologies*, ed. by G. Ranganathan, J. Chen, A. Rocha. Lecture Notes in Networks and Systems, vol. 145 (Springer, Singapore), pp. 549–560. https://doi.org/10.1007/978-981-15-7345-3_47
5. A. Nayak, K. Dutta, Impacts of machine learning and artificial intelligence on mankind, in *2017 International Conference on Intelligent Computing and Control (I2C2)*, Coimbatore (2017)
6. S. Ray, A quick review of machine learning algorithms, in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad (2019), pp. 35–39
7. P. Sethi, V. Bhandari, B. Kohli, SMS spam detection and comparison of various machine learning algorithms, in *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, Gurgaon (2017), pp. 28–31
8. I. (Abe) M. Elfadel, D.S. Boning, X. Li (eds.), *Machine Learning in VLSI Computer-Aided Design* (Springer International Publishing, Springer Nature Switzerland AG, 2019)
9. R. Choudhary, H.K. Gianey, Comprehensive review on supervised machine learning algorithms, in *2017 International Conference on Machine Learning and Data Science (MLDS)*, Noida (2017), pp. 37–43
10. A.-N. Du, B.-X. Fang, Comparison of machine learning algorithms in Chinese web filtering, in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, vol. 4, Shanghai (2004), pp. 2526–2531
11. K.H. Yeap, H. Nisar, *Introductory Chapter: VLSI*. <https://doi.org/10.5772/intechopen.69188>
12. H.D. Foster, A.C. Krolik, D.J. Lacey, *Assertion-Based Design*, 2nd edn. (Springer Publishing)
13. S. Hertz, D. Sheridan, S. Vasudevan, Mining hardware assertions with guidance from staticanalysis. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **32**, 952–965 (2013)
14. J. Han, M. Kamber, *Data Mining: Concepts and Techniques* (Morgan Kaufmann Publishers Inc., San Francisco, 2000).
15. C. Visweswariah, K. Ravindran, K. Kalafala, S.G. Walker, S. Narayan, First-order incremental block-based statistical timing analysis, in *DAC*, San Diego, CA, June 2004, pp. 331–336
16. U. Gandhi, I. Bustany, W. Swartz, L. Behjat, A reinforcement learning-based framework for solving physical design routing problem in the absence of large test sets, in *2019 ACM/IEEE 1st Workshop on Machine Learning for CAD (MLCAD)*, Canmore, AB (2019), pp. 1–6
17. S. Mantik, G. Posser, W.-K. Chow, Y. Ding, W.-H. Liu, “ISPD 2018 initial detailed routing contest and benchmarks, in *Proceedings of the 2018 International Symposium on Physical Design, ISPD’18* (ACM, New York, NY, USA, 2018), pp. 140–143
18. D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. Van Den Driessche, T. Graepel, D. Hassabis, Mastering the game of Go without human knowledge. *Nature* **550**(7676), 354–359 (2017)
19. T.C. Chen, P.Y. Lee, T.C. Chen, Automatic floorplanning for AI SoCs, in *2020 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu (2020), pp. 1–2
20. S. Bian, M. Hiromoto, M. Shintani, T. Sato, LSTA: learning-based static timing analysis for high-dimensional correlated on-chip variations, in *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Austin, TX (2017), pp. 1–6

21. A.B. Kahng, M. Luo, S. Nath, SI for free: machine learning of interconnect coupling delay and transition effects, in *Proceedings of SLIP* (2015), pp. 1–8
22. K. Hasegawa, M. Oya, M. Yanagisawa, N. Togawa, Hardware trojans classification for gate-level netlists based on machine learning, in *2016 IEEE 22nd International Symposium on On-Line Testing and Robust System Design (IOLTS)*, Sant Feliu de Guixols (2016), pp. 203–206
23. M. Bansal, M. Nanda, M.N. Husain, Security and privacy aspects for Internet of Things (IoT), in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, pp. 199–204 (2021). <https://doi.org/10.1109/ICICT50816.2021.9358665>
24. M. Bansal, S. Garg, Internet of Things (IoT) based assistive devices, in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, Coimbatore, India, pp. 1006–1009 (2021). <https://doi.org/10.1109/ICICT50816.2021.9358662>