A Heterogeneous Graph Framework for ML Applications in Electric Circuits

Ahmed K. Khamis
ECE Dept., University at Albany, SUNY
Albany NY, USA

Mohammed Agamy
ECE Dept., University at Albany, SUNY
Albany NY, USA

IECON 2024 - 50th Annual Conference of the IEEE Industrial Electronics Society | 978-1-6654-6454-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/IECON55916.2024.10905665

***Abstract*—Machine learning (ML) has the potential to revolutionize the design and optimization of electronic systems, but mainly requires high-quality and diverse datasets that capture the features and performance of various electrical circuits. Previously, there were no generic method applicable to any electric circuit whether continuous circuit like resonant LC circuits or switching circuits like buck, boost and buck-boost that allows circuit connections and components separate representation when circuits are fed to ML models. Moreover, all other methods and models are inefficiently built to process either limited complexity circuits, i.e. limited to a certain circuit order, or use a fixed circuit topology so that the generated dataset is of a definite size. In this paper, a novel circuit representation that is based on heterogeneous graph is presented, which is a graphical representation of the energy flow in a physical system including electrical, mechanical and even chemical interactions. Moreover, heterogeneous graph neural network (Hetero-GNN) is applied to the proposed representation of heterogeneous graphs representing electric circuits, allowing for the different ML tasks on different electric circuit and converter applications.**

I. Introduction

Many research works have addressed circuit representation for ML application in the form of a graph and with the use of GNNs. [1]–[5] represented circuits as graphs that are arbitrarily generated from circuit components and connections, where every component is represented as node, while edges represent connections between these components. A complete breakdown of the representation technique, feature assignment, graph representation and model assessment is in [6]. Since the structure of the circuit graph is arbitrary and doesn't represent the actual circuit connection or configuration, the ML model process is only trained for a single circuit connection, does not consider the effect of circuit components or the physical arrangement of the circuit internal connections. Circuit structure based predictions was first introduced in [7]–[9], and was further expanded to include continuous and switching circuits in CCM and DCM in[6]. This mapping offered a diverse and comprehensive guidelines to transform the structural behavior of electric circuit based on the bond graph circuit representation [10]–[13]. Furthermore, the scalability and usability of the graph framework was analysed and tested in [14], while showing a case study of three phase inverter performance prediction. However, this structural representation lacked the physical representation of energy exchange between electric component interaction between circuit elements and circuit node. In this paper, a novel circuit representation technique allowing for the decoupling of the physical interactions as well the circuit configuration of electric circuit is proposed. This is an outcome of representing bond graphs as heterogeneous graph, which preserves the physical and structural meanings of circuit components. The following are the main contributions of this paper: 1) Heterogeneous graph representation of continuous and switching electric circuits by decoupling the physical and structural properties of circuits, allowing independent feature processing that enables the model to focus on specific aspects of the circuit representation. 2) A dataset generation algorithm that converts circuit netlist to an optimized hetero-graph representation, as well as feature assignment framework for assigning node and edge features the captures the physical properties of electric circuits. 3) Application of the proposed Hetero-GNN model in various ML tasks on continuous and switching circuits, including differentiation between three circuits of the same order, components and connections and circuit dynamics prediction based on component variations. The proposed graph representation is tested on a classification task, where the target is to classify three circuits of the same components and component values but different component order. The extreme similarity between the three circuits represent a challenge for the classifier if it cannot differentiate between the structural and energy exchange domain of circuits. Additionally, a regression case study implemented on a buck converter highlights the usability of proposed framework for advanced ML applications.

II. Circuits to Machine Learning Problem

Mapping circuit structure, performance and operating conditions into a formulation where Machine Learning algorithms can be used is completed in several steps: 1) Transforming the circuit structure to a heterogeneous graph representation, 2) Assigning graph features, and then 3) Applying Hetero-GNN model tailored to the target task.

