The Application of Artificial Neural Network on Predicting Brazil Nut Effect Phenomena

Muhammad Iqbal Rahmadhan Putra[*1], Nadhira Azzahra[1], Sparisoma Viridi[1]

Corresponding author: miqbalrp@gmail.com

[1]Department of Physics,
Faculty of Mathematics and Natural Sciences,
Bandung Institute of Technology, Bandung, Indonesia

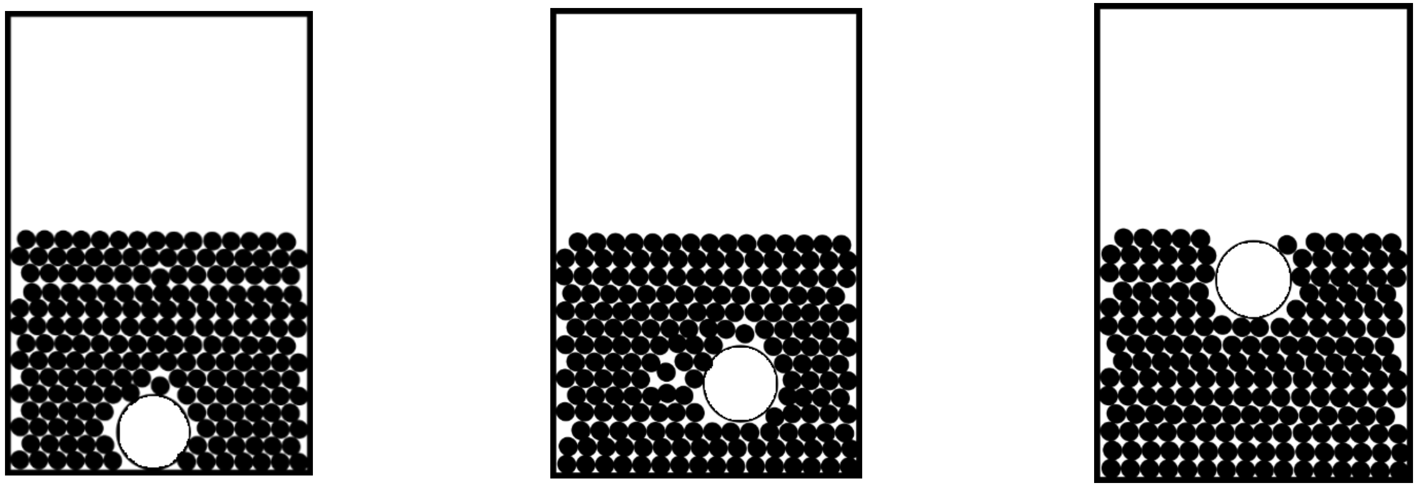
ICCSE2019
International Conference
on Computation in Science and Engineering

ABSTRCT

Brazil nut effect is a phenomenon in which large-size particle lift up when vibrated vertically on granular system. This phenomenon has been studied both through experiments and computational simulations. One parameter that can be extracted from this phenomenon is the final height of the large-size particle. However, due to the large number of input parameters that affect the final results, the way to predict is still the subject of study. In this study, we used artificial neural network (ANN) as a method to predict the final results of experiments through molecular dynamic simulations with different input parameters. The different input parameters were created by generating random numbers of the appropriate range. After the simulation was carried out and the final result was obtained, the input and final parameter data were used as a database for training ANN. To get optimal results, a comparison is made of various ANN model architectures.

INTRODUCTION TO BNE

If a mixture of two different types of granular materials is subjected to **vertical vibrations**, larger granules (or usually known as **intruders**) will rise to the surface. This phenomenon is known as the **Brazil-Nut Effect** (BNE)

