**5. Ranking for Machine Learning Performances for Each BNF**

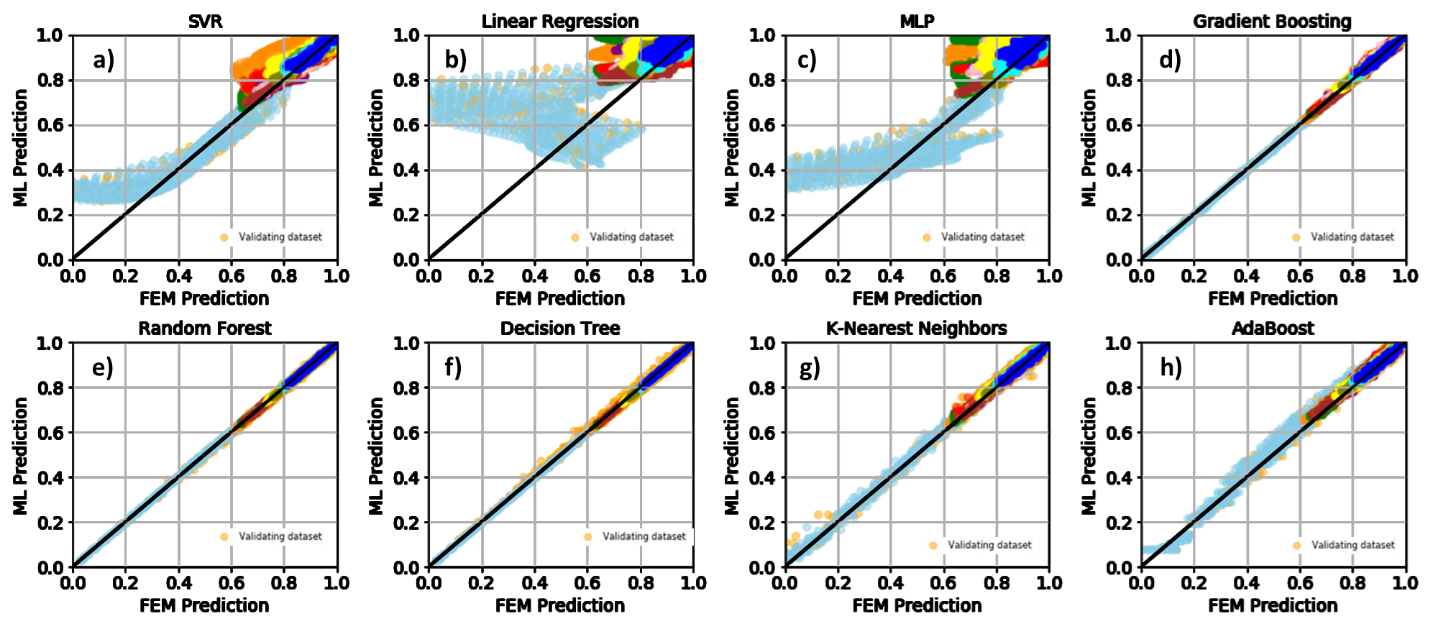
Ranking and recommending the best prediction for BNFs for each algorithm is based on R2 and RSME measurement and all BNFs are provided in Supporting Data Table and Supporting Data Plots in the GitHub folder of *“EMIT-all-BNFs-ML-Performances.docx“*. The ML setup is based on a fixed training split of 70%, a validation split of 15%, and a testing split of 15% with the for each ML model mentioned above. The selected features in the training are BNF class, VF, and reduced wave vectors.

**Table A1**. R2 score for different BNFs for low features training (yellow: best performance)

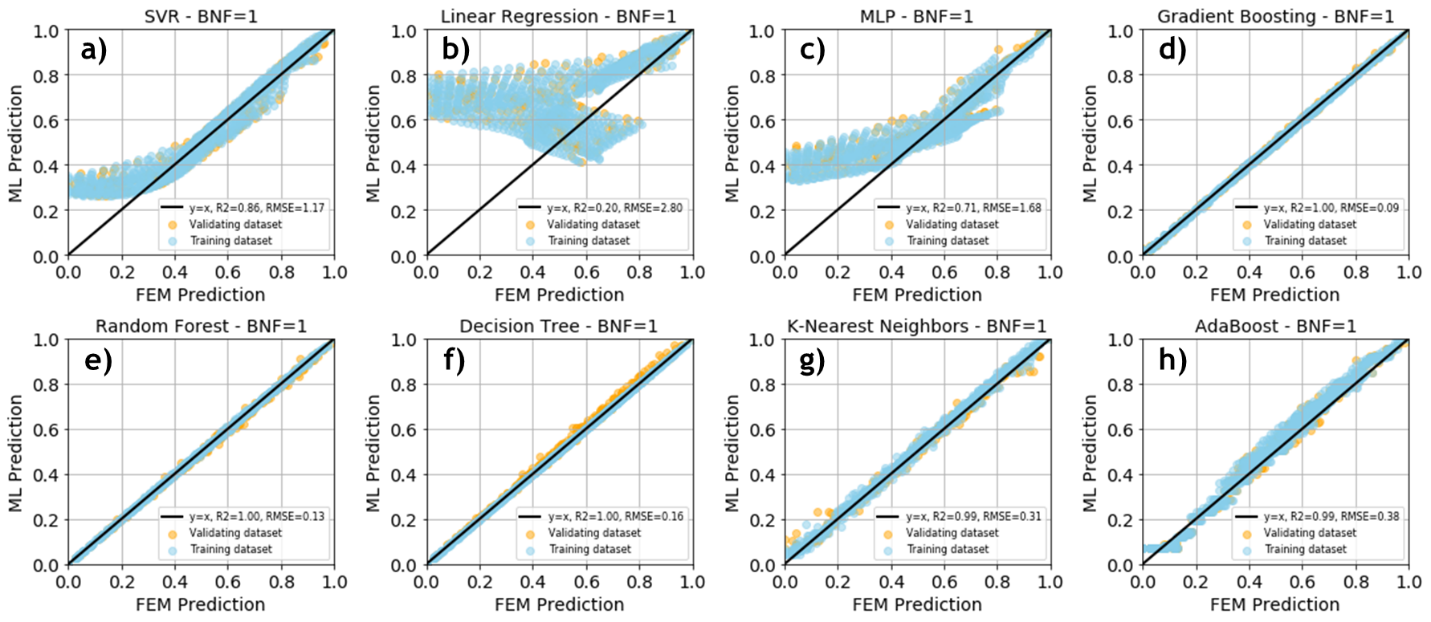
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BNF** | **SVR** | **LR** | **MLP** | **GB** | **RF** | **DT** | **KNN** | **AB** |
| **1** | 0.859287 | 0.196861 | 0.719917 | 0.999078 | 0.998404 | 0.997414 | 0.989877 | 0.986203 |
| **2** | 0.521595 | 0.043459 | 0.04135 | 0.998225 | 0.99824 | 0.997471 | 0.982737 | 0.972518 |
| **3** | 0.950707 | 0.251069 | 0.487641 | 0.998565 | 0.99902 | 0.998783 | 0.993494 | 0.991624 |
| **4** | 0.701809 | 0.385748 | 0.386515 | 0.977437 | 0.993471 | 0.989097 | 0.968029 | 0.937439 |
| **5** | 0.849483 | 0.122399 | 0.284297 | 0.968065 | 0.993972 | 0.992988 | 0.980753 | 0.948555 |
| **6** | 0.517968 | 0.290618 | 0.318174 | 0.966832 | 0.993622 | 0.989436 | 0.964319 | 0.941528 |
| **7** | 0.517968 | 0.993622 | 0.318174 | 0.966832 | 0.993622 | 0.989436 | 0.964319 | 0.941528 |
| **8** | 0.580401 | 0.384939 | 0.413199 | 0.972886 | 0.990984 | 0.984374 | 0.955169 | 0.927915 |
| **9** | 0.870744 | 0.571194 | 0.770362 | 0.982817 | 0.995858 | 0.993308 | 0.981369 | 0.954532 |
| **10** | 0.885919 | 0.551281 | 0.742104 | 0.990749 | 0.996732 | 0.993852 | 0.982177 | 0.9668 |
| **11** | 0.892152 | 0.382491 | 0.684151 | 0.968902 | 0.994746 | 0.990283 | 0.981498 | 0.946899 |
| **12** | 0.861669 | 0.465734 | 0.718946 | 0.982246 | 0.992317 | 0.989106 | 0.974773 | 0.937612 |
| **All** | 0.750808 | 0.386618 | 0.490402 | 0.981053 | 0.995082 | 0.992129 | 0.976543 | 0.954429 |