A. Heterogeneous vs Homogeneous Circuit Graphs

Homogeneous graphs are graphs where nodes and edges belong to the same type or category, and this uniformity in node and edge types simplifies the graph structure and allows for consistent processing and analysis. On the other hand, heterogeneous graphs, are graphs where nodes and edges can belong to different types or categories, which allows nodes and edges to have diverse attributes and relationships, reflecting the complex and varied nature of real-world networks and systems if represented as heterogeneous graphs. Fig. 1 shows an RLC

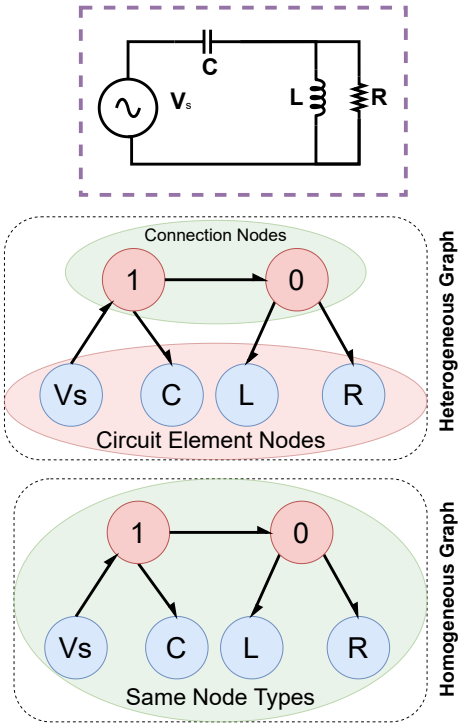
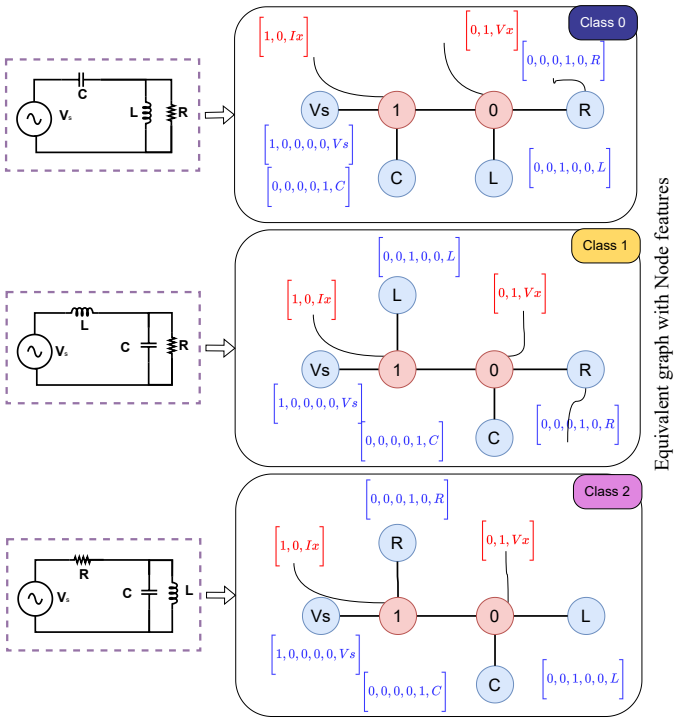


Fig. 1 – Homogeneous vs heterogeneous graph representation of an RLC circuit

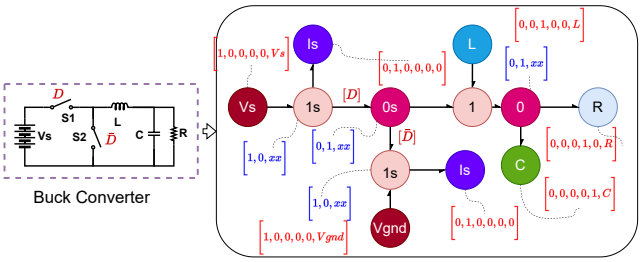
circuit in homogeneous and heterogeneous representation. In the heterogeneous representation, junction and circuit elements nodes are separated by their node type, while in homogeneous representation nodes have the same node type.

