MACHINE LEARNING MODEL COMPARISON BASED ON SOME METRICS

Safa ORHAN
Computer Engineering Student
Istanbul Kultur University
Istanbul, Turkey

Eyüp USTA
Computer Engineering Student
Istanbul Kultur University
Istanbul, Turkey

I. PAPER IMPLEMENTATION

1. Comparison of Classification Algorithms for Detection of Phishing Websites – Same Dataset

K-Nearest Neighbors: # of neighbors = 5, weights = uniform, algorithm = auto

Paper Accuracy: 94.81% --- Our Accuracy: 93.72%

Multilayer Perceptron: Hidden Layers = 30, max iterations = 3000

Paper Accuracy: 97.22% --- Our Accuracy: 95.41%

Multilayer Perceptron: Hidden Layers = 150, max iterations = 1000

Paper Accuracy: 90.28% --- Our Accuracy: 96.98%

Multilayer Perceptron: Hidden Layers = 100, max iterations = 1000

Paper Accuracy: 96.71% --- Our Accuracy: 97.04%

Random Forest: # of estimators = 7, max depth = 11, criteria = entropy

Paper Accuracy: 95.25% --- Our Accuracy: 95.17%

Random Forest: # of estimators = 7, max depth = 8, criteria = entropy

Paper Accuracy: 89.16% --- Our Accuracy: 95.38%

SVC: C = 1.0, kernel = linear

Paper Accuracy: 92.71% --- Our Accuracy: 92.67%

SVC: C = 1.0, kernel = polynomial, degree = 1

Paper Accuracy: 92.57% --- Our Accuracy: 92.58%

SVC: C = 1.0, kernel = polynomial, degree = 2

Paper Accuracy: 93.88% --- Our Accuracy 94.21%

SVC: C = 1.0, kernel = polynomial, degree = 3 -- Accuracy : 95.17 %

SVC: C = 1.0, kernel = polynomial, degree = 4 -- Accuracy : 95.62 %

SVC: C = 1.0, kernel = polynomial, degree = 5 -- Accuracy : 95.99 %

SVC: C = 1.0, kernel = polynomial, degree = 6 -- Accuracy : 96.20 %

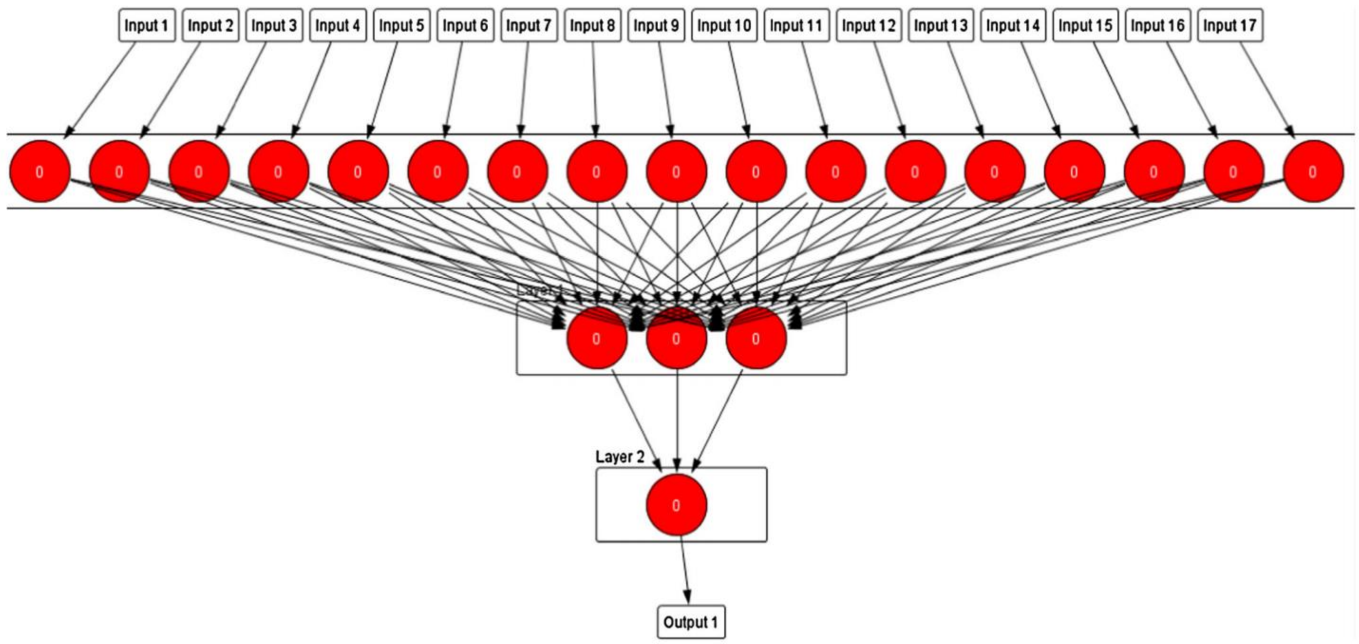
SVC: C = 1.0, kernel = polynomial, degree = 7 -- Accuracy : 96.38 %

SVC: C = 1.0, kernel = polynomial, degree = 8 -- Accuracy : 96.56 %

SVC: C = 1.0, kernel = polynomial, degree = 9 -- Accuracy : 96.71 %

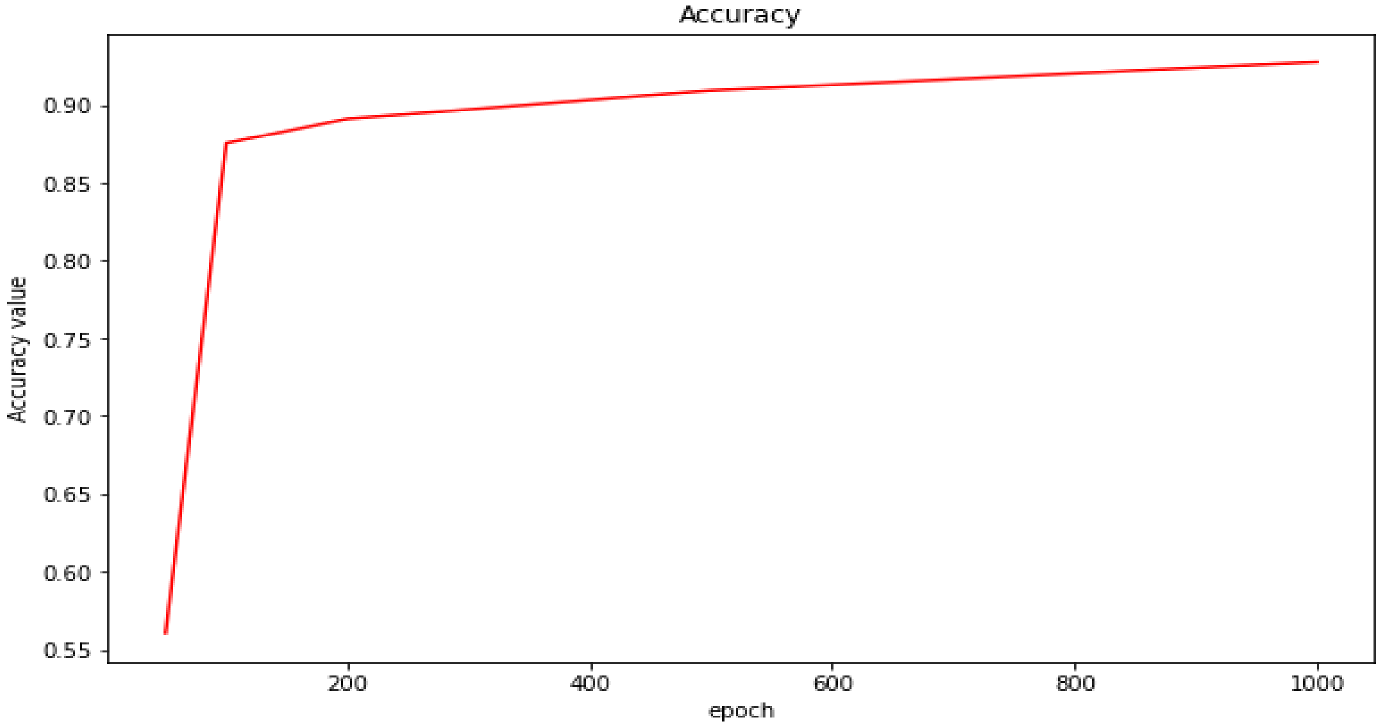
SVC: C = 1.0, kernel = polynomial, degree = 10 -- Accuracy : 96.59 %

