

Supplementary Material

Machine Learning Based Pearl Chain Detection

Input:

Super set- $(Tr_i) = \{Tr_1, Tr_2, \dots, Tr_{10}\}$ Where Tr = Training Sets ($i = 1$ to 10)

Super set $(Te_i) = \{Te_1, Te_2, \dots, Te_{10}\}$ Where Te = Test Sets ($i = 1$ to 10)

Machine Learning Algorithms:

LR, RF, KNN, SVM, NN (ML_i)

Where:

LR-Linear Regression

RF-Random Forest

KNN-Knowledge nearest neighbor

SVM – Support vector machine

NN-Neural Network

Output:

1, Best Approach Algorithm

2, Dominance in Feature Extraction

3, Predicted Values in best approach algorithms.

```
1: BEGIN
2:   FETCH the datasets from source
3:   PREPROCESS the training & test datasets ( $DS_i$ ) as  $Tr_i, Te_i$ 
4:   CONVERT Categorical values ( $C_i$ ) into Numerical values ( $N_i$ )
5:   FOR  $i = 1$  to 10 DO
6:     CHOOSE the target ( $T_i$ ) variable in  $Tr_i$ 
7:     CHOOSE the Original ( $Or_i$ ) variable in  $Te_i$ 
8:     FEED Features of ( $Tr_i$ ) for FEATURE EXTRACTION ( $Fe_i$ )
9:     IDENTIFY the  $Fe_i$  values with Greater Probability to cover maximum  $Fe_i$  values
10:    CHOOSE the Higher Dominance of  $Fe_i$  from  $Tr_i$ 
11:  END FOR
12:  FOR  $j = 1$  to 5 DO
13:    FEED  $Fe_j$  into Machine learning Algorithms ( $ML_j$ )
14:    FIND the Predicted values ( $P_j$ ) from  $ML_j$ 
15:  END FOR
16:  FOR  $i = 1$  to 10 DO
17:    IF  $Or_i == P_i$  THEN
18:       $A_i = 0$ 
19:       $A_i(\text{Accuracy}) = A_i + 1$ 
20:    ELSE IF  $Or_i > P_i || Or_i < P_i$  THEN
21:       $A_i(\text{Accuracy}) = A_i - 1$ 
22:    END IF
23:  END FOR
24:  FOUND ACCURACY ( $A_1, A_2, \dots, A_5$ )
25:  FOR  $i = 1$  to 10 DO
26:     $Gr_i = A_i$ 
27:    IF  $A_i > Gr_i$  THEN
28:       $Gr_i = A_i$ 
29:    EVALUATE greater ( $Gr_i$ ) for the best Machine Learning algorithms
30:  END IF
31: END FOR
32: END
```

