LEARNING LATENT REPRESENTATIONS FOR OPERATIONAL NITROGEN RESPONSE RATE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning latent representations has aided operational decision-making in several disciplines. Its advantages include uncovering hidden interactions in data and automating procedures which were performed manually in the past. Representation learning is also being adopted by earth and environmental sciences. However, there are still subfields that depend on manual feature engineering based on expert knowledge and the use of algorithms which do not utilize the latent space. Relying on those techniques can inhibit operational decision-making since they impose data constraints and inhibit automation. In this work, we adopt a case study for nitrogen response rate prediction and examine if representation learning can be used for operational use. We compare a Multilayer Perceptron, an Autoencoder, and a dual-head Autoencoder with a reference Random Forest model for nitrogen response rate prediction. To bring the predictions closer to an operational setting we assume absence of future weather data, and we are evaluating the models using error metrics and a domain-derived error threshold. The results show that learning latent representations can provide operational nitrogen response rate predictions by offering performance equal and sometimes better than the reference model.

1 INTRODUCTION

Latent representation learning has been adopted in several disciplines to extract and handle hidden interactions between the input variables allowing for more informed decisions. In geosciences, representation learning algorithms emerge (Jean et al., 2018) that perform visual analogies in the latent space, similar to how Word2vec can be leveraged to learn how words appear in similar contexts. In medicine, latent representation learning is used (Zhou et al., 2019b) to work with incomplete multimodality data to learn independent representations for the prediction of Alzheimer's appearance. In biology, latent representations are used to model unmeasured quantities like pain and stress (Kopf & Claassen, 2021). Representation learning has also found its way to the earth and environmental sciences. Examples include learning better representations of 2D coordinates (Mai et al., 2022), and extracting unknown basin characteristics (Ghosh et al., 2021). However, it has been observed (Neumann et al., 2019) that this is not the case for several subfields, where practitioners prefer to use features based on expert knowledge and already proven algorithms that do not explore the latent space. This creates missed opportunities to examine whether improved predictive performance can be achieved, or new interactions to be found, or even to automate prediction pipelines. A representative case of such a missed opportunity is with estimating nitrogen application for fertilization purposes.

Nitrogen is the nutrient that crops and pasture draw from the soil in the greatest quantities (Rivai et al., 2021) and thus it becomes a growth-limiting factor (Zhou et al., 2019a). Nitrogen deficiency has been associated with low yields (Zhang et al., 2015b), and pastures in several countries suffer from it (Rotz et al., 2005; Whitehead, 1995). Farmers apply nitrogen-containing fertilizer to increase pasture growth rates but environmental concerns rise as nitrogen has been linked to soil(Han et al., 2015), freshwater and atmosphere pollution(Zhang et al., 2015a). Subsequently, agricultural practitioners are asked to control nitrogen application with precise doses based on nitrogen response rate[1](NRR). To control nitrogen application, research is being directed towards modern systems like digital twins (Nasirahmadi & Hensel, 2022) which can aid decision support through automation and

[1]Amount of extra kg of yield for every kg of nitrogen applied ($kg_{yield}/ha/kg_{nitrogen}$)

data integration. However, digital twins require components, such as process-based and machine learning (ML) models, that are able to predict NRR across several months in the future to be considered operational. Process-based models that can calculate NRR exist but they are of limited use as the weather months after nitrogen application is unknown yet required to run the model. Also, while NRR observations exist from experiments, they are sparse and not enough to train ML models.

In a recent study (Pylianidis et al., 2022), the authors attempted to tackle these data related problems by training ML models based on process-based model output. They predicted pasture NRR two months ahead of the prediction date, assuming absence of intermediate weather data. However, they performed common practices of environmental sciences like selecting features solely based on expert knowledge, averaging weather variables, and feeding all those to Random Forest (RF) (Breiman, 2001). Hence, the latent space of their data was not explored, and it was left unchecked if they could achieve similar performance with higher resolution data and an approach that learns the latent space. That would be important to examine since methods that learn the latent space have shown to perform equally or better than approaches which do not, as they may capture interactions which are not yet understood. Also, it would promote automation in systems like digital twins by removing the step of manual feature extraction. In this work, we are going to treat this study as a stepping stone, as it proved that we can have accurate NRR predictions in limited data settings, in a situation where an ML model and a process-based model alone were not operational.