2. **Predicting phishing websites based on self-structuring neural network -- Different Datasets Collection**

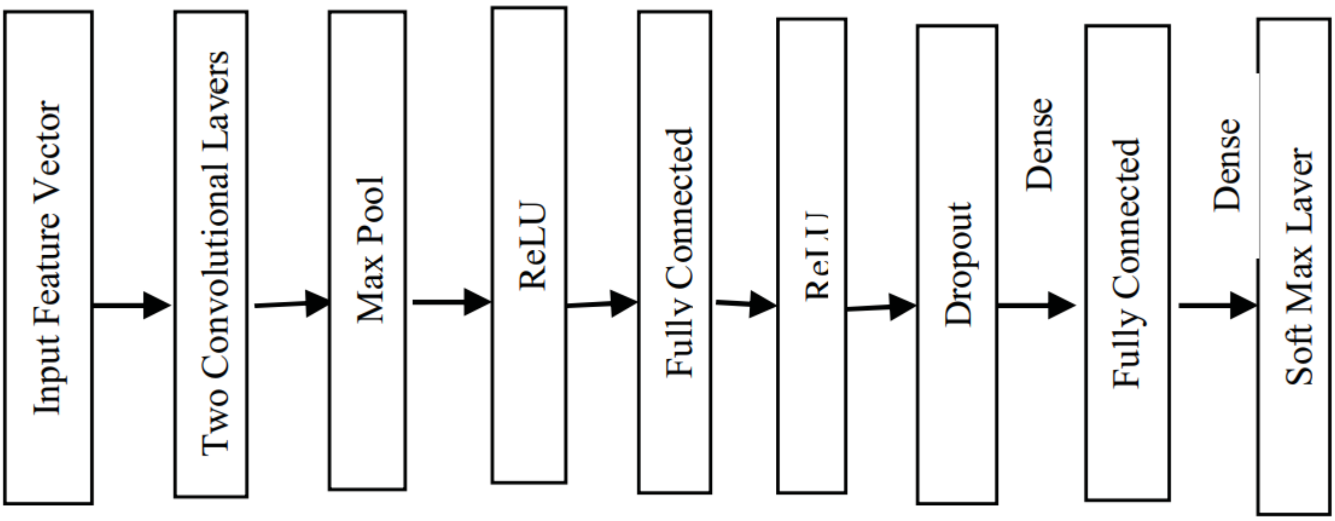


Features = having_IP_Address', 'URL_Length', 'having_At_Symbol', 'Prefix_Suffix', 'Abnormal_URL', 'SFH', 'HTTPS_token', 'Iframe', 'RightClick', 'popUpWidnow', ‘having_Sub_Domain', 'Request_URL', ‘URL_of_Anchor', 'Redirect', 'age_of_domain', 'DNSRecord', 'web_traffic', 'Result'

	Ours	**Papers**
Epoch: 50 -------------------------	**56.09% ---------**	**91.32%**
Epoch: 100 ------------------------	**87.52% ---------**	**92.33%**
Epoch: 200 ------------------------	**89.07% ---------**	**93.07%**
Epoch: 500 ------------------------	**90.90% ---------**	**93.45%**
Epoch: 1000 ------------------------	**92.72% ---------**	**94.07%**



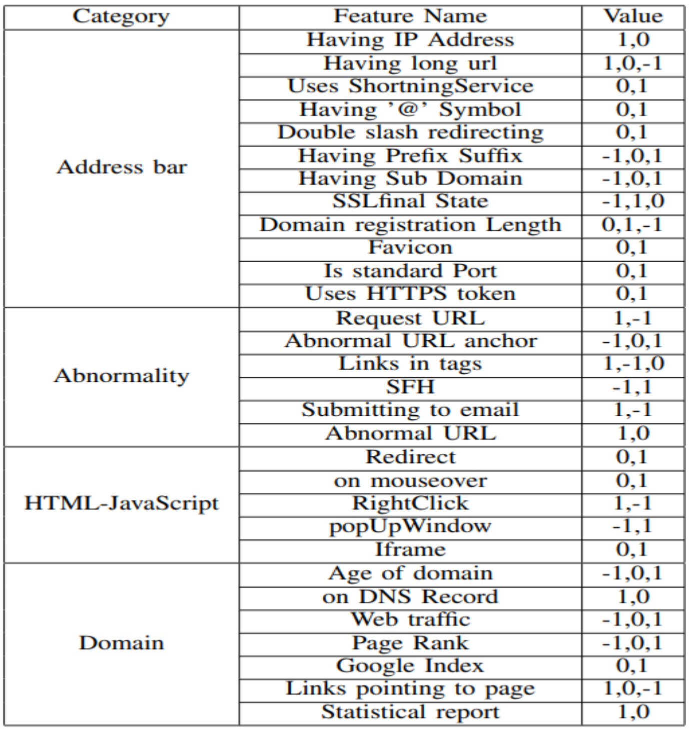
3. Automated Prediction of Phishing Websites Using Deep Convolutional Neural Network – Same Dataset



Accuracy of Paper: 99.3%

Accuracy that we found: 55.43%

4. Phishing Website Detection Using Effective Classifiers and Feature Selection Techniques – Same Dataset



Category	Feature Name	Value
Address bar	Having IP Address	1,0
	Having long url	1,0,-1
	Uses ShortningService	0,1
	Having '@' Symbol	0,1
	Double slash redirecting	0,1
	Having Prefix Suffix	-1,0,1
	Having Sub Domain	-1,0,1
	SSLfinal State	-1,1,0
	Domain registration Length	0,1,-1
	Favicon	0,1
	Is standard Port	0,1
	Uses HTTPS token	0,1
Abnormality	Request URL	1,-1
	Abnormal URL anchor	-1,0,1
	Links in tags	1,-1,0
	SFH	-1,1
	Submitting to email	1,-1
	Abnormal URL	1,0
HTML-JavaScript	Redirect	0,1
	on mouseover	0,1
	RightClick	1,-1
	popUpWindow	-1,1
	Iframe	0,1
Domain	Age of domain	-1,0,1
	on DNS Record	1,0
	Web traffic	-1,0,1
	Page Rank	-1,0,1
	Google Index	0,1
	Links pointing to page	1,0,-1
	Statistical report	1,0

