Journal of Radiation Oncology Informatics

Machine Learning and Radiomics to Predict Local Recurrence of Lung Cancer After Radiotherapy

Alli Jan[1†,‡], Peter Wright[2], Dale Glennan[1], Andrew Miller[3]

[1]University of Wollongong, Graduate School of Medicine, NSW Australia
[2]University of Illinois, Springfield, Illinois USA
[3]Illawarra Cancer Care Centre, Wollongong Hospital, Wollongong Australia

Abstract

Purpose: To assess the likelihood of local recurrence of lung malignancies following stereotactic ablative radiotherapy (SABR) by evaluating clinical and radiomic features with machine learning and novel use of deep learning methods.
Methods: Pre-treatment CT images were obtained from 70 patients with primary lung malignancies. The malignancy was segmented by the treating radiation oncologist, and 107 radiomic features were extracted from the image. The data underwent feature reduction via Spearman's correlation and selection with adapted least absolute shrinkage and selection operator regression analysis. A random forest model and a multi-layer perceptron (MLP) with a cost-sensitive classifier were independently used to assess for the local recurrence of malignancy. The recurrence likelihood predictions from each of these were used to stratify patients into groups with a high and low risk of recurrence. These were assessed for time-to-event predictions using Kaplan–Meier analyses and Gray's test to evaluate the separation between the high- and low-risk groups. The prognostic capacity of the models was evaluated with a concordance index, 95% confidence intervals and bootstrapping (10,000 iterations).
Results: In the context of a small sample size, the MLP was able to predict the recurrence of malignancy with 100% sensitivity and 91% specificity (area under the receiver operating characteristic curve 0.95). The MLP predictions showed a statistically significant separation of high- and low-risk patients, and a robust model fit (p=0.04, c=0.79), which outperformed random forest model predictions (p=0.15, c=0.41) that did not reach statistical significance.
Conclusions: Radiomic data analysis with an MLP showed improved prediction potential within this dataset compared to random forest models for predicting the endpoint. More studies with larger populations and a longer duration of follow-up are required to further assess the functionality of these methods of analysis for predicting the local recurrence of lung cancer after SABR.

Keywords: radiomics, machine learning, artificial intelligence, artificial neural network, lung cancer, local treatment failure, SABR

1 Introduction

Lung cancer is the leading global cause of cancer-related deaths [1]. Stereotactic ablative radiotherapy (SABR) is a current pillar of lung cancer treatment that enables some patients to avoid surgery and allows others who are non-surgical candidates to access curative treatment. SABR is very effective for treating early-stage lung cancer and leads to the remission rates of 90%–95% [2]. Patients at a higher risk of local recurrence would benefit from alterations to their treatment regime such as modified radiotherapy dosing, surgical management, the addition of adjuvant therapy or closer follow-up. Predicting which patients will experience local recurrence through clinical means is difficult and often inaccurate [4]. This paper discusses image analysis with radiomics and artificial intelligence (AI) as an alternate means of identifying high-risk patients to enable early adaptations to be made to their care [5].

† The trained MLP is available on the author's GitHub at github.com/allijan45.
‡ All data are available from the author upon reasonable request, pending the approval of the local ethics board.

This article is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA. Creative Commons Attribution 4.0 International license.

Radiomic feature extraction allows a large number of features in images to be measured and analysed using predefined algorithms that produce a quantitative output representing markers of density, intensity, fine texture, coarse texture and morphology [5]. An explanation of these features is available in *Appendix 1*. Using radiomic features enables the analysis of characteristics of the region of interest (ROI) that are not able to be accurately assessed by gross inspection and aids in the identification of clinically significant, non-obvious patterns. The analysis of radiomic features with AI facilitates the construction of models that can be trained to predict the study endpoint [4]. For the purpose of reporting radiomics results with the highest possible utility, this paper employs a standardised set of features that are compliant with the Imaging Biomarker Standardisation Initiative (IBSI) and reports the statistical analysis in accordance with best-practice guidelines recommended by the Radiomics Quality Score [6].

The findings of the previous literature are shown in *Appendix 2* [1,2,4,7–9]. Four of these studies identified the markers of the local recurrence of lung cancer with varying levels of significance. Only one of the six studies identified was able to predict local recurrence using a model with a combination of CT and PET radiomic features [2]. Five of these six papers focused on their other endpoints, such as overall survival, which they were all able to predict more accurately due to the higher incidence of events. The use of overall survival as analogous to treatment failure in a cohort with a high expected cure rate and who are likely to be elderly and have other comorbidities can cause the overprediction of results and impact bias. Despite the practical limitations in predicting local recurrence, focusing on it as an endpoint could potentially have a higher clinical utility than overall survival due to the opportunity to change patient management with a goal of preventing treatment failure. The existing literature confirms a known machine learning consensus: it can be very difficult to predict results in unbalanced data (data with a low number of events, e.g. 5% recurrence rates).

