Analysis of the Breast Cancer Dataset

Marco Ravaioli, s345349

December 11, 2024

Contents

1 Introduction

This report focuses on analyzing a breast cancer dataset to explore the effectiveness of various machine learning techniques. The dataset contains a mix of categorical and numerical attributes, making preprocessing a critical step to ensure compatibility with the chosen models. The main objective is to answer specific analytical questions while comparing the performance of different algorithms like Decision Trees, Naïve Bayes, and K-Nearest Neighbors (K-NN).

The report is structured into three main sections:

- **Preprocessing**: Here, we explain the methods used to encode and prepare the dataset for analysis, including handling categorical data and numerical ranges.
- **Analysis**: This section addresses the assignment questions by evaluating model performance, analyzing feature relationships, and visualizing results like decision trees and correlation matrices.
- **Conclusions**: Finally, we summarize the key findings, discuss the strengths and limitations of the approaches used.

Through this analysis, we aim to understand not only how these models perform but also how the data itself impacts their results. By carefully tuning parameters and interpreting the outcomes, we can draw meaningful conclusions about the dataset and the models' behavior.

Table 1: Random sampling of 15 rows of the initial dataset.

age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	breast	breast-quad	irradiat	Class
'40-49'	'ge40'	'40-44'	'15-17'	'yes'	'2'	'right'	'left$_up'$	'yes'	'no-recurrence-events'
'30-39'	'premeno'	'35-39'	'0-2'	'no'	'3'	'left'	'left$_low'$	'no'	'recurrence-events'
'50-59'	'premeno'	'20-24'	'3-5'	'yes'	'2'	'left'	'left$_low'$	'no'	'no-recurrence-events'
'40-49'	'premeno'	'20-24'	'0-2'	'no'	'2'	'left'	'left$_low'$	'no'	'no-recurrence-events'
'40-49'	'ge40'	'30-34'	'0-2'	'no'	'2'	'left'	'left$_up'$	'yes'	'no-recurrence-events'
'60-69'	'lt40'	'10-14'	'0-2'	'no'	'1'	'left'	'right$_up'$	'no'	'no-recurrence-events'
'30-39'	'premeno'	'30-34'	'6-8'	'yes'	'2'	'right'	'right$_up'$	'no'	'no-recurrence-events'
'50-59'	'ge40'	'30-34'	'6-8'	'yes'	'3'	'left'	'right$_low'$	'no'	'recurrence-events'
'30-39'	'premeno'	'30-34'	'0-2'	'no'	'1'	'right'	'left$_up'$	'no'	'recurrence-events'
'50-59'	'ge40'	'20-24'	'0-2'	'no'	'3'	'right'	'left$_up'$	'no'	'no-recurrence-events'
'60-69'	'ge40'	'15-19'	'0-2'	'no'	'1'	'left'	'right$_low'$	'no'	'no-recurrence-events'
'30-39'	'premeno'	'10-14'	'0-2'	'no'	'2'	'left'	'right$_low'$	'no'	'no-recurrence-events'
'30-39'	'premeno'	'30-34'	'9-11'	'no'	'2'	'right'	'left$_up'$	'yes'	'recurrence-events'
'50-59'	'ge40'	'0-4'	'0-2'	'no'	'2'	'left'	'central'	'no'	'no-recurrence-events'
'50-59'	'premeno'	'25-29'	'3-5'	'no'	'2'	'right'	'left$_up'$	'yes'	'no-recurrence-events'

2 Preprocessing

The preprocessing phase was crucial to prepare the dataset for machine learning models. The dataset contained a mix of categorical attributes (e.g., 'node-caps', 'breast') and numerical ranges (e.g., 'age', 'tumor-size'), which needed to be encoded appropriately. The primary goal was to transform all features into numerical formats while preserving their inherent relationships and meanings.

Handling Duplicates and Splitting Data

During preprocessing, duplicate entries were identified and removed to avoid redundant information that could bias the models. After cleaning the data, the dataset was divided into:

- **Features (X)**: All columns containing input variables.
- **Target (y)**: The `Class` column, representing the outcome (`recurrence-events` or `no-recur...`).

