Journal of Theoretical and Applied Information Technology
15st June 2026. Vol.104. No.11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GAE-GCN: A DEEP GRAPH LEARNING MODEL FOR POWER PREDICTION IN CMOS VLSI CIRCUITS

PERIYASAMY K[1], G. Y. RAJAA VIKHRAM[2]

[1]Research Scholar. Department of Electronics and Instrumentation Engineering, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, Tamil Nadu, India.

[2]Assistant Professor. Department of Electronics and Instrumentation Engineering, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, Tamil Nadu, India.

E-mail: [1]periyasenth@gmail.com, [2]rajaaviy@srmist.edu.in

ABSTRACT

In very large-scale integration (VLSI) design, power consumption estimation has become something very crucial, having a direct effect on reliability, energy efficiency, management of thermal properties, and other performances at a higher level. With such complementary metal-oxide-semiconductor (CMOS) technology scaling and the complex nature of the integrated circuit, power prediction at an early stage is now of prime importance for design optimization of circuits. Everything with simulation was considered better in terms of power estimation, but it has always been too slow for design iterations. In view of resolving issues in current power estimation systems, this study introduces a novel power estimation scheme using Graph Autoencoder (GAE) combined with Graph Convolutional Network (GCN). Such a model exploits the graph nature of CMOS VLSI circuits, where logic gates and their interconnections are treated as nodes and edges, respectively. The GAE encodes the circuit graphs into low-dimensional latent structural features with preservation of both local and global dependency relationships, while the GCN learns these features to predict power consumption at the circuit level with high accuracy. Initially, gate-level attributes like gate types, flip-flops, inputs, and outputs were considered from the ISCAS'89 benchmark dataset for training and evaluating the model. Experimental results signify the excellent performance of the proposed GAE-GCN model as it has yielded a prediction accuracy with a regression coefficient of 0.9999, RMSE of 0.00010, and a correlation coefficient of 0.999. The results are compared with existing models outlined in the survey for validation purposes. The results comparison indicates that the developed GAE-GCN model outperformed all other models.

Keywords: *VLSI Design Circuits, Power Prediction, GAE, GCN, ISCAS'89 Dataset*

1. INTRODUCTION

The growing utilization of portable electronic devices, mobile phones, and other wireless devices necessitates breakthroughs in low-power technology. Currently, the CMOS logic circuit design is the preferred method for realizing low-power devices in most digital electronics. As logic and clock speeds escalate to satisfy emerging performance demands, the power supply voltage needs for such circuits are becoming a crucial consideration in the design of devices. Power consumption in digital circuits has emerged as a vital factor in high-speed and compact technological applications [1].

Low-power VLSI is a critical research domain for the contemporary electronics industry to investigate potential solutions for substantial leakage current reduction. The creation of low-power VLSI circuits improves the overall efficiency of battery-operated portable devices. CMOS logic is frequently utilized in integrated circuits due to its significant advantages over alternative logic families [2]. The primary advantages of CMOS logic include high noise immunity, minimal leakage power, simple fabrication, increased device density, minimal complexity, and full rail-to-rail outputs. CMOS logic consists of two types of power consumption: static and dynamic power consumption. The currents that occur during logic transitions are classified as dynamic current, which leads to dynamic power consumption. Static current circulates during the static logic states and contributes to the static power consumption. Dynamic power consumption constitutes a

significant portion of total power consumption when technology nodes exceed 100 nm [3] [4].

Table 1: Low Power Design Strategies for VLSI Circuits.

Levels	Strategy
Logic/Circuit level	Logic styles, transistor sizing, and energy recovery
Operating System Level	Partitioning and Power Down Mode
Technology Level	Threshold reduction, multi-threshold devices
Software Level	Regularity, locality, concurrency
Architecture Level	Pipelining, Redundancy, Data encoding

Diverse techniques for minimizing power consumption are possible at different phases of the VLSI design, as illustrated in Table 1. Employing diverse solutions at various phases of the VLSI design might yield efficient power management. Consequently, designers must implement a smart strategy to optimize power consumption in their designs [5].

Preliminary power consumption prediction during the design phase is essential for power management. The four primary methodologies for estimating power consumption are: learning-based techniques, simulation-based techniques, measurement-based techniques, and statistical-based techniques. Learning-based techniques are, by definition, variants of artificial intelligence (AI) technologies that provide autonomous power consumption estimation, derived and enhanced from experience with no explicit storage or programming. Typically, with these procedures, data sets, referred to as training sets, linking inputs to outputs must be gathered and utilized before starting learning processes [6]. Figure 1 presents an overview of the many categories of power consumption [7].

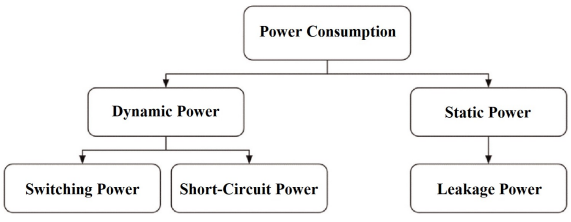


Figure 1: Different Categories of CMOS Circuit Power Consumption Modes

Machine learning provides new opportunities to improve VLSI design and electronic design automation (EDA). As technology advances, in accordance with Moore's law and the diminishment of transistor dimensions, the escalation in design complexity persists. The incorporation of learning methods has improved EDA tools, leading to significant advancements in diverse applications. These innovations have enhanced chip performance, facilitated power leak detection, optimized chip area, reduced power consumption, identified probable fabrication faults, and improved the accuracy of results [8]. Due to their efficacy in addressing difficult problems previously considered intractable, Machine Learning (ML)-based solutions are now prevalent in contemporary industry. The accessibility of high-speed processors and graphical processing units, which accelerate essential computations, is a pivotal component contributing to the growth in ML, particularly in deep learning (DL) [9].

Power analysis in CMOS VLSI circuits at the early design stage is one of the critical and challenging areas since it severely affects design optimization, thermal reliability, and energy efficiency. Conventional approaches, such as SPICE simulation and Monte Carlo-based methods, are accurate but computationally expensive and ineffective for faster design cycles. Another limitation of ML techniques, such as BPNN, ANFIS, or random forests, is that a complex structural dependency relates different subcomponents within a VLSI circuit. These methods treat circuit components in isolation without making use of the underlying graph topology. Apart from this, a huge gap exists in research about the employment of DL models for power estimation that exploit the graph nature of VLSI circuits. Also, very few current models include advanced graph-based architectures to learn local gate-level interactions and global circuit-wide dependencies simultaneously. The proposed research introduces a new paradigm, which integrates GAE for unsupervised structural feature extraction, with GCN for more accurate power prediction and thus acts as a scalable and data-efficient alternative to usual methods.

Integrating GAE and GCN constitutes a core concept of this research to incorporate structure-aware learning for accurate power estimation of CMOS VLSI circuits. The GAE provides essential functionality for extracting meaningful low-dimensional latent features from the high-dimensional circuit graphs, capturing both local node-level properties and global topological patterns without manual feature engineering. These compressed representations would retain critical

structure-level information of the circuit, such as connectivity, gate types, and I/O relationships. The latter is pooled with the GCN to perform the supervised task by means of message passing between nodes to model inter-gate dependencies and mapping functionalities governing their power behavior. Unlike traditional models, which model features as flat vectors, the GCN respects the spatial layout of the circuit, leading to much more informed and accurate predictions [10].

The research represents its novelty through the development of a structure-aware deep graph learning framework by combining the GAE and GCN models to predict the power consumption of CMOS VLSI circuits. The proposed framework models VLSI circuits as graph structures, which use nodes to represent logic gates and interconnections between them. The learning process can acquire knowledge about power behavior through two different types of dependencies, including local gate interactions and global topological dependencies. The GAE performs unsupervised extraction of hidden structural patterns from circuit graphs, while the GCN employs these learned patterns to predict power usage at the graph level. The research work provides new knowledge through its use of graph-based representation learning to develop CMOS power estimation methods, which have not been explored in existing VLSI power prediction studies. The work aims to develop scalable early power estimation techniques and evaluate CMOS benchmark circuits at their initial development stage while extending their evaluation capabilities.

The main objectives of this research include:

- To develop a graph-based representation model for CMOS VLSI circuits with gate-level structural information extracted from the ISCAS'89 benchmark dataset.
- To implement the GAE model for the unsupervised extraction of low-dimensional latent features that describe both local and global dependencies in circuits.
- To employ the GCN model for the final power prediction task through latent features learned from circuit graphs.
- To perform the evaluation phase of the GAE-GCN model on known performance measures such as RMSE, Regression Coefficient, and Correlation Coefficient (R).
- To compare the prediction performance of the proposed method against the existing machine-learning and simulation-based power estimation techniques.
- To demonstrate the scalability and robustness of the GAE-GCN framework with complex and diverse circuits.
- To discuss future directions with detailed advantages and limitations for the proposed research.

The research provides a specific contribution by focusing on estimating power requirements for initial stages of development through graph-based structural analysis of CMOS sequential benchmark circuits. The research work does not focus on transistor-level physical modeling, timing-aware dynamic switching analysis, post-layout optimization, and real-time hardware implementation. Indeed, this research contributes mainly toward improving prediction accuracy and structural learning capability in data-driven VLSI power estimation methodologies.

The subsequent sections of the work are organized as follows. Section 2 provides a concise examination of pertinent literature regarding the estimation of power in VLSI CMOS circuits, emphasizing contemporary models and synthesizing prior research. Section 3 delineates the developed framework for employing a prediction methodology utilizing GAE for extracting features and GCN for power estimation. Section 4 contains the analysis of results and comparison analysis, whilst Section 5 presents the conclusion of the research along with future suggestions.