**Table A2.** Root mean square error (RSME) for different BNFs for low features training

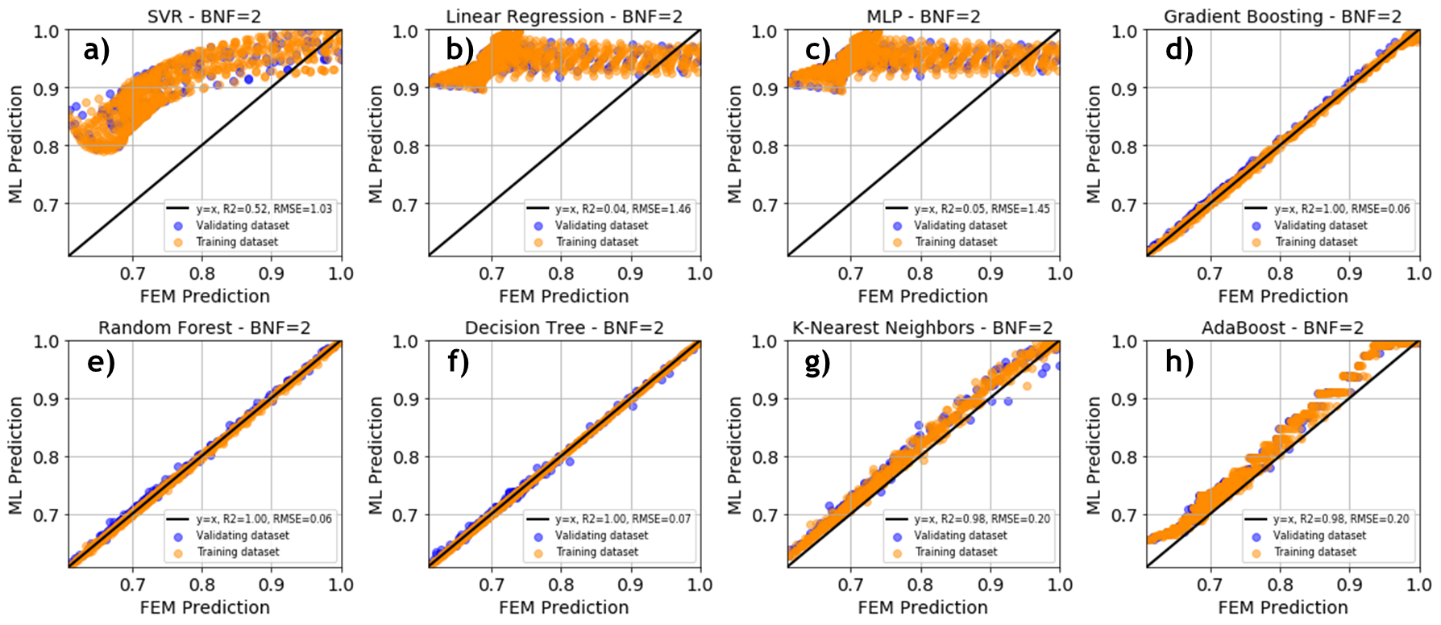
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BNF** | **SVR** | **LR** | **MLP** | **GB** | **RF** | **DT** | **KNN** | **AB** |
| **1** | 2.799875 | 1.171953 | 1.123507 | 0.094866 | 0.118675 | 0.158883 | 0.314334 | 0.372453 |
| **2** | 1.029858 | 1.456234 | 1.450738 | 0.062737 | 0.057753 | 0.074881 | 0.195633 | 0.200450 |
| **3** | 0.455370 | 1.774966 | 1.775372 | 0.077695 | 0.068446 | 0.071561 | 0.165437 | 0.184973 |
| **4** | 0.697648 | 1.001296 | 0.947905 | 0.191905 | 0.106030 | 0.133107 | 0.228438 | 0.320587 |
| **5** | 0.413242 | 0.997840 | 0.809122 | 0.190348 | 0.076153 | 0.089191 | 0.147771 | 0.252175 |
| **6** | 0.843595 | 1.023378 | 0.996336 | 0.221288 | 0.100215 | 0.123921 | 0.229516 | 0.300200 |
| **7** | 0.474297 | 1.110877 | 1.095903 | 0.122384 | 0.102666 | 0.125072 | 0.161223 | 0.320187 |
| **8** | 0.914224 | 1.106865 | 1.090657 | 0.232399 | 0.126107 | 0.176432 | 0.298831 | 0.372709 |
| **9** | 0.491773 | 0.895716 | 0.524801 | 0.179306 | 0.091243 | 0.111234 | 0.186707 | 0.276188 |
| **10** | 0.385026 | 0.763609 | 0.494798 | 0.109644 | 0.063350 | 0.091625 | 0.152187 | 0.205020 |
| **11** | 0.391257 | 0.936219 | 0.614560 | 0.210098 | 0.088446 | 0.117442 | 0.162054 | 0.297273 |
| **12** | 0.457175 | 0.898466 | 0.651655 | 0.163783 | 0.107740 | 0.128297 | 0.195233 | 0.307025 |
| **All** | 0.779445 | 1.094785 | 0.964613 | 0.154704 | 0.092235 | 0.116804 | 0.203114 | 0.284103 |



