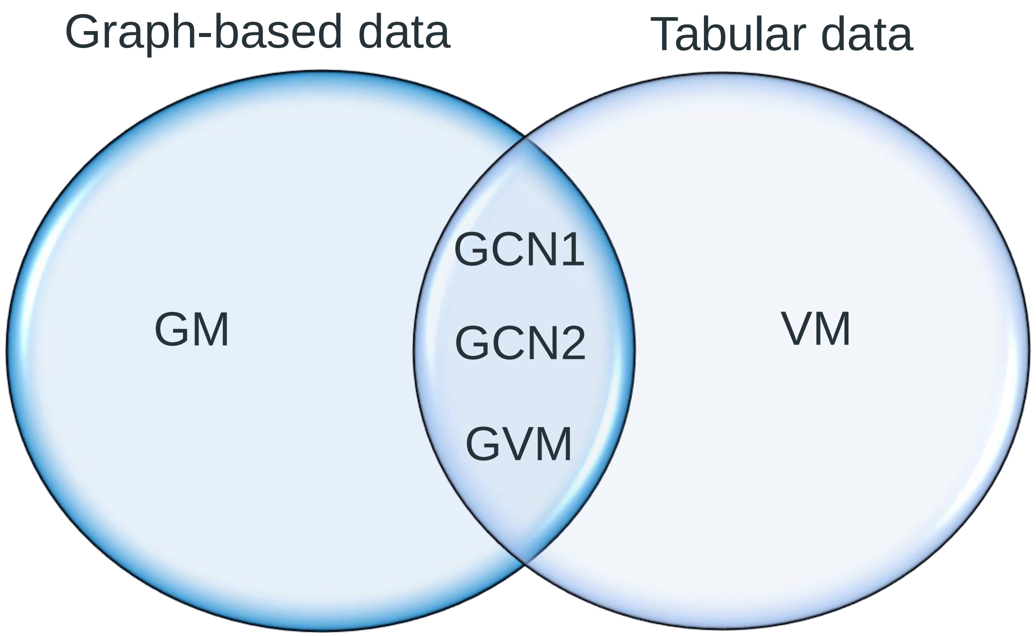
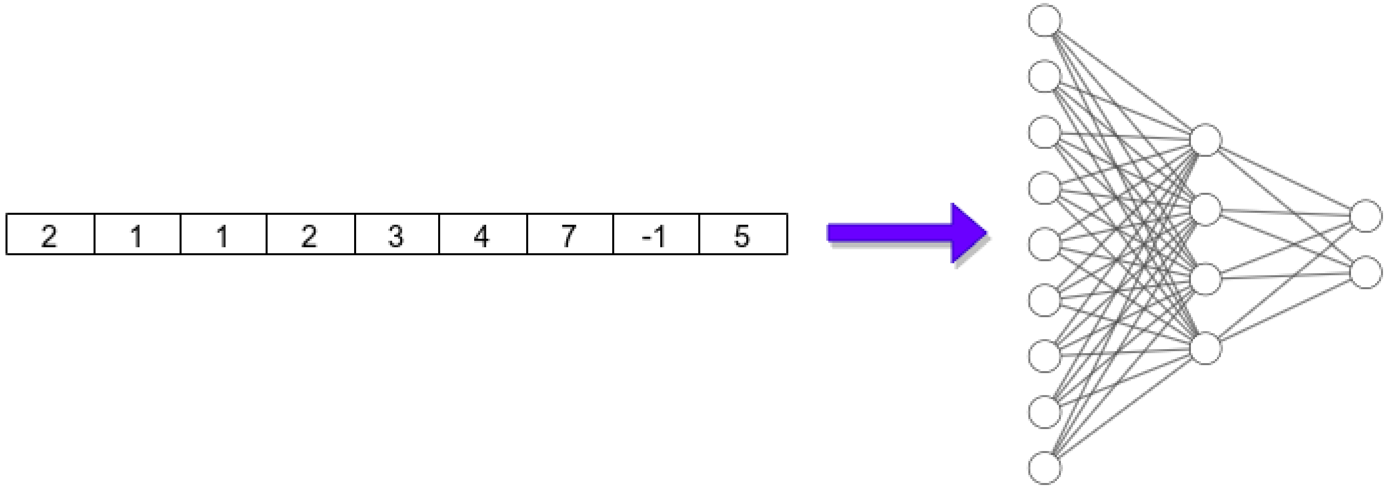
Several models:



