**INTERNSHIP REPORT**

The aim of the internship was to build a Machine Learning application to achieve clustering of digital circuits using the idea of graph neural networks considering every logic gate(AND,OR,XOR) as a node in the graph and every wire in the circuit or in other words the connection between 2 gates as the edge of the graph. To begin with, one needs to understand what a graph is. Hence, firstly, the basics of graph data structure were studied – the core concept and various ways of representation (adjacency matrix, adjacency list, etc.) using some online videos. Furthermore, it is imperative that the working of graph neural networks is clear in order to make manipulations on the same and build a clustering algorithm. The same was studied using blogs on Medium. Now, one can move to studying basics of Deep Graph Library (DGL). DGL has 3 options to use as backend: PyTorch, MXNet and Tensorflow. Here, we have used PyTorch as backend. This project demands in-depth understanding of PyTorch tensors hence that was studied using the official PyTorch website. The conversion of PyTorch tensors to numpy arrays and vice-versa is to be laid special emphasis on. There are 2 ways to create a DGL graph: one can be specifying a source and another destination lists and the other way can be creating an empty graph first and subsequently adding a fixed number of nodes and edges between them. Next, clustering using Karate Club example was done using Graph Neural Networks and changing hyperparameters (no. of epochs, convolutional layers) to observe the effect of the same on clustering and inferring changes in clustering. Moreover, to get a deep idea of clustering algorithms in ML, various techniques of clustering like K-Means Clustering, DBSCAN, Affinity Propagation and Agglomerative Hierarchical Clustering along with their scikit-learn implementations were studied. For a generalised program, the number of sets (no. of clusters = 2\*no. of sets) and output was produced for 2 sets (4 clusters). Now comes the transition to the domain of digital electronics. Basics of ASIC Design EDA Flow and VLSI Partitioning were studied to get an idea of the VLSI Domain. An open-source UCLA Netlist data (ispd2005) was downloaded which comprises of 8 netlists. These netlists are what we have used for clustering digital circuits. Next comes perhaps the most important step of the entire project: parsing the netlist files to convert them into a format which can be fed to create a DGL Graph which can then be used for clustering. Successful creation of graph from the netlists. Moving on to the process of training GCN for clustering specifying the numbers of clusters as 20 using Google Colab (TPU). Tried using networkx for visualising clusters, but not good result. Hence used GEPHI for visualisation of clusters which has Java as its dependency. Defined metrics for analysing quality of clustering: (1) Number of nodes per cluster distribution, (2) Number of intercluster edges between any 2 nodes represented as **Inter-Cluster Score**, which is defined as:

**IC Score = No. of edges between 2 clusters (say c1 and c2) / (Number of nodes in c1 + Number of nodes in c2).**

This was followed by Python Programming to calculate the said metrics for the entire graph and displaying them using 3 bar charts. Now, to make the application truly robust and user-friendly, the entire Python script was converted to a master driver script to be run by the user as an object-oriented model like a module in which the user can specify the filename, the number of epochs to train the graph neural network for, the learning rate for training procedure, and the number of clusters needed as parameters and create an object of the DigitalClustering class and call its 2 member functions:

* Run (): Training neural network and calculating the metrics.
* Visualise (): Plotting the 3 bar graphs.