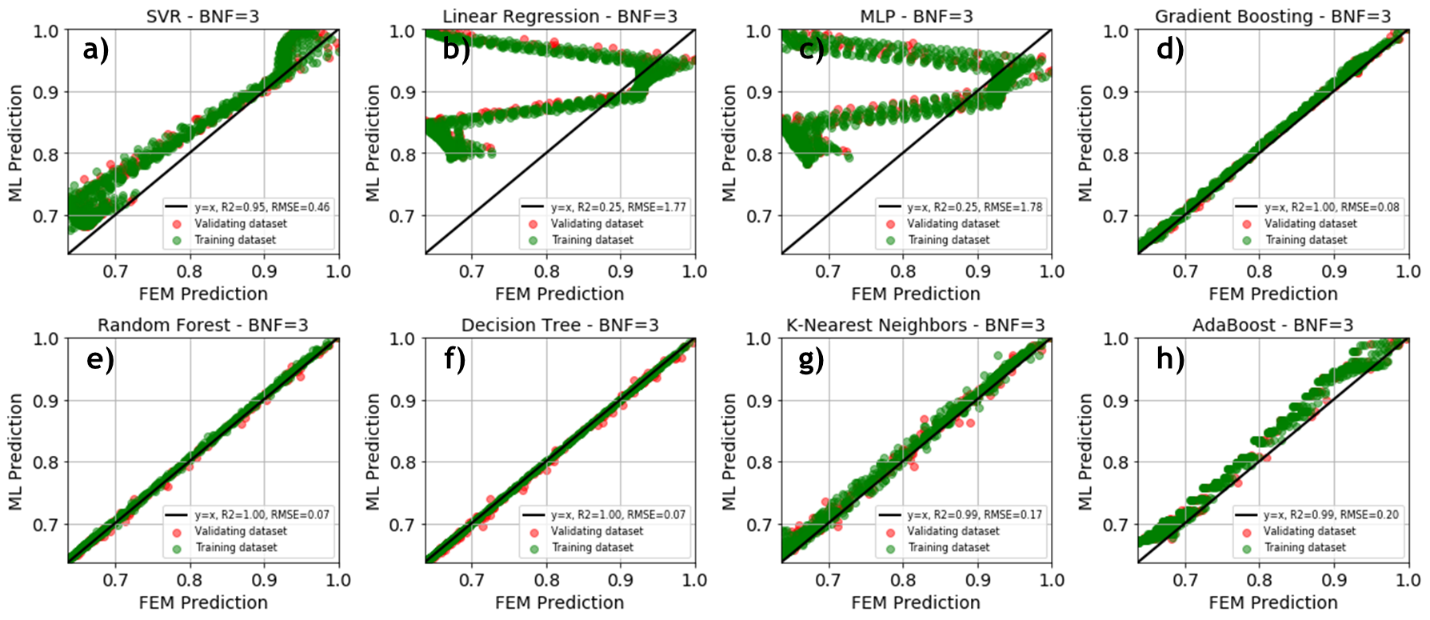
Different BNFs subclusters influence overall ML accuracy for **a)** support vector regression, **b)** linear regression, **c)** multi-layer-perceptrons, **d)** gradient boosting, **e)** random first, **f)** decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

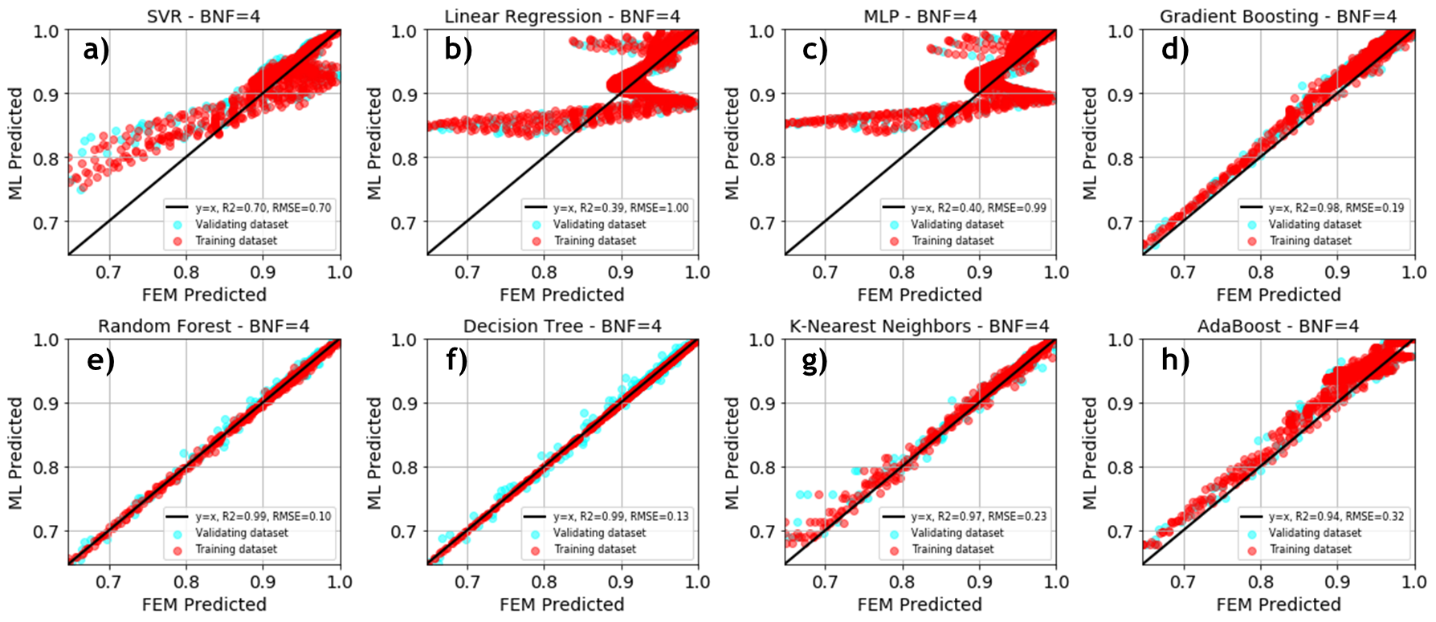
Training result and validation of BNF=1 subcluster for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