Transformations Applied

The following transformations were performed:

- **Age, Tumor Size, Inv-Nodes**: These features consisted of ranges (e.g., '40-49'). Each range was converted to its numerical midpoint (e.g., '40-49' $\rightarrow$ 45) to represent them as continuous values.
- **Menopause**: This ordinal feature was encoded as integers to reflect its natural progression (`lt40` $\rightarrow$ 1, `premeno` $\rightarrow$ 2, `ge40` $\rightarrow$ 3).
- **Node-Caps**: This binary attribute was converted to 0 and 1 (`yes/no` $\rightarrow$ 1/0).
- **Breast-Quad, Breast**: These nominal features were encoded using one-hot encoding, creating separate binary columns for each category.
- **Target Variable (Class)**: The target was encoded as 0 (`no-recurrence-events`) and 1 (`recurrence-events`).

Results of Preprocessing

After preprocessing, all features were successfully converted into numerical values, ensuring compatibility with machine learning models. Each transformation was validated to preserve the dataset's original structure and meaning, with 15 columns because of OneHotEncoding.

Table 2: Random sampling of 15 rows of X_encoded.

age	menopause	tumor-size	inv-nodes	node-caps	deg-malig	irradiat	breast$_{'left'}$	breast$_{'right'}$	breast-quad$_{'central'}$	breast-quad$_{'left_low'}$	breast-quad$_{'left_up'}$	breast-quad$_{'right_low'}$	breast-quad$_{'right_up'}$	breast-quad$_{?}$
54.5	3	27.0	1.0	0	3	0	True	False	False	False	False	False	True	False
54.5	2	27.0	1.0	1	2	0	True	False	False	False	True	False	False	False
64.5	3	17.0	1.0	0	2	1	True	False	False	False	True	False	False	False
44.5	2	32.0	1.0	0	3	0	False	True	False	False	False	False	True	False
44.5	2	27.0	1.0	0	1	0	False	True	False	False	False	True	False	False
44.5	2	32.0	4.0	0	2	0	False	True	False	False	True	False	False	False
64.5	3	47.0	1.0	0	1	1	False	True	False	False	False	False	True	False
54.5	2	52.0	1.0	1	2	1	False	True	False	False	True	False	False	False
54.5	2	32.0	1.0	0	3	0	True	False	False	True	False	False	False	False
54.5	3	22.0	1.0	0	2	0	True	False	False	False	True	False	False	False
54.5	2	27.0	1.0	0	2	0	False	True	False	False	False	True	False	False
34.5	2	32.0	7.0	1	2	0	False	True	False	False	False	False	True	False
44.5	2	32.0	13.0	1	3	1	True	False	False	False	True	False	False	False
44.5	2	42.0	1.0	0	1	0	False	True	False	False	True	False	False	False
54.5	3	22.0	1.0	0	2	0	False	True	True	False	False	False	False	False

3 Analysis and Answers to Questions

3.1 Decision Tree Analysis

A Decision Tree with a maximum depth of 5 was trained to balance complexity and generalization. The most discriminative features identified were **tumor-size** (importance = 0.241) and **deg-malig** (importance = 0.239). These attributes played a key role in classifying the dataset. The tree's height was capped at 5, matching the predefined limit, ensuring it captured relevant patterns without overfitting.

A pure partition was observed in one of the leaf nodes, where all samples belonged to a single class. This illustrates the model's ability to confidently predict outcomes under certain conditions.

	Feature	Importance
2	tumor-size	0.241993
5	deg-malig	0.239404
3	inv-nodes	0.130641
4	node-caps	0.128183
10	breast-quad_'left_low'	0.079771
0	age	0.074128
12	breast-quad_'right_low'	0.051499
8	breast_'right'	0.030100
1	menopause	0.024280
6	irradiat	0.000000
7	breast_'left'	0.000000

Figure 1: Feature importances from the Decision Tree.

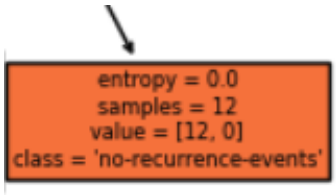

Figure 2: Example of a pure partition.

3.2 Impact of Decision Tree Parameters

The tree's performance was evaluated using various hyperparameters:

- **Maximal Depth**: Deeper trees captured complex patterns but risked overfitting, while shallow trees generalized better but missed details.
- **Minimum Samples Split and Leaf**: Higher thresholds reduced overfitting by limiting small splits but sometimes overlooked fine patterns.
- **Minimal Impurity Decrease**: Higher values simplified the tree, balancing interpretability and accuracy.

