Sudhanshu, 2026, 14:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Analyzing Model Generalization: A Study of Overfitting, Underfitting, and Bias–Variance Trade-Off

Sudhanshu, Nitin Gupta, Dev Button, Jyoti Bansal, Vinayak Sharma
HMR Institute of technology and management Guru Gobind Singh Indraprastha University New Delhi, India,

Abstract- Machine learning models usually work better or worse depending on how well they can use what they learned from the training data to make predictions about new data. Overfitting and underfitting are the two biggest problems in this area. Both can have a big impact on how reliable and accurate the model is at making predictions. This is a meta-analysis of overfitting, underfitting, and generalization that looks at a lot of different machine learning models and datasets. The meta-analysis integrates theoretical frameworks and empirical experiments to examine the training and testing performances across various model complexities. The generalization gap is the main way to measure how well a model works, and the bias-variance tradeoff is the main idea behind the whole paper. The research studies used standard benchmark datasets and a range of algorithms, such as linear models, decision trees, ensemble methods, and neural networks. Error curves, comparative metrics, and statistical summaries were all used to show the patterns in model performance.

Keywords: Overfitting, Underfitting, Machine Learning.

I. INTRODUCTION

Machine learning is a key part of modern intelligent systems. It can be used for many things, such as recognizing images and making predictions. One of the biggest problems with machine learning is still how to make sure that the models work well with data they haven't seen before. If a model only works well on training data but not on new data, it is unreliable [4]. This is why model generalization has become one of the most important areas of research. Overfitting and underfitting are two of the main reasons why machine learning models fail to generalize.

Overfitting is a case when the model is overly aligned with the training data set, such that it captures not only the underlying patterns but also the noise and other random variations in the dataset that make it perform poorly on unseen data[6]. A different scenario arises with underfitting, which occurs when the model cannot capture the data structure which leads to a low ability of the model when it comes to following the dataset. These two problems have their origins from the bias–variance trade off through which one explains the behavior of the models theoretically.

The problems associated with overfitting and generalization have been studied quite extensively for a long time in literature and various studies have been published by researchers. However, most of the papers in the literature are either theoretical analyses or describe the performance of a single model type realized on a particular dataset [2]. To the best of our knowledge, no publication presents a unified, data-driven analysis examining overfitting, underfitting, and generalization simultaneously for a variety of models and datasets in a formal manner. This is hindering the deriving of general findings about model performance in real-world cases.

- To close this gap, we have undertaken a study that systematically investigates overfitting, underfitting, and generalization in machine learning models. The study integrates theoretical understanding with empirical work by testing several algorithms, such as linear models, decision trees, [1] ensemble methods, and neural networks. It relies on standard benchmark datasets and measures model performance using training and testing errors and generalization gaps as the main quantification metric of model behaviors.

- Structured empirical study of the phenomena of overfitting and underfitting across various types of models
- Model behaviors visualization using performance curves and comparative graphs
- Our objective is to take research on model behaviors toward a more holistic and practical direction, which are helpful in the modeling and selection stages of the machine learning lifecycle for better generalization.

The objective of this research is to have a complete and practical comprehension of how models behave to assist in selecting a model that would be more generalizable

II. BACKGROUND AND THEORETICAL FOUNDATIONS

To understand the behavior of machine learning models, it is crucial to distinguish between the model's performance on the training data and the [5] ability to generalize to new data. In supervised learning, a model is prepared on a dataset and then tested on separate, unseen data to determine how well it can make predictions. The difference between these two results leads to concepts such as overfitting, underfitting, and generalization.

Overfitting and Underfitting

Overfitting takes place when a model obtains too much detail from the training set to the point that it reacts to noise and other inaccuracies present in the data. There is an imbalance between training and test errors, so the model that predicted accurately on the training remains unable to match testing results [24]. Generally, the phenomenon just described is attributable to the excessive model complexity looking at a certain number of examples.

Underfitting is the contrary of overfitting. The model does not have enough expressive power to capture the data pattern even in the training set. It is possible to observe that there is a similarity in the training and testing performance which is however quite low.

These phenomena are two extremes of model complexity [29]and they explain the need for a model suitable to the data in which the model has enough flexibility to learn but at the same time it generalizes well to new data.

Generalization and Generalization Gap

Generalization is the model's skill to make predictions on new data from the same source as the training data. The main indicator for assessing generalization is the generalization gap which is defined as the training error minus the test error [4].