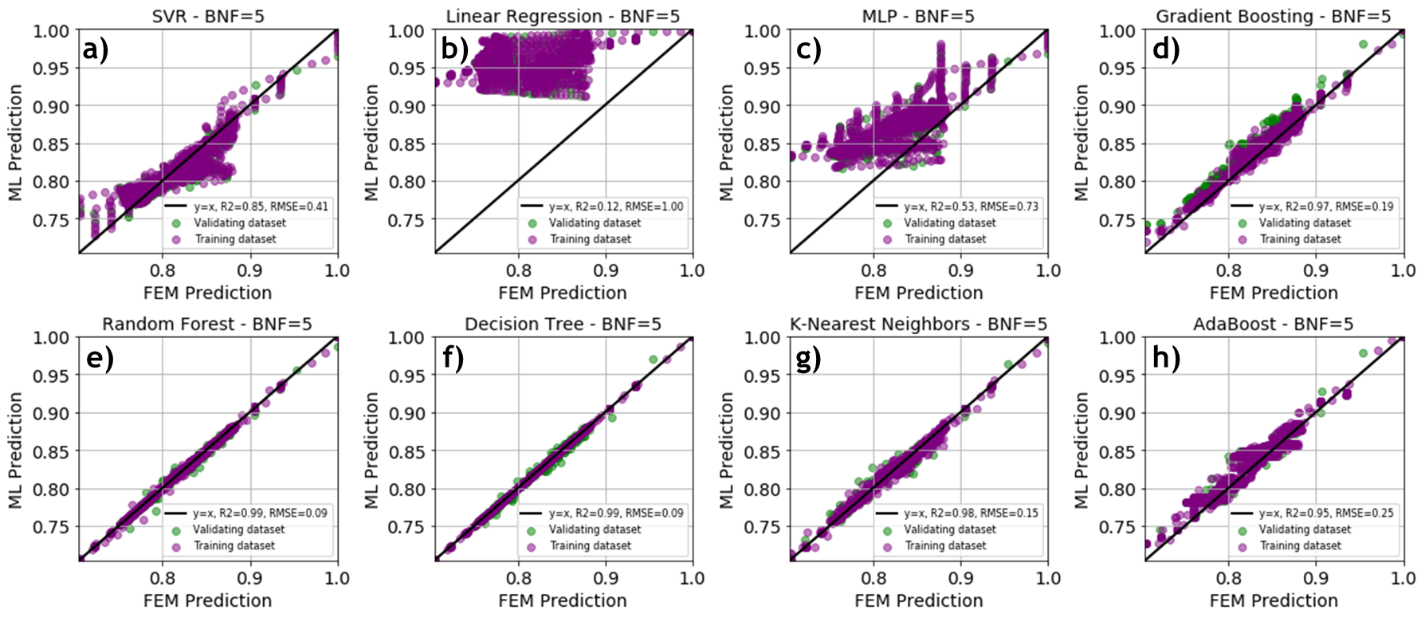
Training and validation results of BNF=2 subcluster for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

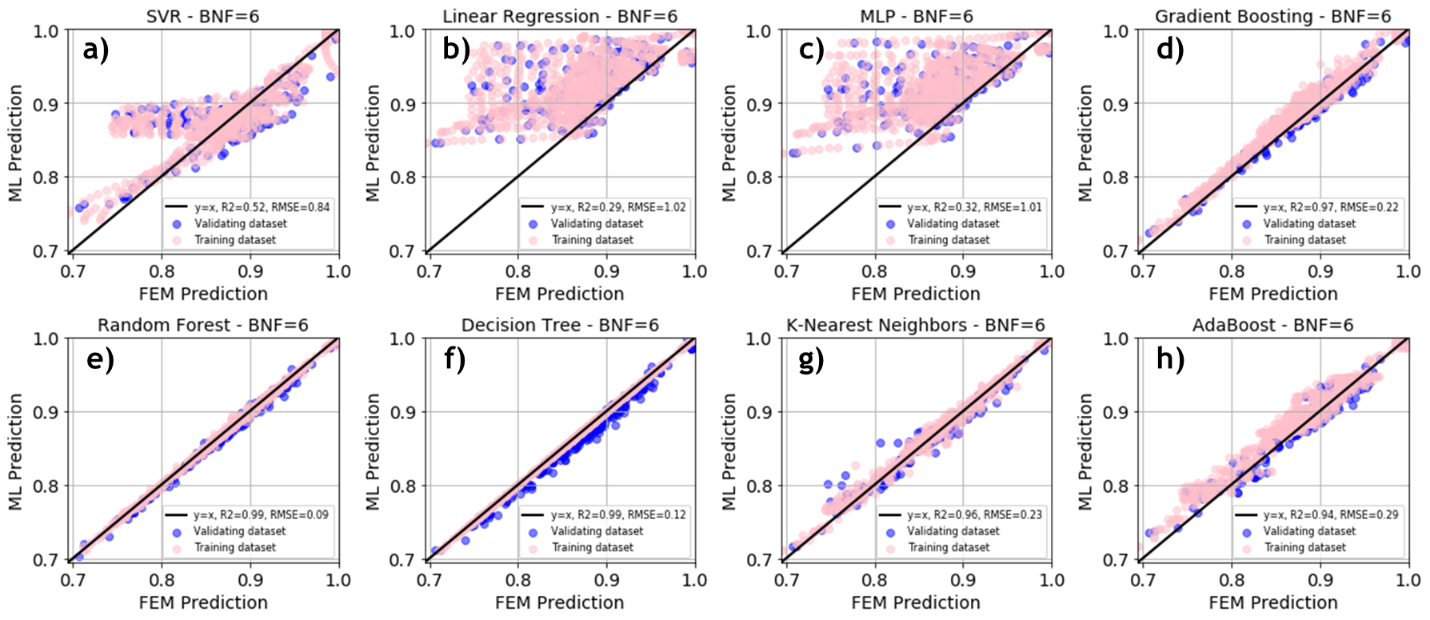
Training and validation results of BNF=3 for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