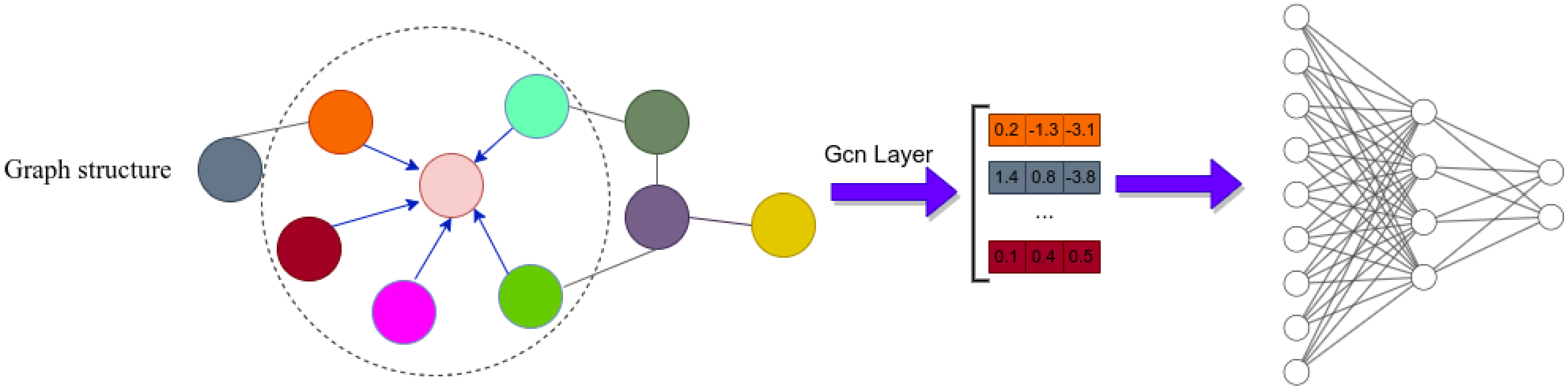
Acronym	Meaning
VM	Values model
GM	Graph model
GCN1	Graph convolutional network with one layer
GCN2	Graph convolutional network with two layers
GVM	Graph&Values model

VM – Values model



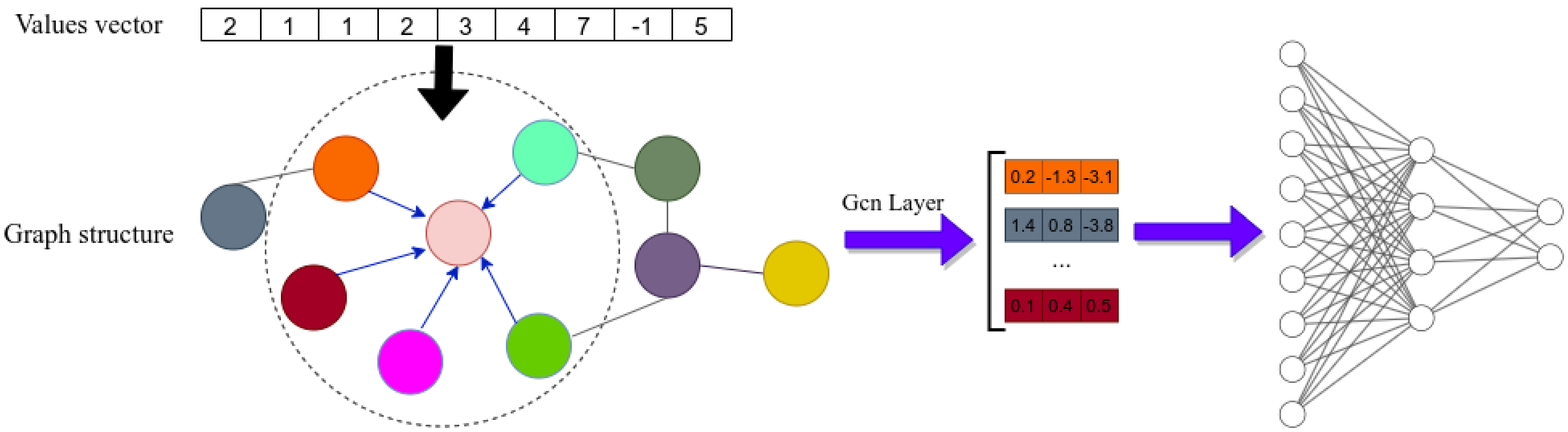
This model applies two fully-connected layers, each followed by an activation function. In this model, the graph structure is ignored, and only the values are used as in a classical tabular ML method.