This paper will demonstrate an alternate approach to predicting the recurrence of lung cancer after SABR with the use of two methods of AI: machine learning with a random forest model and deep learning with a multi-layer perceptron (MLP). A random forest model is a type of supervised machine learning that generates hundreds of decision trees to average the results and predict the most likely outcome. Deep learning is an intricate structure of advanced algorithms that identify patterns and trains itself to interpret outputs through complex model construction. An MLP is a type of deep learning that uses several layers to train the neurons within an artificial neural network. The comparison between the machine learning and deep learning methods on the same patient population will serve to demonstrate a novel method of predicting local recurrence.

2 Methods

2.1 Participant Characteristics

Participant selection occurred at two sites within the Illawarra Shoalhaven Local Health District. Participant data were pooled, and no distinction between sites was made. All patients provided individual consent for the future use of their deidentified medical data in research. This study was approved by the institutional research ethics committee. Inclusion criteria were patients aged over 18 who underwent SABR treatment for early-stage, primary lung cancer in the health district between 2017 and 2020. Exclusion criteria were participants who had recurrences after surgery, metastatic cancers, or who did not have complete, structured clinical data characteristics. Participants were classified as having either no local failure or local failure, which was defined as a recurrence of malignancy within 2 cm of the initial gross tumour volume (GTV). Local failure was confirmed radiologically by the treating radiation oncologist.

2.2 CT Imaging and Malignancy Contouring

The free-breathing, non-contrast, pre-treatment CT images of all patients were acquired with Siemens SOMATOM Confidence CT scanners (120 kv, 2 mm slices). The scanners were calibrated with identical protocols to minimise inter-scanner variance.

There were two methods of delineation used in this paper: manual and to voxel value. Manual delineation was done in Pinnacle 3D® [10], and auto-contouring to voxel value segmentation was performed in OnkoDICOM [11]. In CT imaging, the voxel value of the normal lung is known to be between 100 and 400, which can be used to find an exact border of normal tissue. Each contour was verified by a single radiation oncologist prior to radiomic feature extraction.

2.3 Radiomic Feature Extraction

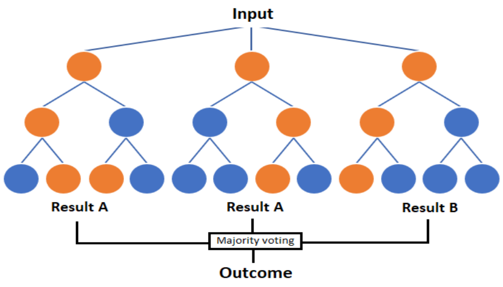
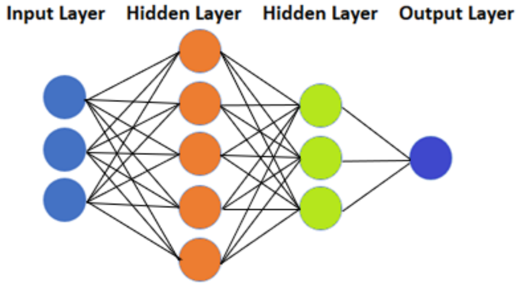
A total of 107 radiomic features were extracted from the images using OnkoDICOM. All the features extracted are part of the Pyradiomics Python library of features, and all comply with algorithms defined by IBSI.

2.4 Feature Reduction and Selection

Feature reduction was used to reduce model complexity and to minimise the number of parameters tested to avoid collinearity, overfitting the data and developing overly optimistic results. The data were normalised to aggregate the scales by converting all values to a z-score. A Spearman's rank correlation coefficient was calculated on each value to determine similarity to other values [12]. A correlation coefficient >0.90 was deemed "very strong," and, as such, all variables with a high correlation to other variables were removed. The feature that was removed was chosen based on which of the two interacting variables had very strong correlation with the highest number of other variables.

Variable regularisation and selection were performed with an adaptive least absolute shrinkage and selection

Table 1: Comparison of Random Forest vs MLP. Optimisation and training explanations are unique to this method. Implementation of the methodology including the code used can be seen in *Appendix 4*.