Visualizations of trees with different configurations are included below to highlight these trade-offs.

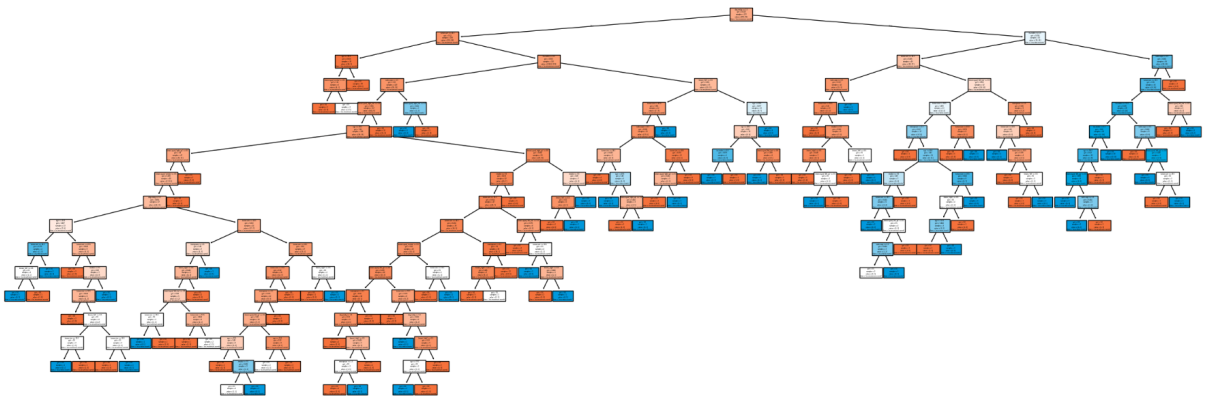

Figure 3: Default config (test score: 0.610). It results in a highly complex tree with excessive depth, leading to overfitting and a lack of generalization.

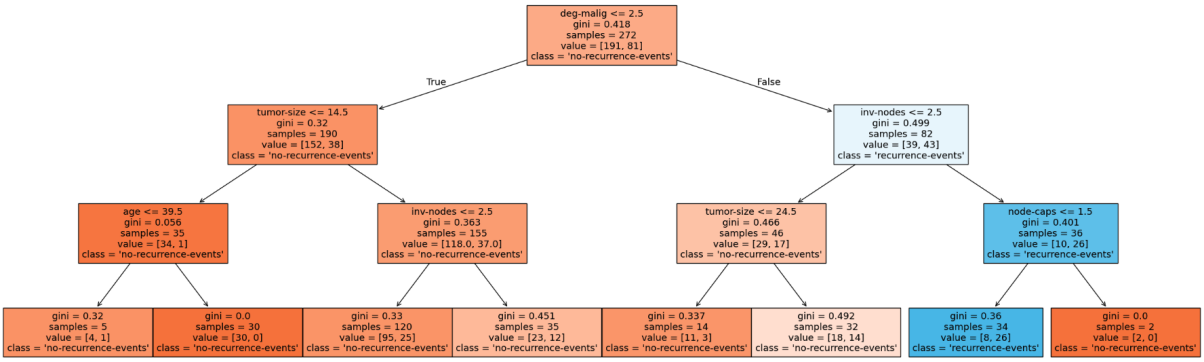

Figure 4: Max_depth = 3 config (test score: 0.684). Limiting the maximum depth simplifies the tree, focusing on broad patterns while avoiding noise. This leads to a more interpretable model.

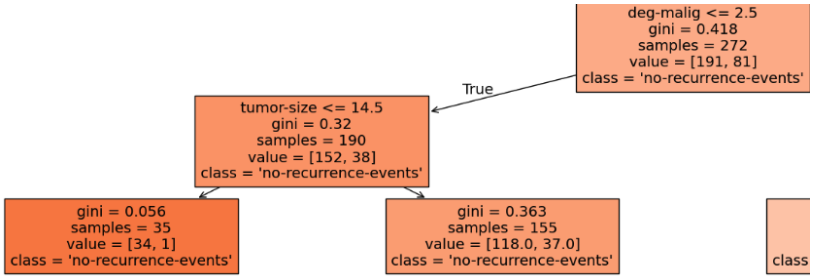

Figure 5: min_impurity_decrease = 0.005 config (test score: 0.706). Introducing a minimal impurity decrease constrains splits to only significant reductions, resulting in a more balanced and generalized tree.