2. LITERATURE REVIEW

This section reviews the recent models developed for the prediction and estimation of VLSI circuits' power consumption with various state-of-the-art techniques. Table 2 highlights the critical analysis of the reviewed models with their advantages & limitations. A Back-Propagation Neural Network (BPNN) and an Adaptive Neuro Fuzzy Inference System (ANFIS) model were proposed in [11] to accurately estimate power consumption in CMOS VLSI circuits, independent of circuit architecture and interconnections. The application of ANFIS for power estimation was relatively novel. Power estimate via ANFIS was conducted by developing initial FIS models by combining back-propagation (BP) and hybrid optimization procedures, employing both linear and constant methods. It was concluded that ANFIS, utilizing the hybrid

optimization strategy with the linear approach, yielded better outcomes.

A hybrid full adder utilizing CMOS logic, transmission gates, and pass transistor logic (PTL) was developed in [12]. The circuit was executed in both 45 nm and 90 nm technologies with Cadence tools. To evaluate the efficiency of the designs, sixteen current full adders were analyzed. Furthermore, the 1-bit full adder was expanded to considerable lengths to evaluate its performance on a big scale. The circuit design demonstrated superior performance at the 1-bit level and in its updated version compared to traditional full adders.

A method for predicting the consumption of power in CMOS VLSI circuits utilizing supervised learning was developed in [13]. The random forest (RF) model was not based on any predetermined empirical mathematical parameters or equations, unlike conventional approaches like the Simulation Program with Integrated Circuit Emphasis (SPICE) circuit modelling. RF generated precise outcomes without necessitating user focus on circuit architecture or connectivity.

An approach employing DL for fault identification was proposed in sequential and combinational circuits for stuck-at faults [14]. The unsupervised process of learning seeks to extract fundamental concepts from data with a deep sparse AE. The supervised approach delineated classification rules utilized to minimize characteristics for identifying various stuck-at defects in circuits. The model was evaluated on nine sequential and ten combinational circuits from the ISCAS'85 and ISCAS'89 benchmarks. In tests, Stacked Sparse AE recovered more resilient latent characteristics from the source database of sequential circuits, whereas Deep Sparse AE proved sufficient for combinational circuits.

A method for power estimation of CMOS VLSI circuits utilizing BPNN was proposed in [15]. The power estimation method utilizing BPNN was utilized for combinational and sequential circuits. The model was trained using 11 distinct training techniques. The BPNN was trained using five distinct qualities for circuits with combinational logic. The findings from the experiment indicated that the outcomes were nearer to the optimal levels, and the BPNN-based technique accurately calculated power.

A power estimation approach utilizing input patterns with predetermined statistical features was developed in [16] to examine the average power consumption of various intellectual property cores and buses/interconnects in system-on-chip design. The genetic algorithm was utilized to generate patterns of input signals throughout the power estimate process. The genetic algorithm simultaneously improved the input signal attributes that affect the outcome of the pattern. The Monte-Carlo zero-delay simulation was subsequently conducted for every individual bus and IP core at the highest level. The simple incorporation of these buses/cores enabled power prediction through a new macro modelling function.

An approach for fault identification utilizing an artificial neural network (ANN) was proposed in [17] to identify stuck-at faults in the 27-channel interrupt controller and circuits from the ISCAS'85 benchmarks. The ATALANTA program was employed in the technique to generate test vectors for every fault. The Stacked Sparse-AE (SSAE) was integrated with a softmax classifier as the last hidden layer to execute the supervised learning phase, utilizing the fault mask for every test pattern provided by the ATALANTA tool. The methodology was executed on 8 combinational circuits from ISCAS'85. The SSAE network achieved optimal accuracy for validation in a feature reduction of patterns of testing for 8 combinational circuits.

A system-level ASIC power assessment methodology utilizing machine learning called NeuPow was developed in [18]. The methodology delivered rapid and precise power estimations throughout the initial design phase. The methodology utilized Artificial Neural Networks (ANNs) to analyze the performance and behavior of a collection of arithmetic components frequently employed in applications of signal processing. The results demonstrated excellent accuracy in estimation with minimal relative error, regardless of the method used and the design's size or layers.

The research in [19] proposed an approach utilizing unate decomposition to implement a mixed static Domino circuit. An approach for dissecting a "Boolean circuit" into unate and binate sub-blocks was created. The K-Means clustering algorithm was devised to identify cluster centroids. Employing an Influence table methodology, the decomposition algorithm derived the maximal unate set, encompassing states capable of realizing a Domino block. Subsequently, the research identified an optimal segment of the unate set to be implemented with Domino logic, while the remainder was executed using static logic. The outcome exhibited an ideal decomposition, resulting in optimal overall circuit performance.

A method for power assessment of ISCAS'89 Benchmark circuits utilizing an ANN, specifically incorporating BPNN and Radial Basis Function Neural Network (RBFNN), was proposed in [20]. The method rapidly and accurately estimated power using the Inputs and Outputs and gate data of the VLSI circuit, without necessitating comprehensive structural details and interconnections of the circuit. The power estimation for the evaluated circuits was corroborated by regression analysis. The RBFNN demonstrated better performance and produced regression outcomes that closely approximate optimal values.

A machine-based circuit power estimating (ML-CPE) approach for power prediction of VLSI circuits was proposed in [21]. Initially, to determine testing with dissipation of power, it predicted the performance of the testing. Secondly, the variations in energy and movement data were associated with semiconductor design, highlighting minor issues. Various categories of ML algorithms were employed. The algorithms utilized testing as well as training datasets from the database. Essential features were derived from the ML model, resulting in the final anticipated energy as the output. The ML-CPE system, utilizing an ML model, demonstrated enhanced system efficiency.

A method for predicting power dissipation in integrated circuits through an ML methodology was implemented in [22], which was applicable in both post-layout and pre-layout stages. The research employed supervised models and algorithms, including Linear Regression, K-Nearest Neighbors, Support Vector Machines, and Random Forest, for power prediction. The average error was below 4% when comparing power estimates derived from ML with the Cadence EDA tool, indicating that Random Forest was an appropriate technique for power estimation in integrated circuits with a minimal error percentage.

A Stochastic Spider Monkey Optimizer-based High-Level Synthesis (SSMO-HLS) framework was proposed in [23] to enhance performance through improved runtime adaptability. In the SSMO-HLS model, behavioural input was gathered, and data flow evaluation was conducted to transform the input into a data flow diagram. The compilation process identified an error in the functional unit within the data flow diagram. SMO was employed to identify the best functional unit for executing binding, scheduling, and allocation processes. Post-binding, the output circuit was produced with enhanced runtime adaptability to process-voltage-temperature variations. The SSMO-HLS model's performance was evaluated using the ISCAS'89 dataset, demonstrating enhanced functional unit selection accuracy, reduced error rates, and decreased circuit adaptation time.

The study in [24] introduced AgileTransfer and AgileDevelop, effective ML-based methodologies for power model development. AgileDevelop reduced the overhead associated with developing models from the ground up through coverage-based sampling, which included both power space and signal space. AgileTransfer facilitated the transfer of current models to enhance design RTLs with small supplementary expenses by producing pseudo labels from source layout datasets and adjusting them with sampling ground truth power. The solutions substantially decreased the label creation bottleneck from three days to ten minutes, all while preserving good accuracy.

The literature review presents different viewpoints about the effective method to forecast CMOS VLSI power requirements despite advances in machine learning power estimation techniques. The traditional ML models like Random Forest, ANN, and ANFIS do perform well in prediction tasks, but researchers have found these models fail to model the intricate structural relationships that exist in circuit topologies. Current deep learning methods enhance representation capabilities; however, they fail to capture the spatial relationships that VLSI netlists establish through their graph structure. The research field requires ongoing investigation to establish a proper balance between prediction accuracy, system scalability, interpretability, and computational complexity. Therefore, this research proposed a method to estimate CMOS VLSI power consumption by combining graph-based representation learning with graph convolutional prediction techniques.

Table 2: Critical Analysis of Reviewed Current Models.