	Random Forest model	**Multilayer Perceptron model**
Concept	Supervised machine learning that generates hundreds of decision trees and averages results to predict the most likely outcome.	Deep learning within an artificial neural network that trains itself to predict patterns and outputs through using algorithms in hidden layers.
Depiction	 Input Result A Result A Result B Majority voting Outcome	
Optimisation	Hyperparameters (number of decision trees, number of features sampled at each branch, and tree complexity) were optimised through use of a grid search to minimise error or maximise the model fit.	Stochastic gradient descent: model runs through and calculates error between the predicted and actual outcome and continues the epochs of model while changing parameters to minimise errors in output. Early stopping, dropout and random noise for regularisation.
Training	Model was trained on 70% of the data and tested on 30% of the data.	Model used 10 folds of cross-validation: model repeatedly splits into several training/validation datasets to allow the use of all data for training.

operator (LASSO) regression analysis. The method penalises binomial logistic regression by shrinking coefficient magnitudes towards zero if they are not strongly associated with the outcome. The adaptive LASSO has oracle properties and counteracts known biases in regular LASSO by weighting the coefficients used. Features that failed to show importance in the adaptive LASSO analysis were removed.

As the existing literature shows the superiority of combined clinical/radiomic feature modelling to separate models [9,13,14], all models were built with a combination of radiomic features and clinical features.

2.5 Random Survival Forest Model

The eight radiomic features significant for local recurrence and the nine clinical features were used to construct a random forest–supervised machine learning algorithm in RStudio [15] using the programming language R [16]. Random forest models perform better without noisy data, which was the reason for the use of the reduced set of radiomic features. All of the R packages used are listed in *Appendix 3*. The data were randomly partitioned into a training set (70% of the data) and a testing set (30%). The model hyperparameters m_{try}, the number of trees, and minimum node size were optimised with a grid search to improve the robustness of the model. Using the optimal hyperparameters, the model was trained and then run on the test data with the aim of predicting which participants in the test group would develop local recurrence.

2.6 Multilayer Perceptron Model

An MLP was built in Weka 3.8.5 [17]. The MLP used a cost-sensitive classifier with a 2 × 2 cost matrix penalising false negatives at 14 times the cost of a false positive and no penalty for true positives or true negatives. The classifier DL4jMlp [18] was used with two hidden layers: one dense layer with an ActivationReLU activation function and eight outputs, and an output layer with an ActivationSigmoid activation function, a LossBinaryXCENT loss function and one output. The MLP was created to run with eight epochs and an early stopping function to cease computations after two epochs with no improvement in results as a regularisation method to avoid model overfitting. Other regularisation methods included the use of a dropout layer to ensure that the multiple internal representations of the endpoint were learned by the model, and the addition of random noise to improve error generalisation and structure mapping. Stochastic gradient descent was used for the optimisation algorithm with no gradient normalisation method used.

The input layers consisted of all of the 107 radiomic features and the nine clinical features. The output layer was a sigmoid function with one neuron that classified patients as a number between 0 (no local recurrence) or 1 (local recurrence).

The training of the MLP was performed with 10 folds of cross-validation and the model was assessed with an area under the receiver operating characteristic curve (AUC). Random forests and MLPs were the only types of machine learning models built for this paper. MLP hyperparameters were altered to adjust model output in accordance with standard practices. Most MLP changes between models were the internal adjustments of the weights and biases of inputs through the use of stochastic gradient descent.

The prediction results from the random forest model and the MLP were used to stratify the participants into high- and low-risk groups around the median [19]. This part of the method was performed separately to the model's case recurrence predictions. The random forest model and the MLP both classified how likely someone was to develop local recurrence, and the 35 participants with the highest risk from each model output (regardless of whether the model predicted local recurrence or not) were classified as "high risk" and the lowest 35 were classified as "low risk." This

Table 2: Participant clinical characteristics: Ordinal data are displayed as a median or as median, (range), nominal data are depicted as the total number (percentages of cohort).

	Total (n=70)	Local Failure (n=3)	No Local Failure (n=67)
Size (mm)	19.5 (7–46)	25.6	19.2
GTVp (ml)	5.97 (0.85–21.9)	9.24	5.83
SUVp	5.91 (1.3–19.8)	7.53	5.72
Radiation dose (Gy)			
BED For Prescription	108 (52.5–151.2)	105.6	108.1
BED of PTV max	106.2 (52.8–325.8)	102.4	106.4
BED of PTV min	177.3 (79.4–461.3)	164.2	178.6
Age at diagnosis	73.68 (45.11–92.79)	70.14	73.9
Gender			
Female	43 (61%)	1 (33%)	42 (64%)
Male	27 (38%)	2 (67%)	25 (36%)
Grade			
1	11 (16%)	0	11 (16%)
2	19 (27%)	1 (33%)	18 (28%)
Unknown	40 (57%)	2 (67%)	38 (56%)
Location			
Upper lobe	37 (53%)	1 (33%)	36 (55%)
Middle lobe	6 (9%)	0	6 (9%)
Lower lobe	27 (38%)	2 (67%)	25 (36%)
Histology			
Adenocarcinoma	45 (64%)	1 (33%)	44 (66%)
Squamous cell carcinoma	13 (19%)	1 (33%)	12 (18%)
Other	12 (17%)	1 (33%)	11 (16%)
Duration of follow-up (years)	1.93 (0.41–4.1)	1.43	1.95