GM – Graph model



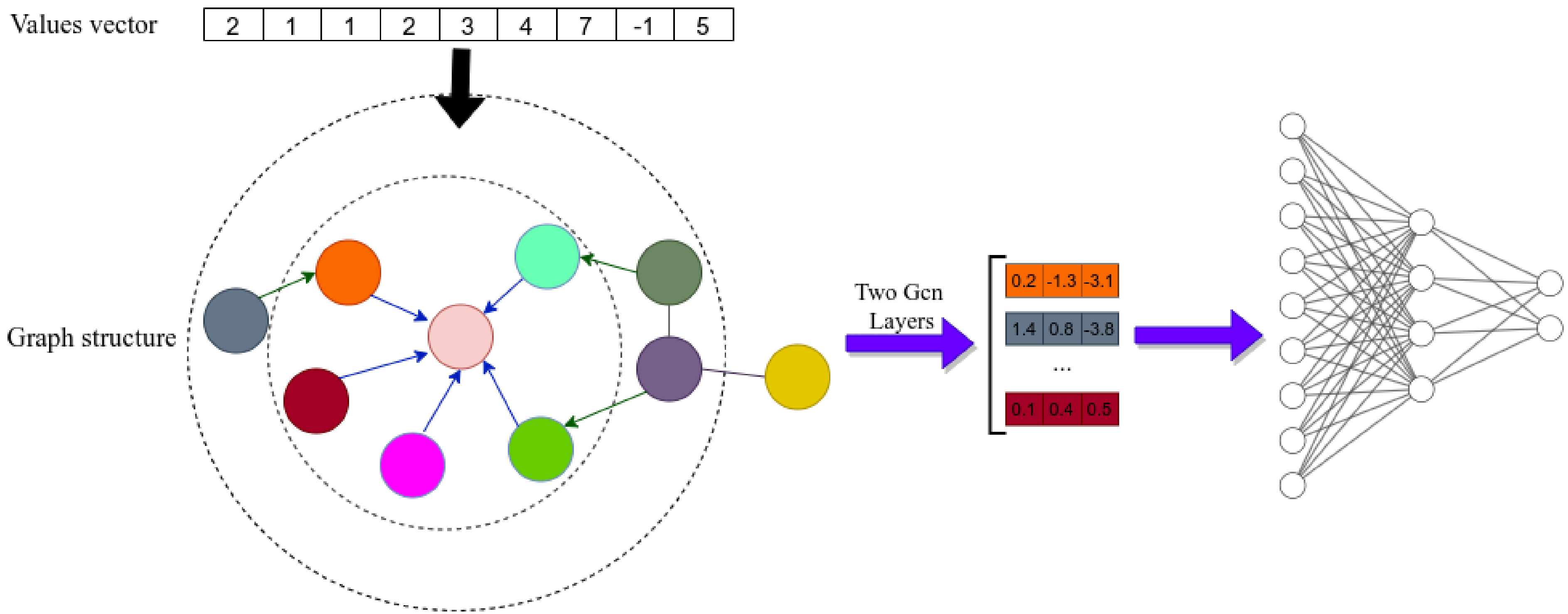
This model applies GCN layer, but it ignores the values within the vertices. We assign each existing vertex with the same value (=1). Thus, we do not pay attention to the tabular values. This method basically tries to learn from the graph topology ignoring the tabular values.