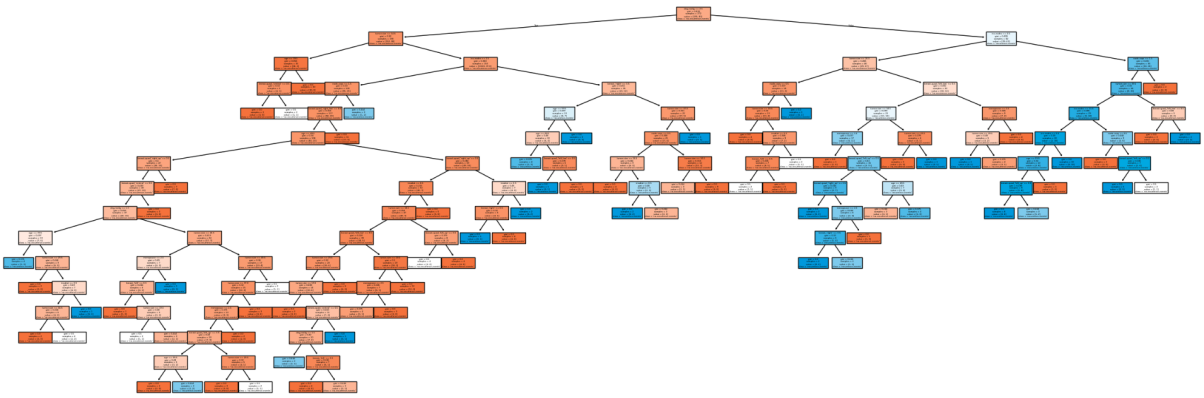

Figure 6: min_samples_split = 5 config (test score: 0.636). Restricting minimum samples per split prevents overfitting by ensuring splits occur only on sufficiently large data subsets, resulting in a less complex tree.

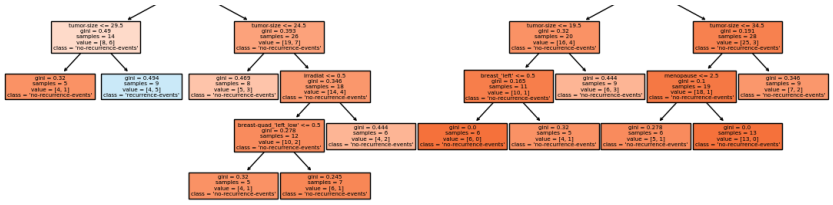

Figure 7: min_samples_leaf = 5 config (test score: 0.669). Enforcing a minimum number of samples per leaf node balances the tree structure, reducing noise sensitivity while maintaining simplicity.

3.3 Impact of Parameters on Average Accuracy

The performance of the Decision Tree was evaluated under five different parameter configurations using 10-fold Stratified Cross-Validation. This analysis explores how key hyperparameters impact the model's complexity, generalization, and average accuracy.
The following configurations were tested using 10-fold Stratified Cross-Validation to evaluate how different parameters impact Decision Tree performance:

1. **Default Configuration**: The default tree captures detailed patterns but risks overfitting due to excessive splits and a lack of constraints. This results in imbalanced predictions and moderate accuracy, with higher variability in predictions.

2. **Max Depth Modified**: Limiting the tree's depth to 3 simplifies the structure by focusing on broader patterns. This improves generalization and balances predictions but may miss finer details and critical patterns, leading to occasional misclassifications and reduced accuracy for edge cases.

3. **Minimal Impurity Decrease**: Imposing a minimum impurity decrease threshold (e.g., 0.005) prevents insignificant splits, resulting in a more streamlined tree. This adjustment enhances interpretability, reduces unnecessary complexity, and maintains balanced predictions while avoiding overfitting.

4. **Minimum Samples Split Modified**: Increasing the minimum samples required for a split (e.g., 5) prevents the tree from overfitting small subsets of data. This adjustment simplifies the tree structure and reduces noise sensitivity, improving generalization but potentially underfitting by ignoring smaller yet important patterns.