Here, we perform a systematic comparison of different architectures to learn the latent space of a synthetic dataset for NRR prediction. We adopt the case study and data provided by (Pylianidis et al., 2022) and we use RF as a reference for comparing the performance of the architectures. We learn the latent representations of the inputs/outputs of a process-based model and predict NRR with a Multilayer Perceptron (MLP), an autoencoder (AE), and a dual-head autoencoder (DAE). We perform multiple runs for each architecture as well as RF to verify the robustness of each model. We then evaluate the results using error metrics as well as a domain-derived error threshold.

2 Materials and Methods

2.1 Case study & Data generation

The case study was concerned with finding the pasture NRR for two sites (Fig. 4) in New Zealand. The prediction target was the NRR of pasture dry matter grown in the two months after fertilizer application. Data generation was performed with APSIM (Holzworth et al., 2014). The simulation parameters of APSIM covered conditions which are known to affect pasture growth. The full factorial (Antony, 2014) of those parameters was created and put to APSIM. The range of each parameter can be seen in Table 2.

2.2 Data preprocessing

The generated data were processed to form a regression problem. The target variable was the NRR and the input variables were the weather, fertilizer amount, fertilization month, irrigation and a subset of biophysical variables produced by APSIM. From the generated daily data, only the data within the first 28 days prior to fertilization were preserved because pasture is supposed to 'lose memory' of past conditions after that time frame. **Weather data after that these 28 days were also discarded as they would be unavailable in operational conditions**. The remaining data were split into 67.5% training, 12.5% validation, and 20% test sets, based on years, to avoid information leakage during later processing stages. The validation set included the years [1979, 1987, 1999, 2007], the training set years [1979-2010] excluding the validation years, and the test set [2011-2018].

2.3 Architectures

2.3.1 Multilayer Perceptron

An MLP was put in the comparison to examine how its latent space learning capabilities compared with learning compressed representations of an AE. The loss was given by equation 1. The network topology can be seen in Fig. 1. Training parameters can be found in Appendix C.

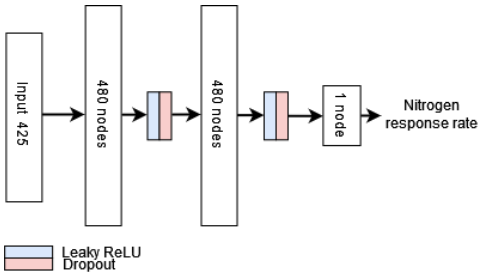

Figure 1: The topology of the MLP.

2.3.2 AUTOENCODER

An AE was selected to create a compressed representation of the input variables. The AE included skip connections similarly to (Li et al., 2018) from the encoder to the decoder to lessen degradation (He et al., 2016). The reconstruction loss was given by equation 2. After training the AE, the decoder was removed and replaced by an MLP. Training was performed again for the MLP (with loss given by equation 1, and a frozen encoder) to learn to predict NRR. The autoencoder topology can be seen inside the dashed line of Fig. 2.

$$L_{nrr} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2 \qquad (1) \qquad\qquad L_{rec} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{425}(y_{ij} - \hat{y_{ij}})^2 \qquad (2)$$

where $N = batch\ size$.

2.3.3 DUAL-HEAD AUTOENCODER

The encoder and decoder parts were the same as of the 'simple' AE. The addition was that the compressed representation was then directed to an MLP which carried out the NRR prediction task. The network topology can be seen in Fig. 2. Both the AE and the MLP were trained simultaneously, with the total loss being the summation of the equations 1, 2.