Ref	Models	Application	Advantages	Limitations
[11]	BPNN + ANFIS with hybrid optimization	CMOS VLSI power estimation independent of architecture	Accurate power estimation with novel ANFIS use; flexible across architectures.	Complexity in FIS model setup; limited generalization to unseen designs.
[12]	CMOS, TG, and PTL-based hybrid full adder	Low-power full adder design across 45 nm and 90 nm	High performance and efficiency in adder logic.	Focused on hardware design, not predictive modelling.
[13]	RF Regression	Power prediction in CMOS VLSI without structural constraints	No need for circuit interconnect details; accurate prediction.	Lacks explainability; performance dropped for unseen data patterns.
[14]	DL with Deep Sparse and Stacked Sparse AEs	Fault detection in sequential and combinational circuits (ISCAS'85 & ISCAS'89)	Robust latent feature extraction; effective fault classification.	Requires separate AE configurations for sequential vs. combinational circuits.
[15]	BPNN trained with 11 algorithms	Power estimation for combinational and sequential logic	Near-optimal estimation accuracy with BPNN flexibility.	Dependent on the choice of training strategy, possible overfitting.
[16]	Genetic Algorithm-based Input Pattern Generator + Monte Carlo	Average power estimation in SoC IPs and buses	Incorporates statistical properties of inputs; scalable to system-level.	Time-consuming simulations; limited real-time applicability.
[17]	ANN + SSAE + Softmax Classifier	Fault identification using ISCAS'85 circuits	High validation accuracy; effective feature reduction.	Limited to stuck-at faults; dataset-specific tuning required.
[18]	NeuPow (ANN-based)	ASIC power estimation in early design phases	Fast and accurate predictions; architecture-independent.	Focused on arithmetic components; limited scalability to full SoC.
[19]	K-Means + Unate Decomposition + Influence Table	Domino logic implementation in static CMOS designs	Efficient Boolean partitioning; optimized performance.	Complex decomposition logic; limited to logic transformation.
[20]	BPNN + RBFNN	Power estimation using IOs and gate info	Accurate and rapid estimation; RBFNN showed better regression.	Ignores deep structural features; suffers from a generalization issue.
[21]	ML-CPE with multiple ML models	Power prediction from energy/motion data and test patterns	Effective feature selection; robust ML integration.	Lacks detail on model architecture; not scaled with large designs.
[22]	ML models (LR, KNN, SVM, RF)	Pre- and post-layout power prediction	High accuracy (<4% error); suitable for layout-aware estimation.	Limited interpretability; requires full datasets for training.
[23]	SSMO-HLS	HLS circuit binding, scheduling, allocation	Adaptive to PVT variations; fast and accurate synthesis.	Complex optimization procedure; resource-intensive.
[24]	AgileDevelop + AgileTransfer	Fast power model development and transfer learning	Reduces labelling time (3 days → 10 mins); accurate with low cost.	Struggled with high variation in target designs.

2.1 Problem Statement

The existing methods for CMOS VLSI power estimation through ML and deep learning methods face major issues because they lack proper structural awareness, scalability, and prediction generalization. The traditional simulation methods including SPICE and Monte Carlo techniques provide dependable results; however, their computational demands and execution durations make them impractical for early-stage design optimization. Conventional ML models such as ANN, ANFIS, Random Forest, and SVM primarily utilized manual numerical features and generally fail to capture the complex topological dependencies between interconnected circuit components. The recent deep learning techniques enhance their ability to learn features but still use circuit data processing methods that consider data as separate feature vectors instead of using graph-based structures. The research field still lacks a

scalable graph-based deep learning framework, which can simultaneously learn local gate interactions and global circuit relationships to predict power usage in CMOS VLSI circuits during their early development stage.

This research hypothesizes that representing CMOS VLSI circuits as graph structures and integrating GAE with GCN models can significantly improve power prediction accuracy compared with conventional ML and simulation-based approaches.

3. PROPOSED ESTIMATION MODEL

This research proposes a novel graph-based DL framework to predict the power in CMOS VLSI circuits. The workflow diagram of the model illustrated in Figure 2 represents the end-to-end pipeline for power estimation in CMOS VLSI circuits using GAE and GCN models, using the ISCAS'89 dataset as a case. First, the ISCAS'89 dataset consists of benchmark circuits with gate-level information, which includes logic gate types, flip-flops, counts of input/output, and Monte Carlo simulated power values. These circuit descriptions pass through data preprocessing and normalization, during which the circuits are converted into graph representations. In this graph, logic gates constitute nodes, and the interconnections between them (derived from the netlist) form edges. The features at the node level are then extracted from gate type (one-hot encoded) to different structural characteristics, along with normalization values to standardize their scaling across the entire dataset.

Following the preprocessing, the dataset was split into the training and test datasets so that supervised learning and validation of the proposed model could occur. The dataset is partitioned into an 80:20 ratio for training and evaluating the developed model. The training set is then passed to the GAE-based feature extraction stage, during which the encoder part of the GAE uses the GCN layers to encode each gate (node) into a low-dimensional latent vector that captures the local properties of the gate and the global structure of the circuit. These latent features are important for dimensionality reduction while preserving meaningful structural relationships in the graph. These latent features are then given as inputs to a prediction model using GCN layers. The model further processes the graph using graph convolutional layers to learn hierarchical dependencies and outputs a continuous value representing the estimate of power consumption of the entire circuit. Thus, GCN utilization allows the model to comprehensively capture the complex forms of interactions and correlations among circuit components, which are very much needed for power estimations of any credibility. Eventually, the model output is analyzed for its performance, where the predicted values of power are tested against the original set of values from the dataset. With the computation of evaluation metrics, the performance of the proposed model in comparison with other models can be validated for accuracy and robustness.

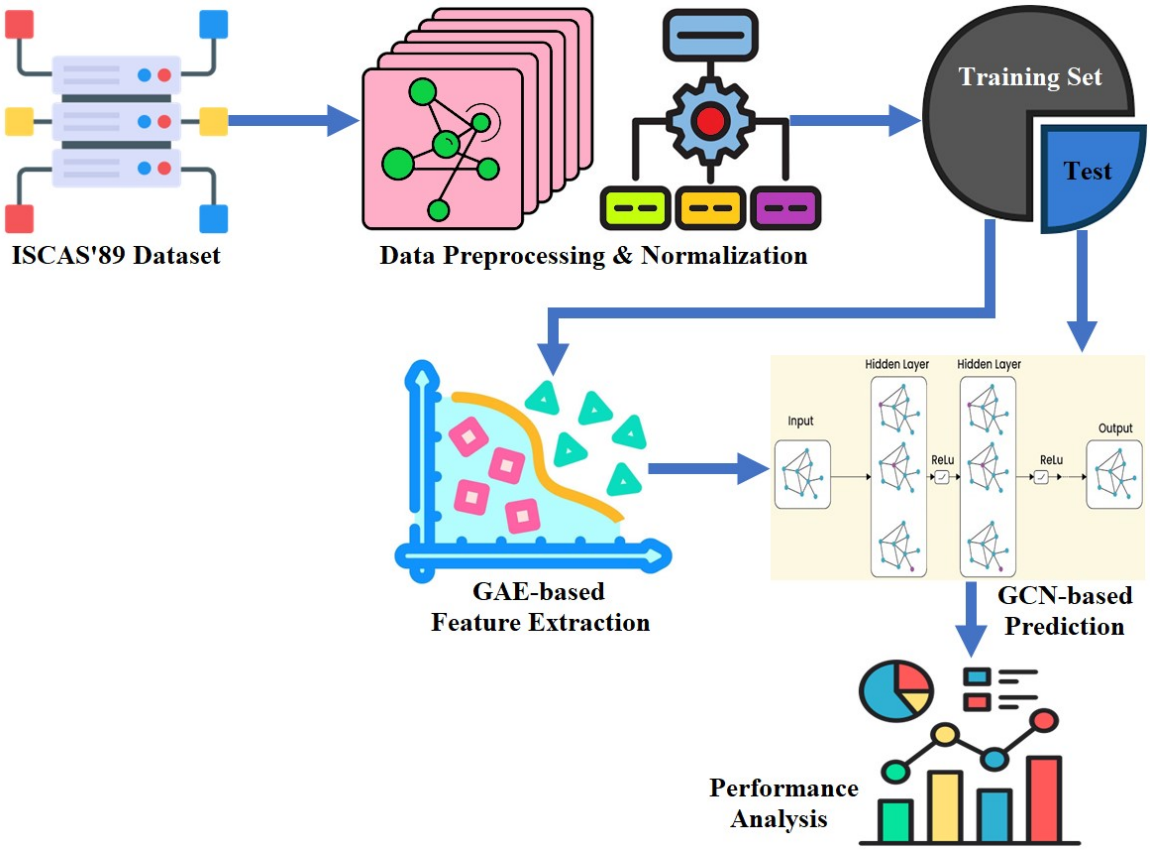


Figure 2: Workflow of the Research Model

This entire workflow ensures a complete and efficient pipeline for power estimation of the VLSI circuits with graph-based DL techniques. To perform power prediction for CMOS VLSI circuits through the use of GAE integrated with GCN is the core novel idea, a comparatively less-mapped yet very pivotal area of study in VLSI design automation. The traditional methods of power estimation either utilize simulation tools like SPICE, which are tedious and include ineffective computational capabilities, or shallow machine learning models that do not recognize the deep structural interdependencies within a circuit. Other learning-based models also consider the components of a circuit as independent units, disregarding the graph-based nature of VLSI layouts.

The proposed method develops a novel graph-based framework for the representation of VLSI circuits as graphs, where gates and their interconnections are transformed into nodes and edges. It can be stated that the novelty emerges from the two-stage technique of learning: first, the GAE learns these features in an unsupervised way, capturing rich, latent structural information from the graphs to preserve, in other words, the local structural as well as global information; second, the resultant compressed embeddings are fed into a supervised GCN that consequently trains to predict circuit-level power consumptions. Hence, it is a novel integration whereby GAE retains good predictive feature extraction, independent of the task, and GCN conducts task-specific regression, thus improving generalization and accuracy.

The combination of GAE and GCN in this study brings meaningful improvements to model capabilities and practical applicability in VLSI design:

- Structure-Aware Learning: Since graphs can be used to represent circuit structure, the model is aware of spatial and logical structure, which is essential for power prediction.
- Dimensionality Reduction with Feature Preservation: GAE projects high-dimensional node features into meaningful embeddings to reduce computational cost, but still preserves useful patterns for prediction.