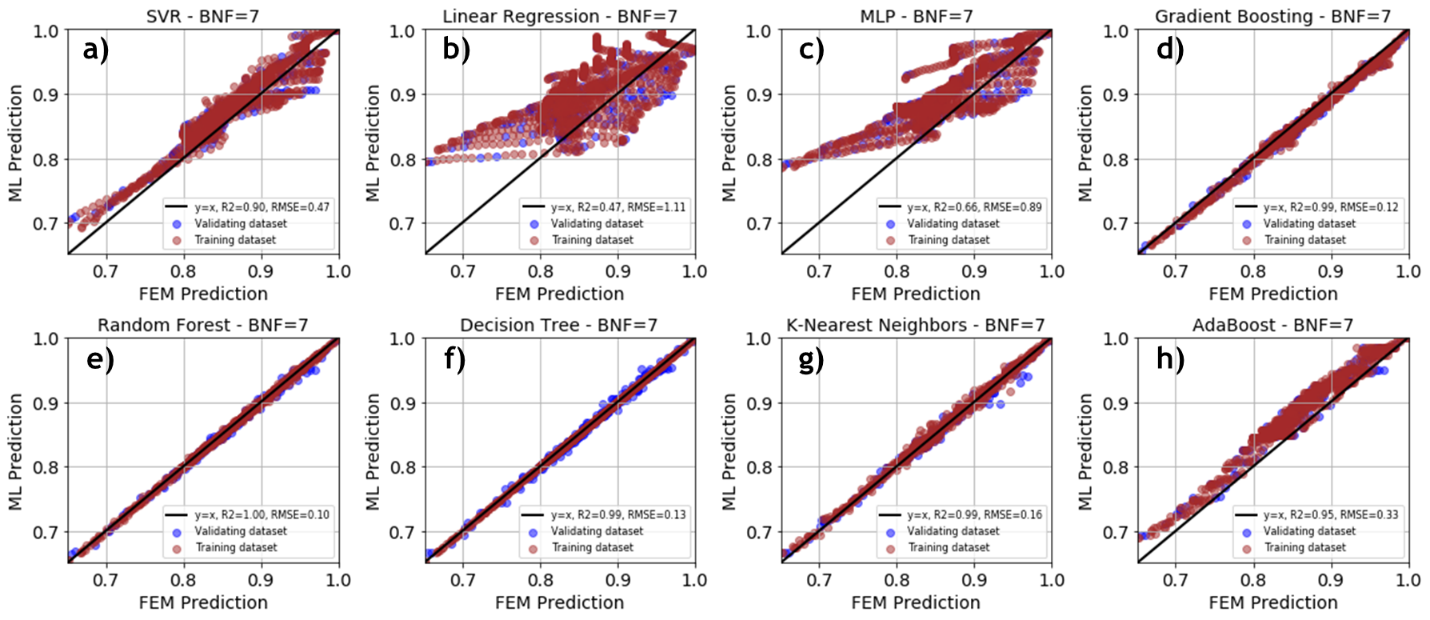
Training and validation results of BNF=4 for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

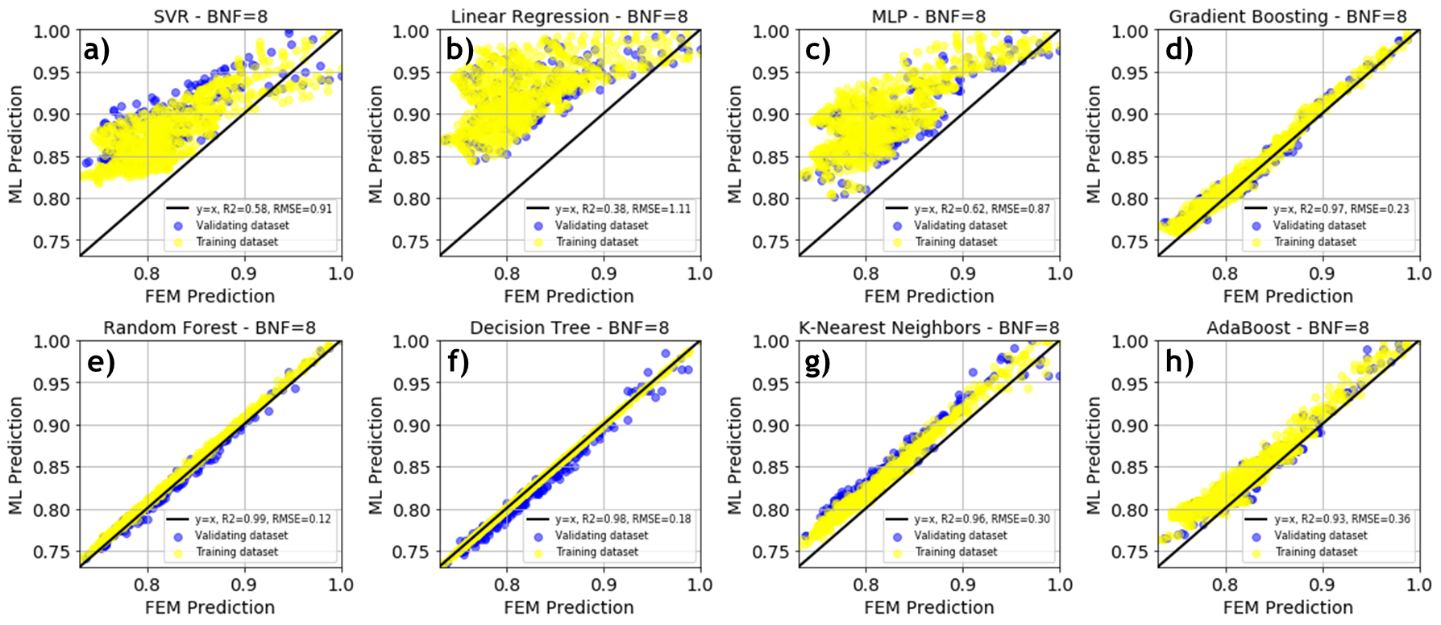
Training and validation results of BNF=5 for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