1) Homogeneous Circuit Graphs

The methodology provided in [6]–[9], [15] represents the electric circuits as homogeneous graphs, i.e graphs including the same type of nodes, where the components of the circuit and junctions representing connections are modelled as nodes, while connections between components and junctions are edges. The homo-graph representation captures the physical relationships as well as connectivity between circuit node, providing a structured format for machine learning analysis. Also, a unified node feature assignment algorithm that assigns features to each node in the graph. This algorithm is designed to be scalable and topology-generic, with inherent capability to handle circuits of varying sizes and complexities. However, the proposed homogeneous graph representation lacked the separation between the physical and structural representation of electric circuit, which leads the developed ML models to struggle in capturing the intricate relationships between physical components and structural connections with the physical behavior of electric circuit. Moreover, in graph generation tasks, the model may produce graphs that lack coherence between physical components and their connections, affecting the realism of the generated output.



(a)



(b)

Fig. 2 – Heterogeneous graph representation of: a) Three continuous RLC circuits, b) Buck converter

2) Hetero-Graph Representation

Mapping physical electric circuits represented as bond graphs onto a heterogeneous graph for further processing by a heterogeneous Graph Neural Network (GNN) involves a systematic translation of the circuit elements and junctions into nodes, and the bonds into edges. Fig. 2 shows RLC circuits in addition to buck converter as examples of continuous and switching circuits in hetero-graph representation. As for node Representation, circuit elements (V, I, L, C, R) and junctions (0 , 1) are represented as two distinct node types. Each node type is characterized by specific features. Circuit Element (CE) nodes are encoded with a one-hot vector representing the type of circuit element, accompanied by the analog value of the component. The one-hot encoding is a binary vector with a high (1) bit on the corresponding element and all the other elements are set to low (0), ensuring that each element type is uniquely identifiable. Furthermore, junction (J) nodes are assigned a one-hot encoding indicating whether the junction is a zero (common voltage) or one (common current) junction,

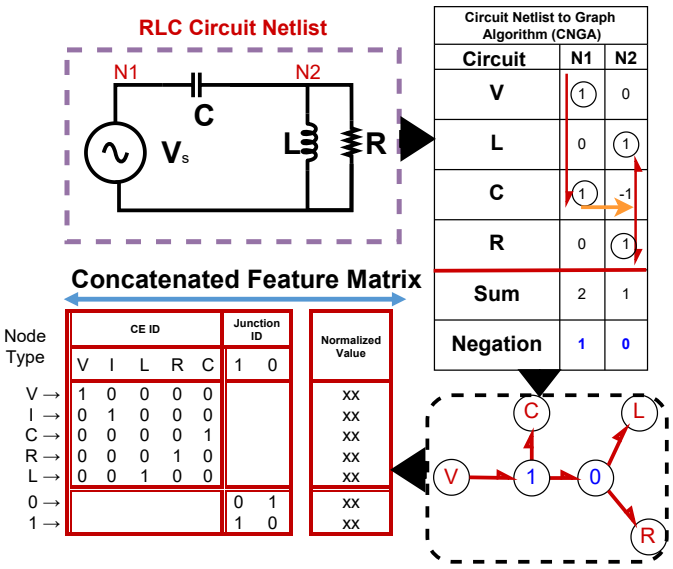


Fig. 3 – CNGA flowchart with feature assignment matrix

cascaded with the analog value of the current or voltage. Node features are included by introducing the concept of **Element ID**. By identifying circuit elements or junctions as graph nodes, an algorithm assigns the appropriate **one-hot encoding** ID out of five circuit nodes (voltage source, current source, Inductance, Capacitance, Resistance) or the two junction nodes (zero or one junction). Then, element ID and a normalized one run time simulation of circuit are concatenated to from the feature matrix of the whole graph with dimension $N \times d_{in}$, where N is number of CE or Junction nodes and d_{in} is the dimension of feature vector. Regarding the edge representation, edges in the heterogeneous graph represent the energy exchange between circuit elements and junctions, as well as the connections between junctions themselves. Energy Exchange Edges between circuit elements and junctions carry features representing voltage and current values that signify the magnitude of energy exchange. Likewise, connection/wire edges symbolize the physical connections or wires in the circuit, and are characterized by features of voltage, current, and duty cycle. A full feature analysis is performed for circuit assigned as "class 1" is shown in Table I, where 'xx' values represent features that are concatenated with the one-hot encoding.