PERFORMANCE OF CLASSIFIERS FOR ADDRESS BAR BASED FEATURES ONLY

Naive Bayes: 89.59%   Paper: 89.95%

Decision Tree: 90.32%   Paper: 90.19%

PERFORMANCE OF CLASSIFIERS FOR ABNORMAL BASED FEATURES ONLY

Naive Bayes: 72.20%   Paper: 88.45%

Decision Tree: 87.63%   Paper: 89.05%

PERFORMANCE OF CLASSIFIERS FOR JAVASCRIPT and HTML BASED FEATURES ONLY

Naive Bayes: 56.01%   Paper: 54.12%

Decision Tree: 57.43%   Paper: 58.02%

PERFORMANCE OF CLASSIFIERS FOR DOMAIN BAR BASED FEATURES ONLY

Naive Bayes: 69.88%   Paper: 80.35%

Decision Tree: 72.74%   Paper: 81.55%

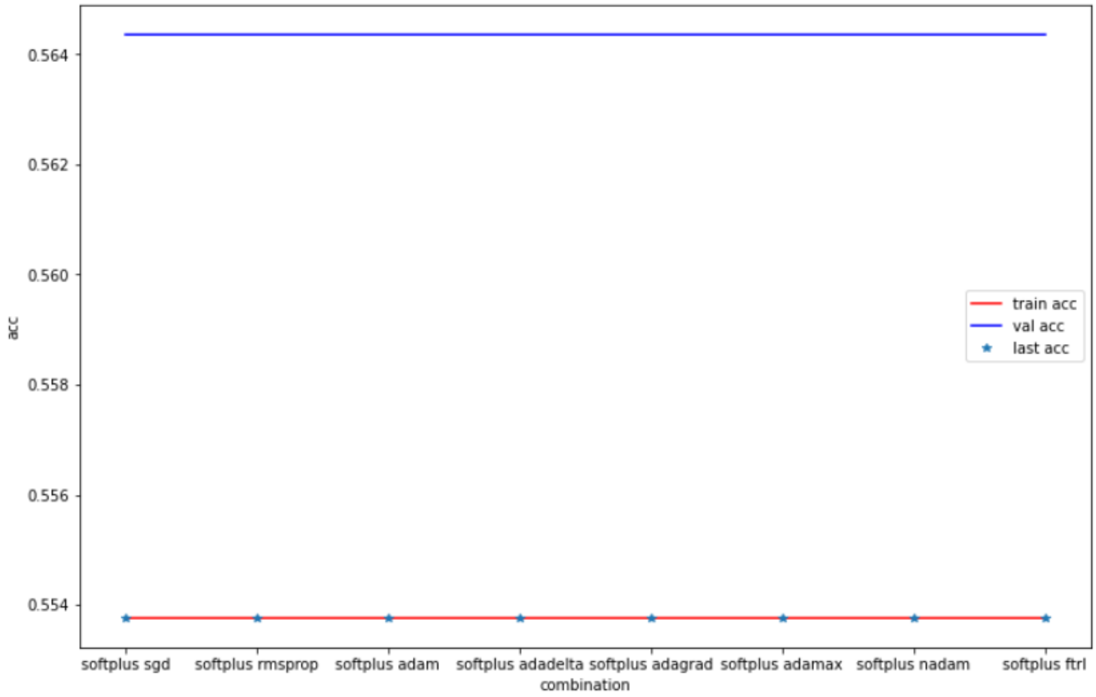
Deep Learning Models

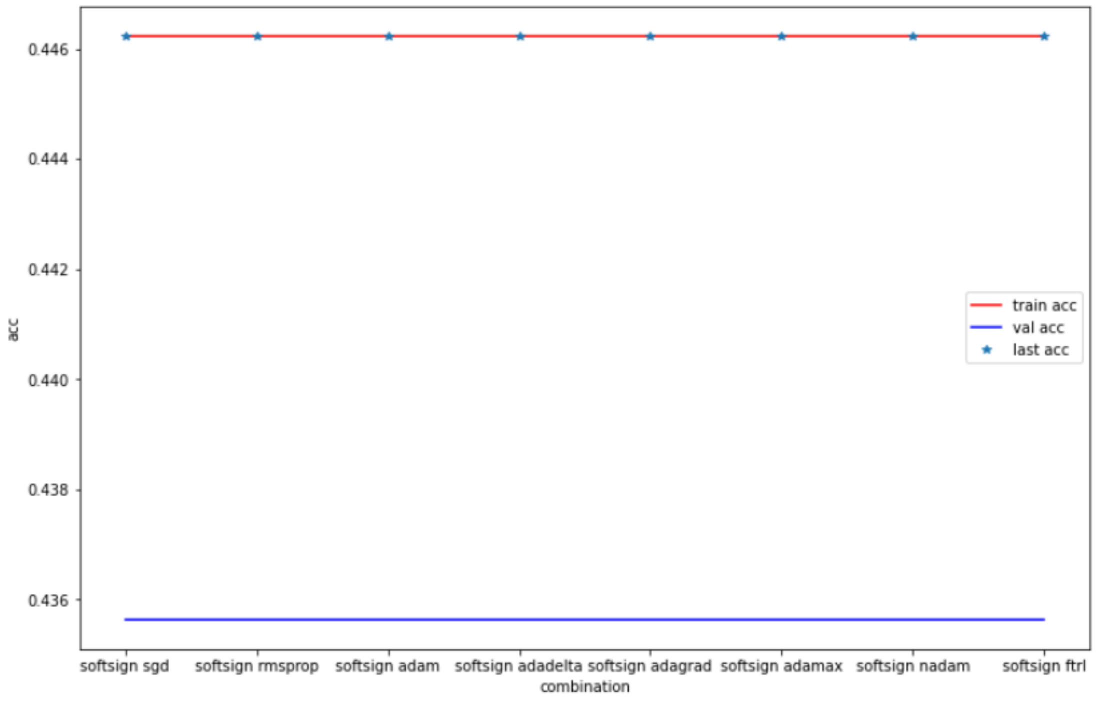
Activation = 'softplus', 'softsign', 'selu', 'elu', 'exponential', 'sigmoid', 'relu', 'tanh'
Optimizers = 'sgd', 'rmsprop', 'adam', 'adadelta', 'adagrad', 'adamax', 'nadam', 'ftrl'
Binary Loss = 'binary_crossentropy'

Model Architecture

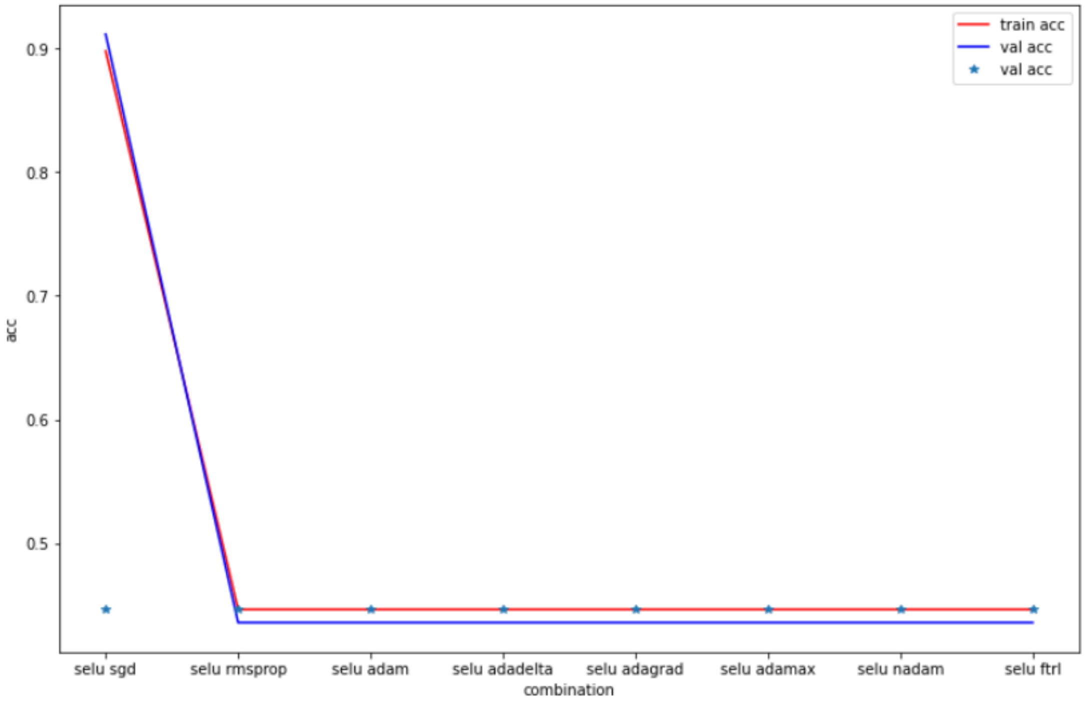
```
Dense(64, input_shape = (30,),  activation='relu')
Dense(128, kernel_regularizer=l2(0.0001), activation='relu')
Dense(128, activation='relu')
Dense(128,  activation='relu')
Dense(1)
```

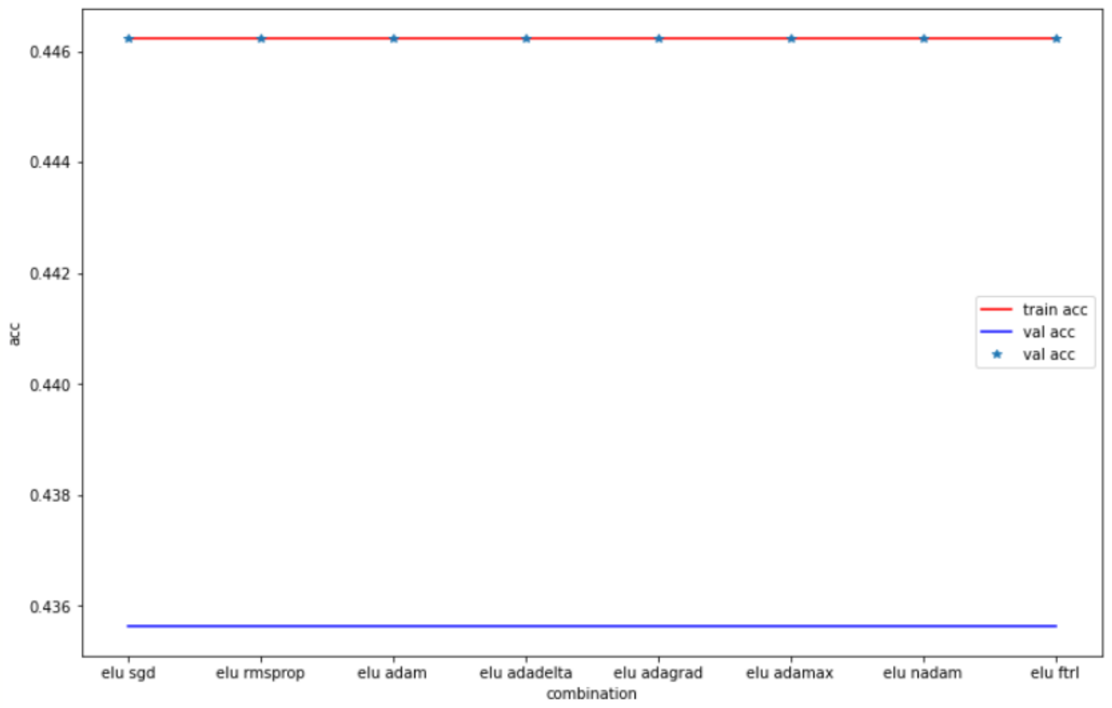
Results:

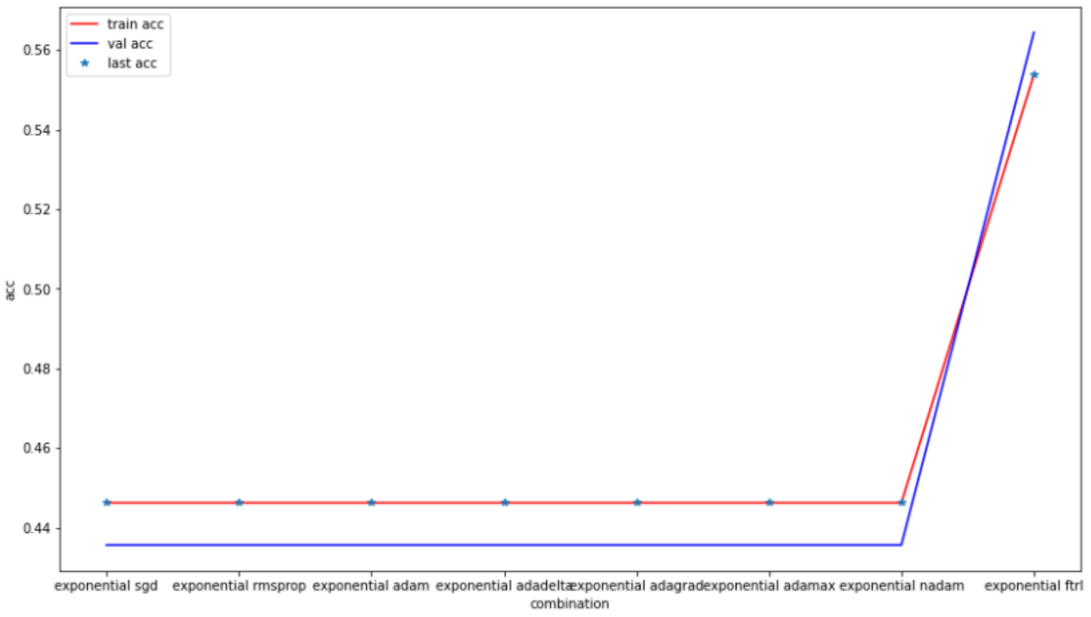


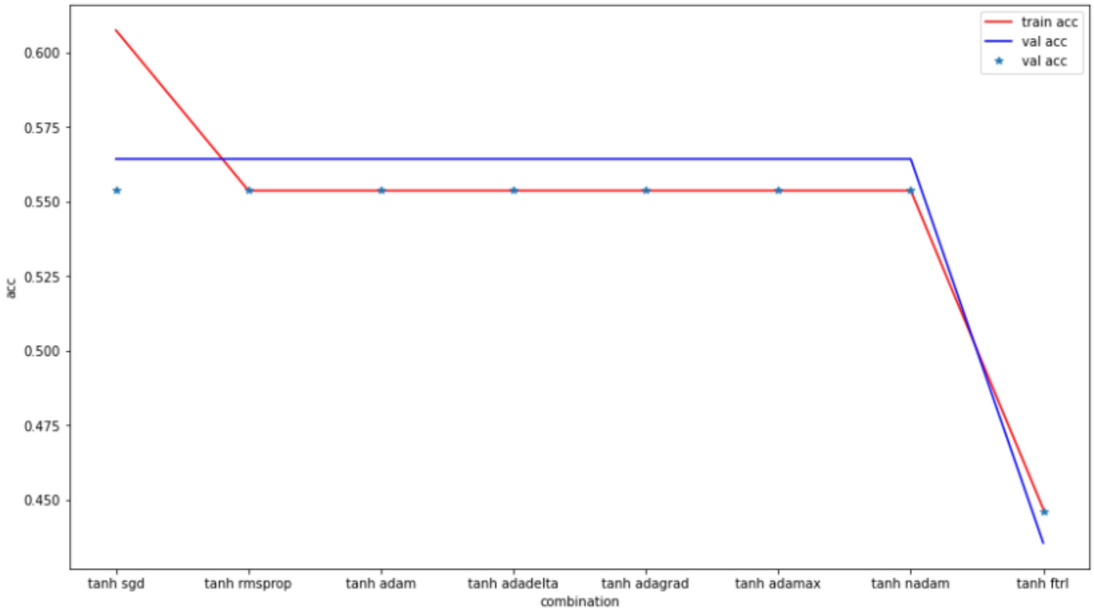

0.446
0.444
0.442
0.440
0.438
0.436
acc
train acc
val acc
last acc
softsign sgd
softsign rmsprop
softsign adam
softsign adadelta
softsign adagrad
softsign adamax
softsign nadam
softsign ftrl
combination

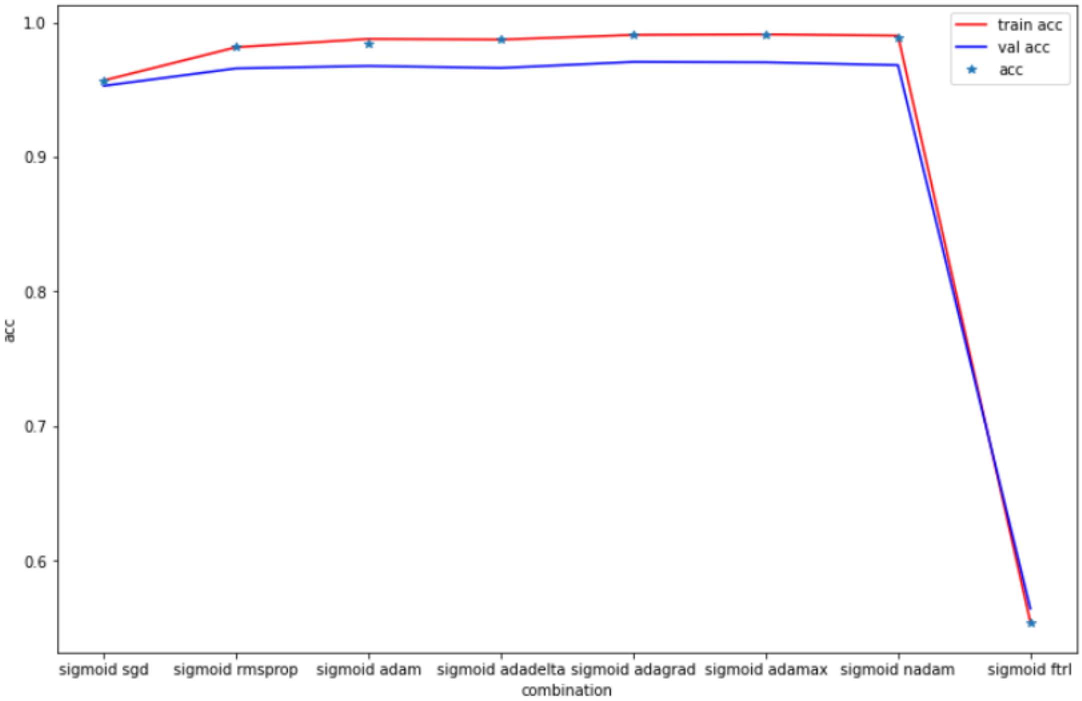
```
<matplotlib.legend.Legend at 0x7f[illegible]>
```