- Better Generalization: This unsupervised manner of training the GAE allows the characteristics of circuits to be captured generally, and so it will be good on designs that are previously unseen or slightly varied.
- Effective Prediction: After training, the prediction of power is performed with the GCN by using graph-level labels instead of performing a detailed simulation, so that power prediction can be performed in the early design stages.
- Scalable and Flexible: This model can scale among circuits of different sizes and complexities and can be modified slightly to perform related tasks such as delay estimation, fault analysis, or thermal prediction.

3.1 Dataset Details

The ISCAS'89 benchmark circuit dataset was employed to train and assess the developed GAE-GCN model, as illustrated in the following tables. The GAE-GCN model was trained on 20 sequential benchmark circuits, with 6 circuits designated for testing. These sets include features pertinent to sequential circuits, including the total number of outputs (OUT), inputs (IN), total gates, D flip-flops (DFF), inverters (INV), NAND, NOR, AND, and OR gates. This dataset is collected from the Kaggle repository. The process entailed monitoring power consumption for a designated input set utilizing SIS, a significant tool for creating and idealizing sequential circuits. This tool employs zero latency, makes distributions uniformly, samples for node switching, and functions at a clock frequency of 20 MHz. The findings of the dataset were calculated using the Monte Carlo method [11] [22].

Table 3: Dataset Training Samples.

Circuits	IN	Gate	NOR	AND	INV	NAND	DFF	OR	OUT	MC Simulation Power (mw)
S38584	12	11448	1185	5516	7805	2126	1452	2621	278	1.87987
S38417	28	8709	2279	4154	13470	2050	1636	226	106	1.14518
S35932	35	12204	0	4032	3861	7020	1728	1152	320	1.22048
S15850	14	3448	151	1619	6324	968	597	710	87	0.51991
S13207	31	2573	98	1114	5378	849	669	512	121	0.35400
S9234	19	2027	113	955	3570	528	228	431	22	0.28004
S5378	35	1004	765	0	1775	0	179	239	49	0.23357
S1494	8	558	0	354	89	0	6	204	19	0.06018
S1423	17	490	92	197	167	64	74	137	5	0.07181
S1238	14	428	57	134	80	125	18	112	14	0.06347
S953	16	311	112	49	84	114	29	36	23	0.02458
S838	34	288	70	105	158	57	32	56	1	0.01292
S820	18	256	66	76	33	54	5	60	19	0.02831
S713	35	139	0	94	254	28	19	17	23	0.03743
S444	3	119	34	13	62	58	21	14	6	0.01172
S420	18	160	34	49	78	29	16	28	1	0.00903
S400	3	106	34	11	58	36	21	25	6	0.01065
S349	9	104	31	44	57	19	15	10	11	0.01856
S298	3	75	19	31	44	9	14	16	6	0.00912
S208	10	66	16	21	38	15	8	14	1	0.00698

Table 4: Dataset Test Samples.

Circuits	IN	Gate	AND	NOR	INV	NAND	DFF	OR	OUT
S27	4	8	1	4	2	1	3	2	1
S344	9	101	44	30	59	18	15	9	11
S386	7	118	83	0	41	0	6	35	7
S382	3	99	1	34	59	30	21	24	6
S641	35	107	90	0	272	4	19	13	24
S1488	8	550	350	0	103	0	6	200	19

Tables 3 and 4 represent the data employed for training and evaluation. The selected set of standard sequential circuits for both training and evaluation in this research was determined by the quality and accessibility of the ISCAS'89 data set, a recognized standard for assessing VLSI power prediction models. These circuits encompass a wide variety of dimensions and intricacies, offering a solid basis for training and evaluation. Although the dataset is relatively small in size, its quality is crucial for attaining high accuracy in prediction. The ISCAS'89 circuits are meticulously structured and encompass accurate netlists and gate-level representations, guaranteeing dependable and uniform extraction of features for model training [25]. Furthermore, the application of the GAE for the extraction of features improves the model's generalization capacity by effectively collecting latent patterns from the inputs, even with limited data.

3.2 Data Preprocessing

In this research, each VLSI circuit from the ISCAS'89 benchmark dataset is treated as a graph G = (V, E), where V stands for the set of nodes representing the logic elements of the circuit (such as AND, OR, NAND, NOR, INV gates, and D flip-flops), and E stands for the set of edges denoting interconnections or signal flow between these logic elements based on the corresponding circuit netlist. This graph-based representation fully captures structural and functional dependencies in digital circuit layouts that need to be learned as contextual relations affecting power consumption. For each node $v_i \in V$, a feature vector gets generated with gate-level information considered pertinent. The feature vector contains gate type (one-hot encoded), input, and output counts. Those are features that enter the GAE, giving the encoder the facility of learning compact, high-level representations of each gate, with an injected notion of topology carried by the adjacency matrix. This manner of structured graph formulation guarantees that the downstream GCN is properly exploiting all local and global interactions of the circuit towards estimating power consumption accurately.

Normalization of data is a pre-processing technique employed to scale or convert data, guaranteeing every characteristic contributes equally to the research. Normalizing the data adjusts the values of various aspects to a uniform scale, hence avoiding any specific feature from emphasizing the evaluation due to its huge scale. This normalization approach enhances the results and convergence of the developed model by ensuring every feature's impact is equilibrated and significantly contributes to the overall study. The significance of data normalization in developing precise predictions has been applied across the developed model. Data normalization is essential for preparing input data for modelling, as it reduces the effects of varying sizes and feature distribution.

This research employs a Z-score normalization method to enhance the predictive accuracy of power usage. The normalization method aims to preprocess the data that is input data and improve the performance and efficacy of the GCN model. This normalization approach aims to accommodate attributes of the data and customize the input features to align more effectively with the GCN architecture. Z-score standardization, or normalization, is a prevalent technique for data normalization in predictive modelling and statistical research. It entails normalizing the data to ensure the standard deviation (STD) equals one and the mean of the feature equals zero. This procedure normalizes the data to a mean of zero and adjusts it to possess a STD of one.

The equation for z-score normalization of a feature x' is expressed as follows in (1):

$$x' = x - \mu(x)/\sigma(x) \quad (1)$$

Here, x represents the real value, x' denotes the standardized value, $\mu(x)$ signifies the mean of the features x, and $\sigma(x)$ indicates the STD of x. A notable advantage of z-score normalization is its resilience to outliers in comparison to alternative normalization techniques. Outliers are anomalous values that markedly deviate from the predominant data points and could bias the dataset's statistical features [26].

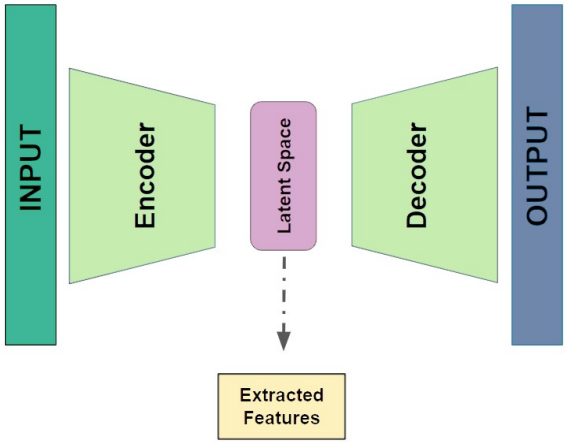
3.3 Stacked Auto Encoder Model

The method used here extracts the key features of node features through a GAE. This GAE undertakes dimensionality reduction of node-level features, preceding their input into GCN layers. Given raw node features (e.g., gate type, fan-in/fan-out, etc.), the GAE model generates a latent embedding for each node capturing structural and semantic information. The GAE model compresses high-dimensional sparse features into meaningful representations. Using the embeddings built with structural information, the GCN-based regression model could better generalize from a dataset like ISCAS'89.

Figure 3: Architecture of GAE

Figure 3 depicts the architecture of a GAE, which has been utilized in this work to effectively extract

features from CMOS VLSI circuit data. The input to the GAE is a circuit graph where nodes represent logic gates and the edges represent interconnections. The encoder, which is generally a combination of GCN layers, aggregates neighborhood information and learns to represent each node with meaningful low-dimensional embeddings. These embeddings are then stored in the latent space, which captures the structural and functional patterns implicitly encoded in the circuit graphs. From this latent space, extracted features can be isolated and handed over to downstream predictive models like a GCN regressor for power estimation. That is, the decoder reconstructs the original input structure (e.g., adjacency matrix or node features) to enforce encoded features to retain enough graph information. Thus, such an end-to-end unsupervised learning mechanism enables the model to condense the abstract and complex circuits into extremely small yet meaningful feature representations that promote the generalizability and accuracy of power prediction for VLSI design [27].



Typically, GAE comprises two components: the encoder and the decoder. Initially, for a graph G = (V, E, X) where V and E represent the sets of vertices and edges, respectively, the encoder allocates a latent vector z_i to every node $v_i \in V$. This research typically chooses GCN as the encoder because of its computational efficiency and exceptional network modeling capabilities. The GCN can be expressed as given in the following equation (2).

$$Z = \hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}XW + b \quad (2)$$

Here, Z represents the embedding matrix, $\hat{A} = A + I$ signifies the matrix of adjacency augmented with self-loops, $D_{ii} = \sum_{j=0}\hat{A}_{ij}$ denotes the diagonal degree matrices, X is the node's feature matrix, and W and b are variables to be optimized. From a node-wise perspective, the layer of convolution could be regarded as the accumulation of data from adjacent nodes. The decoder seeks to reconstruct A using the encoder's output Z, with the fundamental decoding process relying on the inner result of latent interpretations. For both nodes v_i, $v_j \in V$, a larger inner product z_j indicates a greater likelihood of a connection between nodes v_i and v_j in the graph. The matrix of adjacency of the reconstruction graph 'A', derived from A, could be expressed as $\hat{A} = \sigma(Z^T Z)$, and the process of reconstruction illustrates the underlying concept of the AE: If the matrix of adjacency can be accurately reproduced by the hidden vectors, the acquired hidden vectors must retain adequate data from the initial graph [28].