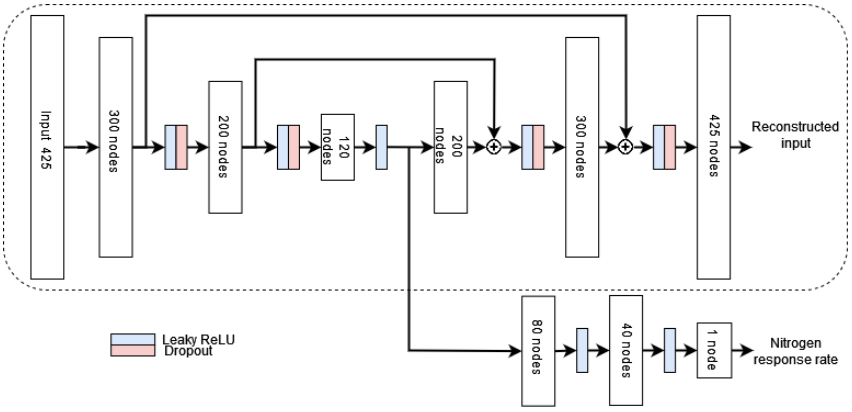

Figure 2: The topologies of the AE (inside the dashed border), and the DAE (altogether).

2.4 EVALUATION

The performance of the different architectures was compared using the *mean absolute error* (MAE), the variance learned from the latent representations using R^2, and the standard deviation of the predictions. Also, the predictive capacity of the models was assessed using a domain-derived error

threshold of 5 $kg_{yield}/ha/kg_{nitrogen}$. Prediction residuals systematically above that threshold constituted a model incapable for operational use. Each architecture, as well as RF, were ran 5 times with different seeds to verify the robustness of the results.

3 RESULTS

In Table 1, we see the error metrics for each architecture and location aggregated over the runs. RF has the lowest error and highest explained variance for both locations. AE has the largest error and lowest R^2. DAE has the lowest errors between the architectures with just a slight edge over MLP. Regarding the standard deviations of the predictions, AE has the lowest deviation and DAE the highest.

Table 1: Error metrics for each architecture and RF aggregated over the runs. σ refers to the standard deviation of the predictions of the runs.

	RF			**MLP**			**AE**			**DAE**		
	MAE	R^2	σ	MAE	R^2	σ	MAE	R^2	σ	MAE	R^2	σ
Waiotu	1.55	0.68	3.53	1.85	0.62	3.63	2.26	0.45	3.34	1.72	0.65	3.62
Mahana	1.87	0.61	4.16	2.19	0.53	4.57	2.72	0.38	4.02	2.07	0.5	4.9

In Fig. 3, we can see how the residuals of the different architectures compared to RF across months. The residuals were aggregated over years and the five runs. For the first location, Waiotu, we observe that all candle bodies are below the domain-derived threshold that we set, with some upper whiskers overcoming the threshold. DAE seems to be the best performing architecture, since it has the shortest body of the three and also lower medians. Also, DAE appears to have slightly lower errors than RF in several cases. AE appears to have the largest errors, with large candles and extended upper whiskers. For Mahana, most candles are below our threshold but with larger bodies than Waiotu and taller upper whiskers. For January and December AE is above and close to the threshold respectively, generally having the highest errors. MLP and DAE seem to outperform RF for January, February and December. Again, DAE has the best performance from the three architectures with candles being lower than the rest and lower medians.

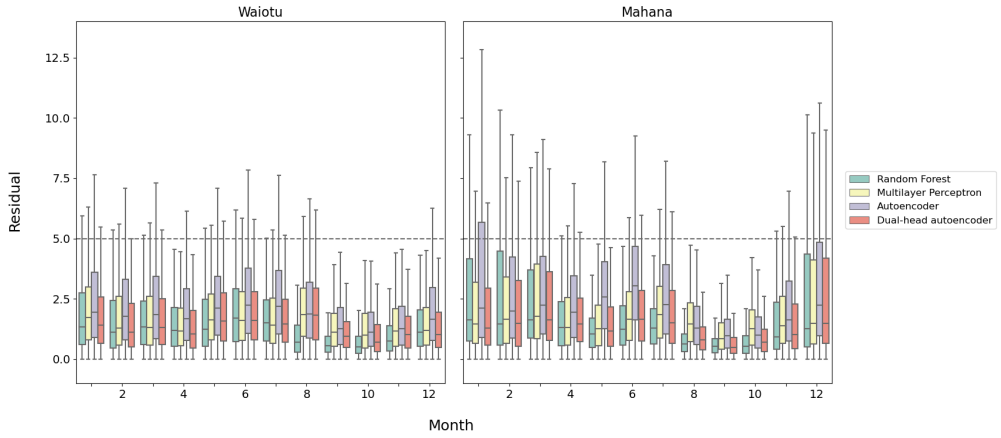

Figure 3: Residuals for each architecture and RF aggregated over years and runs. The horizontal dashed line indicates the domain-derived threshold. The body of the candles represents 50% of the values, and the bottom and top whiskers 25% each. The horizontal lines inside the candles show the median.