5. **Minimum Samples Leaf Modified**: Requiring a minimum number of samples per leaf (e.g., 5) ensures larger and more stable leaf nodes. This reduces noise sensitivity and improves robustness. However, this also leads to smoother decision boundaries, which can overlook finer details in the data, balancing stability and granularity.

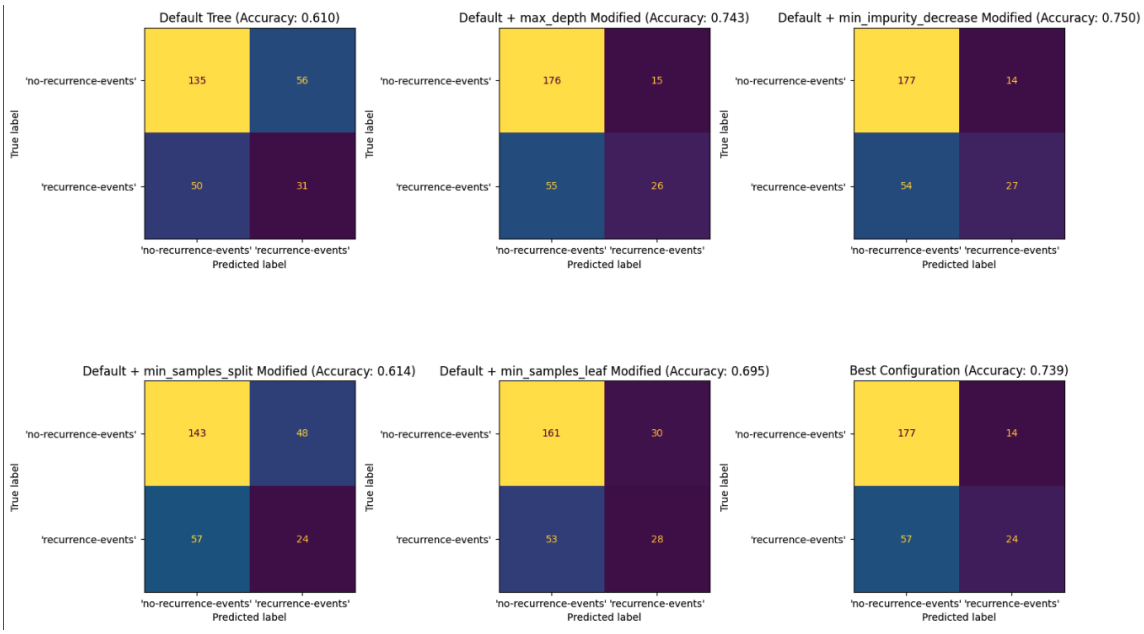

Figure 8: Combined confusion matrices for different configurations. Each matrix highlights the distribution of predictions and misclassifications, illustrating the impact of parameter changes on model performance.

3.4 K-NN and Naïve Bayes Analysis

Impact of Parameter K on K-NN Accuracy

The performance of the K-NN classifier was evaluated using 10-fold Stratified Cross-Validation across different values of K. Lower K values ($K = 5$) resulted in higher sensitivity to noise, leading to misclassifications. Increasing K ($K = 20, 25$) improved stability and generalization by averaging predictions over larger neighborhoods, though overly large K values ($K = 30$) caused slight underfitting by smoothing decision boundaries excessively.