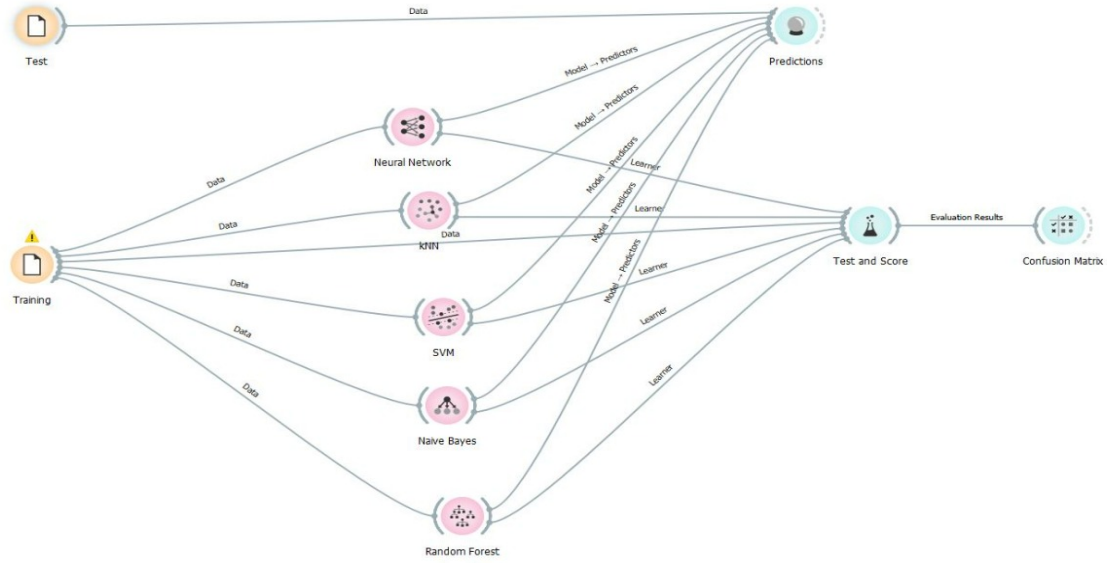


Fig. S1. Applying different machine learning algorithms on the pearl chain dataset using Orange
Input / Output

As input for datasets, we have a training set and a test set (Tr_i & Te_i), respectively. The prediction results are based on five machine learning algorithms. It finds dominant values in feature extracted, best approach algorithms among machine learning algorithms, and the best way to predict the values of test sets.

Preprocessing and Higher Dominance

Initially fetch the training and test datasets from the source. Preprocess the dataset by converting categorical values (C_i) into numerical values (N_i). Identify the target feature (T_i) from the training sets (Tr_i). Find the original predicting feature (Or_i) from the test sets (Te_i). Use the training sets (Tr_i) of feature extraction (Fe_i). Identify the Fe_i values with Greater probability to cover maximum features values. Finally, find the higher dominance features (Fe_i) from the training sets (Tr_i).

Accuracy Findings

Feed the higher dominance feature extracted values (Fe_j) into 5 Machine learning algorithms (ML_j). Comparing the accuracy of all algorithms based on the original value (Or_i) present in the test set and the predicted values (P_j) found using all algorithms (ML_j). Analyze and choose the most accurate machine learning algorithms based on the datasets collected from the source.