3.4 GCN Modelling

In contrast to conventional Neural Networks, the application of Graph Neural Networks (GNNs) for predicting the power at nodes in CMOS VLSI circuits offers a theoretical benefit through the interchange of data among interconnected nodes. In conventional Neural Networks, samples of input are regarded as individual entities. To the stated application and dataset involving CMOS VLSI circuits, treating individual logic gates or circuit modules as separate and independent entities would be an oversimplification. Essentially, power consumption in a VLSI circuit is a function of diverse structural and functional features, which are inherently correlated amongst the interconnected components. On one hand, we have logic gate elements (AND, OR, NAND, etc.) and flip-flops that have correlations of a topological nature; in the switching activity of one logic gate and in the fan-in and fan-out relationships of the adjacent gates, load sharing is an important parameter and affects those neighboring gates through signal propagation. On the other hand, in terms of functional dependencies, blocks have shared timing paths, control signals, or synchronized clock domains, such that gating or activity in one domain influences gating or activity in another, in the same logic cone. Hence, since GCNs can capture such complicated interdependencies characterizing inter-node relationships along with structural dependencies in circuit graphs, these networks have been applied for a finer, more context-aware prediction of power consumption in CMOS VLSI circuits. Figure 4 depicts the GCN architecture implemented in this research [29].

GNNs function on a graph $G = (E, V)$, where edges and V signifies the nodes. The features of the

Journal of Theoretical and Applied Information Technology
15st June 2026. Vol.104. No.11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

node v are depicted as a feature vector. The collection of every node is depicted as the feature matrix of dimensions $d \times n$, where n denotes the total count of nodes and d signifies the feature dimensions. The edges are depicted as a matrix of adjacency A of dimensions $n \times n$, where $A_{u,v} = 1$ indicates that nodes u and v are related, whereas $A_{u,v} = 0$ signifies otherwise. In the weighted graph $A_{u,v}$ is defined as the weight corresponding to the edge between the nodes u and v.

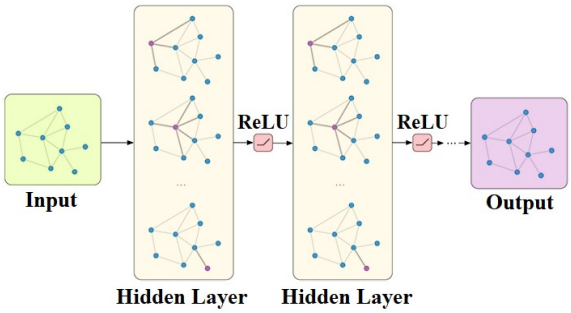


Figure 4: Architecture of GCN

In response to the major achievement of Convolutional Neural Networks (CNNs), GCN has been designed to extend the convolutions to represent graph data. The GCN method employs spatial graph convolutions, similar to ordinary convolutions, whereby every node v in every layer collects the features of its neighboring nodes. Like conventional CNNs that learn filters, GCNs also combine neighbouring features utilizing learnable weighted matrices. As all the nodes consolidate data from their immediate neighbors in every layer, the size of the receptive field of the network expands by one for every successive layer. The subsequent illustration of fundamental spatial graph convolution has been described as follows in equation (3):

$$h_u^{(l+1)} = \sigma\left(W_1 h_u^{(l)} + W_2 \sum_{v \in N(u)} h_v^{(l)}\right) \quad (3)$$

Here, $h_u^{(l)} \in R^{d^{(l)}}$ signifies the d-dimensional representation of a node u in layer l, while $N(u)$ delineates the neighborhood of node u. The variables W_1 and W_2 denote shared learnable weighted matrices, while σ represents the activation function, including ReLU. In the initial layer, h_u represents the features of the input x_u of node u. In the graph convolutions, all the neighbours are assigned an identical weight during neighbourhood aggregation. Utilizing the matrix of adjacency and the syntax $H^{(l)} = \left(h_1^{(l)}, \ldots, h_n^{(l)}\right)$, equation (3) could be expressed in the form of a matrix as a product with the preceding adjacency matrix A:

$$H^{(l+1)} = \sigma\left(W_1 H^{(l)} + W_2 H^{(l)} A\right) \quad (4)$$

Observe that A in equation (4) is devoid of self-loops, indicating that the diagonal elements are assigned a value of zero. The target node's features are modified utilizing a matrix W_1. Standard graph convolutions could be augmented with attention techniques to ascertain specific weights for every neighbour. The updated node representations for node i modify equation (3) as follows:

$$h_u^{(l+1)} = \sigma\left(W_1 h_u^{(l)} + \sum_{v \in N(u)} \alpha_{u,v} W_2 h_v^{(l)}\right) \quad (5)$$

Unlike the conventional convolution presented in equation (3), each neighbor v was assigned a weight determined by an attention coefficient $\alpha_{u,v}$ as in (5), calculated specifically for every node pair. The coefficient, indicating the significance of one node relative to another, was calculated as follows:

$$\alpha_{u,v} = softmax\left(\frac{(W_3 h_u)^T W_4 h_v}{\sqrt{d_l}}\right) \quad (6)$$

In equation (6), d_l denotes the quantity of outcomes for the hidden layers l, representing the hidden dimensions, and $\sum_{v \in V} \alpha_{u,v} = 1$. While α often vary for every node and its neighbors, the weights utilized to calculate α are consistent across all nodes. To incorporate edge features into the attention mechanisms, the equation was modified as follows:

$$h_u^{(l+1)} = \sigma\left(W_1 h_u^{(l)} + \sum_{v \in N(u)} \alpha_{u,v} \left(W_2 h_v^{(l)} + W_5 e_{u,v}\right)\right) \quad (7)$$

Here in equation (7), $e_{u,v}$ represents the weight linked to the edge connecting node u and node v, while W_5 refers to the learnable weight matrix utilized for the computation of α, incorporating edge weights as given in the following equation (8).

$$\alpha_{u,v} = softmax\left(\frac{(W_3 h_u)^T (W_4 h_v W_5 e_{u,v})}{\sqrt{d_l}}\right) \quad (8)$$

Multi-head attention could be employed to discern various relationships among nodes within a single layer. The attention mechanism was employed repeatedly, with each instance acquiring its distinct weights. The resultant feature maps, referred to as attention heads, are subsequently concatenated. The concept resembles the generation of numerous feature maps in conventional CNNs. Multiple convolutional layers stacking constructs a GCN that generates a graph with novel node representations, applicable for specialized predictive tasks like node classification or regression. Concerning this, the node embeddings

could be utilized directly through the application of an appropriate activation function, or they could go through further transformation via an appropriate readout network. A typical network comprises a series of completely connected layers that individually change every node [30]. The pseudocode for the developed GAE-GCN model power prediction task is given in the following.

Table 5: Hyperparameter Details of GAE-GCN.

Hyperparameters	Value/Description
Encoder Layers	2
Hidden Units	[64, 32]
Latent Dimension	16
Decoder Type	Inner Product
Batch Size	1 graph
Dropout	0.2
Input Features	Latent Features from GAE (16)
GCN Layers	2
Hidden Units	[64, 32]
Activation Function	ReLU
Optimizer	Adam
Learning Rate	0.001
Epochs	300

Algorithm: GAE-GCN Model

Initialization
LOAD Dataset:
- *Load ISCAS'89 benchmark dataset*
- *Split into Training Set and Test Set*

PREPROCESS Data:
- *For each circuit:*
  Construct Graph G (V, E) where:
    V = Nodes representing gates (AND, OR, NAND, etc.)
    E = Edges representing netlist connections
  Assign feature vector to each node (e.g., gate type, fan-in/fan-out)
  Normalize features

FEATURE EXTRACTION using Graph Autoencoder:
- *Initialize GAE with:*
  - *Encoder: Graph Convolutional Layers*
  - *Latent Space: Low-dimensional embeddings*
  - *Decoder: Inner product decoder to reconstruct graph structure*
- *Train GAE on training graphs using reconstruction loss*
- *Extract Latent Features (Z) from the encoder for each node*

POWER PREDICTION using GCN:
- *Input: Latent features (Z) + graph structure*
- *Define GCN with multiple graph convolutional layers*
- *Add global pooling/readout to aggregate node features into graph-level embedding*
- *Add dense layers for regression output*
- *Train GCN to minimize prediction loss*

EVALUATE Model:
- *Predict power values for test circuits using trained GCN*

OUTPUT Results:
- *Predicted power results*

END

The proposed research framework for power estimation in CMOS VLSI circuits introduces the organized pipeline of data preprocessing, feature extraction, and regression modeling. The graph representation of the circuit from the ISCAS'89 dataset consists of nodes representing logic gates and edges for netlist connections. GAE thus extracts meaningful latent features from the graph after encoding such features with a two-layer GCN and decoding using an inner product. These latent features are then leveraged by a separate GCN regression model predicting circuit-level power. It consists of two graph convolutional layers, followed by global pooling and a fully connected regression output layer. As mentioned in Table 4, both GAE and GCN are trained using Adam with a 0.001 learning rate and dropout for regularization. Other associated hyperparameters, number of hidden units [64, 32], latent dimension 16, and number of epochs 300 are chosen, balancing model complexity and generalization ability. The model can yield precise power prediction for CMOS VLSI designs.