4 DISCUSSION

From a performance oriented perspective, we could deduct that RF is the best model by looking at the error metrics. However, we cannot judge how much better it is from MLP and DAE or how well the different architectures learned because their errors and standard deviations were similar. A more clear case is that of AE, which underperforms the rest of the models considerably. The standard deviation of its predictions might be the smallest but this may be due to learning a small part of the lower dimensional manifold, created on the output of the encoder, and thus not being able to offer varied predictions.

Examining the residuals of the architectures, we observe that they provide predictions mostly within our domain-derived threshold. The multiple runs and yearly aggregation demonstrate the stability[2] of the models, showcasing their robustness. This conveys that the models were able to extract latent representations which allow them to be potentially used in an operational setting. AE appeared to be the weakest model since its candles were generally larger, exceeded our threshold in Mahana. The two stage training (first autoencoder, then replacing then decoder with an MLP) may have caused it to weigh more on learning how to reconstruct its inputs rather than how NRR is connected with them. On the other hand, the MLP was able to extract more meaningful representations for NRR predictions something evident from the fact that in many months it was on par and sometimes better than RF. Similarly, DAE performed equal or better than RF in most months for both locations. This may imply that the latent space that MLP and DAE learned covered aspects which were not represented in the manually derived expert features of RF. Also, the performance gap between AE and DAE showed that optimizing simultaneously for two tasks when one task depends on the other can make the network learn better representations in the contexnt of this study.

An aspect potentially affecting the results of the architectures is how well input features can be represented in the latent space learned by the models. APSIM has a binary input variable to control the existence of irrigation which materially changes NRR. This variable is the only signal outside of APSIM that indicates this type of change. In our architectures there are several layers and this signal may be difficult to be preserved and projected in the latent space. On the contrary, for algorithms like RF this signal is not lost and can easily change how predictions are made. This may be a reason for not having higher performance with the different tested architectures and something to be accounted for when learning latent representations from environmental data.

5 CONCLUSION AND FUTURE WORK

In this study, we assessed the ability of three neural network architectures to learn the latent space of process-based model output for operational decision support. We compared the results with those of RF which was already proved operational in another study. The results were promising since all architectures were able to learn representations that captured enough variation to be considered operational. The MLP and DAE outperformed RF in certain cases, showing that they can uncover latent factors from the input space which accounted for more variability than manually selected features based on domain knowledge. This is an important step towards providing operational decision support in modern systems like digital twins, avoiding feature engineering in certain cases and automating prediction pipelines.

In the future, we would like to experiment with more synthetic datasets to examine if we can generalize our findings to other case studies. Also, we would like to validate the models using observation data to further verify how operational the created models are. Another important aspect would be to experiment with architectures that provide explicit interpretability of the latent space and examine how this space compares with expert-derived features.

ACKNOWLEDGMENTS

This work has been supported by the European Union Horizon 2020 Research and Innovation program (Grant #810775, "Dragon") and the Wageningen University and Research Investment Program "Digital Twins".

[2]Variation between the months exists due to seasonality. December to February is summer in New Zealand with conditions that increase uncertainty for pasture growth and thus NRR errors.

REFERENCES

Jiju Antony. 6 - Full Factorial Designs. In Jiju Antony (ed.), *Design of Experiments for Engineers and Scientists (Second Edition)*, pp. 63–85. Elsevier, Oxford, 2014. ISBN 978-0-08-099417-8. doi: https://doi.org/10.1016/B978-0-08-099417-8.00006-7. URL https://www.sciencedirect.com/science/article/pii/B9780080994178000067.

Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Rahul Ghosh, Arvind Renganathan, Ankush Khandelwal, Xiaowei Jia, Xiang Li, John Neiber, Chris Duffy, and Vipin Kumar. Knowledge-guided Self-supervised Learning for estimating River-Basin Characteristics. 9 2021. URL http://arxiv.org/abs/2109.06429.