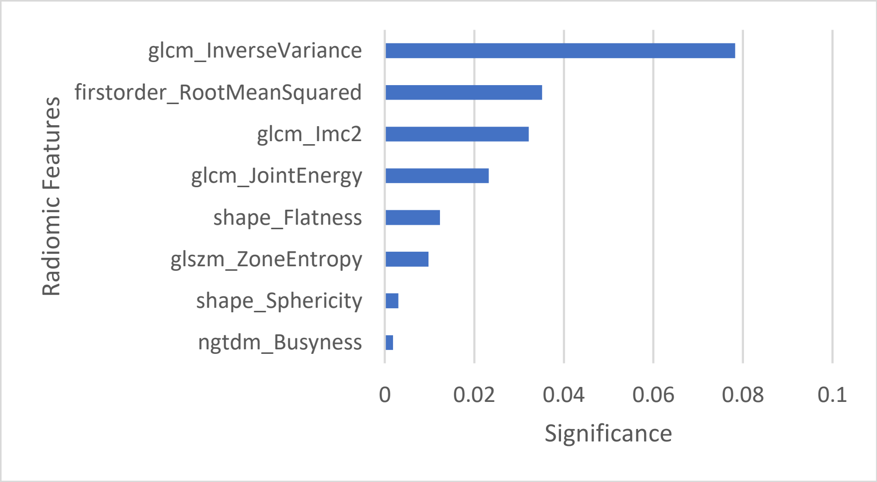


Figure 1 – Significance of Radiomic Features in Adaptive LASSO. Figure 1 depicts the significance various features had compared to the endpoint of local recurrence after adaptive LASSO selection. Glcm_InverseVariance, firstorder_RootMeanSquared, and glcm_Imc2 were the most significant features. A full list of features is in *Appendix 1* and features not listed in the figure had a significance of "0" on Adaptive LASSO.

Table 3 – Features used in AI modelling. Clinical features were selected based on the completion of structured datasets within the electronic medical record. Radiomic features were selected by adaptive LASSO.

Radiomic features selected by adaptive LASSO	Clinical features
Shape-Flatness	Gender
Shape-Sphericity	Grade
Firstorder-RootMeanSquared	Age at diagnosis
GLCM-Imc2	SUVp
GLCM-InverseVariance	GTVp (ml)
GLCM-JointEnergy	Axis size (mm)
GLSZM-ZoneEntropy	BED for prescription
NGTDM-Busyness	BED for PTV max
	BED for PTV min

conservative risk group determination was done specifically to prevent the reporting of overly optimistic results in accordance with the Radiomics Quality Score [6]. The high- and low-risk groups were assessed on Kaplan–Meier plots [20] for time-to-event prediction, and the significance of separation of the groups was assessed with Gray's test [21]. The model fit was analysed with an adjusted Harrel's concordance index (c-index) [22]. The accuracy of predictions was evaluated with 95% confidence intervals (Cis) with 10,000 iterations of bootstrapping.

3. Results

3.1 Participant Characteristics

A total of 70 participants were eligible for this study between the two sites. The median age of patients at diagnosis was 73 years, and there were 43 female and 27 male patients. Patients were treated with a mean BED for a prescription of 108 Gy, and the median duration of treatment was 8 days. The mean duration of follow-up was 1.93 years. Three cases of local failure were identified (4.2%).

3.2 Feature Reduction and Selection

Of the 107 radiomic features evaluated, 57 were removed for having inter-feature correlation >90%. There was no significant correlation of any clinical features with any radiomic features, and thus, none of these were removed from the analysis. The adaptive LASSO analysis selected eight radiomic features that were significant for the local

Table 4 – Confusion Matrix of MLP Predictions. MLP model was built to assess for the local recurrence of lung cancer after SABR. Model predicted 18 cases of local recurrence, including all actual cases of local recurrence.

		Predicted	
		Negative	Positive
Actual	Negative	52	15
	Positive	0	3

Specificity	**Sensitivity**	**Precision**	**Recall**	**F-Measure**	**AUC-ROC**
0.78	1.00	0.78	0.78	0.85	0.88

Accuracy	**RMSE**
78.6%	0.43

recurrence with the other 49 features failing to predict for the endpoint.

3.3 Random Forest Model Predictions

The random forest model was not able to accurately predict recurrence of lung cancer in the testing group. The random forest model maintained a low overall error rate by misclassifying all the cases of local recurrence.