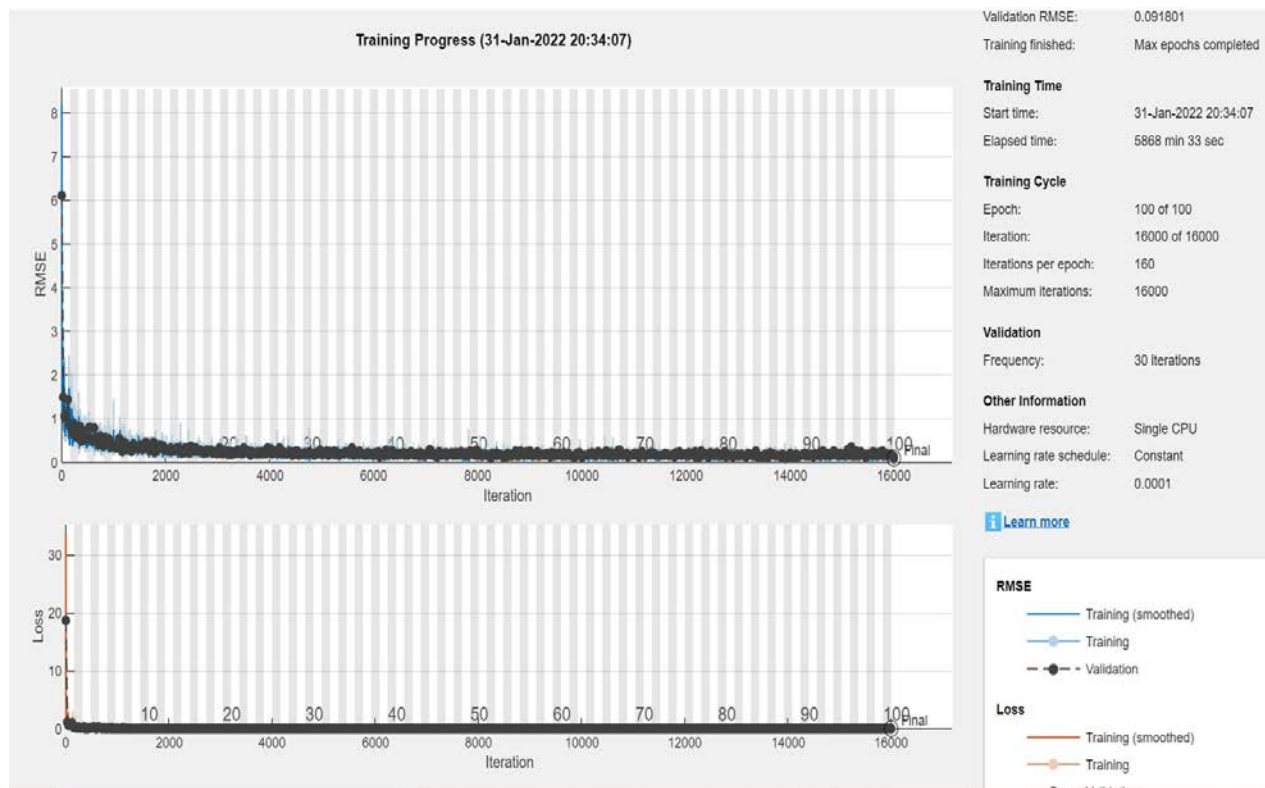


Fig. S2. Training Progress shows the Validation RMSE of AlexNet with ADAM

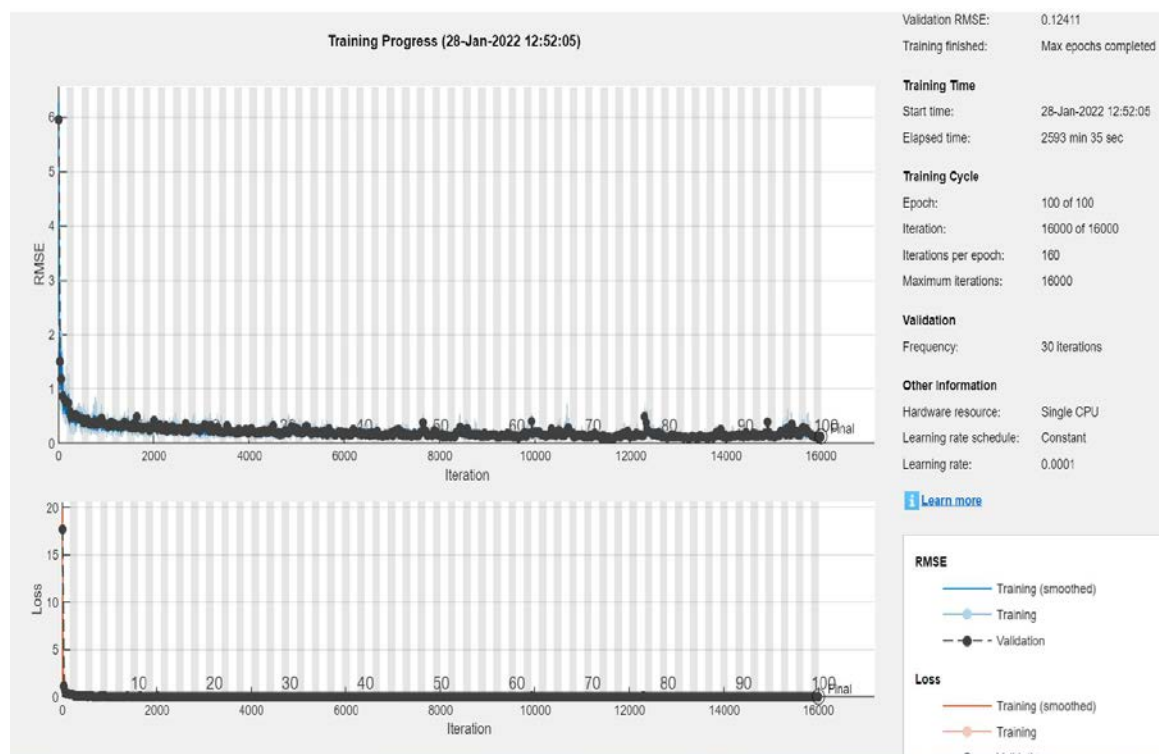


Fig. S3. Training Progress shows the Validation RMSE of ResNet-50 with RMSProp

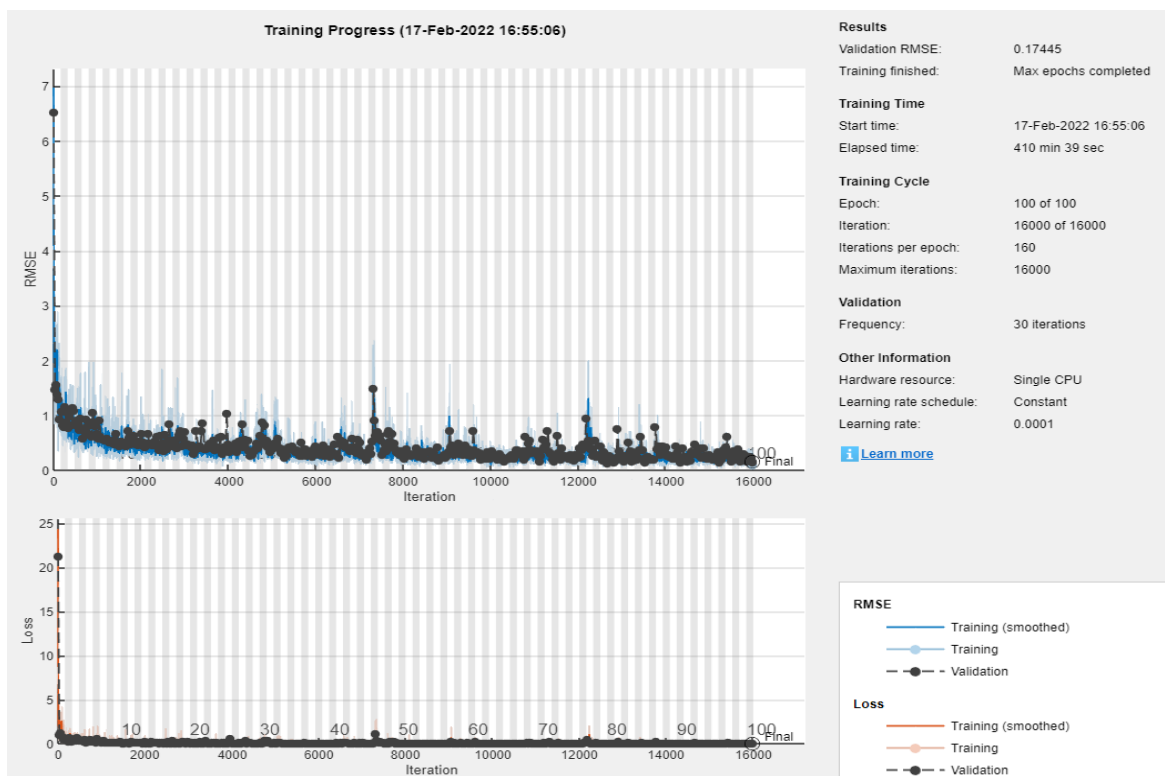


Fig. S4. Training Progress shows the Validation RMSE of AlexNet with ADAM

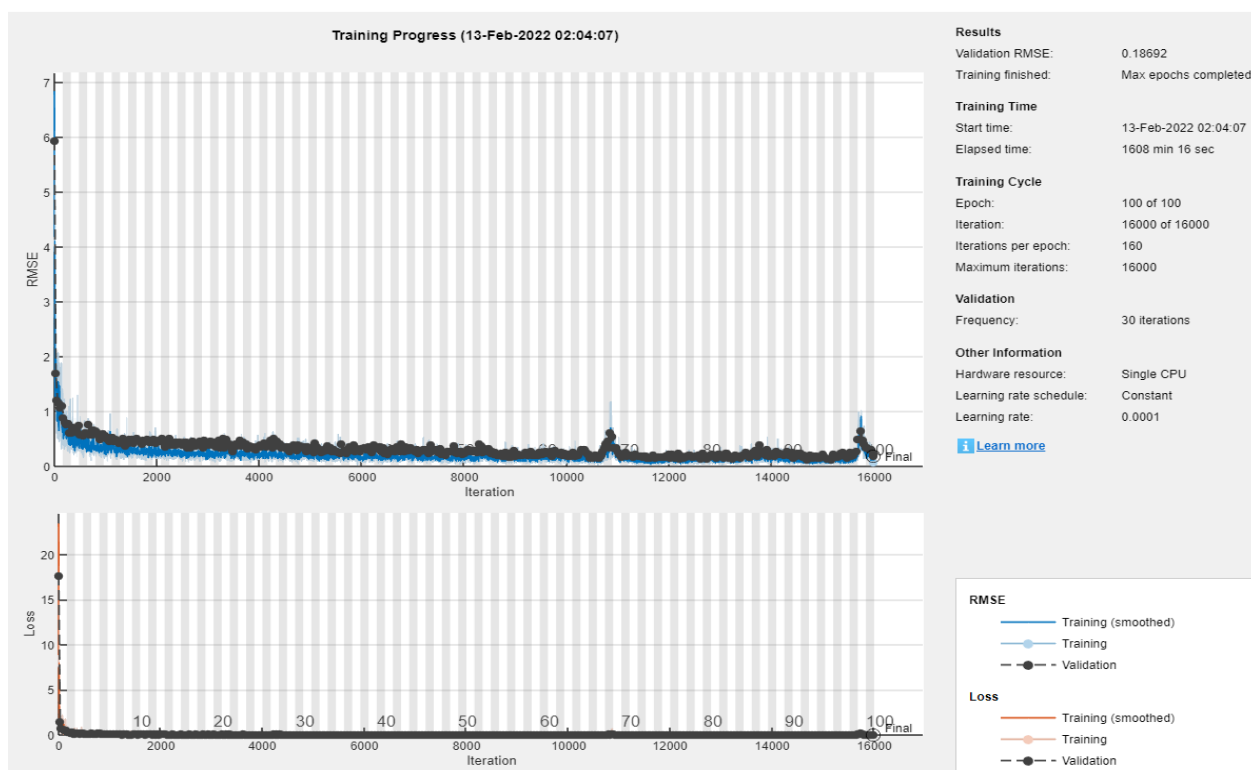