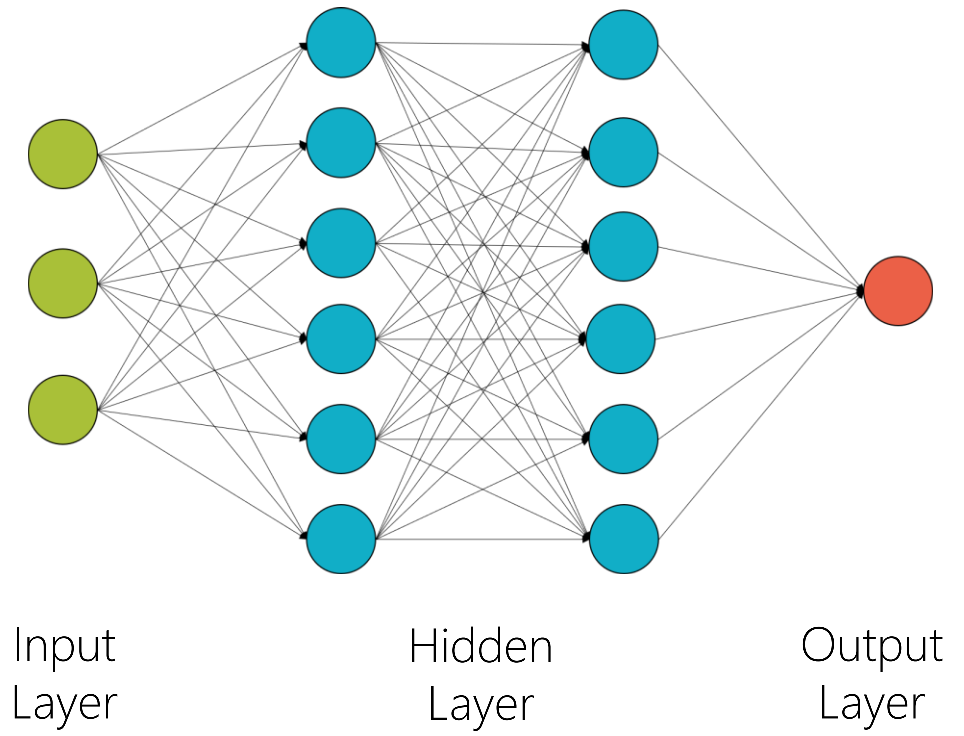


$t = 1$ $t = 2$ $t = 3$

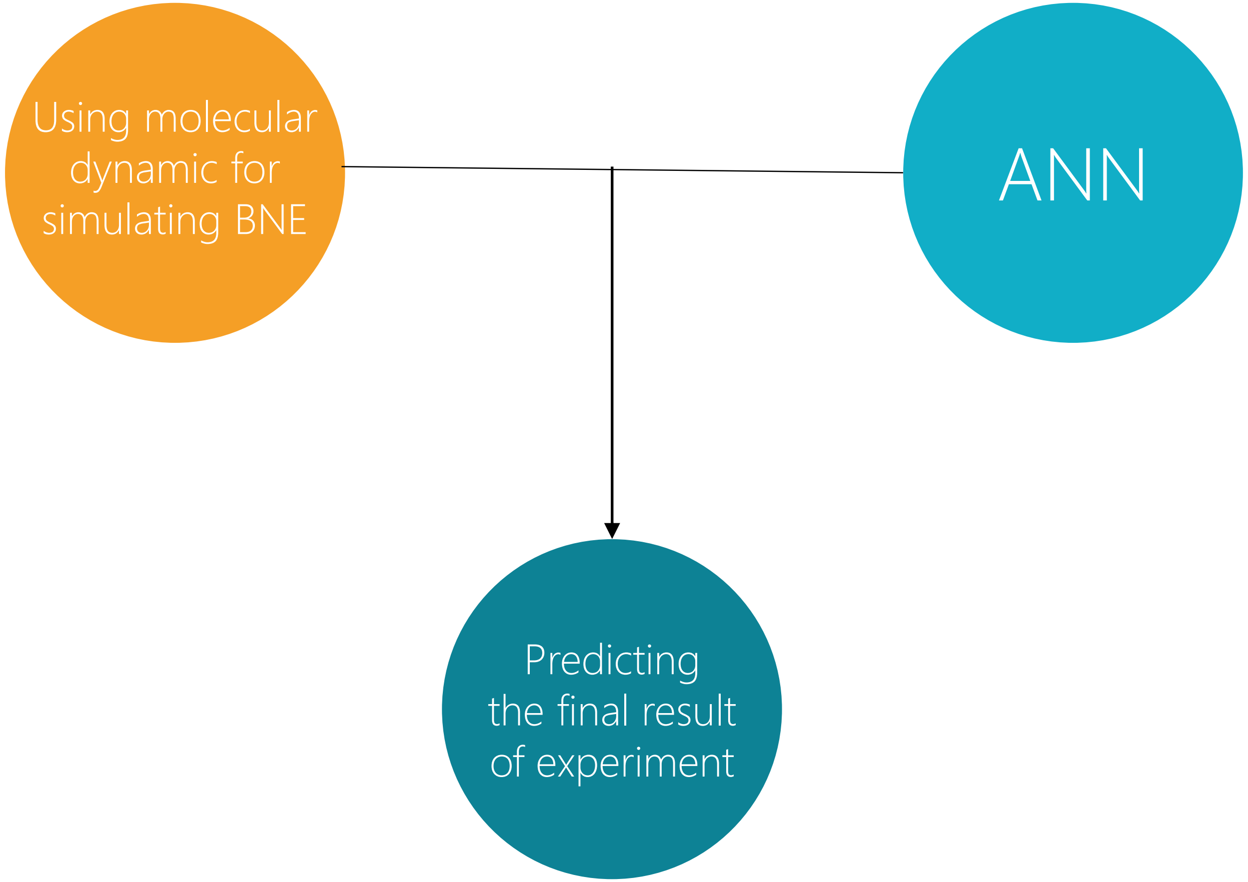
The factors that can influence BNE are the mass and diameter ratios of both types of granular material, the amplitude and frequency of the vibration, the arrangement of the bed particles and the initial configuration of the granular mixtures.

INTRODUCTION TO ANN

Artificial Neural Network (ANN) is a method of processing information that is inspired by the capabilities of the human nervous system.
ANN is usually designed for specific purposes, for example for **fitting** (prediction and estimation), **pattern recognition** (for example letter recognition or speech recognition) or **classification of data groups** (clustering)