3.4 Deep Learning Model Predictions

The MLP analysis was able to correctly classify 78.6% of all results and correctly predicted all cases of local recurrence. The AUC was 0.88 with a root mean squared error of 0.46. For the purposes of this paper, a value of above 0.7 for the AUC [23,24] or c-index [24] demonstrated a model of sufficient predictive capability.

3.5 Outcome Predictions

Following risk stratification into high- and low-risk groups, the random forest model classified all the cases of local recurrence in the high-risk group. Gray's test of the random forest model had a p-value of 0.15 and thus was not able to demonstrate statistically significant separation between the groups. The c-index of the model was 0.41 which demonstrates poor fit of the model.

For time-to-event analysis of the MLP predictions the model was able to demonstrate statistically significant separation via analysis with Gray's test (p=0.04). A p-value of <0.05 was considered significant for the purpose of this paper. The high- and low-risk groups had a difference between the mean time before an event (local recurrence or censoring) of 0.44 years (95% CI = 0.03–0.86). The model performed well with a c-index of 0.80.

4. Discussion

This paper was able to successfully differentiate between patients with a high- and low-risk of lung cancer recurrence after SABR via an MLP with a cost-sensitive classifier. This method was able to demonstrate the improved prediction of outcomes compared to a random forest model that was unable to show statistical significance. To the authors' knowledge, this is the first use of an artificial neural network in combination with radiomic data to predict lung malignancy recurrence, and this is the first accurate prediction of this risk using only radiomic data from pre-treatment CT imaging.

Recent trends in machine learning have led to the use of deep learning neural networks as the dominant methodology in a number of perceptual classification competitions, wherein deep learning consistently outperforms probabilistic modelling, kernel methods and tree models [41]. Similar to this consensus, this paper found improved modelling with the deep learning methods compared to random forest models. While the purpose of this paper is not to claim methodological superiority, the cautious reporting of positive results serves to demonstrate the proof of concept of the methodology for this indication. The use of MLPs has been established as a valid methodology in predicting outcomes based on genetic markers [25] and in using radiomics to identify disease [26] or to differentiate tumour subtypes [27]. These results are preliminary but contribute to a growing body of evidence demonstrating the potential for the use of deep learning to analyse radiomic data and suggest a benefit to this method in assessing for the local recurrence of lung malignancies.

The MLP was able to accurately categorise 78.6% of all patients and predict 100% of local recurrence (AUC=0.88). The modelling had a 100% negative predictive value and a 17% positive predictive value. The 15 participants who were classified as false positives by the MLP could potentially be at a highest risk of experiencing local recurrence in the future. The short follow-up time before data collection for the participants in the paper likely means that some of the other participants who would experience local recurrence had not yet been identified. The benefit of the cost-sensitive analysis within the MLP can be noted here. The random forest model, although unable to identify any cases of local recurrence, maintained an overall accuracy of >95%. Due to having few cases of local recurrence, the model tends to misclassify those results to maintain the lowest possible