4. EXPERIMENTATION ANALYSIS

4.1 Experimental Setup

In this work, a deep learning model based on GAE-GCN was proposed to predict the power consumption in CMOS VLSI circuits. The model used graph-based representations of the circuit structure, whereby the input parameters comprise key circuit features such as the count of inputs and outputs, logic gates (AND, OR, NAND, NOR, Inverter), and D Flip-flops. For these parameters and structures, the ISCAS'89 benchmark circuit was used. The dataset was divided into training and testing sets using an 80:20 split ratio. These parameters (nodes) and connections between these nodes (edges) serve as the input to the GAE, which thereby extracts the critical features, which then become the input to a GCN for power predictions.

The simulation and implementation of the proposed method were carried out using Python, utilizing popular deep learning libraries like PyTorch and PyTorch Geometric. The experimental settings are running on a laptop with an Intel i7 processor with 16GB RAM, which has enough computational ability to digest graph-based datasets

and perform model training along with evaluation. This model provides smooth processing of the preprocessing, model training, and performance analysis procedures for the ISCAS'89 circuits, allowing for power consumption estimation that is reliable and consistent.

4.2 Performance Metrics

The GAE-GCN model proposed for power estimation in CMOS VLSI circuits is quantitatively evaluated in terms of three principal metrics: Percentage Error (Error %), Root Mean Square Error (RMSE), and Correlation Coefficient (R). These three indicators establish how far off the expected power values are from the actual values in terms of accuracy, precision, and correlation, respectively. The actual measurement of power is done through Monte Carlo simulation on the ISCAS'89 benchmark circuits [11].

The percentage error shows how much the actual power values (obtained through simulation) differ from their predicted values (estimated through the GAE-GCN model). This is given by:

$$Err\% = \frac{Act\ Value - Est\ Value}{Est\ Value} \times 100 \quad (9)$$

Here, the Actual Value is parametric to power consumption by Monte Carlo simulation, whereas the Estimated Value would have been the prediction given by the GAE-GCN.

The RMSE is computed to check the model's overall prediction accuracy, measuring the average magnitude of errors in predicting, hence reflecting how much predicted values fit actual values:

$$RMSE = \sqrt{\frac{\sum_{I=1}^{N}(Y_I^O - Y_I^C)^2}{N}} \quad (10)$$

Where Y_I^O is the actual value of power obtained from simulation, Y_I^C is the predicted value; and N is the number of data points considered for the study. The smaller the value of RMSE, the better the accuracy, with a value of zero indicating perfect prediction.

Another factor used to track the exactness of the analysis is the correlation coefficient; it shows the strength and direction of the linear relationship. It is calculated by the following expression:

$$R = \frac{\sum_{I=1}^{N}\left(Y_I^O - Y^O\right)\left(Y_I^C - Y^C\right)}{\sqrt{\sum_{I=1}^{N}(Y_I^O - Y^O)^2 \sum_{I=1}^{N}(Y_I^C - Y^C)^2}} \quad (11)$$

Where Y^O and Y^C are the mean values of the observed and predicted values, respectively. A strong positive correlation (with an R close to 1) would imply that the model's predictions were perfectly aligned with the 'actual' trend in the data [21].

The assessment metrics provide different evaluation results, which demonstrate the successful functioning of the developed power prediction model. The model utilized RMSE to determine the average prediction errors and evaluates the accuracy of numerical estimation. The regression coefficient measures how well the model predicts actual outcomes while the correlation coefficient measures the effectiveness of the linear relationship between actual and predicted values. Percentage Error measures the prediction errors between actual circuit performance and predicted results. The research uses these metrics as standard measurements to create benchmarks that allow equitable assessments of current methodologies.

4.3 Results & Discussion

The results and discussion section discusses the experimental results for the proposed GAE-GCN power estimation model for CMOS VLSI circuits based on the ISCAS'89 benchmark dataset. It compares the predicted power values against the actual values obtained through Monte Carlo simulations for the evaluation of predictive accuracy of the proposed model. Several performance parameters, such as Error%, Root Mean Square Error (RMSE), and Correlation Coefficient (R), were considered for testing the model's efficiency. The test results establish that the model is able to methodically encapsulate the structural and functional dependencies that exist within the circuits, thereby confirming the proposed method's robustness and generalizability in power prediction.

Table 6 presents a comprehensive result of predicted power values with relative accuracies, error percentages, and corresponding benchmark CMOS VLSI circuits. The model's ability to provide an accurate estimation of power at its early stage, a critical factor for VLSI design in terms of energy optimization and circuit reliability, is fully emphasized. The actual power values in the table are Monte Carlo simulation results that are treated as the ground truth for comparison. Predicted power is obtained by means of the GAE-GCN approach, which models circuits using graphs to reflect high-level topological and functional dependencies among components such as logic gates and flip-flops. The Error % was derived based on the relative difference of actual versus predicted values, thus providing an insight into the level of accuracy of the prediction method. From S27, one of the simplest ISCAS'89 benchmark circuits, the

prediction is a power value of 0.05530 mW as against the actual 0.05536 mW, thus registering a minimal error of 0.11%. This minor error is a testament to the fact that the model can generalize for basic circuits, hence capturing low-level structural behavior.

Table 6: Predicted and Error Results of GAE-GCN Model.

Test Circuit	Real Power Value (mW)	Predicted Power Value (mW)	Error in %
S27	0.05536	0.05530	0.11
S344	0.01846	0.01849	0.16
S382	0.01048	0.01045	0.29
S386	0.01620	0.01624	0.25
S641	0.03629	0.03635	0.17
S1488	0.07624	0.07618	0.08

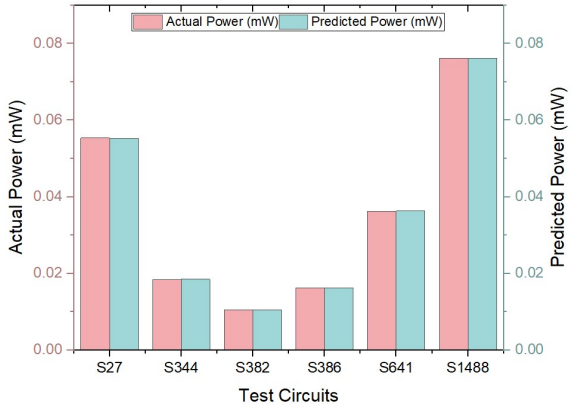


Figure 5: Graphical Chart of Actual and Predicted Power

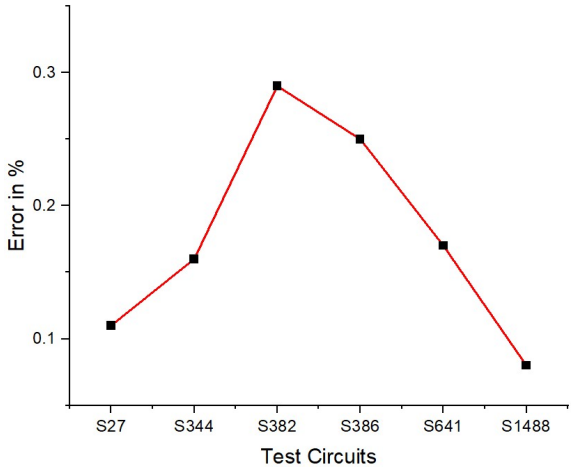


Figure 6: Graphical Chart of Error %

For S344, the power was computed to be 0.01846 mW and predictions made to be 0.01849 mW, yielding an error of 0.16%. This is, however, with a slight increase in circuit complexity than S27; still, high prediction fidelity is maintained by the model. This performance insinuates that the graph embeddings learned through the GAE capture not only gate-level characteristics but also the spatial configuration and interaction between components.

For S382, whose actual power is 0.01048 mW and predicted value is slightly less at 0.01045 mW, the error of 0.29% is the largest for the circuits listed. Even then, this is a very small magnitude, again highlighting the sturdiness of the model. The increase in deviation might be indicative of the circuit's behavior sensitively changing due to its unique structure or differences in actual switch activities that were not adequately captured during feature extraction. However, the prediction error below 0.3% is by all means acceptable in a design-stage prediction.

For S386, the value of power is 0.01620 mW, while the predicted value is 0.01624 mW with an error of 0.25%. In this case, since the predicted value is very close to the actual value, the GCN component of the model very likely learns and transfers information along the graph structure by considering the dependency relations between different logic blocks in a contextually relevant manner.

As for S641, power was measured, and the power predicted at 0.03629 mW and 0.03635 mW, respectively, leading to a 0.17% error. This shows that the model is scalable and can handle larger and more interconnected circuit topologies. The low error means that the GAE-GCN pipeline is able to generalize across levels of complexity, maintaining precision and consistency.

For S1488, the model attains the best prediction performance, with actual power at 0.07624 mW and predicted power at 0.07618 mW, having an error of 0.08%, the lowest. Such high accuracy for a complex circuit demonstrates that the model extracts meaningful latent features via GAE and then effectively models spatial relations via GCN. Hence, GAE-GCN also accommodates hierarchical structures and long-range dependencies that prevail in such large-scale circuits.

In general, the error rates for all six circuits range from 0.08% to 0.29%, which is a testament to the model's excellent prediction capability. The low errors substantiate not only the choice of GAE for feature extraction and GCN for relational learning, but also the application of such a model to practical deployment in early VLSI design flows, besides testifying to the model's power to generalize well, from small- to large-sized circuits, and to maintain robustness against a varying diversity of structural patterns. Figures 5 and 6 depict the graphical illustrations for predicted power and error % results.