RESEARCH OBJECTIVES



RESEARCH METHODS

Collecting data through molecular dynamic simulation

↓

Data preprocessing

- Scaling
- One-Hot encoding
- Folding

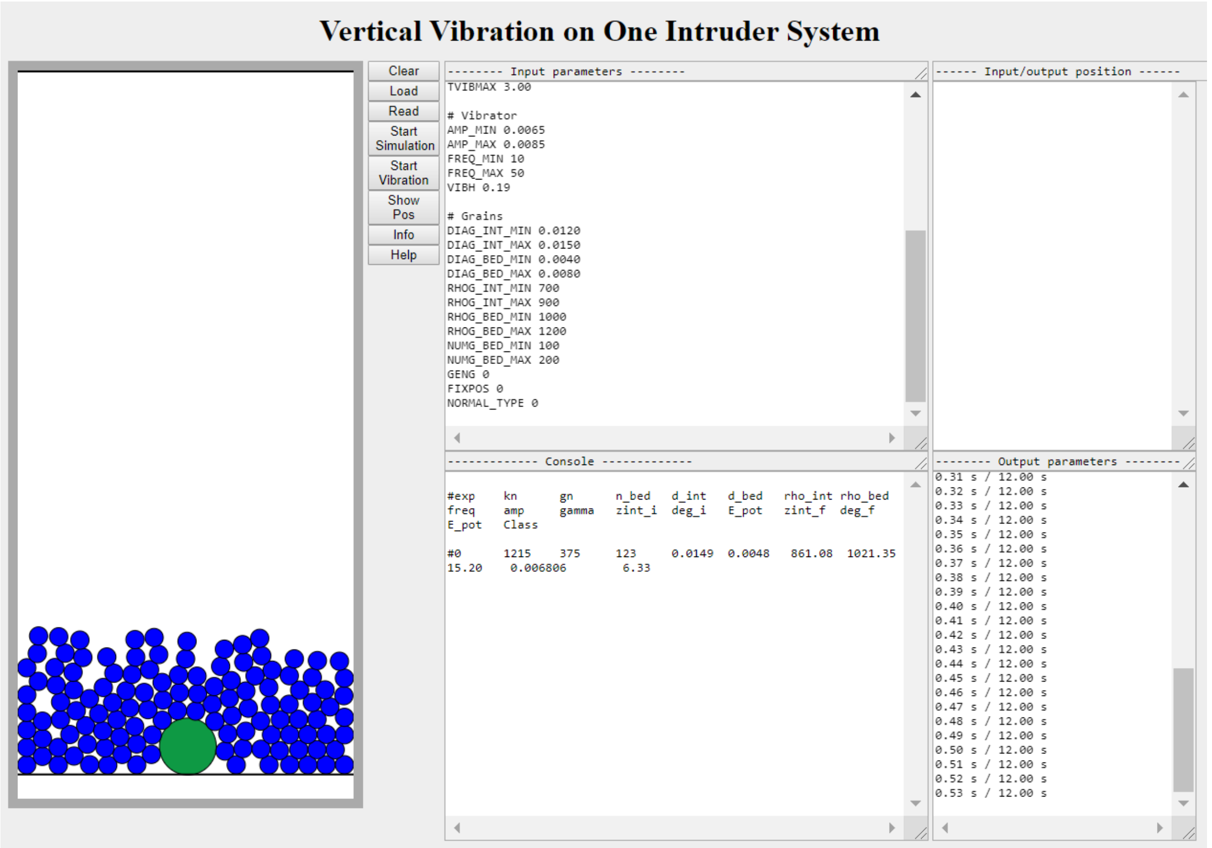
↓

Building ANN model

↓

Training model with K-Fold Validation

COLLECTING DATA



Start
for $n = 0$ to N_{exp}
    generate random value of N_g, d_{bed}, ρ_{bed}, d_{int}, ρ_{int}, k_n, γ_n, A, and f
    generate initial positions and velocities of each particles
    $t_{vib} = 0$
    for $t = 0$ to t_{max}
        for each particle i
            while $t > t_{delay}$
                calculate $\theta(t_{vib})$ and $\theta'(t_{vib})$
                $t_{vib} = t_{vib} + 1$
            calculate F_i
            calculate $a_i(t)$, $v_i(t + \Delta t)$, and $r_i(t + \Delta t)$
        end for
    end for
    print $r_{intruder}$ in z-direction
end for

NORMAL FORCE BETWEEN GRAIN

$$\delta_{ij} = a_i + a_j - |\boldsymbol{r}_i - \boldsymbol{r}_j|$$

$$\boldsymbol{v}_{n_{ij}} = ((\boldsymbol{v}_i - \boldsymbol{v}_j) \cdot \boldsymbol{n}_{ij})\boldsymbol{n}_{ij}$$