Generalization Gap = Ltrain(f) – Ltest(f)

In case the generalization gap is lower, then it means that the model produces steady results on both the train and the test sets. Having a larger gap is usually the sign of overfitting [2][29]. But if there is an elevated error in the training and test sets, the model is suspected to be underfitting.

Bias–Variance Trade off

Bias–variance tradeoff is a theory which helps to explain the reasons of overfitting and underfitting [7]. It divides the average prediction error into three components:

Error=Bias2+Variance +Noise (2)

- Bias is the error that arises when a real-world problem is substituted by a simpler model. When bias is high, underfitting is the result.
- Variance is the extent to which a model is influenced by the changes in the training data. Overfitting happens [9] when the variance is high.
- Noise is the error in data that is irreducible in nature.

If bias as well as variance were kept in check, one would get a model that has minimal total error.

Loss Functions and Error Measurement

Loss functions play an important role in the quantitative assessment of model performance. MSE (Mean Squared Error) [12] is the most widely used loss function for regression problems:

(MSE):L= $\mathbf{1}\sum_{i=1}^{n}(y_i - \hat{y}_i)\mathbf{2}$ (3)

Accuracy, cross-entropy, F1-score are some of the measures that are most used for classification cases [14]. The use of the following indicators forms the starting point for comparison of performances between different models.

Complexity of Model and Learning Behavior

The complexity of the model can have a significant influence on the overfitting and underfitting problem in the model. As the complexity increases:

- The training error decreases
- The testing error decreases for some time but then starts increasing

This description represents the commonly known U-shaped curve, depicting the journey from underfitting to overfitting [16]. It becomes essential to identify this point for optimal learning to attain good results regarding generalization.

III. METHODOLOGY

In this study, the experiment makes use of empirical data to investigate the problems associated with overfitting, underfitting, and generalization in the context of machine learning using a controlled experimentation environment. The objective is to measure the performance of the model at various complexities and levels of generalization. [12]

Dataset Description

We will employ the Breast Cancer Wisconsin dataset for our experimentation purposes. It is an extremely common dataset used for binary classification tasks [15]. The data is made up of attributes obtained from digitized mammogram images of breast lumps, which need to be classified as either malignant or benign.

For ensuring consistency in model training, first the whole data set is normalized, so that the training instances are on the same level. It is a very important step for obtaining good results from the models.

Data Preprocessing

Then the data set of this information will be split into training set and testing dataset in the ratio 80:20. The training set is used to train the models while the test set is utilized for testing. Feature scaling was executed using a standard technique [17], which means that the data is first mean cantered then scaled to unit variance. This ensures the equal scaling of all features, which improves the performance of analytical algorithms.

This is crucial for models like logistic regression and neural networks, which are sensitive to the magnitudes of features.

Model Selection

To demonstrate various learning scenarios, [13]four machine learning algorithms are chosen described below.

Machine learning algorithm	
Logistic Regression	Linear algorithm having high bias, which aids in determining under fitting scenarios.
Decision Tree	Non-linear algorithm with increased risk of overfitting as complexity grows [18]
Random Forest	Ensemble algorithm to reduce variance and improve generalization.[5][17]
Neural Network (MLP-Classifier)	Flexible algorithms to learn complicated patterns.[20]

For each configuration, the model is trained on the training dataset and evaluated on both training and testing datasets.

Experimental Design

This is achieved by tuning model complexity via key parameters:

- Logistic Regression: uses a regularization parameter
- Decision Tree: about the maximum depth
- Random Forest: tweak both tree depth and the number of estimators
- Neural Network: change up the number of hidden neurons

With each setup, you train the model on your training data[19], then test how it performs on both the training and testing datasets.

Evaluation Metrics

Classification accuracy was calculated for both training and testing samples. To measure

generalization behaviours quantitatively, the generalization gap was calculated as:

Gap = Accuracy(train) – Accuracy(test)

A big difference between the training and testing performance suggests overfitting, that is, the model [28] gets so good at learning the training data that it fails to generalize to new (or unseen) data. A small gap indicates good generalization, and the model will perform equally well on both training and other datasets. On the other hand, if both training error and testing error are high, this means we have underfitted our model is too simple to depict checkers [24].

Analysis Framework

This research is divided into three levels of analysis:

1. **Model-wise-Analysis:**
   We take each model on its own and see how its performance shifts as it gets more complex.
2. **Comparative-Analysis:**
   Here, we pit the models against each other, comparing how well they generalize and how accurate they are.
3. **Statistical-Observation:**
   Finally, we pull back to look at the big picture—analyzing overall trends to spot clear patterns of overfitting and underfitting.