Mei Han, Mamoru Okamoto, Perrin H Beatty, Steven J Rothstein, and Allen G Good. The Genetics of Nitrogen Use Efficiency in Crop Plants. *Annual Review of Genetics*, 49(1):269–289, 11 2015. ISSN 0066-4197. doi: 10.1146/annurev-genet-112414-055037. URL https://doi.org/10.1146/annurev-genet-112414-055037.

K He, X Zhang, S Ren, and J Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. ISBN 1063-6919. doi: 10.1109/CVPR.2016.90.

Dean P. Holzworth, Neil I. Huth, Peter G. deVoil, Eric J. Zurcher, Neville I. Herrmann, Greg McLean, Karine Chenu, Erik J. van Oosterom, Val Snow, Chris Murphy, Andrew D. Moore, Hamish Brown, Jeremy P.M. Whish, Shaun Verrall, Justin Fainges, Lindsay W. Bell, Allan S. Peake, Perry L. Poulton, Zvi Hochman, Peter J. Thorburn, Donald S. Gaydon, Neal P. Dalgliesh, Daniel Rodriguez, Howard Cox, Scott Chapman, Alastair Doherty, Edmar Teixeira, Joanna Sharp, Rogerio Cichota, Iris Vogeler, Frank Y. Li, Enli Wang, Graeme L. Hammer, Michael J. Robertson, John P. Dimes, Anthony M. Whitbread, James Hunt, Harm van Rees, Tim McClelland, Peter S. Carberry, John N.G. Hargreaves, Neil MacLeod, Cam McDonald, Justin Harsdorf, Sara Wedgwood, and Brian A. Keating. APSIM - Evolution towards a new generation of agricultural systems simulation. *Environmental Modelling and Software*, 62:327–350, 12 2014. ISSN 13648152. doi: 10.1016/j.envsoft.2014.07.009.

Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2Vec: Unsupervised Representation Learning for Spatially Distributed Data. Technical report, 2018. URL www.aaai.org.

Andreas Kopf and Manfred Claassen. Latent representation learning in biology and translational medicine. *Patterns*, 2(3):100198, 2021. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2021.100198. URL https://www.sciencedirect.com/science/article/pii/S2666389921000015.

Lianfa Li, Ying Fang, Jun Wu, and Jinfeng Wang. Autoencoder Based Residual Deep Networks for Robust Regression Prediction and Spatiotemporal Estimation, 2018.

Gengchen Mai, Yao Xuan, Wenyun Zuo, Krzysztof Janowicz, and Ni Lao. Sphere2Vec: Multi-Scale Representation Learning over a Spherical Surface for Geospatial Predictions. 1 2022. URL http://arxiv.org/abs/2201.10489.

Abozar Nasirahmadi and Oliver Hensel. Toward the Next Generation of Digitalization in Agriculture Based on Digital Twin Paradigm. *Sensors*, 22(2), 1 2022. ISSN 14248220. doi: 10.3390/s22020498.

Maxim Neumann, Andre Susano Pinto, Xiaohua Zhai, and Neil Houlsby. In-domain representation learning for remote sensing. 11 2019. URL http://arxiv.org/abs/1911.06721.

Christos Pylianidis, Val Snow, Hiske Overweg, Sjoukje Osinga, John Kean, and Ioannis N Athanasiadis. Simulation-assisted machine learning for operational digital twins. *Environmental Modelling & Software*, 148:105274, 2022. ISSN 1364-8152. doi: https://doi.org/10.1016/j.envsoft.2021.105274. URL https://www.sciencedirect.com/science/article/pii/S1364815221003169.

Reza Ramdan Rivai, Takuji Miyamoto, Tatsuya Awano, Rie Takada, Yuki Tobimatsu, Toshiaki Umezawa, and Masaru Kobayashi. Nitrogen deficiency results in changes to cell wall composition of sorghum seedlings. *Scientific Reports*, 11(1):23309, 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-02570-y. URL https://doi.org/10.1038/s41598-021-02570-y.