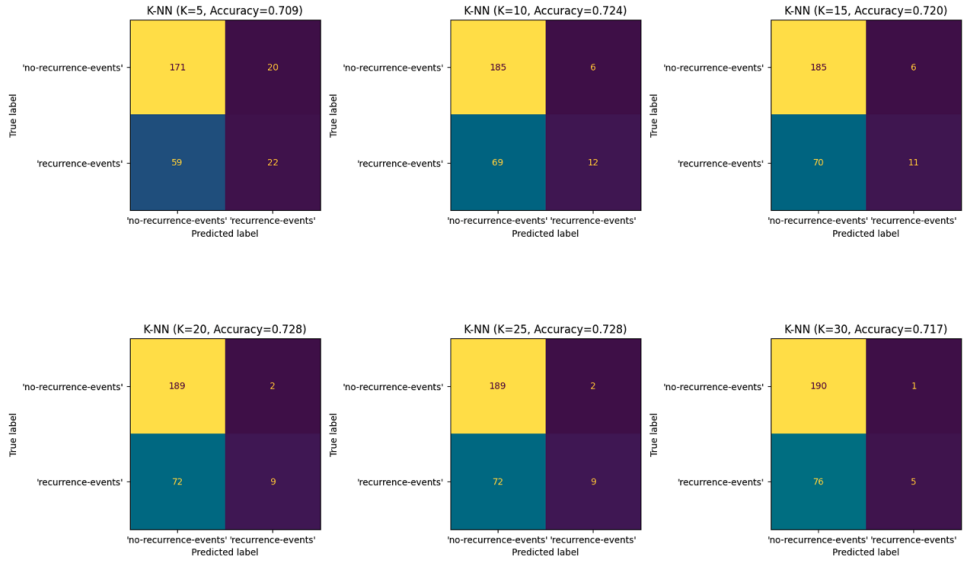

Figure 9: Confusion matrices for K-NN with different K values.

Naïve Bayes achieved an accuracy of 0.691, showcasing its robustness despite violations of the independence assumption. This simplicity makes it efficient for datasets with correlated features, as it assumes conditional independence given the class.

K-NN, which relies on local patterns in the feature space, does not assume independence among features. Its performance depends heavily on the choice of K and the data distribution, allowing it to capture complex relationships. However, it is computationally more intensive compared to Naïve Bayes.

While Naïve Bayes is faster and easier to tune, K-NN provides greater flexibility, making it better suited for datasets with strong dependencies among features.

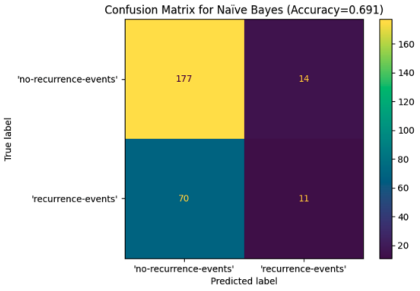

Figure 10: Confusion matrix for Naïve Bayes.

3.5 Correlation Matrix and Naïve Independence Assumption

The correlation matrix highlights pairwise relationships between numerical attributes in the Breast dataset. The strongest correlation is observed between **age** and **menopause** ($r = 0.63$), reflecting a natural dependency where menopausal status is influenced by age. This violates the Naïve independence assumption, which assumes conditional independence among features given the target. Despite this, Naïve Bayes achieves reasonable accuracy, showcasing its robustness.

The correlation analysis also shows low correlations between most features and the target, indicating that no single attribute strongly predicts the outcome. This reinforces the importance of leveraging multiple features for classification.

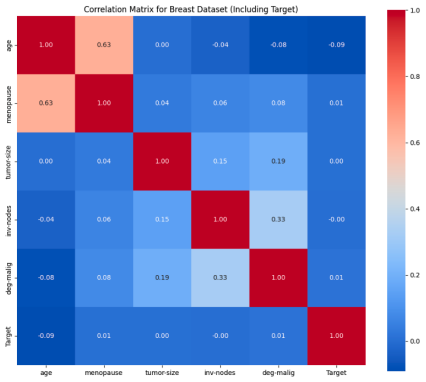

Figure 11: Correlation Matrix for Breast Dataset. High correlation between **age** and **menopause** challenges the Naïve independence assumption.

4 Conclusions

This analysis emphasized the importance of thoughtful preprocessing and hyperparameter tuning in machine learning. Key findings include:

- Decision Trees demonstrated their effectiveness, identifying `tumor-size` and `deg-malig` as the most significant features. Parameter adjustments, such as setting a `max_depth` of 5, helped balance generalization and complexity, while additional tuning improved accuracy and interpretability.

- K-NN surpassed Naïve Bayes in performance when properly tuned, leveraging local patterns effectively without assuming independence among features. However, Naïve Bayes remained robust and computationally efficient despite violating the independence assumption.

- The correlation matrix highlighted dependencies, such as between `age` and `menopause`, reinforcing the need for models capable of handling feature interactions.

Additionally, GridSearchCV was employed to identify the best parameter combinations for Decision Trees and K-NN, optimizing their performance. However, due to space constraints, the detailed exploration of these results was not included in this homework.

In conclusion, the study demonstrates that model success depends not only on algorithm choice but also on the careful handling of data and tailored parameter optimization.