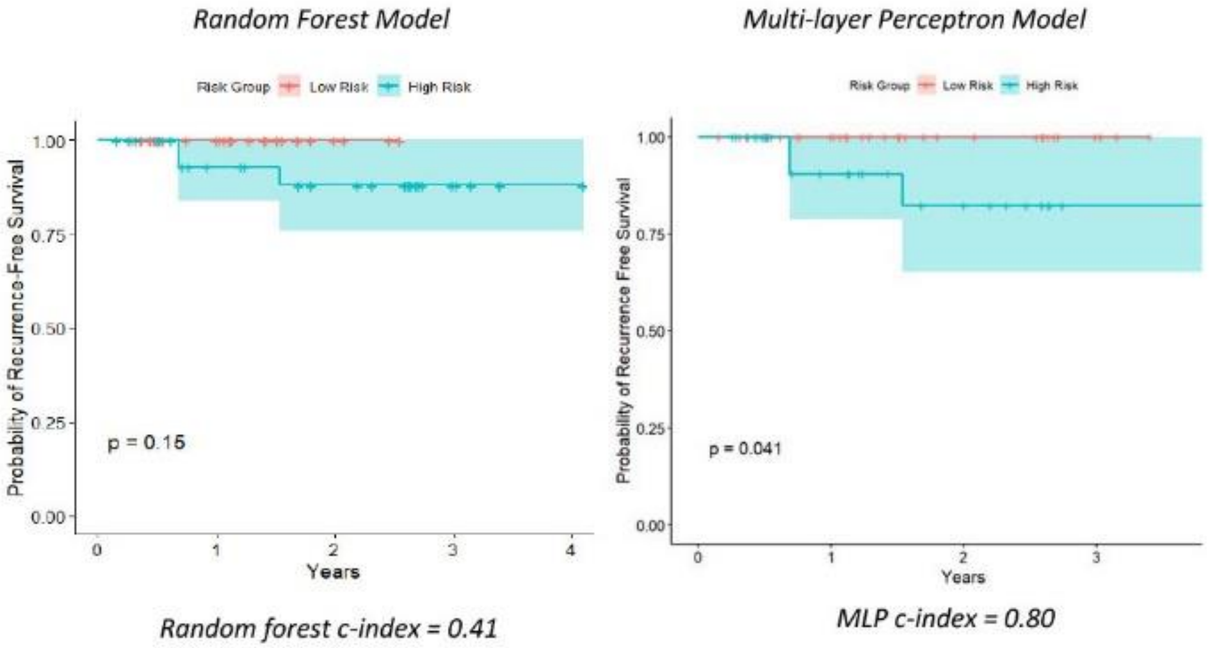


Figure 2 – Probability of Local Recurrence of Malignancy in Combined Clinical/Radiomic Feature Models. Time-to-event outcome prediction for the random forest and MLP models based on high- and low-risk groups. Random forest model was not able to show statistically significant separation between high- and low-risk groups. MLP was able to show separation between groups with p=0.04.

error. This is common in machine learning modelling with unbalanced data. The cost-sensitive analysis in the MLP penalised false negatives, which improved true-positive prediction and increased false-positive prediction. This decreased the accuracy of the model to 78.6% but increased the sensitivity of the model to 100%.

While these results appear very successful at predicting the endpoint, using the MLP solely for predicting cases of local recurrence is falsely optimistic. Models built on a limited set of data become overfit to the initial dataset, and thus, it is likely that this overfitting would make this model less successful at predicting local recurrences in new data. This decreases the reliability of the high sensitivity and the AUC.

A different metric of outcome for the MLP predictions was used for a more accurate impression of the model's actual capacity to assess patients. The likelihood of risk was assessed by differentiating patients into high- and low-risk groups around the median. As this then includes 17 participants who the model did not predict would experience local recurrences (but did deem at a higher risk), it greatly decreases the effects of model overfitting on the analysis. The analysis was able to demonstrate a significant separation between the high- and low-risk groups (p=0.04, c=0.80). There was an average of 5 months longer until a participant in the low-risk group experienced an event compared to the participants in the high-risk group. The MLP outperformed the random forest model by having more significant time-to-event prediction capabilities. The random forest model was not able to differentiate accurately between high- and low-risk participants and demonstrated a poor model fit (p=0.15, c=0.48).

In the existing literature, five papers were able to identify markers that were significant for local recurrence, with only one paper being able to predict local recurrence. These methods included statistical analysis with the Cox proportional hazard model or machine learning with random forest modelling. These outcomes are similar to the results of this paper, wherein the adaptive LASSO identified markers significant for local recurrence and the random forest modelling was unable to accurately predict local recurrence or high-risk patients. The use of AI in predicting the local recurrence of cancer can also forego assessing radiomic features entirely with the use of the types of AI that directly assess images, such as convolutional neural networks (CNNs). These have the benefits of not being limited to pre-defined features and enable the model to develop unique features that can be more predictive of the endpoint. The development of unique features within the model are unable to be externally recreated due to the nature of the AI, which can limit reproducibility. Of course, the model itself can be externally validated on new datasets although this does depend upon the authors publishing their models. A review of the existing literature only identified one CNN used for the local recurrence of lung malignancies after SABR, which was unable to predict the endpoint accurately (c=0.38) [8]. A table comparing the methods, scope, and outcomes of six papers predicting local recurrence after SABR is available in *Appendix 2*. None of these datasets were available for use or had the appropriate CT scanner calibrations for an accurate, direct comparison of radiomic features.

In the adaptive LASSO, glcm-Inverse-variance was the feature with the highest correlation to local recurrence. Interestingly, and beyond the scope of this paper that did not account for histological findings, glcm-Inverse-variance has been reported to have a strong correlation with the Ki-67 proliferation index (p=0.00) [28], which suggests more aggressive tumours that would be more likely to recur. Flatness was identified as a significant predictor in this paper's random forest model; this feature was also identified as significant in the existing literature [2].

The interpretation of these results should be performed with caution due to the small sample size and short duration of follow-up. The other limitations of this paper include the manual delineation of the GTV and the use of non-harmonised images from multiple CT scanners. These limitations can be addressed in future research by using only images with GTVs delineated to the pixel value [1], trialling methods for CT image harmonisation to ensure that the CT images are standardised across different scanners [29] and imaging patients at different time points to accommodate temporal variabilities. The results could be improved upon by expanding the number of sites used to include a greater number and diversity of patients.