*B. Circuit Netlist to Graph Algorithm (**CNGA**)*

The CNGA algorithm is a novel dataset generation algorithm that converts circuit netlist to an optimized bond graph. It can capture the features and performance of various electrical circuits in a scalable and general way. If the circuit has switch connected to ground, a voltage source of value zero is added to one terminal of the switch. Fig. 3 shows an example of obtaining a graph from an RLC circuit netlist of class 0 indicated in Fig. 2 when fed to CNGA. The algorithm works as follows:

1) The algorithm takes the circuit netlist as input, which is a description of the components and connections of the circuit. The netlist contains the names, values, and nodes of the components, such as resistors, capacitors, inductors, sources, etc. The nodes are numbered from N_0 to N_M, where 0 is the ground node and M is the total number of nodes in the circuit.
2) The algorithm assigns 1 and -1 values at the corresponding circuit nodes that the components are connected to. The ground node 0 is ignored for an optimized graph output. For example, voltage source (V) is connected to node N1 and ground, so N1 field is assigned 1 while the ground net is ignored.
3) The algorithm sums the values at each net, and if the sum is even number, it is considered 0 and if odd number it is 1. and then takes the binary negation of the sum. The binary negation output represents the type of junction to which the node is assigned.
4) The algorithm outputs an optimized bond graph by merging every consecutive 1 or 0 nodes. The bond graph consists of bond graph junction which the CNGA extracted and the circuit elements, which are connected to these junctions by directed edges that represent the power flow

Table I – Features per node and edge type

Node	Node Type	Features
V	Element	[1, 0, 0, 0, 0, Vx]
1	Junction	[0, 1, xx]
L	Element	[0, 0, 0, 0, 1, Lx]
0	Junction	[1, 0, xx]
C	Element	[0, 0, 0, 1, 0, Cx]
R	Element	[0, 0, 1, 0, 0, Rx]
Edge	**Edge Type**	**Features**
V→1	CE $\xrightarrow{\text{Energy Ex.}}$ J	[D]
1→L	J $\xrightarrow{\text{Energy Ex.}}$ CE	[D]
1→0	J $\xrightarrow{\text{Wire}}$ J	[D]
0→C	J $\xrightarrow{\text{Energy Ex.}}$ CE	[D]
0→R	J $\xrightarrow{\text{Energy Ex.}}$ CE	[D]

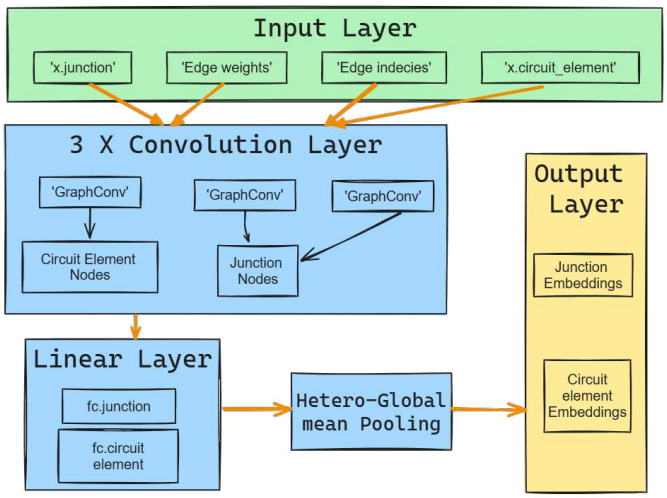
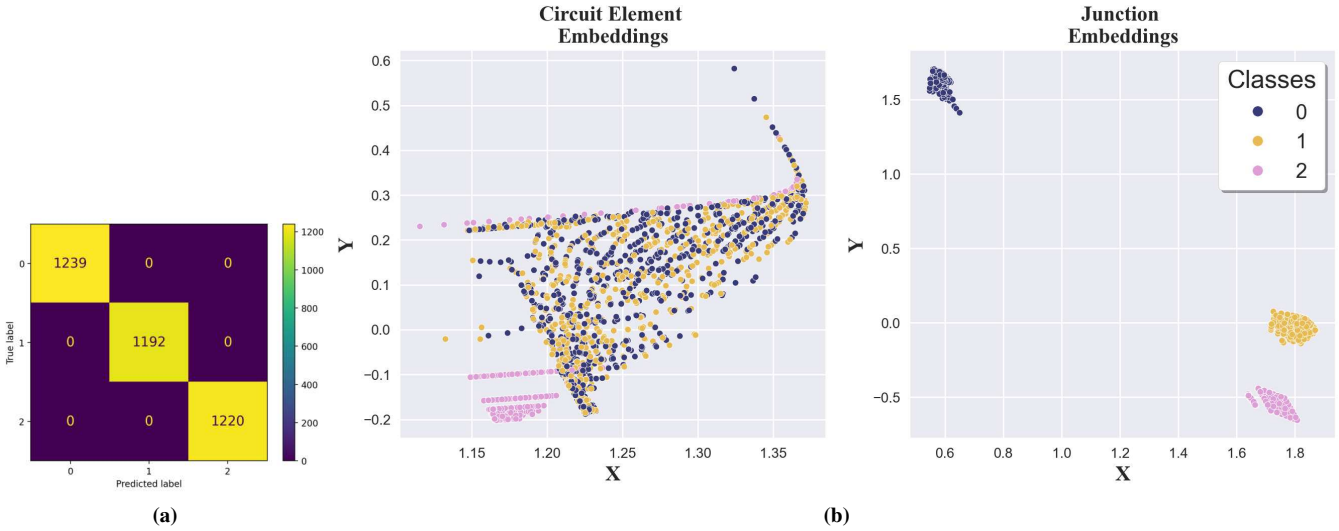