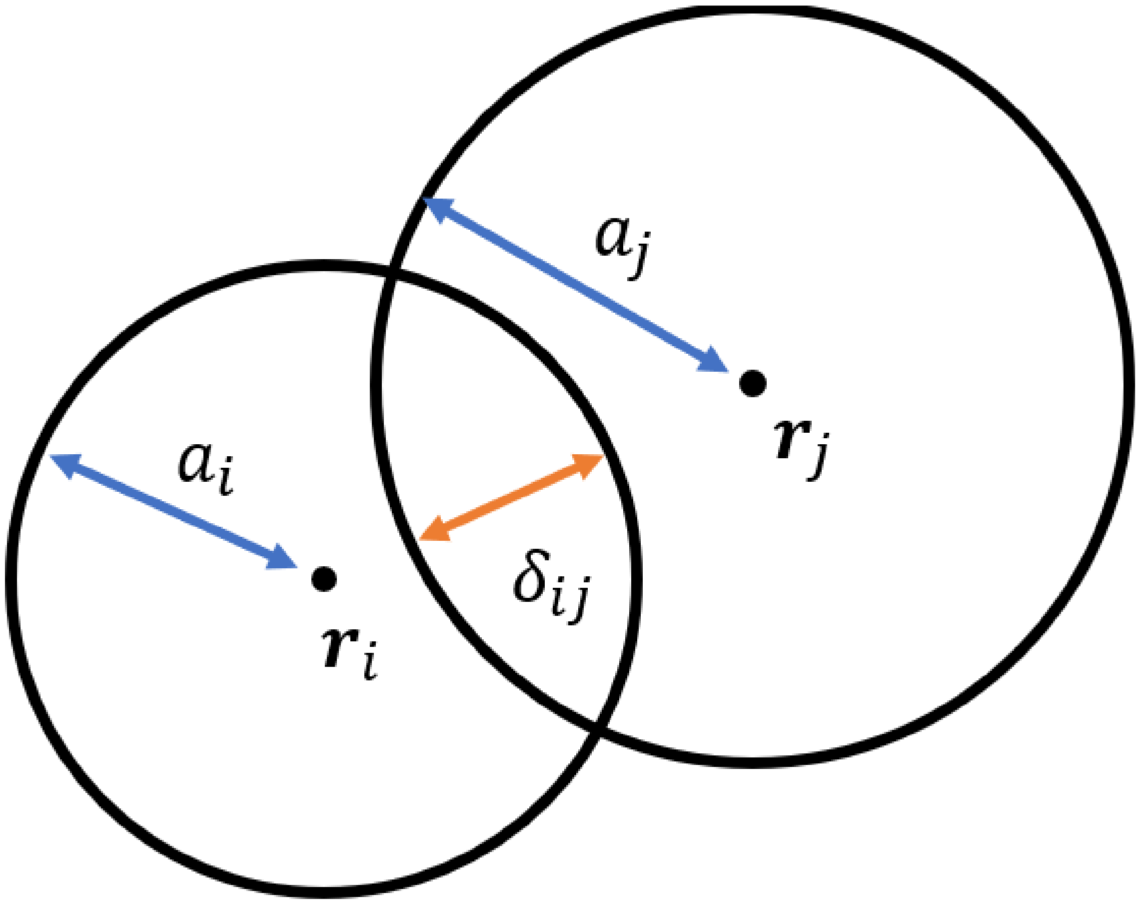
$$m_{eff} = \frac{m_i m_j}{m_i + m_j}$$

$$\boldsymbol{n}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j / |\boldsymbol{r}_i - \boldsymbol{r}_j|$$

Spring-dashpot model in general [5]:

$$\boldsymbol{F}_{n_{ij}} = k_n \delta_{ij} \boldsymbol{n}_{ij} - \gamma_n m_{eff} \boldsymbol{v}_{\boldsymbol{n}_{ij}}$$

k_n : Grains normal stiffness coefficient (N/m)
γ_n : Grains normal damping coefficient (s^{-1})



DATASET

Physical parameters	Values
Number of beds N_g	Random in range [100, 200]
Beds diameter d_{bed}	Random in range [0.004, 0.008] (m)
Beds density ρ_{bed}	Random in range [1000, 2000] ($\mathrm{kg/m^3}$)
Intruder diameter d_{int}	Random in range [0.0120, 0.0150] (m)
Intruder density ρ_{int}	Random in range [700, 900] ($\mathrm{kg/m^3}$)
Stiffness coefficient k_n	Random in range [1000, 2000] (N/m)
Damping coefficient γ_n	Random in range [300, 400] ($\mathrm{s^{-1}}$)
Vibration amplitude A	Random in range [0.0065, 0.0085] (m)
Vibration frequency f	Random in range [10, 50] (Hz)
Gravitational acceleration g	9.807 ($\mathrm{m/s^2}$)
Vibration time t_{vib}	10 (s)
Delay time t_{del}	1 (s)