Visualization Strategy

To substantiate the analysis, several plots were prepared, including training versus testing accuracy curves, generalization gap comparisons across different models, and scatter plots of training versus testing performance [25]. These visualizations are useful for identifying existing trends and for verifying theoretical concepts such as the bias–variance trade-off.

IV. RESULTS AND ANALYSIS

The paper dives into how different machine learning models perform, especially as their complexity changes. To figure out what's really going on under the hood, the authors look at training accuracy, testing accuracy, and the generalization gap.

Logistic Regression Analysis

The results of logistic regression analysis show that there is a smooth way to learn when the regularization parameters change where the models are trained [30]. The graph that shows accuracy shows that both training and testing accuracies go up as the model gets more complex, from C=0.01 to C=0.01. At C=0.1, the testing accuracy is at its highest point. After that, it starts to go down.

The generalization gap hardly moves from zero over the entire range of values implying the model generalizes very well. At the lowest end of C values, the model shows small underfitting with both training and testing accuracy [27]. As C increases, the model becomes more flexible, achieving optimal performance before showing a marginal increase in the generalization gap at higher values.

These observations suggest that Logistic Regression maintains a good balance between bias and variance, making it a robust baseline model.

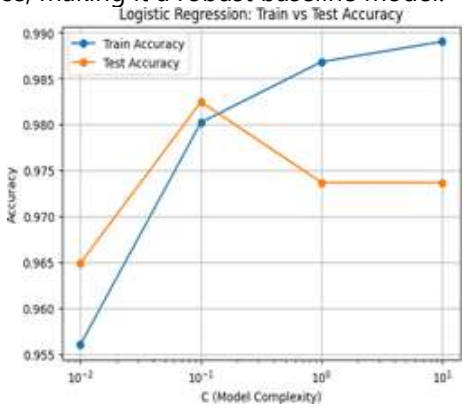

Figure 1 Train vs Test Accuracy

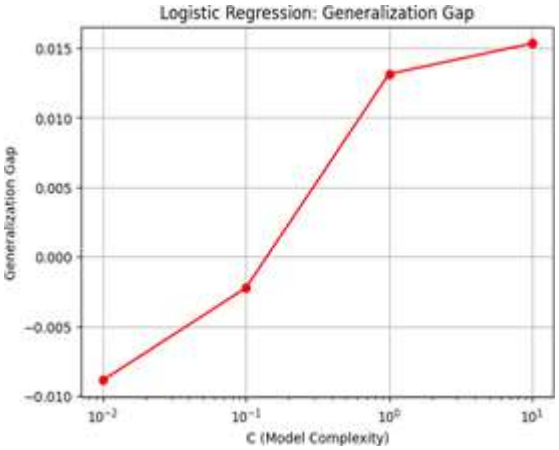

Figure 2 Generalization Gap

Decision Tree Analysis

The behavior exhibiting a growth of overfitting in Decision Tree as model complexity was increased is very evident. As the tree grows, the training accuracy is almost touching 100% very quickly, further suggesting that the model is memorizing the training data.

The test accuracy, on the other hand, remains at the same level after some depth of the tree has been reached, thereby producing a huge generalization gap, especially at larger depths.

The graph of the gap also makes it clear that the gap is growing very fast with increasing depth. This confirms again that Decision Trees are the ones that suffer from overfitting the most if not regularly properly.

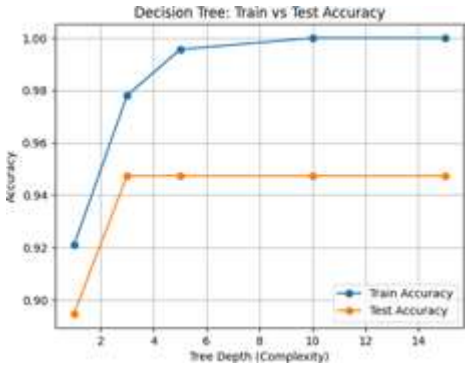

Figure 3: Train vs Test Accuracy

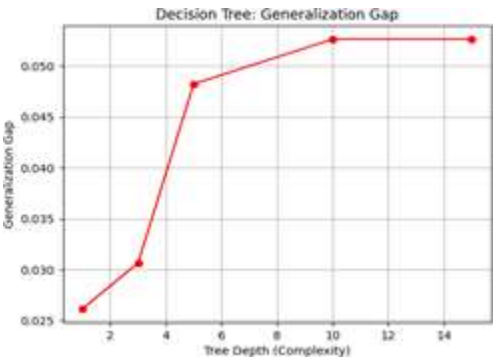

Figure 4: Generalization Gap

Random Forest Analysis

Random Forest is better in generalization when compared to a single Decision Tree. The training accuracy increases with the depth of the tree, but it does not reach the extreme levels seen in the Decision Tree model.