Fig. 4 – Hetero-GNN structure



(a) (b)

Fig. 5 – a) Confusion matrix of classifier output, b) 2D representation of the classified circuits

between the circuit elements to the circuit.

5) The algorithm identifies the series junctions and assign them the components that have 1 in their node column. Series junctions have the highest priority, so they can claim the components first. Series junctions are the nodes that have a sum of zero or an even number.
6) Parallel junctions are assigned the remaining components that have 1 in their node column. Parallel junctions are nodes that have a sum of an odd number.
7) Bond graph is built by starting from the highest sum value and connecting the junctions and components by bonds. Rows that have 1 & -1 indicate a jump from junction to junction (orange arrow).

C. Graph Neural Network Model

GNNs are are scalable and permutation invariant, i.e input layer is not fixed and node re-ordering will not affect the NN layer output, which satisfies the requirements needed for physical circuits representations. This allows each node to have knowledge about self information, surrounding neighbour information, and knowledge about graph structure. The graphical representation of physical systems as Bond graphs that capture the energy exchange and conservation among different components is not directly equivalent to heterogeneous graphs, and require a proper mapping and interpretation to be used as input data for GNN models. Moreover, designing a GNN model that can learn from heterogeneous graphs representing physical systems can be cumbersome [16], since accommodating different types of circuit elements, junctions, and edges, as well as their features and physical interactions is not trivial. Heterogeneous graphs are more complex and diverse than homogeneous graphs, and require special handling of the different types and features of nodes and edges.

III. ML Applications Using Hetero-GNN

The general structure of the GNN model shown in Fig. 4 utilizes three GCN layers to exchange messages across nodes. The GCN layers, aggregates feature vectors representing CE and junction features in the neighborhood nodes by message-passing algorithm, then passes the result to a dense NN layer, then apply a RELU non-linearity as an activation function. The output is fed to the Hetero-global mean pooling (GMP) layer, which returns the average node features of each node type across the node features dimension. The GMP layer output two vectors representing CE and junction features, each is of size d. The output of this general GNN structure can be used in classification and regression tasks with minor modifications to suit the application needs.

A. Circuit Structure Identification

To test the capability of the proposed framework of decoupling and differentiating between the physical and structural properties of electric circuits, a circuit classifier that takes three continuous filter circuits with identical circuit elements but different arrangement is built. The classifier application can be further utilized for digital twin applications. The classifier is fed circuits in graph forms (G) with different number of nodes (N), along with their corresponding node features (X), and adjacency matrix (A) as inputs and outputs the probability (Y) of a converter to belong to a certain class (C).