Class	Position of Intruder
A	> 75 %
B	50 - 75 %
C	25 – 50 %
D	< 25 %

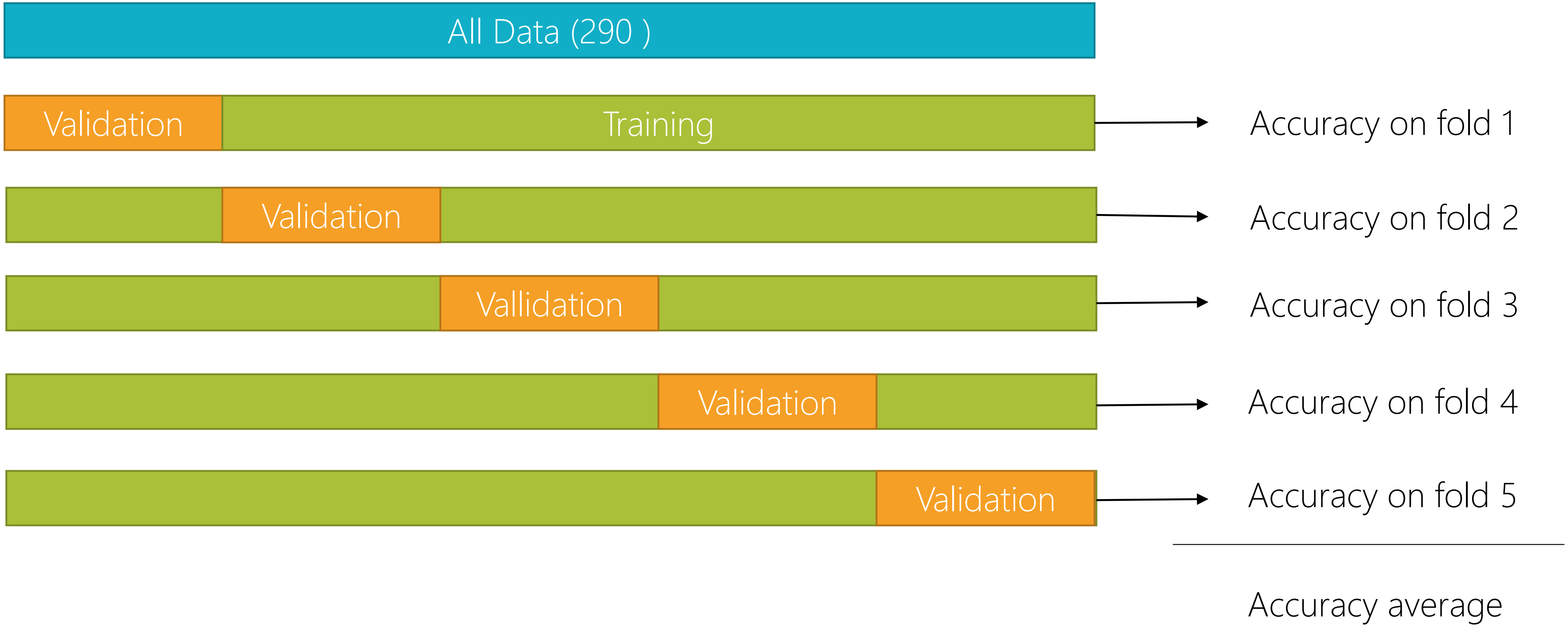
Features

- k_n
- γ_n
- d_{ratio}
- ρ_{ratio}
- contact density
- f
- $\Gamma = \frac{(2\pi f)^2 A}{g}$