Fig. S5. Training Progress shows the Validation RMSE of ResNet-50 with ADAM optimizer

Table S1. CNN Deep Regression Model Performance on Yeast Cells for various Architectures

Architecture	Optimizer	Accuracy	MRE
AlexNet	ADAM	0.9550	0.0551
	SGDM	0.9575	0.0541
	RMSProp	0.9750	0.0191
ResNet-50	ADAM	0.9925	0.0210
	SGDM	0.9825	0.0452
	RMSProp	0.9925	0.0168
MobileNetV2	ADAM	0.9925	0.0255
	SGDM	0.9525	0.0562
	RMSProp	0.9975	0.0187
GoogLeNet	ADAM	0.9950	0.0247
	SGDM	0.9900	0.0306
	RMSProp	0.9925	0.0217

Table S2. CNN Deep Regression Model Performance on PS Microbeads for various Architectures.

Architecture	Optimizer	Accuracy	MRE
AlexNet	ADAM	0.9675	0.0257
	SGDM	0.8875	0.0803
	RMSProp	0.9750	0.0411
ResNet-50	ADAM	0.9600	0.0268
	SGDM	0.9200	0.0500
	RMSProp	0.9075	0.0379
MobileNetV2	ADAM	0.9075	0.0339
	SGDM	0.9125	0.0682
	RMSProp	0.9250	0.0204
GoogLeNet	ADAM	0.9750	0.0245
	SGDM	0.9225	0.0411
	RMSProp	0.9500	0.0230