With the classifier GNN model structure implemented, a fully connected (FC) layer takes CE and junction embeddings from the (GMP) layer and outputs and generates a score for each circuit belonging to every defined class, while (Softmax) output layer is used to calculate the probability in range of (0-1) of the possibility of each circuit belonging to certain class. The classifier is fed with a 12000 graph dataset representing the three converter topologies with 70% to 30% split ratio between training and testing datasets, and the loss function

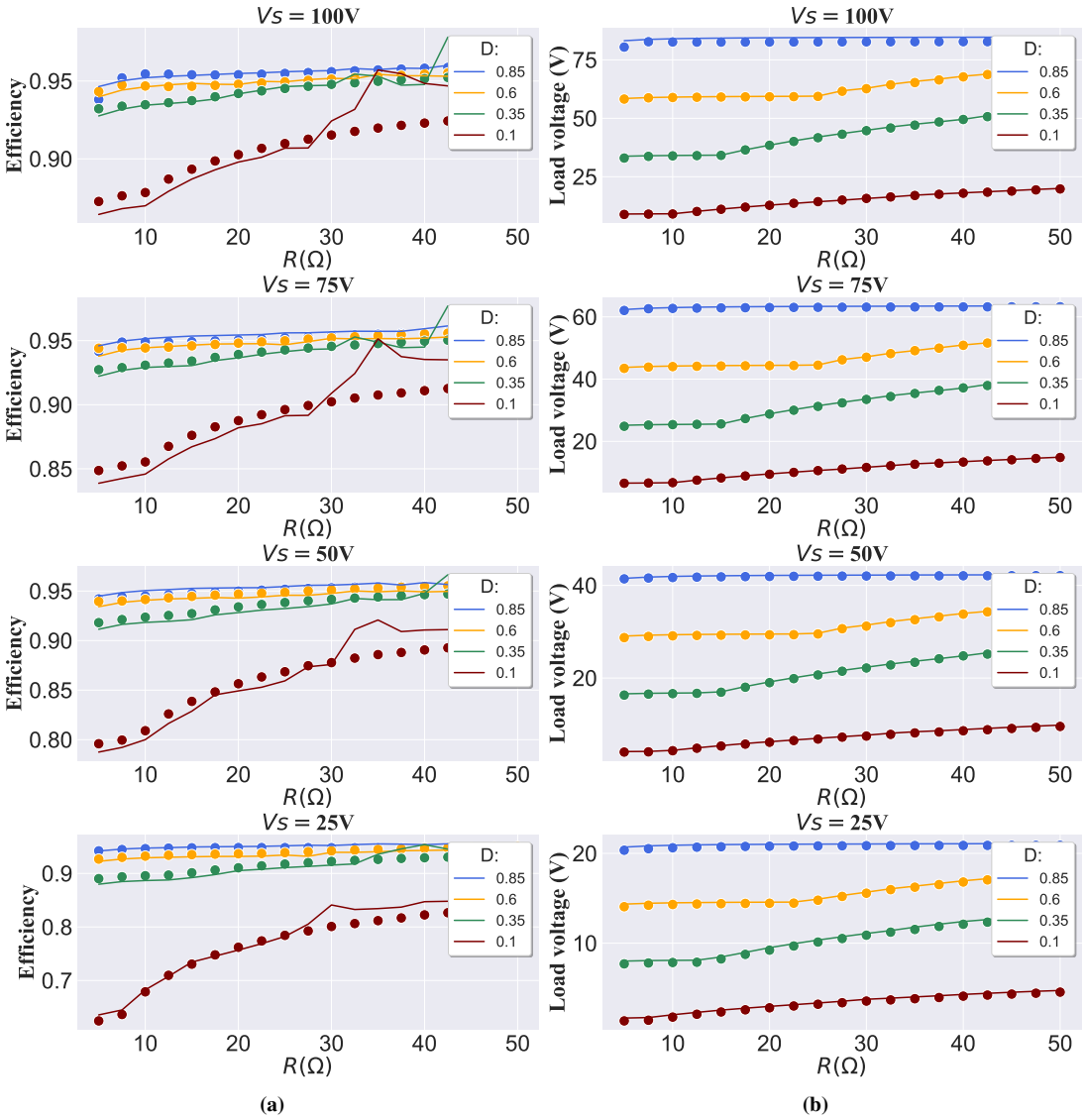


Fig. 6 – Regression model predictions corresponding to converter control and parameter variations. a) Efficiency prediction, b) Output voltage prediction