Target

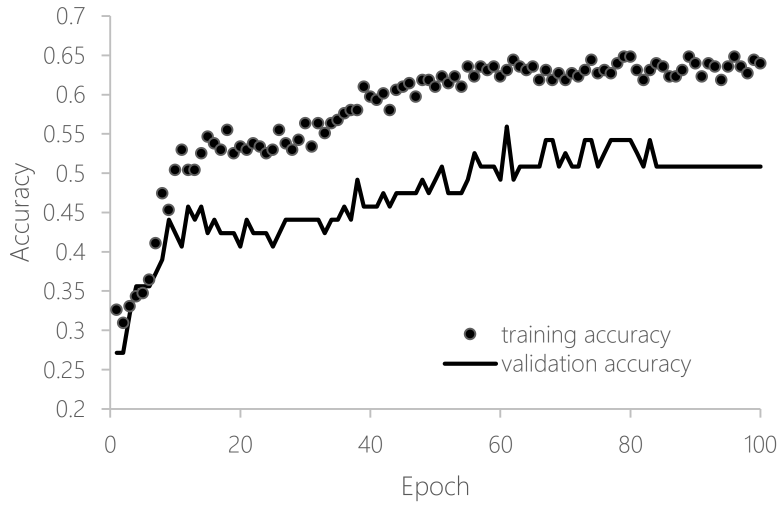
- Class

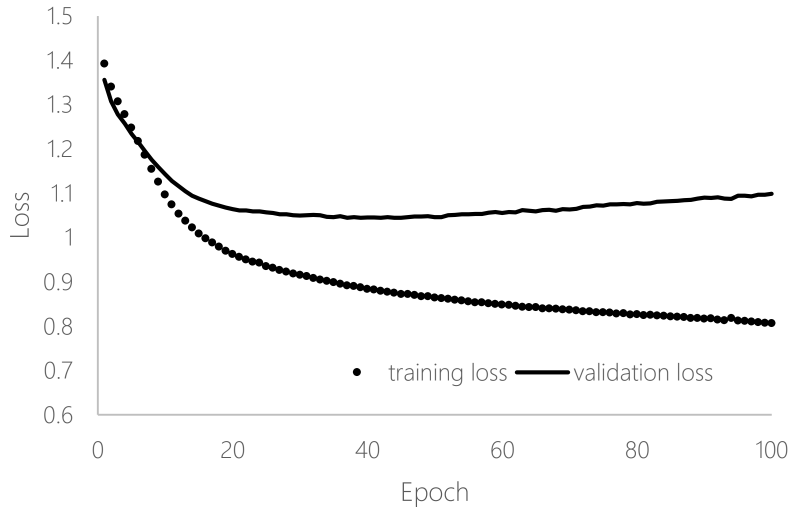
K-FOLD
CROSS VALIDATION



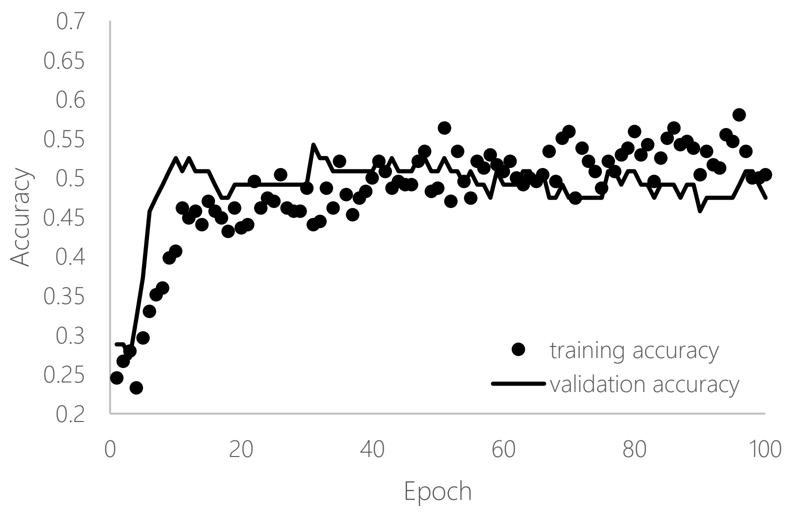
DROP-OUT LAYER

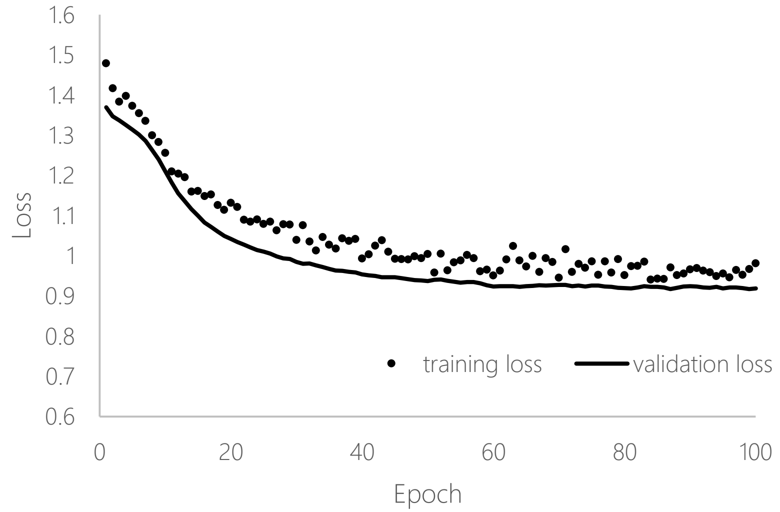
Over-fitting that occurs :