5. Conclusion

In conclusion, the integration of new technology into clinical practice is changing physicians' capacity to tailor patient care. AI models can be built that predict for event outcomes, and these have the potential to be used to modify early treatment at a time when it is most impactful. This paper demonstrated a potential for deep learning models to identify high-risk patients with improved sensitivity and statistical significance compared to traditional machine learning models. While this represents only an assessment of a small cohort of lung cancer patients at two institutions, the MLP used in this paper predicted 100% of the lung cancer local recurrence and identified patients at a higher risk with statistically significant differentiation from the low-risk group. The random forest machine learning method was unable to predict any local recurrence and could not significantly differentiate between high- and low-risk patients. The improved modelling predictions with deep learning analysis demonstrate a potential benefit to the use of this type of modelling for this indication and suggest a need for ongoing research in this area.

References

1. Li, Q. *et al*. Imaging features from pre-treatment CT scans are associated with clinical outcomes in non-small-cell lung cancer patients treated with stereotactic body radiotherapy. *Med Phys* **44**, 4341–4349 (2017).
2. Dissaux, G. *et al*. Pretreatment ^{18} F-FDG PET/CT Radiomics Predict Local Recurrence in Patients Treated with Stereotactic Body Radiotherapy for Early-Stage Non–Small Cell Lung Cancer: A Multicentric Study. *J Nucl Med* **61**, 814–820 (2020).
3. Morias, S. *et al*. Treatment-Related Adverse Effects in Lung Cancer Patients after Stereotactic Ablative Radiation Therapy. *J Oncol* **2018**, 6483626 (2018).
4. Lafata, K. J. *et al*. Association of pre-treatment radiomic features with lung cancer recurrence following stereotactic body radiation therapy. *Phys. Med. Biol.* **64**, 025007 (2019).
5. van Timmeren, J. E., Cester, D., Tanadini-Lang, S., Alkadhi, H. & Baessler, B. Radiomics in medical imaging—"how-to" guide and critical reflection. *Insights Imaging* **11**, 91 (2020).
6. Lambin, P. RQS - Radiomics.world. https://www.radiomics.world/rqs2 (2022).
7. Pyka, T. *et al*. Textural features in pre-treatment [F18]-FDG-PET/CT are correlated with risk of local recurrence and disease-specific survival in early stage NSCLC patients receiving primary stereotactic radiation therapy. *Radiation Oncology* **10**, 100 (2015).
8. Afshar, P. et al. DRTOP: deep learning-based radiomics for the time-to-event outcome prediction in lung cancer. *Sci Rep* **10**, 12366 (2020).
9. Kakino, R. *et al*. Application and limitation of radiomics approach to prognostic prediction for lung stereotactic body radiotherapy using breath-hold CT images with random survival forest: A multi-institutional study. *Medical Physics* **47**, 4634–4643 (2020).
10. Xia, P. & Murray, E. 3D treatment planning system—Pinnacle system. *Medical Dosimetry* **43**, (2018).
11. Maher,Ashley *et al*. *OnkoDICOM*. (2019).
12. Spearman, C. The proof and measurement of association between two things. *The American Journal of Psychology* **15**, 72–101 (1904).
13. Huynh, E. *et al*. CT-based radiomic analysis of stereotactic body radiation therapy patients with lung cancer. *Radiotherapy and Oncology* **120**, 258–266 (2016).
14. Lafata, K. J. *et al*. Association of pre-treatment radiomic features with lung cancer recurrence following stereotactic body radiation therapy. *Phys. Med. Biol.* **64**, 025007 (2019).
15. RStudio Team. *RStudio*. (RStudio,PBC, 2020).
16. Gentleman, R. & Ihaka, R. R. https://www.r-project.org/about.html (1997).
17. Frank, E., Hall, M. & Witten, I. *The WEKA Workbench*. (2016).
18. Lang, S., Bravo-Marquez, F., Beckham, C., Hall, M. & Frank, E. WekaDeeplearning4j: A deep learning package for Weka based on Deeplearning4j. *Knowledge-Based Systems* **178**, 48–50 (2019).
19. Lambin, P. *et al*. Radiomics: the bridge between medical imaging and personalized medicine. *Nat Rev Clin Oncol* **14**, 749–762 (2017).
20. Kaplan, E. L. & Meier, P. Nonparametric Estimation from Incomplete Observations. *Journal of the American Statistical Association* **53**, 457–481 (1958).
21. Gray, R. A Class of K-Sample Tests for Comparing the Cumulative Incidence of a Competing Risk. (1988) doi:10.1214/AOS/1176350951.
22. Harrell, F. E., Califf, R. M., Pryor, D. B., Lee, K. L. & Rosati, R. A. Evaluating the yield of medical tests. *JAMA* **247**, 2543–2546 (1982).
23. Mandrekar, J. N. Receiver Operating Characteristic Curve in Diagnostic Test

Assessment. *Journal of Thoracic Oncology* **5**, 1315–1316 (2010).