GCN1 – Graph Convolution Network with one layer



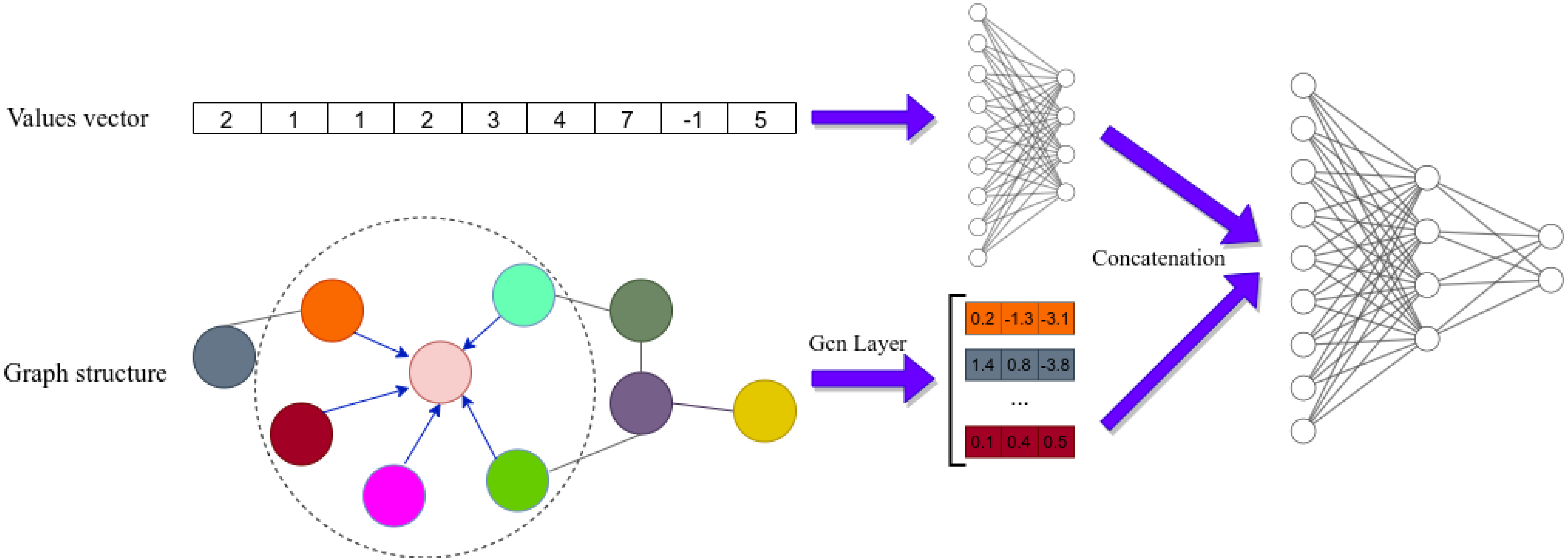
This model applies one layer of convolution to graph structure and vertices' values. The GCN layer propagates values across the neighbors thus transferring information in the graph. The last layer of the GCN is the input to a FCN with a binary output.

GCN2 – Graph Convolution Network with two layers



This model applies two layers of GCN. This is similar to the method above, but with more distant neighbors.

GVM – Graph&Values model



This model concatenates the results of GM and VM models. The concatenated vector is fed into another classification machine (2-layer FCN). In this model, we assumed that each type of information is independent and the two types can be merged as independent variables.

Simulations:

Goal: Examine the different approaches studied in this thesis

Each sample consists of two components:

- A values vector x_i
- A graph structure represented by the adjacency matrix A_i.

The simulation has two parameters:

- σ - The difference in the values distribution among classes
- ϵ - The difference in degrees distribution among classes.

Class 0:

- The values were sampled from $N(0,1)$.
- The graphs were generated using an Erdos Renyi model with $n = 50$ and $p = 0.1$.

Class 1:

- The values were sampled from $N(\mu,1)$, μ differs between dimensions and is sampled from $N(0,\sigma)$.
- The graphs were generated using an Erdos Renyi model with $n = 50$ and $p = 0.1 + \epsilon$.

GCN with learnt α

Basic GCN: $\sigma((A + I)xW)$

Our GCN: $\sigma((A + \alpha I)xW)$

Goal: The motivation is an attempt to **learn** the interaction magnitude between vertex's the propagation from the value in the previous layer and its neighbors' values in the previous layer.