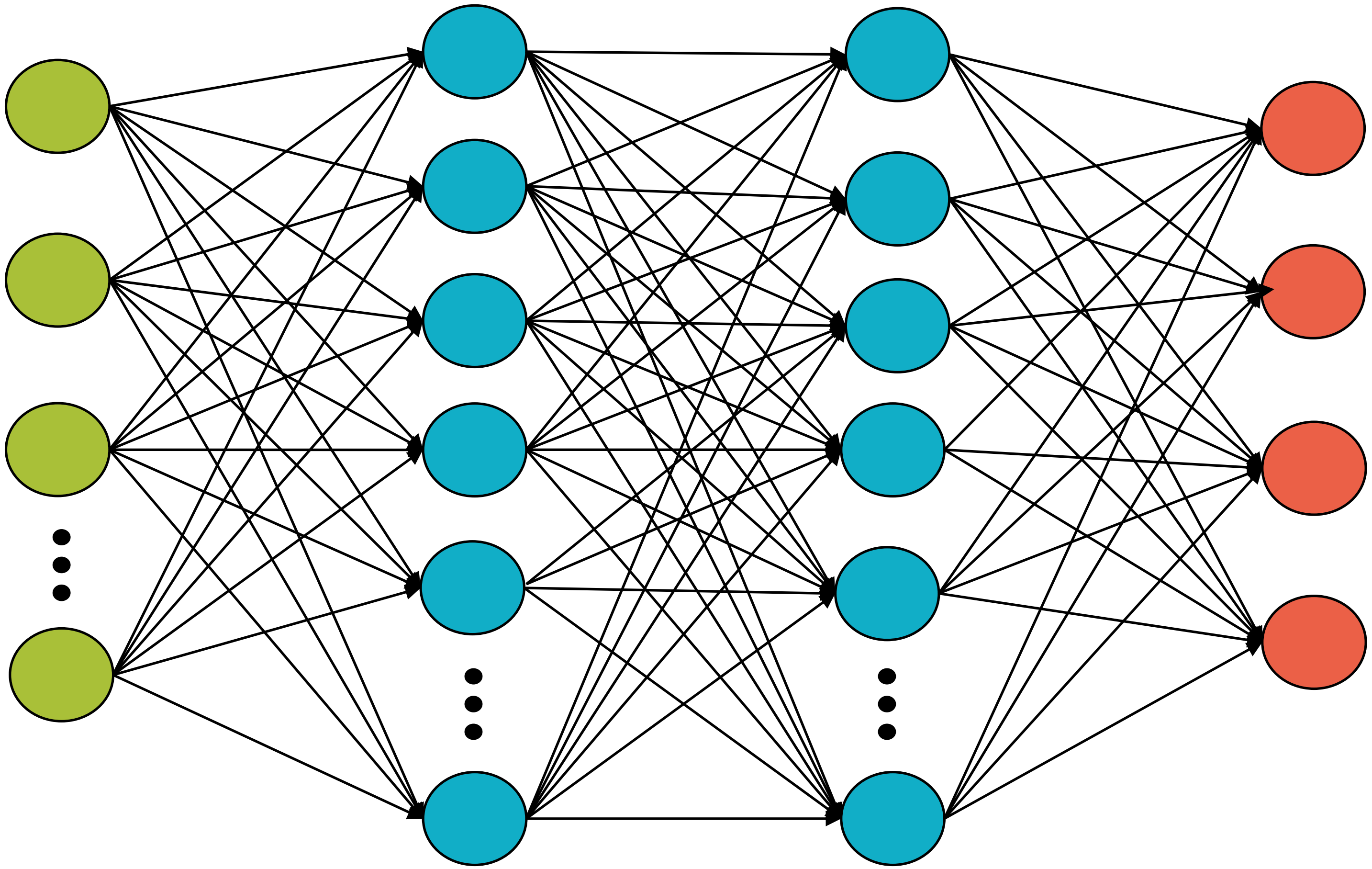


Using drop-out layer :





BUILDING MODELS



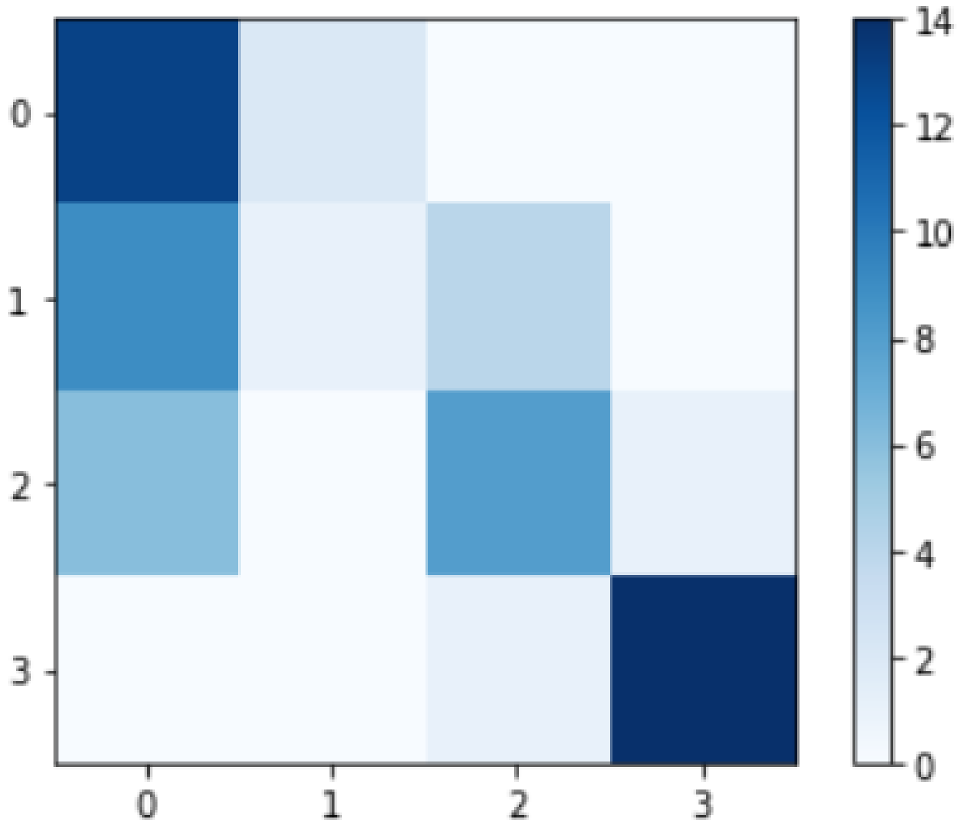
- Input layer
  - 8 features
- Hidden layers
  - N nodes
  - M nodes
  - Rectified Linear Units (ReLU) function
- Output layer
  - 4 nodes as One-hot Encoding
  - Softmax function

TRAINING MODELS

Result of k-Fold cross validation for each layers-nodes possibilities:

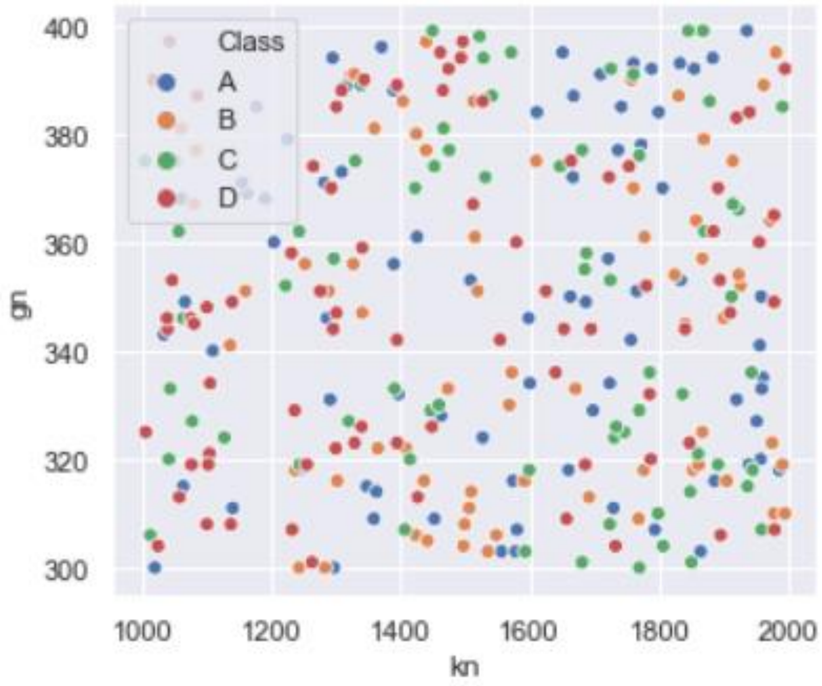
Number of nodes	Number of layers							
	1		2		3		4	
	Average	Std	Average	Std	Average	Std	Average	Std
1	47%	(+/- 4%)	37%	(+/- 11%)	32%	(+/- 15%)	24%	(+/- 2%)
3	48%	(+/- 4%)	45%	(+/- 7%)	41%	(+/- 11%)	48%	(+/- 2%)
5	52%	(+/- 6%)	47%	(+/- 6%)	51%	(+/- 6%)	47%	(+/- 3%)
10	51%	(+/- 4%)	50%	(+/- 4%)	49%	(+/- 4%)	52%	(+/- 7%)
20	51%	(+/- 3%)	51%	(+/- 3%)	51%	(+/- 4%)	52%	(+/- 5%)
50	49%	(+/- 6%)	50%	(+/- 6%)	49%	(+/- 2%)	50%	(+/- 4%)