Table 7: Comparison of GAE-GCN Results with Current Models.

Model	Regression Coefficient	RMSE	R
ANFIS [11]	0.9940	0.00020	0.999
BPNN [11]	0.9993	0.00044	0.846
SSAE [14]	NA	NA	0.998
BPNN [15]	NA	NA	0.992
NeuPow [18]	NA	NA	0.990
RFBNN [20]	NA	NA	0.998
ANN [21]	0.9082	0.00065	0.907
KNN [23]	NA	0.00043	0.857
LR [23]	NA	0.00013	0.899
RF [23]	NA	0.00010	0.998
SVM [23]	NA	0.00011	0.986
AWELM [31]	0.9890	0.00057	0.988
Monte Carlo [32]	0.753	0.00068	0.846
Proposed GAE-GCN	0.9999	0.00010	0.999

Table 7 presents a comparative assessment of the proposed GAE-GCN model against a spectrum of current and previously established machine-learning and soft computing methods that have been used as power estimation models for CMOS VLSI circuits. The comparison is based upon three apparently critical performance parameters, namely, Regression Coefficient, Root Mean Square Error (RMSE), and Correlation Coefficient (R), each parameter relating to a different perspective of model validity and prediction reliability. Models enlisted in this table are extracted from recent and relevant literature; this facilitates strong benchmarking for the model under consideration.

ANFIS correlates well with R = 0.999 and a regression coefficient of 0.9940, but has a greater RMSE (0.00020) than that of the GAE-GCN model. Although ANFIS excels in modeling nonlinear relationships, it is not scalable and exhibits structural unawareness that outperforms that of a GCN. BPNN shows a regression coefficient of 0.9993 with a slightly lower value of R, which is 0.846. On the contrary, while it is quite close on average, it will be rather weak in following the actual trends consistently. This further highlights the lack of robustness in capturing the circuit behavior. The SSAE, BPNN, NeuPow, and RFBNN models do not furnish Regression Coefficients or RMSEs, but present R values from 0.990 to 0.998. The deep learning models of SSAE and RFBNN exhibit fine goodness-of-fit values with R = 0.998, indicating that they capture deeper patterns. However, without RMSE or regression metrics, their comparative precision remains uncertain.

The ANN-type power estimation model for ASIC NeuPow gives an R value of 0.990, which places it at good but not the best performance. ANN exhibits far inferior performance with a regression coefficient = 0.9082, RMSE = 0.00065, and R = 0.907, showing the inability of simple networks to capture structural relationships without specialized learning architecture. AWELM, adaptive extreme learning machine, performs with R = 0.988 and RMSE = 0.00057 but falls regressive to GAE-GCN in accuracy. KNN, LR, RF, and SVM-based ML models present mixed results: RF and SVM perform well with RMSE values of 0.00010 and 0.00011, respectively, and R values of 0.998 and 0.986, respectively; however, none of these approaches is able to exploit structural dependencies within circuits.

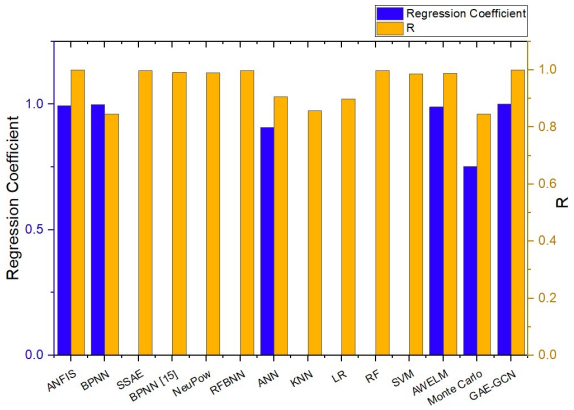


Figure 7: Graphical Chart for Comparison of Coefficient Results

By showing average RMSEs (0.00013 and 0.00043) and moderate R values (0.899 and 0.857), these models are showing weaker alignment with actual power trends, especially for more complex circuits. As a classical simulation-based method, Monte Carlo attains an R value of 0.846 with a relatively high RMSE of 0.00068 and has the lowest regression coefficient (0.7530). It is reliable

but time-expensive and less accurate than data-driven methods, thus unsuitable for fast predictions in the early stages. The GCN-GAE model offers better regression and correlation, almost close to one, and an exceedingly low RMSE, better than all other existing methods, indicating that the model can learn well about both local and global conditionalities in circuit structures via graph-based learning. GAE performs feature extraction, while GCN offers prediction, thereby providing an architectural advantage considered missing from traditional models. Figures 7 and 8 depict the graphical illustrations for comparison of coefficient results and RMSE results.

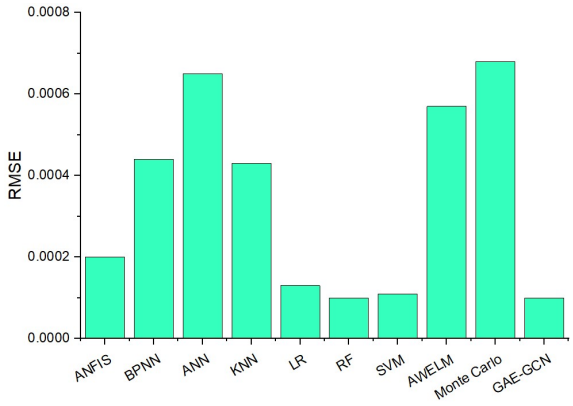


Figure 8: Graphical Chart of RMSE Results

The comparative results demonstrate that the developed GAE-GCN method provides better performance than current models regarding power estimation. It strikes a balance among creating high regression accuracy (0.9999), strong correlation (R = 0.999), and minimum prediction errors (RMSE of 0.00010), better than traditional ML models and other advanced Neural models. Graph-based architecture learning lets the model understand and exploit structural dependencies in the VLSI circuits, which most other models do not. That uniquely makes the proposed method not only reliable and precise but also scalable enough to work in integrated early-stage EDA workflows where speedy and accurate power prediction is needed for design optimization. The comparative analysis further demonstrates that the proposed GAE-GCN framework differs fundamentally from existing studies in terms of structural learning capability and representation methodology. The proposed framework achieves better prediction accuracy than existing models while implementing a scalable structure-aware method, which will be suitable for future intelligent EDA applications. The results highlight that both research objectives and the novelty of the graph-based CMOS power estimation method have been properly validated.

4.4 Advantages & Limitations

The advantages of the research include the following:

- The graph representation facilitates an accurate modeling of circuit topology and captures local and global dependencies between gates and interconnections.
- The proposed GAE-GCN model shows superiority in performance metrics (Regression Coefficient = 0.9999, RMSE = 0.00010, R = 0.999), indicating near-perfect conversion between predicted values and actual simulation results.
- The GAE learns latent features efficiently in a compressed way while keeping them informative, and helps improve the learning and reduce the input dimensionality of the circuit graphs.
- It generally works well for circuits of all sizes and complexities and, therefore, can be used for small-scale and large-scale VLSI designs.
- Estimations from this model will bring much faster decisions at early design stages, substituting Monte Carlo simulation or expensive SPICE analysis.
- The GAE-GCN framework can also be minimally adapted for other VLSI tasks such as delay estimation, fault detection, or thermal analysis.

The limitations of the research include the following:

- Accuracy in predictions is indeed highly dependent on an accurate and fully granular graph representation generated from the netlist. Improper modeling may really reduce performance.
- The ISCAS'89 dataset includes only a few circuits, which in turn restricts the range and intensity of training sets.
- The training of GNNs, especially with the autoencoder variants, is much more memory-intensive and time-consuming compared to simpler alternatives, although inference is quick.

- Static features of the circuit constitute the focus of the current model, whereas dynamic factors such as signal timing or temporal switching activity are not considered.
- Deep learning-based graph models, including GCNs, are, in most cases, black box models, making it hard to discern which particular features mostly affect predictions.
- The proposed model is evaluated only on benchmark sequential circuits from the ISCAS'89 dataset, and therefore, its performance on modern industrial-scale heterogeneous VLSI designs requires additional performance testing.
- The model succeeds in accurate predictions, but the interpretability of graph embeddings and their physical circuit behavior links need further development to assist practical hardware debugging work.

5. CONCLUSION

This research proposed a novel and accurate early-stage power estimation model that integrated the GAE method with the GCN method. The model introduced an organized pipeline of data preprocessing, feature extraction, and regression modeling. The circuit netlists from the ISCAS'89 dataset undergo graph modeling in this approach while exploiting structural dependencies between logic elements, allowing an efficient capture of local and global levels of gate characteristics. The GAE performs feature compression, whereas the power consumption is predicted from these embeddings by the GCN. Tested upon the ISCAS'89 benchmark dataset, the proposed model outperforms the current machine learning and simulation methods by achieving a regression coefficient of 0.9999, RMSE of 0.00010, and R coefficient of 0.999. The results strongly establish the claimed robustness and scalability of the approach, making it appropriate for successful adoption at the early design stages where fast and effective power estimations are important for circuit performance and energy efficiency co-optimization. The experimental findings validate the proposed research hypothesis that graph-based representation learning using integrated GAE-GCN architecture can effectively improve early-stage power prediction accuracy for CMOS VLSI circuits. In future, the work will include addressing dynamic power estimation by adding temporal switching activity and features of the clock domain to the model. The same framework can be used for associated tasks like delay and thermal estimation. Challenges would consist of extending the model for larger real industrial circuits and creating an implementation that integrates smoothly with EDA tools.