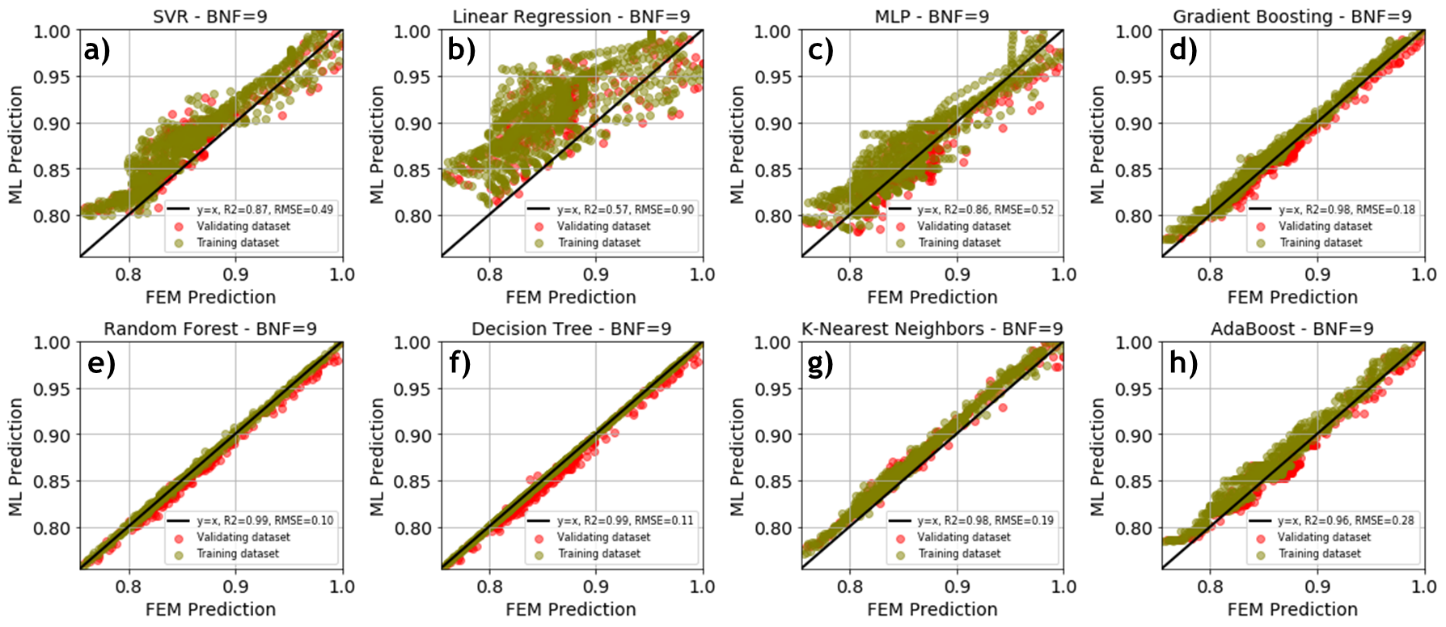
Training and validation results of BNF=6 for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

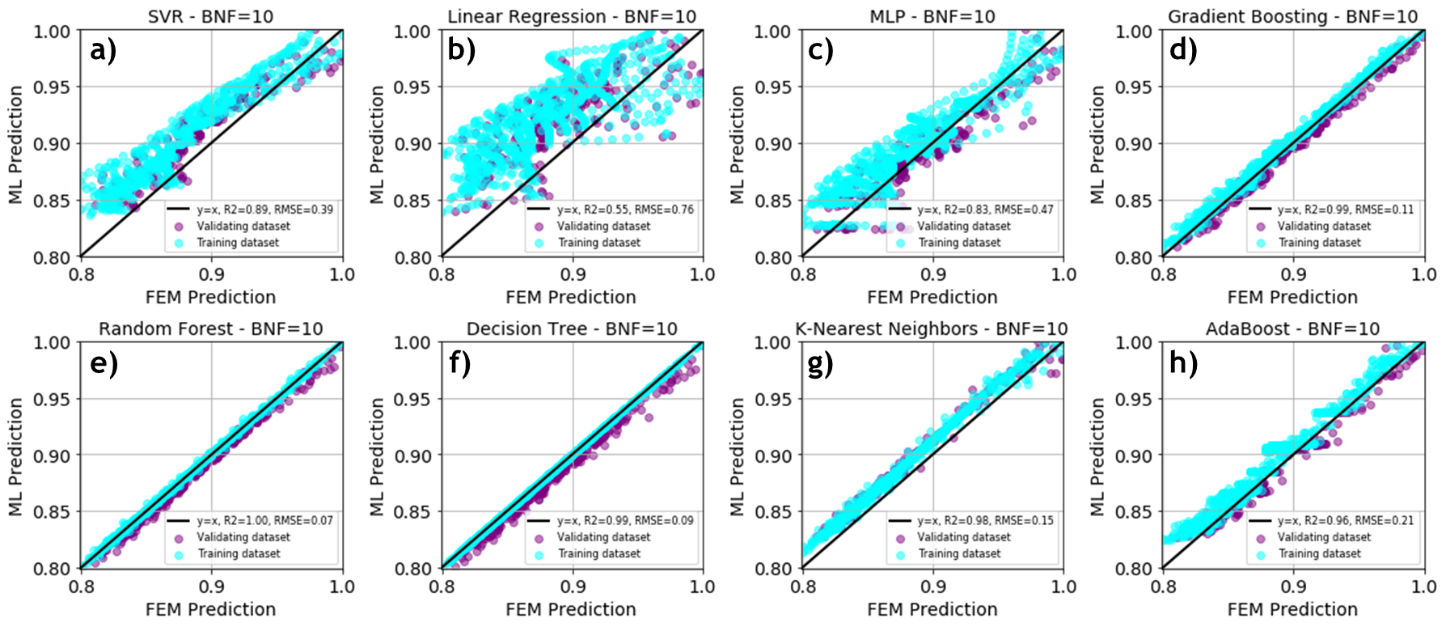
Training and validation results of BNF=7 for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