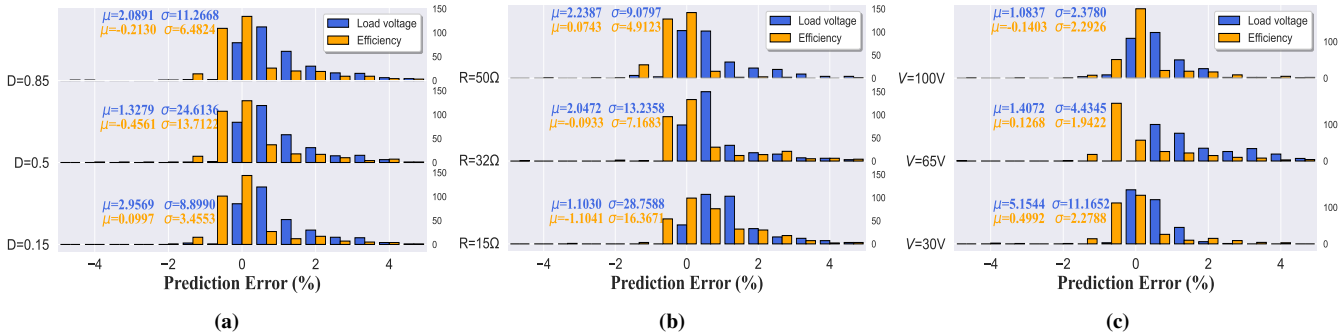


Fig. 7 – Prediction error histograms in load voltage and efficiency

is cross entropy loss function. The trained classifier scored 100% for testing data, when trained for 100 epochs. In Fig. 5, a 2-D output representation of 3600 test dataset graphs are plotted and colorized according to their predicted class. The scatter plots titled "Circuit Element Embeddings" and "Junction Embeddings" provide a visual representation of how the NN model processes and classifies the data. Despite the uniformity in the number and values of circuit elements across the classes i.e having the same type and values of circuit components, the model has differentiated them into distinct clusters, as evident in the "Junction Embeddings" plot. Also, the confusion matrix provided in Fig. 5a indicate the classifier's high level of accuracy, as demonstrated by the substantial values along the main diagonal, which represent the true positives for each class (0, 1, and 2).

B. Converter Performance Prediction

The hetero-GNN based model in Fig. 4 is adjusted in order to obtain converter dynamics predictions based on variable component values. The model takes converter circuits in graph forms (G), node features (X), adjacency matrix (A), edge features (Z) as input, and outputs the predicted variables (Y). Multiple case studies including single and multi-variable regression problems are shown, including obtaining predictions of the most essential outputs of converter circuits, namely Voltage gain and efficiency, with the potential to scale up to include many more variables and more prediction outputs. Two types of variations are tested with the proposed regression model, namely hardware and controller parameter variations. Hardware variation are when circuit component values are changed, which is a real life equivalent of changing resistor values or changing the supply voltage. The model is fed a dataset that contains circuit data after being transformed into its graph forms and assigned node and edge features, as well as information about the prediction targets, obtained from simulations. Model output shown in Fig. 6 highlights the prediction output compared to a straight line representing the ideal case of the model having 100% accuracy, while the dots represents the predicted output at this instant. The overall performance of the model was highly accurate, as indicated by the minimal prediction error. The histogram in Fig. 7 indicates error distribution in model output when exposed to changes in load resistance, supply voltage and duty cycle variations across testing dataset range. The prediction error mean(μ) and standard deviation (σ) are used to assess the model's error, while it is shown that the model can attain high prediction accuracy, while maintaining prediction error percentage less than 10%.