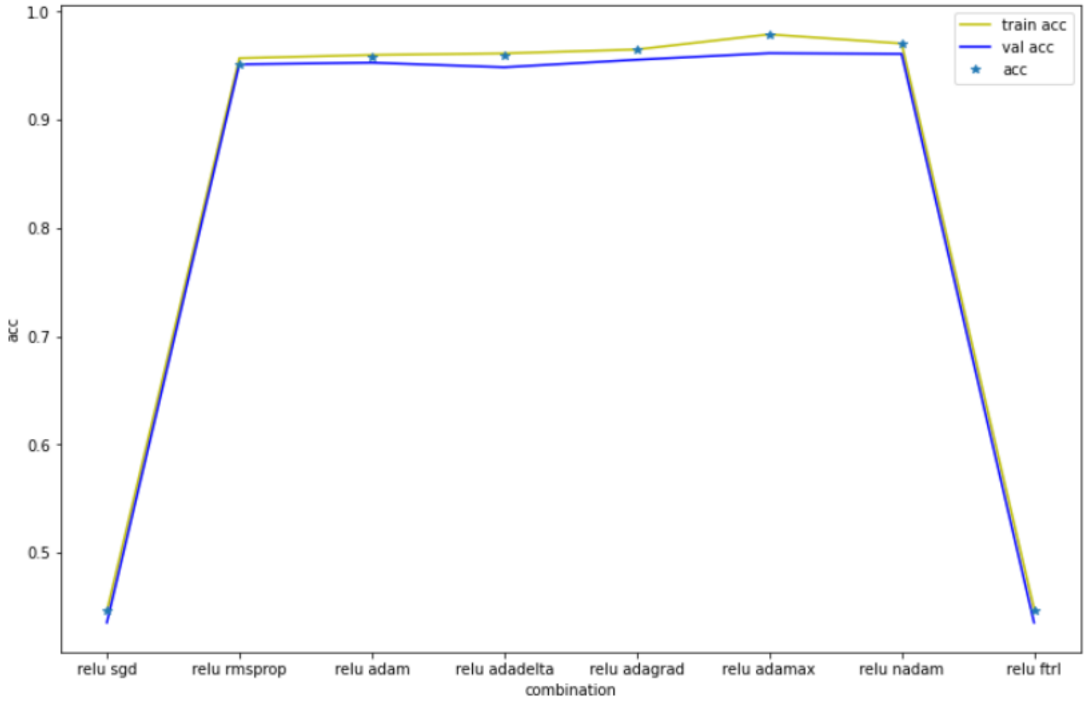

0.9
0.8
0.7
0.6
0.5
acc
train acc
val acc
val acc
selu sgd
selu rmsprop
selu adam
selu adadelta
selu adagrad
selu adamax
selu nadam
selu ftrl
combination

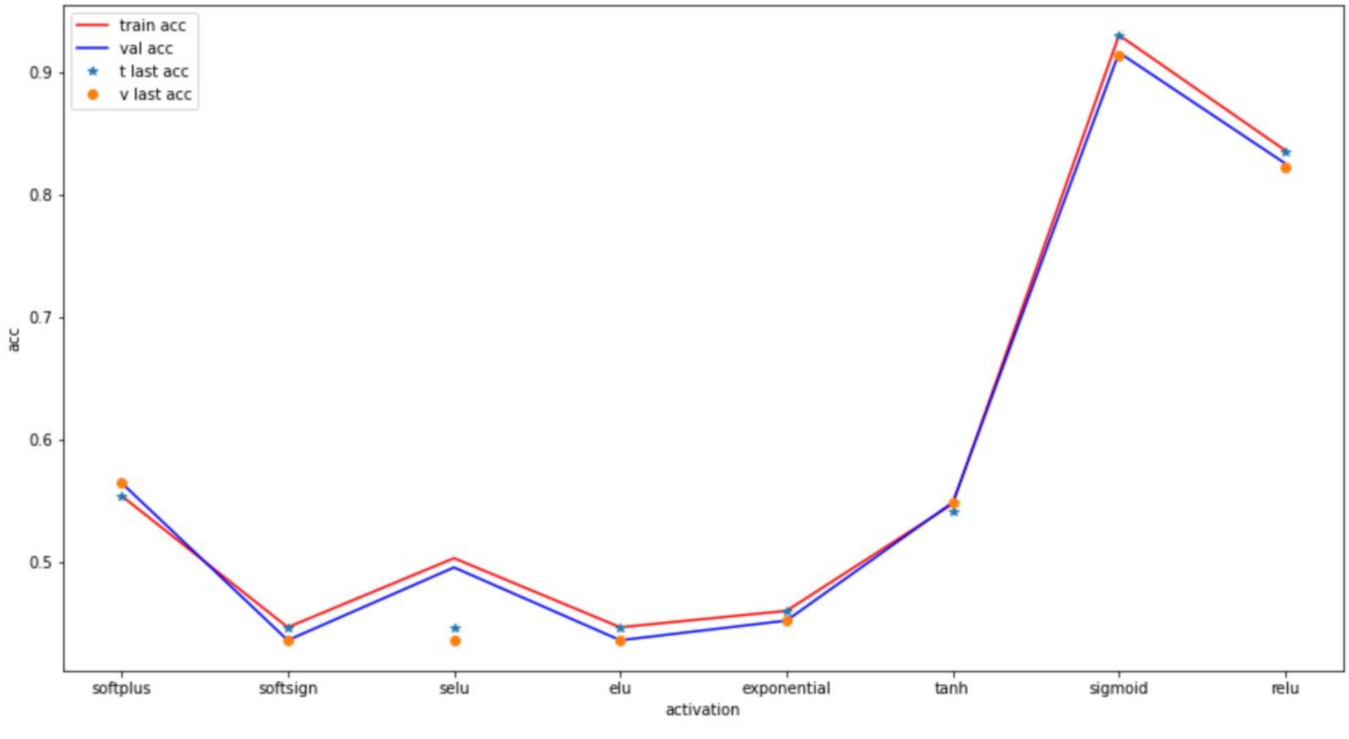

0.446
0.444
0.442
0.440
0.438
0.436
acc
train acc
val acc
val acc
elu sgd
elu rmsprop
elu adam
elu adadelta
elu adagrad
elu adamax
elu nadam
elu ftrl
combination


train acc
val acc
last acc
0.56
0.54
0.52
0.50
0.48
0.46
0.44
acc
exponential sgd
exponential rmsprop
exponential adam
exponential adadelta
exponential adagrad
exponential adamax
exponential nadam
exponential ftrl
combination


train acc
val acc
val acc
acc
0.600
0.575
0.550
0.525
0.500
0.475
0.450
tanh sgd
tanh rmsprop
tanh adam
tanh adadelta
tanh adagrad
tanh adamax
tanh nadam
tanh ftrl
combination

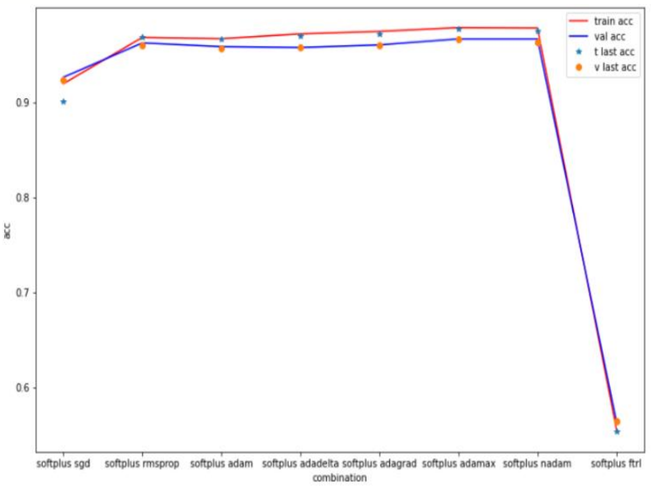

train acc
val acc
acc
acc
1.0
0.9
0.8
0.7
0.6
sigmoid sgd
sigmoid rmsprop
sigmoid adam
sigmoid adadelta
sigmoid adagrad
sigmoid adamax
sigmoid nadam
sigmoid ftrl
combination

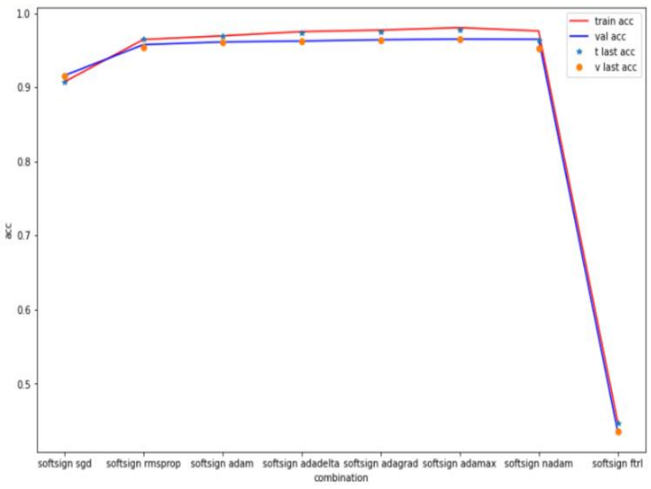

train acc
val acc
acc
acc
1.0
0.9
0.8
0.7
0.6
0.5
relu sgd
relu rmsprop
relu adam
relu adadelta
relu adagrad
relu adamax
relu nadam
relu ftrl
combination