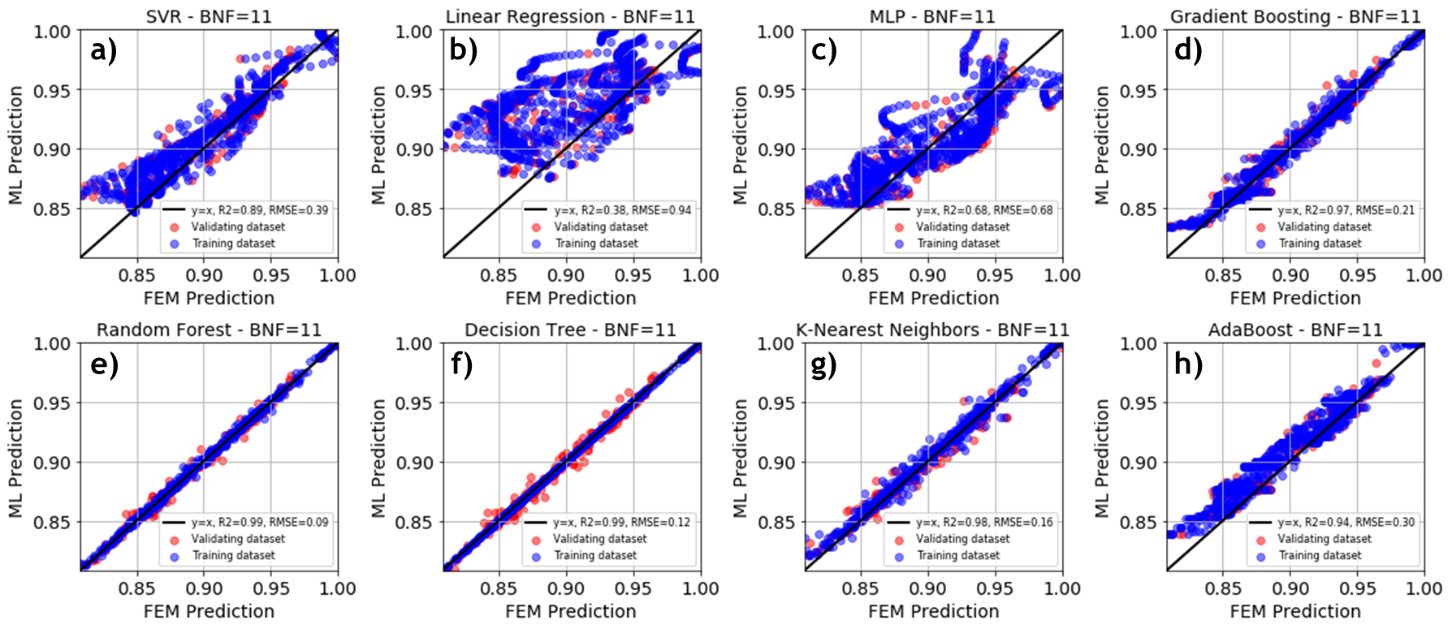
Training and validation results of BNF=8 for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

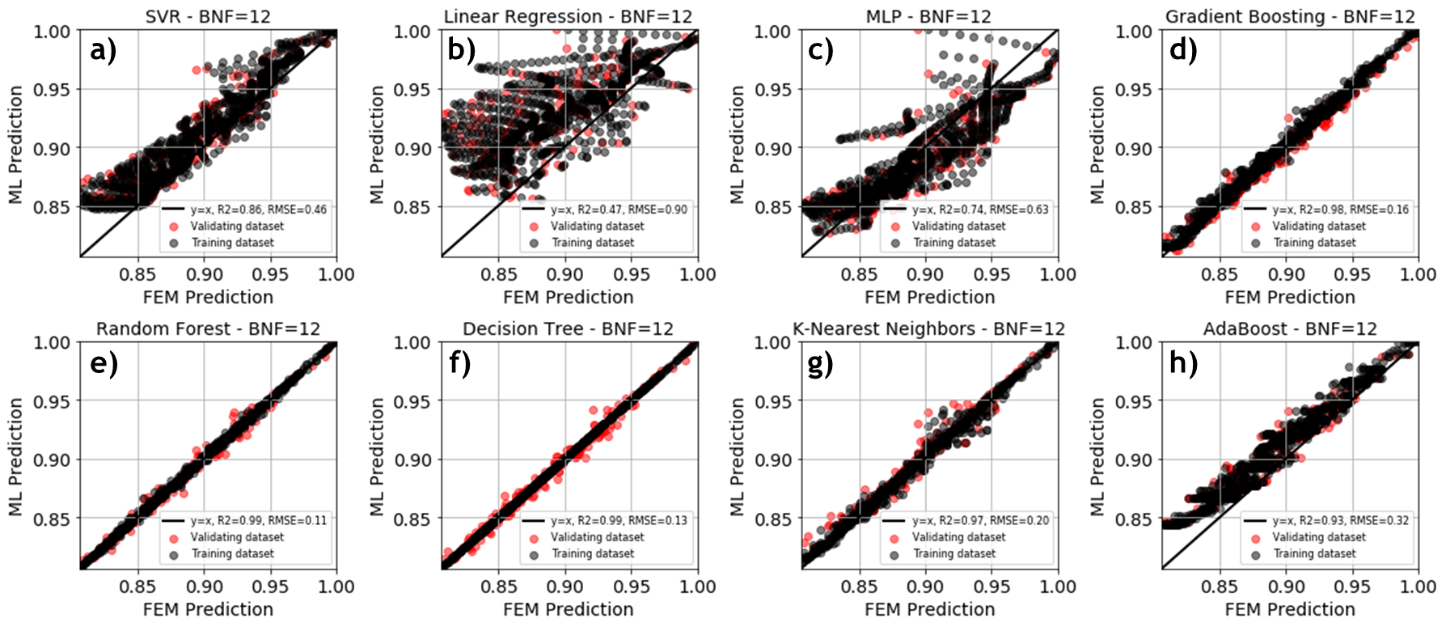
Training and validation results of BNF=9 for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