C A Rotz, F Taube, M P Russelle, J Oenema, M A Sanderson, and M Wachendorf. Whole-farm perspectives of nutrient flows in grassland agriculture. *Crop Science*, 45(6):2139–2159, 11 2005. ISSN 0011-183X. doi: 10.2135/cropsci2004.0523. URL http://www.scopus.com/inward/record.url?scp=27644585386&partnerID=8YFLogxKhttp://www.scopus.com/inward/citedby.url?scp=27644585386&partnerID=8YFLogxK.

David Charles Whitehead. *Grassland nitrogen*. CAB international, 1995. ISBN 9780851989150.

Xin Zhang, Eric A Davidson, Denise L Mauzerall, Timothy D Searchinger, Patrice Dumas, and Ye Shen. Managing nitrogen for sustainable development. *Nature*, 528(7580):51–59, 2015a. ISSN 1476-4687. doi: 10.1038/nature15743. URL https://doi.org/10.1038/nature15743.

Yangjun Zhang, Lubin Tan, Zuofeng Zhu, Lixing Yuan, Daoxin Xie, and Chuanqing Sun. TOND1 confers tolerance to nitrogen deficiency in rice. *The Plant Journal*, 81(3):367–376, 2 2015b. ISSN 0960-7412. doi: https://doi.org/10.1111/tpj.12736. URL https://doi.org/10.1111/tpj.12736.

Chunyan Zhou, Jing Le, Dengxin Hua, Tingyao He, and Jiandong Mao. Imaging analysis of chlorophyll fluorescence induction for monitoring plant water and nitrogen treatments. *Measurement*, 136:478–486, 2019a. ISSN 0263-2241. doi: https://doi.org/10.1016/j.measurement.2018.12.088. URL https://www.sciencedirect.com/science/article/pii/S026322411831234X.

T Zhou, M Liu, K H. Thung, and D Shen. Latent Representation Learning for Alzheimer's Disease Diagnosis With Incomplete Multi-Modality Neuroimaging and Genetic Data. *IEEE Transactions on Medical Imaging*, 38(10):2411–2422, 2019b. ISSN 1558-254X. doi: 10.1109/TMI.2019.2913158.

APPENDIX

A CASE STUDY SITES

Figure 4: New Zealand sites. Sites on the red circles are the ones included in this work.

B APSIM SIMULATION PARAMETERS

Table 2: APSIM simulation parameters and their ranges. The full factorial of those parameters comprised the input to APSIM.

Parameter	**Range**
Weather	daily weather from 3 sites
Soil water	42, 67, 110 and 177 mm of plant-available water
Soil fertility	2, 4, and 6% of carbon concentration
Irrigation	irrigated, non-irrigated
Fertilizer year	1979-2018
Fertilizer month	January-December
Fertilizer day	5^{th}, 15^{th} and 25^{th} of the month
Fertilizer amount	0, 20, 40, 60, 80 and 100 kg N / ha

C TUNING AND TRAINING

The training data were standardized for each location independently. The test and validation data were standardized with the corresponding training scaler, in order to have the same mean and standard deviation.

The number of layers, nodes in each layer, optimizer parameters, and dropout rate for each architecture were based on the results of a preliminary study. The MLP had two hidden layers with 480 nodes each, optimization with Adam (lr=0.001, weight_decay=0.0001), dropout rate 20%, batch size 64 and 100 epochs. The AE had five hidden layers (300, 200, 120, 200, 300 nodes), optimization with AdamW (lr=0.0003, weight_decay=0.01), dropout rate 10%, batch size 64 and 60 epochs. After training, the decoder was replaced with an MLP with two hidden layers of 180 nodes each and training for 60 epochs. The DAE had the same autoencoder and optimizer as AE, with an addition of an MLP connected to the output of the encoder. The MLP had two hidden layers (80, 40 nodes). The whole network was trained with batch size 64, for 100 epochs.

RF took as input weekly aggregated features which were only a few and were considered explanatory so no feature selection took place. Hyperparameter tuning was performed using Bayesian optimization with 25 iterations and the 5-fold cross-validation score as a metric for each iteration. The tuned parameters can be seen in Table 3.

Table 3: The parameters tuned during Bayesian optimization for RF.

Parameters	**Range**
n_estimators	50-800
max_depth	3-12
min_samples_split	30-500
min_samples_leaf	30-500
max_features	0.33