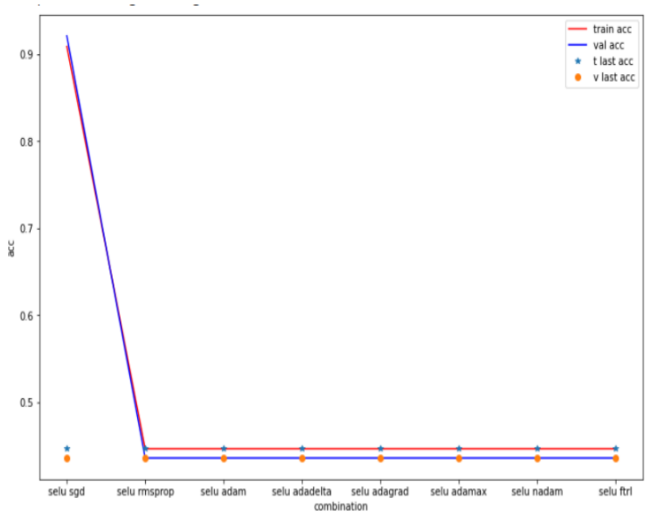

train acc
val acc
t last acc
v last acc
acc
0.9
0.8
0.7
0.6
0.5
softplus
softsign
selu
elu
exponential
tanh
sigmoid
relu
activation

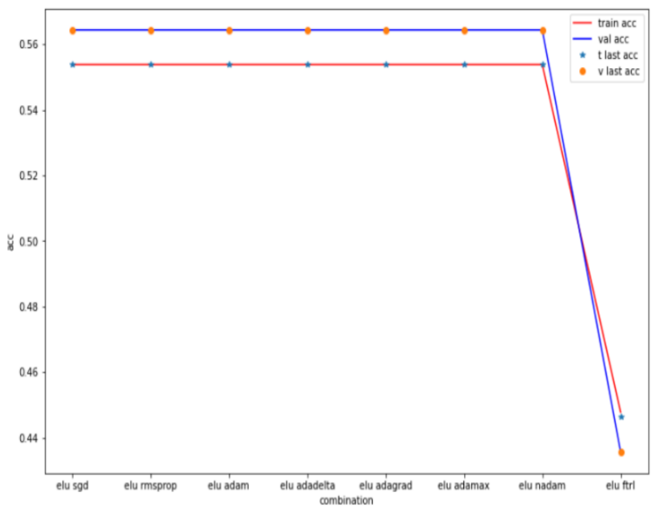
Model Architecture

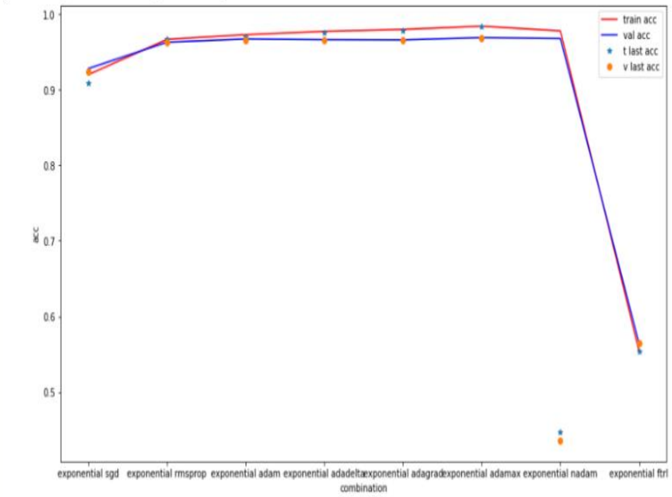
```
keras.layers.LSTM(128, activation='relu',return_sequences=True, input_shape=(1,30))
keras.layers.Dropout(0.2)
keras.layers.LSTM(256, activation='relu')
keras.layers.Dropout(0.2)
keras.layers.Dense(250,activation='relu')
keras.layers.Dense(1)
```

