

SUPPLEMENTARY METHODS

EXTRACTING BULK DEFECT PARAMETERS IN SILICON WAFERS USING MACHINE LEARNING MODELS

Yoann Buratti, Quoc Le Gia, Josef Dick, Yan Zhu and Ziv Hameiri

MACHINE LEARNING HYPER-PARAMETERS - REGRESSION

```
sklearn.ensemble.RandomForestRegressor(  
    bootstrap=True,  
    ccp_alpha=0.0,  
    criterion='mse',  
    max_depth=None,  
    max_features='auto',  
    max_leaf_nodes=None,  
    max_samples=None,  
    min_impurity_decrease=0.0,  
    min_impurity_split=None,  
    min_samples_leaf=1,  
    min_samples_split=2,  
    min_weight_fraction_leaf=0.0,  
    n_estimators=100, n_jobs=-1,  
    oob_score=False,  
    random_state=None,  
    verbose=2,  
    warm_start=False  
)  
sklearn.ensemble.AdaBoostRegressor(  
    base_estimator=DecisionTreeRegressor(  
        ccp_alpha=0.0,  
        criterion='mse',  
        max_depth=None,  
        max_features=None,  
        max_leaf_nodes=None,  
        min_impurity_decrease=0.0,  
        min_impurity_split=None,  
        min_samples_leaf=1,  
        min_samples_split=2,  
        min_weight_fraction_leaf=0.0,  
        presort='deprecated',  
        random_state=None,  
        splitter='best'),  
    learning_rate=1.0,  
    loss='linear',  
    n_estimators=100,  
    random_state=None  
)  
sklearn.ensemble.GradientBoostingRegressor(  
    alpha=0.9,  
    ccp_alpha=0.0,  
    criterion='friedman_mse',  
    init=None,  
    learning_rate=0.1,  
    loss='ls',  
    max_depth=10,  
    max_features=None,  
    max_leaf_nodes=None,  
    min_impurity_decrease=0.0,
```

```

56         min_impurity_split=None,
57         min_samples_leaf=1,
58         min_samples_split=2,
59         min_weight_fraction_leaf=0.0,
60         n_estimators=100,
61         n_iter_no_change=None,
62         presort='deprecated',
63         random_state=None,
64         subsample=1.0,
65         tol=0.0001,
66         validation_fraction=0.1,
67         verbose=2,
68         warm_start=False
69     )
70     sklearn.neural_network.MLPRegressor(
71         activation='relu',
72         alpha=0.001,
73         batch_size='auto',
74         beta_1=0.9,
75         beta_2=0.999,
76         early_stopping=False,
77         epsilon=1e-08,
78         hidden_layer_sizes=(100, 100),
79         learning_rate='adaptive',
80         learning_rate_init=0.001,
81         max_fun=15000,
82         max_iter=200,
83         momentum=0.9,
84         n_iter_no_change=10,
85         nesterovs_momentum=True,
86         power_t=0.5,
87         random_state=None,
88         shuffle=True,
89         solver='adam',
90         tol=0.0001,
91         validation_fraction=0.1,
92         verbose=2,
93         warm_start=False
94     )
95     sklearn.svm.SVR(
96         C=5,
97         cache_size=200,
98         coef0=0.0,
99         degree=3,
100        epsilon=0.1,
101        gamma='auto',
102        kernel='rbf',
103        max_iter=-1,
104        shrinking=True,
105        tol=0.001,
106        verbose=2
107    )
108

```

```
110 sklearn.ensemble.RandomForestClassifier(  
111     bootstrap=True,  
112     ccp_alpha=0.0,  
113     class_weight=None,  
114     criterion='gini',  
115     max_depth=None,  
116     max_features='auto',  
117     max_leaf_nodes=None,  
118     max_samples=None,  
119     min_impurity_decrease=0.0,  
120     min_impurity_split=None,  
121     min_samples_leaf=1,  
122     min_samples_split=2,  
123     min_weight_fraction_leaf=0.0,  
124     n_estimators=100,  
125     n_jobs=-1,  
126     oob_score=False,  
127     random_state=None,  
128     verbose=2,  
129     warm_start=False  
130 )  
131 sklearn.ensemble.AdaBoostClassifier(  
132     algorithm='SAMME.R',  
133     base_estimator=DecisionTreeClassifier(  
134         ccp_alpha=0.0,  
135         class_weight=None,  
136         criterion='gini',  
137         max_depth=None,  
138         max_features=None,  
139         max_leaf_nodes=None,  
140         min_impurity_decrease=0.0,  
141         min_impurity_split=None,  
142         min_samples_leaf=1,  
143         min_samples_split=2,  
144         min_weight_fraction_leaf=0.0,  
145         presort='deprecated',  
146         random_state=None,  
147         splitter='best'  
148     ),  
149     learning_rate=1.0,  
150     n_estimators=10,  
151     random_state=None  
152 )  
153 sklearn.ensemble.GradientBoostingClassifier(  
154     ccp_alpha=0.0,  
155     criterion='friedman_mse',  
156     init=None,  
157     learning_rate=0.1,  
158     loss='deviance',  
159     max_depth=3,  
160     max_features=None,  
161     max_leaf_nodes=None,  
162     min_impurity_decrease=0.0,  
163     min_impurity_split=None,  
164     min_samples_leaf=1,  
165     min_samples_split=2,  
166     min_weight_fraction_leaf=0.0,  
167     n_estimators=100,  
168     n_iter_no_change=None,
```

```

169         presort='deprecated',
170         random_state=None,
171         subsample=1.0,
172         tol=0.0001,
173         validation_fraction=0.1,
174         verbose=2,
175         warm_start=False
176     )
177     sklearn.neural_network.MLPClassifier(
178         activation='relu',
179         alpha=0.001,
180         batch_size='auto',
181         beta_1=0.9,
182         beta_2=0.999,
183         early_stopping=False,
184         epsilon=1e-08,
185         hidden_layer_sizes=(100, 100),
186         learning_rate='adaptive',
187         learning_rate_init=0.001,
188         max_fun=15000,
189         max_iter=200,
190         momentum=0.9,
191         n_iter_no_change=10,
192         nesterovs_momentum=True,
193         power_t=0.5,
194         random_state=None,
195         shuffle=True,
196         solver='adam',
197         tol=0.0001,
198         validation_fraction=0.1,
199         verbose=2,
200         warm_start=False
201     )
202     sklearn.neighbors.KNeighborsClassifier(
203         algorithm='auto',
204         leaf_size=30,
205         metric='minkowski',
206         metric_params=None,
207         n_jobs=-1,
208         n_neighbors=5,
209         p=2,
210         weights='distance'
211     )

```