IV. Conclusion

This paper presented a mapping of the physical and structural properties of electric circuits using heterogeneous graphs. Bond graphs are used as a primary modelling technique to model electric circuit, then are transformed to heterogeneous graphs for heterogeneous GNN based ML applications. Feature extraction and assignment is shown in this paper, with theoretical explanations on how GNN can distinguish between structural and physical explanation. A classifier model was created and trained using the generated dataset and was able to obtain accuracy of 100%, while showing it can distinguish between the energy domains of circuits and their structural configurations. Additionally, a regression model is trained to predict the circuit performance when subjected to variations in operating conditions, load variations and controller variations.

References

[1] G. Zhang, H. He, and D. Katabi, "Circuit-GNN: Graph neural networks for distributed circuit design," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 09–15 Jun 2019, pp. 7364–7373.

[2] H. Wang, K. Wang, J. Yang, L. Shen, N. Sun, H.-S. Lee, and S. Han, "Gcn-rl circuit designer: Transferable transistor sizing with graph neural networks and reinforcement learning," in *Proceedings of the 57th ACM/EDAC/IEEE Design Automation Conference*, ser. DAC '20. IEEE Press, 2020.

[3] H. Wang, J. Yang, H. Lee, and S. Han, "Learning to design circuits," *CoRR*, 2018.

[4] Y. Li, Y. Lin, M. Madhusudan, A. Sharma, W. Xu, S. S. Sapatnekar, R. Harjani, and J. Hu, "A customized graph neural network model for guiding analog ic placement," in *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 2020, pp. 1–9.

[5] D. S. Lopera, L. Servadei, G. N. Kiprit, S. Hazra, R. Wille, and W. Ecker, "A survey of graph neural networks for electronic design automation," in *2021 ACM/IEEE 3rd Workshop on Machine Learning for CAD (MLCAD)*, 2021, pp. 1–6.

[6] Khamis, Ahmed and Agamy, Mohammed, "Comprehensive mapping of continuous/switching circuits in ccm and dcm to machine learning domain using homogeneous graph neural networks," *IEEE Open Journal of Circuits and Systems*, pp. 1–1, 2023.

[7] Khamis, Ahmed and Agamy, Mohammed, "Mapping continuous circuit structures to machine learning space," in *2022 IEEE 31st International Symposium on Industrial Electronics (ISIE)*, 2022, pp. 149–155.

[8] Khamis, Ahmed and Agamy, Mohammed, "Switching converter circuits representation in ccm dcm using graph neural network," in *2023 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2023, pp. 2697–2702.

[9] Khamis, Ahmed and Agamy, Mohammed, "Circuit structure dependent multi-variable regression model based predictions for dc-dc converters," in *2023 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2023, pp. 2703–2708.

[10] V. D. Gebben, "Bond Graph Bibliography for 1961–1976," *Journal of Dynamic Systems, Measurement, and Control*, vol. 99, no. 2, pp. 143–145, 06 1977. [Online]. Available: https://doi.org/10.1115/1.3427087

[11] P. C. Breedveld, "Multibond graph elements in physical systems theory," *Journal of the Franklin Institute*, vol. 319, no. 1-2, pp. 1–36, 1985.

[12] K. Sirivadhna, E. F. Richards, and M. D. Anderson, "The application of bond graphs to electrical machinery and power engineering," *IEEE Power Engineering Review*, vol. PER-3, no. 5, pp. 35–35, 1983.

[13] L. Umanand, "Switched junctions in bondgraph for modelling power electronic systems," *Journal of the Indian Institute of Science*, vol. 80, p. 327, 2013.

[14] A. K. Khamis and M. Agamy, "Circuit dynamics prediction via graph neural network graph framework integration: Three phase inverter case study," *IEEE Open Journal of Power Electronics*, pp. 1–16, 2024.

[15] Khamis, Ahmed and Agamy, Mohammed, "Converter circuits to machine learning: Optimal feature selection," in *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2022, pp. 1–7.

[16] Z. Shao, Y. Xu, W. Wei, F. Wang, Z. Zhang, and F. Zhu, "Heterogeneous graph neural network with multi-view representation learning," 2022.