The testing accuracy is quite stable at various depths, and the generalization gap is a lot smaller. This exemplifies the work of ensemble Random Forest in lowering variance and overfitting reduction.

It is indeed very interesting to see how the model's robustness and reliability gain another level by combining individual trees, even though individual trees can be rather complex.

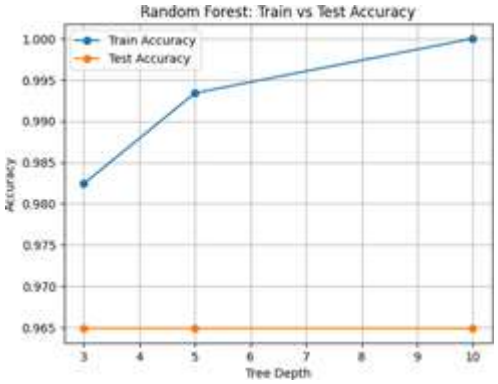

Figure 5:Train vs Test Accuracy

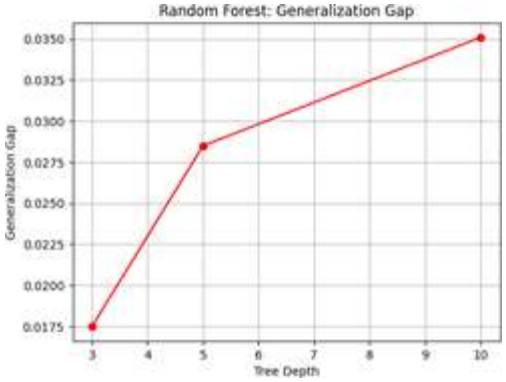

Figure 6: Generalization Gap

Neural Network Analysis

Based on the number of neurons, the Neural Network model changes its learning properties rather drastically. With few neurons, the model is doing quite well but when it comes to high number of neurons training accuracy almost immediately reaches the point of being perfect.

Nonetheless, this increment in training accuracy is not matched by the testing accuracy, rather in higher

neuron counts, it has a decreasing trend. This results in an increment of the generalization gap thereby slightly overfitting the model.

Neural Networks have been proven to be very powerful but according to this study their strength can also become a weakness if the model is not properly architecture for generalization.

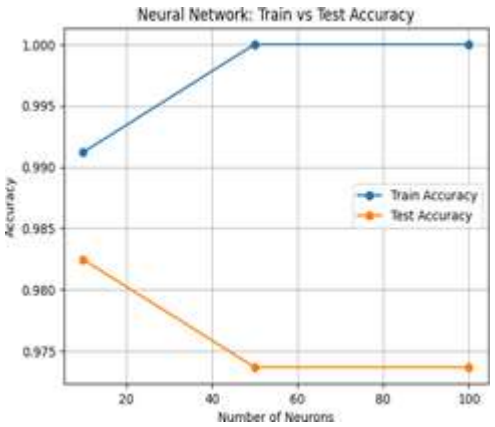

Figure 7: Train vs Test Accuracy

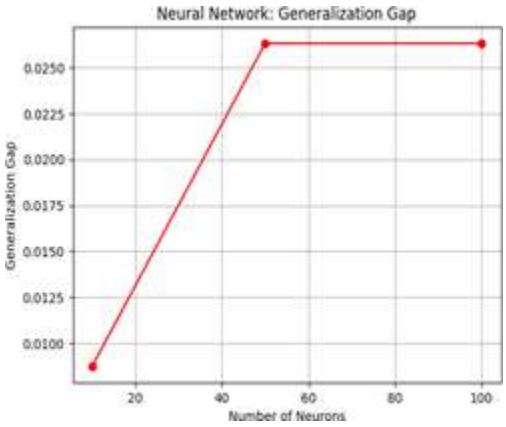

Figure 8: Generalization Gap

Comparative Analysis

When looking at all the different models together, distinct patterns of the models emerged in the following way:

- Logistic Regression exhibits stable performance with minimal overfitting
- Decision Tree shows significant overfitting at higher complexity
- Random Forest achieves a balance between accuracy and generalization
- Neural Network demonstrates flexible behaviours but requires tuning

Out of all the models Random Forest gives the best balance between performance in training and generalization as indicated by the relatively low generalization gap and testing accuracy that does not fluctuate a lot.