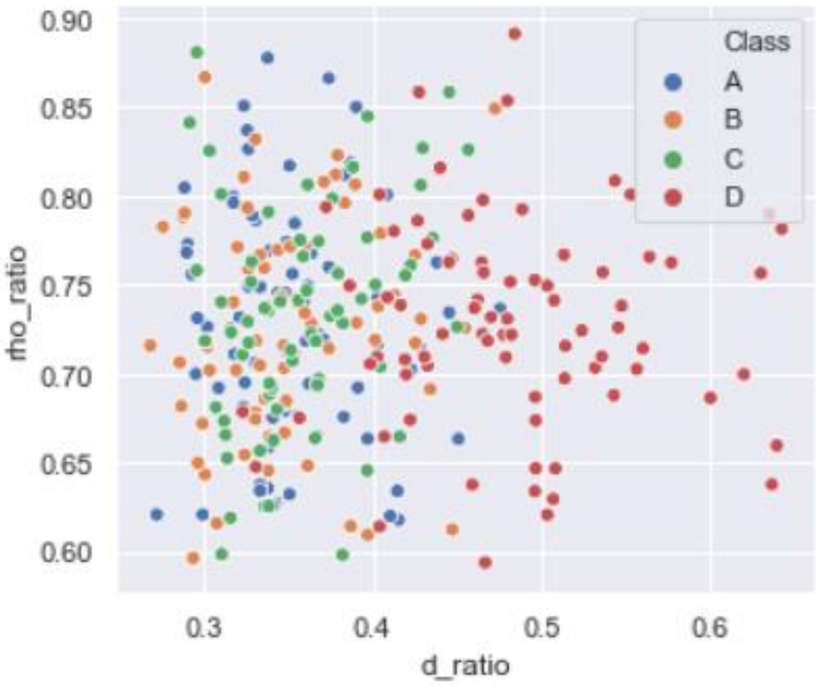
CONFUSION MATRIX

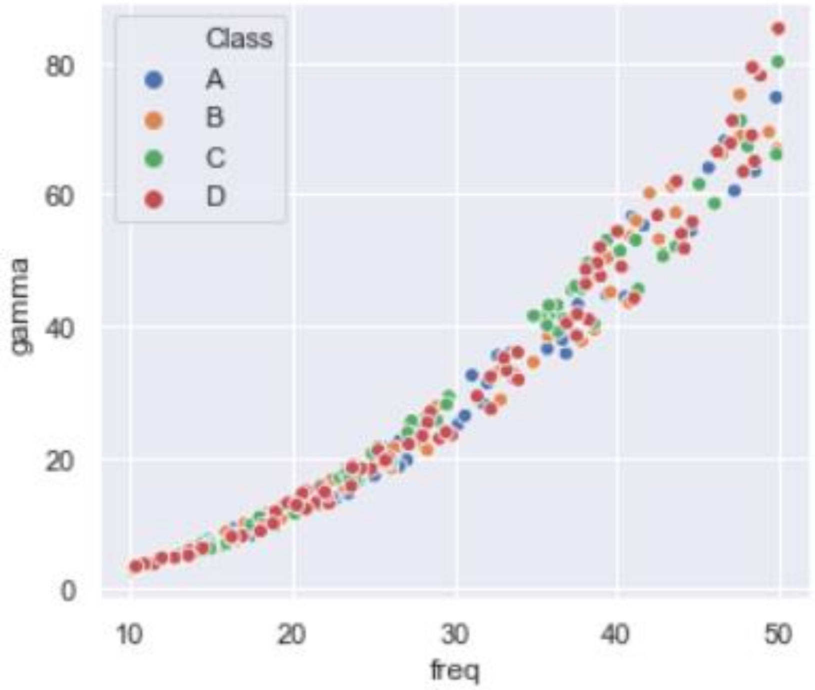


Only class A and class B give accuracy more than 80%

SCATTER PLOT







CONCLUSION

- The model that was built can only predict with 52% accuracy.
  - Only class A and class D give more than 80 accuracy
- For further research, collecting data with another range of parameters is needed for more proportional class division

THANK YOU