REFERENCES:

[1] A. Bhattacharjee, T. Majumdar, and S. Bhownik, A Low Powers Adiabatic Approach for Scaled VLSI Circuit, Journal of VLSI Circuit and Systems, Vol. 6, No. 1, 2024, pp. 1-6. https://doi.org/10.31838/jvcs/06.01.01

[2] A. Bhardwaj, D. Dash, A. Anand, V.M. Bhanusali, A. Kushawah, I. Mishra, and C.K. Panday, Comprehensive Analysis on Complementary FETs, IEEE Access, Vol. 13, 2025, pp. 82554-82572. https://doi.org/10.1109/ACCESS.2025.3568134

[3] V.K. Sharma, A survey of leakages reduction techniques in CMOS digital circuits for nano scale regime, Australian Journals of Electrical and Electronics Engineering, Vol. 18, No. 4, 2021, pp. 217-236. https://doi.org/10.1080/1448837X.2021.1966957

[4] K. Kalpana, S. Ramya, and A. Purusothaman, Designs and implementations of circuits optimizations in VLSI: A review, AG Volumes, 2023, pp. 73-84.

[5] R.D. Kumar, M.R. Prabu, K.L. Priya, M. Renugha, K.S. Kumar, and V. Venisa, An Investigation of Low Powers VLSI Designs Technique, Journal of Sciences & Computing and Engineering Research, Vol. 6, 2023, pp. 5-9. https://doi.org/10.46379/jscer.2023.060402

[6] M. Richa, J.C. Prévatet, M. Dardailon, M. Maroué, and A.E. Samahat, High-levels power estimations technique in embedded system hardware: an overview, The Journal of Supercomputing, Vol. 79, No. 4, 2023, pp. 3771-3790. https://doi.org/10.1007/s11227-022-04798-5

[7] L.L. Ng, K.H. Yep, M.W.C. Goh, and V. Dakullagi, Power consumptions in CMOS circuit, In Electromagnetic Field in Advancing Science and Technology, 2022, pp. 1-17. http://dx.doi.org/10.5772/intechopen.105717

[8] Y. Attaoui, M. Chentauf, Z.E.A.A. Ismail, and A. Elmoarabit, Machine Learning in VLSI Designs: A Comprehensive Review, Journal of

Integrated Circuits and Systems, Vol. 19, No. 2, 2024, pp. 1-14. https://doi.org/10.29292/jics.v19i2.826

[9] G. Thakur, S. Jain, and H. Sohail, Current issue and emerging technique for VLSI testing: A review, Measurement: Sensors, 24, 2022, 100497. https://doi.org/10.1016/j.measen.2022.100497

[10] Y. Li, C. Xue, F. Zaragari, and Y.R. Liu, From graphs theory to graphs neural network (GNN): The opportunities of GNN in power electronics, IEEE Access, Vol. 11, 2023, pp. 145067-145084. https://doi.org/10.1109/ACCESS.2023.3345795

[11] G. Vellingiri and R. Jayabalan, Adaptive neuro fuzzy inference system-based powers estimation method for CMOS VLSI circuits, International Journals of Electronics, Vol. 105, No. 3, 2018, pp. 398-411. https://doi.org/10.1080/00207217.2017.1357763

[12] T. Rajagopal and A. Chakhrapani, A novel high-performances hybrid full adders for VLSI circuit, Circuits, System, and Signal Processing, Vol. 40, No. 11, 2021, pp. 5718-5732. https://doi.org/10.1007/s00034-021-01725-6

[13] G.R. Thippesamy, R. Jayadurgha, and S.K. Sharma, AI-based Machine Learning Algorithm in VLSI Technology Application, ICTACT Journals on Microelectronics, Vol. 8, No. 4, 2023, pp. 1462-1465. https://doi.org/10.21917/ijme.2023.0252

[14] M. Moness, L. Gabber, A.I. Husein, and H.M. Alli, Automated designs errors debugging of digital VLSI circuit, Journal of Electronic Testing, Vol. 38, No. 4, 2022, pp. 395-417. https://doi.org/10.1007/s10836-022-06020-z

[15] V. Govindaraj, B.S. Babhu, and C. Ezhilazhaghan, Back propagations neural network-based powers estimation method for CMOS VLSI circuits, In AIP Conference Proceedings, Vol. 2385, No. 1, 2022, pp. 1-6. https://doi.org/10.1063/5.0070733

[16] Y.A. Durrani and T. Riesago, Efficient powers analysis approach and its applications to systems-on-chips designs, Microprocessors and Microsystems, Vol. 46, 2016, pp. 11-20. https://doi.org/10.1016/j.micpro.2016.09.003

[17] L. Gaber, A.I. Husein, and M. Moneas, Faults detections based on deep learning for digital VLSI circuit, Procedia Computer Science, Vol. 194, 2021, pp. 122-131. https://doi.org/10.1016/j.procs.2021.10.065

[18] Y. Nasser, C. Sua, J.C. Prévottet, T. Fani, F. Palombo, M. Héllard, and L. Rafo, NeuPow: A CAD methodology for high-levels powers estimations based on machine learning, ACM Transaction on Design Automations of Electronic System (TODAES), Vol. 25, No. 5, 2020, pp. 1-29. https://doi.org/10.1145/338814

[19] V. Yadav and S.K. Yadhav, Optimizations of Power Consumptions in CMOS VLSI Circuits Using Different Cluster, Test Engineering and Management, Vol. 83, 2020, pp. 26149–26155.

[20] P. Ramanathan, B. Surendhiran, and P.T. Vanati, Power estimations of benchmarks circuit using artificial neural network, Pensee, Vol. 75, No. 9, 2013, pp. 427-433.

[21] E. Poovannan and S. Karthik, Power Prediction of VLSI Circuits Using Machines Learning, Computer, Materials & Continua, Vol. 74, No. 1, 2023, pp. 2161-2177. https://doi.org/10.32604/cmc.2023.032512

[22] E. Poovannan and S. Karthik, Pre Syntheses and Post Syntheses Powers Estimation of VLSI Circuits Using Machines Learning Approach, Applied Artificial Intelligence, Vol. 36, No. 1, 2022, pp. 2145640. https://doi.org/10.1080/08839514.2022.2145640

[23] M.T. Rani and S.S. Kumar, Spiders monkey optimizations–based high-levels synthesis in VLSI circuit for runtimes adaptability, Concurrency and Computation: Practices and Experiences, Vol. 33, No. 3, 2021, pp. e5280. https://doi.org/10.1002/cpe.5280

[24] Y. Lu, Q. Zheng, and Z. Xei, Unleashing Flexibility of ML-based Powers Estimator Through Efficient Developments Strategies, In Proceeding of the 29th ACM/IEEE International Symposiums on Low Power Electronics and Designs, 2024, pp. 1-6. https://doi.org/10.1145/3665314.3670823

[25] P. Mohanraj, S. Paramashivam, and P. Sathyamurthy, A power trace-based hardware trojans detections using deep artificial neural networks, Analog Integrated Circuit and Signals Processing, Vol. 123, No. 3, 2025. https://doi.org/10.1007/s10470-025-02351-x

[26] Y.S. Kim, M.K. Kimm, N. Fui, J. Lui, J. Wan, and J. Serebric, Investigating the impacts of data normalizations method on predicting electricity consumptions in a building using different artificial neural networks model, Sustainable Cities and Society, Vol. 118, 2025,

105570. https://doi.org/10.1016/j.scs.2024.105570

[27] H. Yu, T. Aii, M. Yang, W. Hung, and L. Harie, A graph autoencoders network to measure the geometrics similarity of drainage network in scaling transformations, International Journal of Digital Earth, Vol. 16, No. 1, 2023, pp. 1828-1852. https://doi.org/10.1080/17538947.2023.2212920

[28] Y. Ye, H. Zha, T. Ciu, R. Yun, L. Zheng, and H. Xing, University Evaluations through Graduates Employments Predictions: An Influence based Graph Autoencoders Approach, IEEE Transaction on Knowledge and Data Engineering, Vol. 36, No. 11, 2024, pp. 7255-7267. https://doi.org/10.1109/TKDE.2024.3402234

[29] A. Sadasivan, K. Gananatan, J.D. Paul Nesamoni Rose Marry, and S. Balasubhramanian, A Systematic Survey of Graphs Convolutional Network for Artificial Intelligences Application, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, Vol. 15, No. 2, 2025, pp. e70012. https://doi.org/10.1002/widm.70012

[30] D. Beinert, C. Holzhütter, J.M. Thomas, and S. Voget, Power flows forecast at transmissions grids node using graphs neural network, Energy and AI, Vol. 14, 2023, 100262. https://doi.org/10.1016/j.egyai.2023.100262

[31] T. Kuntavai and A. Jeevanandham, Adaptive wavelet ELMs-fuzzy inference system-based soft computing model for powers estimation in sustainable CMOS VLSI circuits, Soft Computing, Vol. 24, 2020, pp. 11755-11768. https://doi.org/10.1007/s00500-019-04636-6

[32] N.E. Evmorfopoulos, G.I. Stamoullis, and J.N. Avarisiotis, A Monte Carlo approach for maximum powers estimation based on extreme value theory, IEEE Transactions on Computers-Aided Design of Integrated Circuits and System, 21(4), 2002, pp. 415-432. https://doi.org/10.1109/43.992765