Training and validation results of BNF=10 for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

Training and validation results of BNF=11 for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

****

Training and validation results of BNF=12 for **a)** support vector regression, **b**) linear regression, **c)** multi-layer-perceptron, **d)** gradient boosting, **e)** random forest, **f**) decision tree, **g)** k-nearest neighbors, and **h)** adaboost model.

Increasing the number of selected features for the BNFs alters the training process for each of the ML model performances. The features considered for ML training include VF, cell size, reduced wave vectors (k), and twenty BNF features (from the 1st to the 20th) to predict BNFs using ML. **Table A3** for the R2 score for fitness and **Table A4** for RMSE summarize the performances of different models for the BNFs where the support vector regressor (SVR) highlighted in golden yellow color, has the best performance over twelve BNFs prediction. Then, the RF model highlighted in grey-silver color ranked second, and the GB model highlighted in bronze color ranked third overall.

**Table A3**. R2 Score for different BNFs for more features in training (yellow: best performance)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BNF** | **SVR** | **LR** | **MLP** | **GB** | **RF** | **DT** | **KNN** | **AB** |
| **1** | 0.9940020 | 0.9971397 | 0.9656618 | 0.9977011 | 0.9980064 | 0.997597 | 0.992867 | 0.987132 |
| **2** | 0.9946089 | 0.9929038 | 0.9125616 | 0.9979924 | 0.9971961 | 0.997135 | 0.989889 | 0.982158 |
| **3** | 0.9967236 | 0.9917238 | 0.8827861 | 0.9932057 | 0.9917238 | 0.982983 | 0.977625 | 0.979565 |
| **4** | 0.9890551 | 0.9788758 | 0.8613497 | 0.9872578 | 0.9880089 | 0.984820 | 0.976634 | 0.943647 |
| **5** | 0.9876390 | 0.9707481 | 0.8392304 | 0.9878985 | 0.9909448 | 0.976703 | 0.954942 | 0.922669 |
| **6** | 0.9909803 | 0.9441417 | 0.8995257 | 0.9868177 | 0.9921441 | 0.985006 | 0.979425 | 0.956480 |
| **7** | 0.9951257 | 0.9748066 | 0.8526738 | 0.9898613 | 0.9935039 | 0.986926 | 0.993424 | 0.965494 |
| **8** | 0.9854307 | 0.9563757 | 0.7183280 | 0.9823213 | 0.9851481 | 0.979558 | 0.962538 | 0.938576 |
| **9** | 0.9943648 | 0.9465350 | 0.8441195 | 0.9922594 | 0.9909503 | 0.984852 | 0.982270 | 0.961005 |
| **10** | 0.9929668 | 0.9697455 | 0.9364680 | 0.9921344 | 0.9960203 | 0.995528 | 0.989529 | 0.965601 |
| **11** | 0.9913404 | 0.8867271 | 0.7084459 | 0.9783467 | 0.9805684 | 0.971855 | 0.978518 | 0.928558 |
| **12** | 0.9904509 | 0.9543321 | 0.8654636 | 0.9801142 | 0.9825870 | 0.971352 | 0.968757 | 0.937243 |
| **All** | 0.9918907 | 0.9636712 | 0.8572178 | 0.9888259 | 0.9905668 | 0.984526 | 0.978868 | 0.955677 |

**Table A4.** Root mean square error (RSME) for different BNFs for more features in training

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BNF** | **SVR** | **LR** | **MLP** | **GB** | **RF** | **DT** | **KNN** | **AB** |
| **1** | 0.2477603 | 0.1710936 | 0.5128143 | 0.1533856 | 0.1328396 | 0.194813 | 0.270182 | 0.232886 |
| **2** | 0.1081791 | 0.1241136 | 0.4356701 | 0.0660144 | 0.0780162 | 0.078861 | 0.148147 | 0.196801 |
| **3** | 0.1146891 | 0.1965681 | 0.6859858 | 0.1651564 | 0.1822805 | 0.261371 | 0.299711 | 0.286422 |
| **4** | 0.1364929 | 0.1896248 | 0.4858094 | 0.1472745 | 0.1428678 | 0.160743 | 0.199430 | 0.309714 |
| **5** | 0.1236794 | 0.1902604 | 0.4460396 | 0.1223742 | 0.1058569 | 0.169793 | 0.236132 | 0.309347 |
| **6** | 0.1227956 | 0.3055840 | 0.4098402 | 0.1484503 | 0.1146001 | 0.158321 | 0.185458 | 0.269730 |
| **7** | 0.1125639 | 0.2559093 | 0.6188467 | 0.1623428 | 0.1299476 | 0.184345 | 0.130735 | 0.299494 |
| **8** | 0.1729336 | 0.2992439 | 0.7603843 | 0.1904962 | 0.1746031 | 0.204843 | 0.277304 | 0.355082 |
| **9** | 0.1010333 | 0.3112048 | 0.5313828 | 0.1184124 | 0.1280345 | 0.165646 | 0.179207 | 0.265774 |
| **10** | 0.0960565 | 0.1992259 | 0.2887001 | 0.1015817 | 0.0722558 | 0.076590 | 0.117203 | 0.212433 |
| **11** | 0.1077905 | 0.3898475 | 0.6254484 | 0.1704485 | 0.1614676 | 0.194324 | 0.169772 | 0.309604 |
| **12** | 0.1186819 | 0.2595425 | 0.4454748 | 0.1712672 | 0.1602654 | 0.205562 | 0.214671 | 0.304252 |
| **All** | 0.1302213 | 0.2410182 | 0.5205330 | 0.1431003 | 0.131919 | 0.171268 | 0.202330 | 0.279295 |