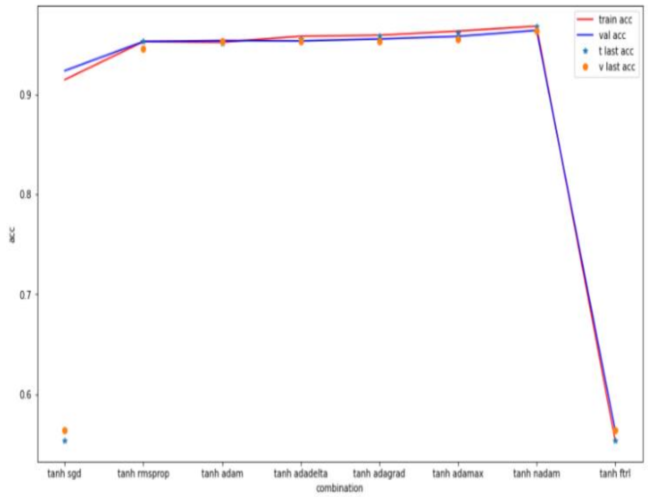


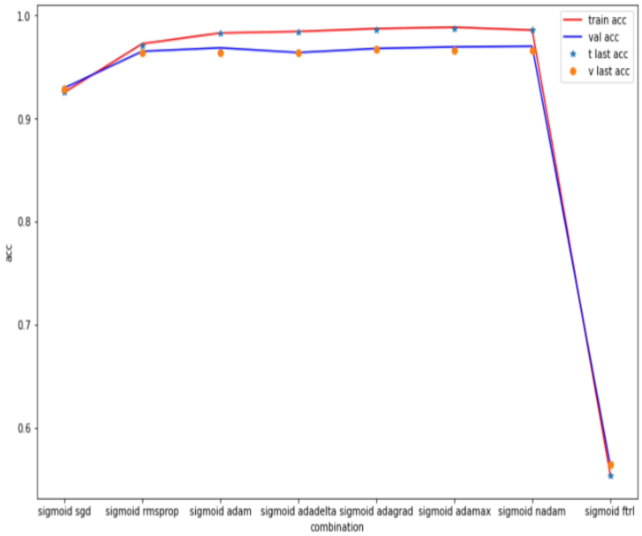


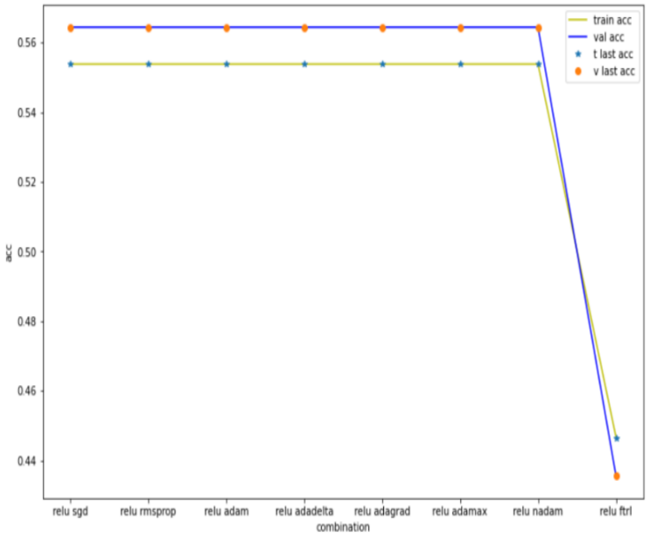


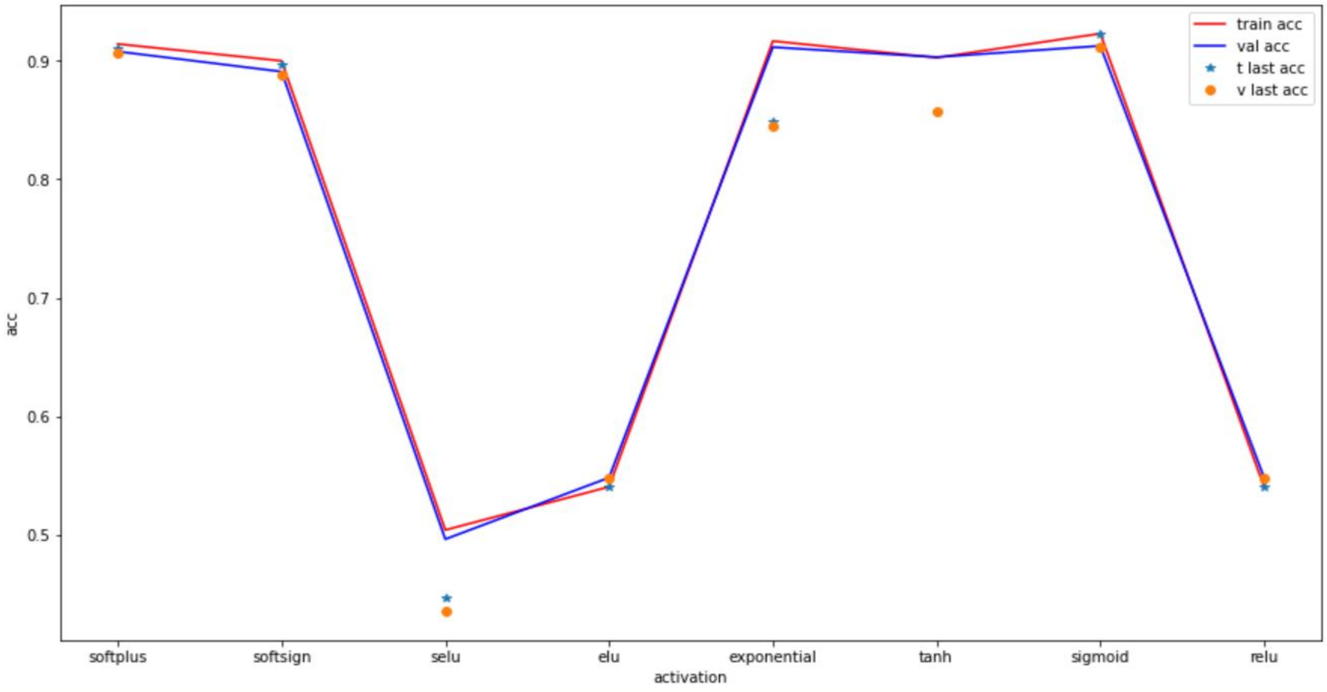



train acc
val acc
t last acc
v last acc
acc
1.0
0.9
0.8
0.7
0.6
0.5
exponential sgd
exponential rmsprop
exponential adam
exponential adadelta
exponential adagrad
exponential adamax
exponential nadam
exponential ftrl
combination


train acc
val acc
t last acc
v last acc
acc
0.9
0.8
0.7
0.6
tanh sgd
tanh rmsprop
tanh adam
tanh adadelta
tanh adagrad
tanh adamax
tanh nadam
tanh ftrl
combination

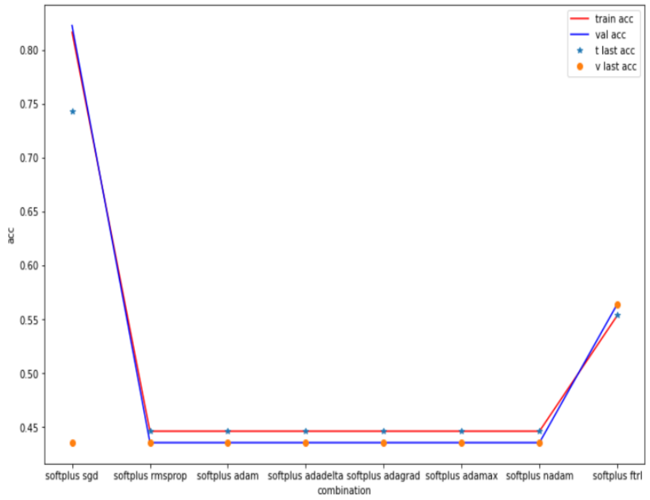

train acc
val acc
t last acc
v last acc
acc
1.0
0.9
0.8
0.7
0.6
sigmoid sgd
sigmoid rmsprop
sigmoid adam
sigmoid adadelta
sigmoid adagrad
sigmoid adamax
sigmoid nadam
sigmoid ftrl
combination

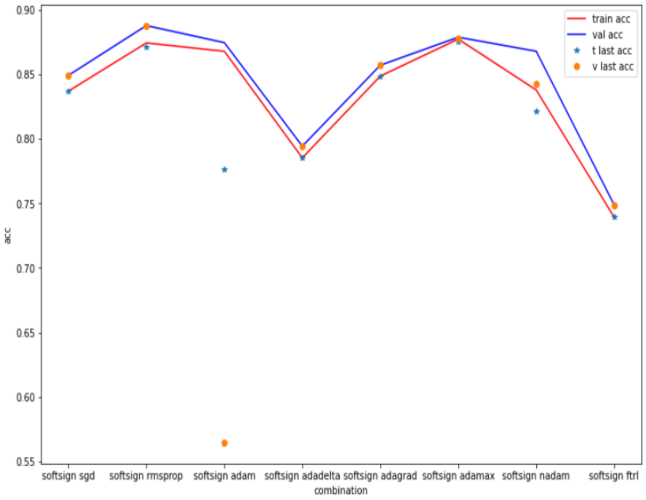

train acc
val acc
t last acc
v last acc
acc
0.56
0.54
0.52
0.50
0.48
0.46
0.44
relu sgd
relu rmsprop
relu adam
relu adadelta
relu adagrad
relu adamax
relu nadam
relu ftrl
combination

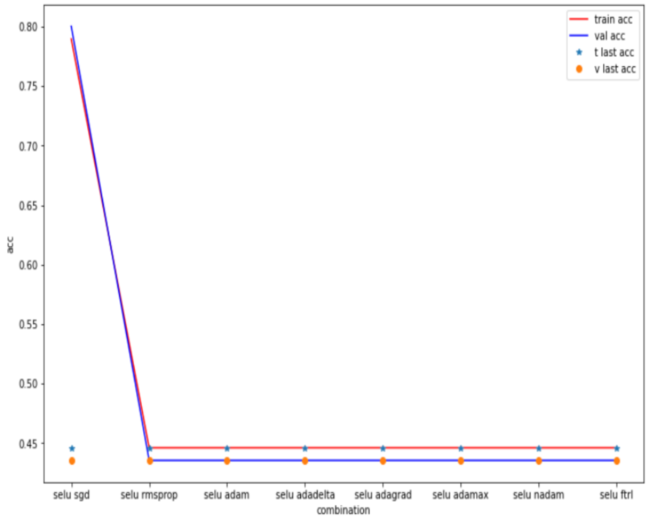

train acc
val acc
t last acc
v last acc
acc
0.9
0.8
0.7
0.6
0.5
softplus
softsign
selu
elu
exponential
tanh
sigmoid
relu
activation

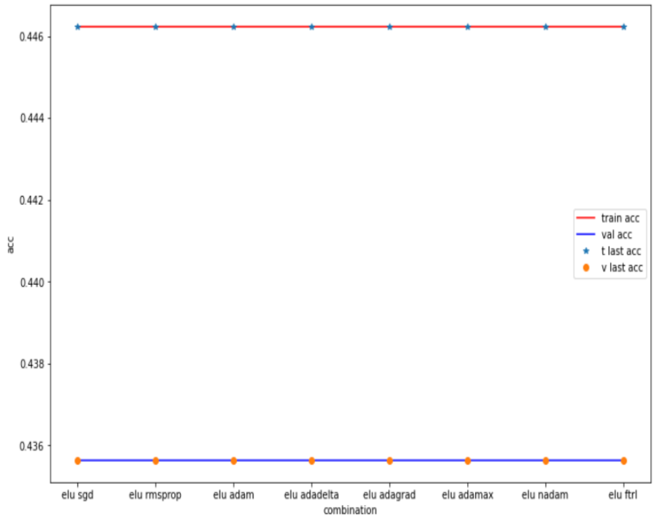
Model Architecture

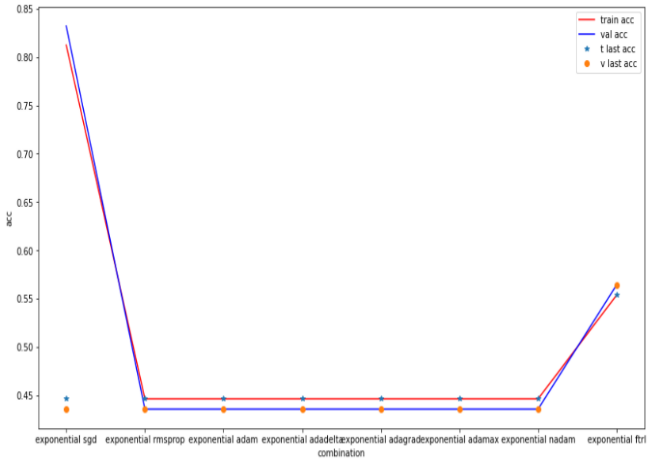
```
keras.layers.Conv2D(32,(1,1), activation='relu',input_shape=(2,5,3))
keras.layers.MaxPool2D(2,2)
keras.layers.Conv2D(64,(1,1),activation='relu')
keras.layers.Conv2D(128,(1,1),activation='relu')
keras.layers.Conv2D(128,(1,1),activation='relu')
keras.layers.Flatten()
keras.layers.Dense(512,activation='relu')
keras.layers.Dense(1)
```