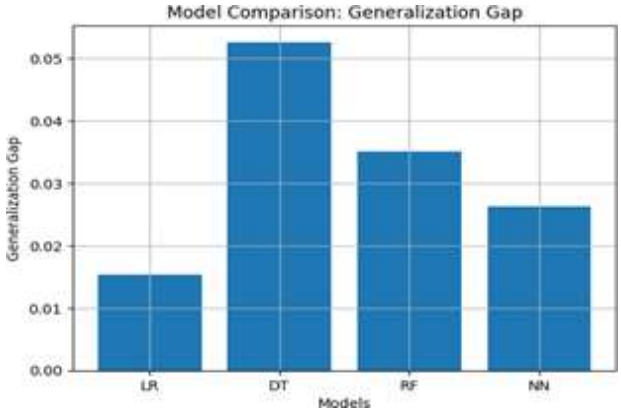

Figure 9: COMPARISON

Key Observations

A few major points can be derived from the results:

- It is possible to attain higher performance on training data by increasing the model complexity, however, that may lead to deterioration in the model generalization
- Overfitting has been demonstrated to be most distinct in Decision Trees which are high-capacity models
- Ensemble methods work very well in reducing overfitting
- Neural networks need good care and architectural design to avoid performance drop phases
- Generalization gap is a very robust and reliable indicator of model behaviour

V. CONCLUSION

A meta-analysis of overfitting, underfitting, and generalization in machine learning models was carried out here on a benchmark dataset. The datasets show that model performance may change a lot with a change in model complexity and that such changes are affected by the degree to which the model balances bias and variance.

Logistic Regression obtained stable results with a very low generalization gap suggesting it is well

generalizable. In contrast, Decision Trees revealed clear signs of overfitting as the model complexity was increased with the training accuracy reaching the level of almost perfect. Random Forest has been shown as a model that can reduce variance and therefore achieve better generalization by means of ensemble learning and Neural Networks had a very complex behavior with high training accuracy but need caution in tuning so as not to overfit.

The analysis shows that higher model complexity does not necessarily lead to better performance on unseen data but rather an optimal balance should be found to minimize the generalization gap. The study also highlights the power of ensemble methods in improving model robustness.

REFERENCES

1. C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.
2. T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning. Springer, 2009.
3. K. P. Murphy, Machine Learning: A Probabilistic Perspective. MIT Press, 2012.
4. S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014.
5. L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
6. D. Opitz and R. Maclin, "Popular Ensemble Methods: An Empirical Study," Journal of Artificial Intelligence Research, vol. 11, pp. 169–198, 1999.
7. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
8. A. Ng, "Machine Learning Yearning," 2018.
9. V. Vapnik, The Nature of Statistical Learning Theory. Springer, 1995.
10. G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning. Springer, 2013.
11. J. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," Annals of Statistics, 2001.
12. R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation," IJCAI, 1995.
13. P. Domingos, "A Few Useful Things to Know About Machine Learning," Communications of the ACM, 2012.
14. L. Bottou, "Stochastic Gradient Learning in Neural Networks," Proceedings of Neuro-Nîmes, 1991.
15. D. Rumelhart, G. Hinton, and R. Williams, "Learning Representations by Backpropagation," Nature, 1986.
16. Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," Nature, 2015.
17. T. Dietterich, "Ensemble Methods in Machine Learning," Multiple Classifier Systems, 2000.
18. J. Quinlan, "Induction of Decision Trees," Machine Learning, 1986.
19. Scikit-learn Developers, "Scikit-learn: Machine Learning in Python," 2024.
20. K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," Neural Networks, 1989.
21. G. Hinton, "Reducing the Dimensionality of Data with Neural Networks," Science, 2006.
22. Y. Freund and R. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," Journal of Computer and System Sciences, 1997.
23. L. Bottou and O. Bousquet, "The Tradeoffs of Large Scale Learning," Advances in Neural Information Processing Systems, 2008.
24. A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," NeurIPS, 2012.
25. C. Cortes and V. Vapnik, "Support-Vector Networks," Machine Learning, 1995.
26. J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," Journal of Machine Learning Research, 2012.
27. M. Kuhn and K. Johnson, Applied Predictive Modeling. Springer, 2013.
28. F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, 2011.
29. A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly, 2019.
30. T. Mitchell, Machine Learning. McGraw-Hill, 1997.