initializer	Activation	#hidden layers	(L1 || L2 ||L1_L2) & (batchN || dorpO)	Optimizer	Loss	val_loss	Accuracy	val_accuracy	f1 score	Val_f1 score	Model number
lecun_normal	Selu	3	No&No	adam	0.4441	0.506	0.7734	0.7360	0.6784	0.6197	1
//	//	//	Yes&No	//	0.5333	0.5420	0.7311	0.7099	0.6308	0.5945	2
//	//	//	Yes&yes	//	0.5579	0.5742	0.7065	0.7028	0.5956	0.6757	3
//	//	5	Yes&yes	//	0.5862	0.5818	0.6982	0.7034	0.5954	0.6829	4
he_normal	relu	3	No&no	Adam	0.3870	0.5642	0.8122	0.7196	0.7341	0.5966	5
//	//	//	Yes&yes	Adam	0.5196	0.5203	0.7358	0.7274	0.6365	0.6514	6
//	elu	5	//+default	//	0.4638	0.4759	0.7604	0.7544	0.6708	0.6495	7
			//(0.1)		Worse	Worse	Worse	Worse	Worse	worse	
//	//	//	Default+ dropout	Adam	0.4795	0.4751	0.7519	0.7605	0.6497	0.6776	8
//	//	3	//	//	The same	-----	----	----	----	------	10
//	//	5	//	Sgd							9
//	//	3	//	//+schd	0.4918	0.4826	0.7381	0.7493	0.6383	0.6779	11
//	//	//	//	Adam+schd							12

Model number	Overfitting (high variance)	Underfitting (high bias)	Model performance (fluctuations)	More train can help ?	comment
1	Yes	Yes More training can give more	Not bad	Underfitting only	
2	Small	Yes (end of the model)	bad	NO	
3	No	YYESS	bad	NO	
4	No	Yes	Bad	No	
5	Yes	Yes More training can give more	Not bad (good)	Underfitting only	
6	No	Yes	Bad	No	
7	Small	Yes	good	Tend to overfitting	
8	No	Yes	Excellent	little	Elu gives the best performance with dropout
9					Bad performance
10					More layer is not usefull
11	no	yes	good	More training was given Slightly better performance	
12					Bad performance

Drop out (0.3 - 0.5) with L1 & L2 and Elu gives us the best performance

More layer is not useful 3 is enough