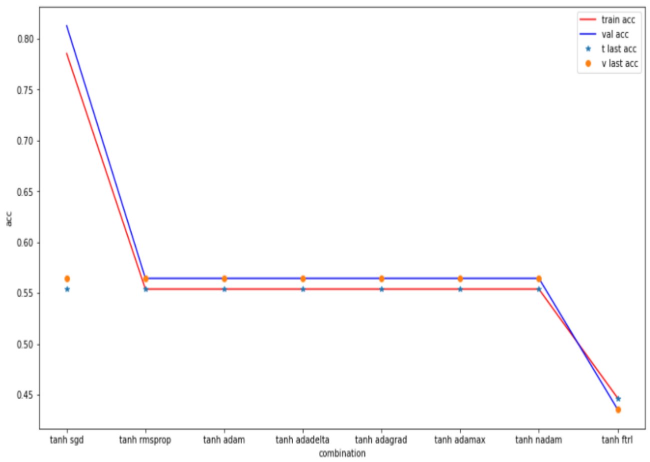


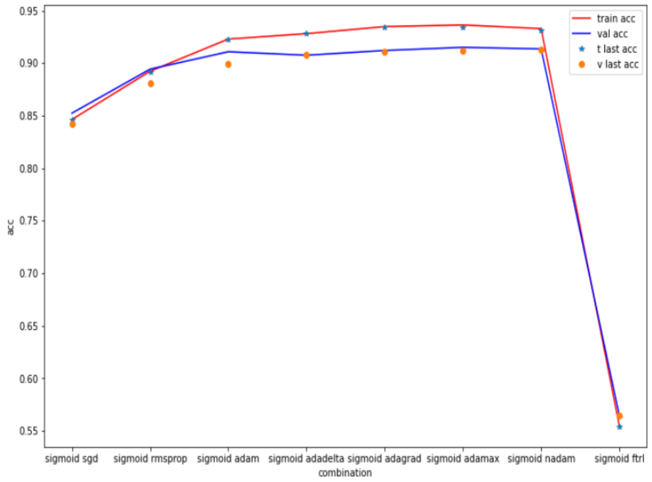


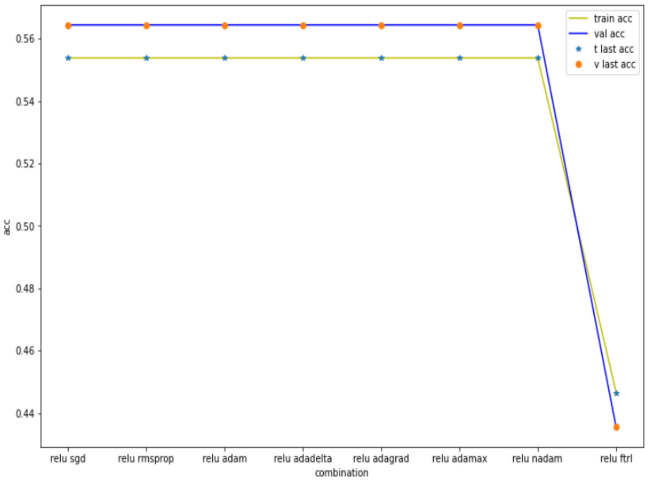


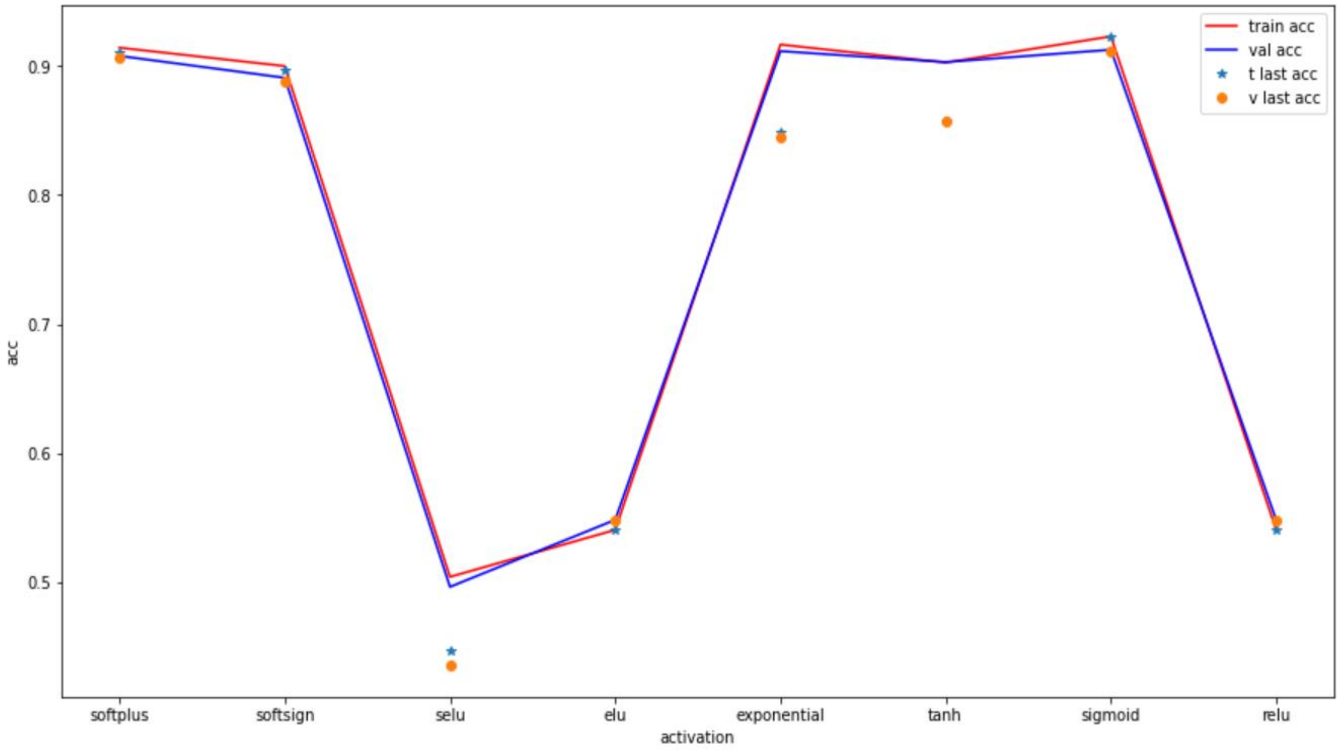



train acc
val acc
t last acc
v last acc
acc
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
exponential sgd
exponential rmsprop
exponential adam
exponential adadelta
exponential adagrad
exponential adamax
exponential nadam
exponential ftrl
combination


train acc
val acc
t last acc
v last acc
acc
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
tanh sgd
tanh rmsprop
tanh adam
tanh adadelta
tanh adagrad
tanh adamax
tanh nadam
tanh ftrl
combination


train acc
val acc
t last acc
v last acc
acc
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
sigmoid sgd
sigmoid rmsprop
sigmoid adam
sigmoid adadelta
sigmoid adagrad
sigmoid adamax
sigmoid nadam
sigmoid ftrl
combination


train acc
val acc
t last acc
v last acc
acc
0.56
0.54
0.52
0.50
0.48
0.46
0.44
relu sgd
relu rmsprop
relu adam
relu adadelta
relu adagrad
relu adamax
relu nadam
relu ftrl
combination


train acc
val acc
t last acc
v last acc
acc
0.9
0.8
0.7
0.6
0.5
softplus
softsign
selu
elu
exponential
tanh
sigmoid
relu
activation