24. Hosmer, D. & Lemeshow, S. *Applied Logistic Regression*. (John Wiley & Sons, 2000).
25. Borkowska, E. *et al*. Artificial neural network in predicting bladder cancer recurrence. *Hereditary Cancer in Clinical Practice* **10**, A3 (2012).
26. Meng, Y. *et al*. Noncontrast Magnetic Resonance Radiomics and Multilayer Perceptron Network Classifier: An approach for Predicting Fibroblast Activation Protein Expression in Patients With Pancreatic Ductal Adenocarcinoma. *J Magn Reson Imaging* **54**, 1432–1443 (2021).
27. Yun, J. *et al*. Radiomic features and multilayer perceptron network classifier: a robust MRI classification strategy for distinguishing glioblastoma from primary central nervous system lymphoma. *Sci Rep* **9**, 5746 (2019).
28. Zhou, B., Xu, J., Tian, Y., Yuan, S. & Li, X. Correlation between radiomic features based on contrast-enhanced computed tomography images and Ki-67 proliferation index in lung cancer: A preliminary study. *Thorac Cancer* **9**, 1235–1240 (2018).
29. Mahon, R. N., Ghita, M., Hugo, G. D. & Weiss, E. ComBat harmonization for radiomic features in independent phantom and lung cancer patient computed tomography datasets. *Phys Med Biol* **65**, 015010 (2020).
30. Zwanenburg, A., Leger, S., Vallières, M. & Löck, S. Image biomarker standardisation initiative. *Radiology* **295**, 328–338 (2020).
31. Kassambara, A. *ggpubr: 'ggplot2' Based Publication Ready Plots*. (2020).
32. Wickham. *tidyverse package*. (2021).
33. Vaughan, D. & RStudio. *workflows: Modeling Workflows*. (2021).
34. Wright, M., Wager, S. & Probst, P. *Package 'ranger'*. (2021).
35. Therneau, T. M., Lumley, Atkinson & Crowson. *Package 'survival'*. (2022).
36. Kassambara, A., Kosinski, M., Biecek, P. & Fabian, S. *survminer: Drawing Survival Curves using 'ggplot2'*. (2021).
37. Wang [aut, W., cre, Chen, K. & Yan, J. *intsurv: Integrative Survival Modeling*. (2021).
38. Ishwaran, H. & Kogalur, U. B. *randomForestSRC: Fast Unified Random Forests for Survival, Regression, and Classification (RF-SRC)*. (2022).
39. Gerds, T. A. *Package 'pec'*. (2022).
40. Gray, B. Package *'cmprsk' - Subdistribution Analysis of Competing Risks*. (2021).
41. Chollet, Francois. *Deep Learning with Python*. (Manning Publications, 2021).
42. Friedman J, Hastie T, Tibshirani R (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software,* **33**(1), 1–22

6. Appendices

Appendix 1 – description of radiomic features

Quantifying intensity describes the density of the tumour by calculating the grey-level histogram and the probability density by calculating the grey-level occurrence matrix [14]. Fine texture features assess homogeneity at the limit of the image resolution by evaluating similarities in intensity between adjacent voxels; the grey-level co-occurrence matrix (GLCM) and neighbouring-grey-tone-difference. Coarse texture features measure the gross homogeneity of the tumour structure. This is done through assessing run lengths, which are the size of a group of adjacent voxels that measure similar greyscale intensities which are known as grey-level-size-zones and grey-level-dependence [30]. Morphological features quantify the shape, surface area and volume of the tumour [14].

Table 5 – 107 Pyradiomic Features used. All features concur with definitions from IBSI.

Intensity Features	Fine Texture Features	Coarse Texture Features	Morphological Features
● First-Order Features: 10-Percentile 90-Percentile Energy Entropy Interquartile Range Kurtosis Maximum Mean-Absolute-Deviation Mean Median Minimum Range Robust-Mean-Absolute-Deviation Root-Mean-Squared Skewness Total-Energy Uniformity Variance	● GLCM features: Autocorrelation Cluster-Prominence Cluster-Shade Cluster-Tendency Contrast Correlation Difference-Average Difference-Entropy Difference-Variance Id Idm Idmn Idn Imc1 Imc2 Inverse-Variance Joint-Average Joint-Energy Joint-Entropy MCC Maximum-Probability Sum-Average Sum-Entropy Sum-Squares ● NGTDM features: Busyness Coarseness Complexity Contrast Strength	● GLDM features: Dependence-Entropy Dependence-Nonuniformity Dependence-Nonuniformity-Normalized Dependence-Variance Gray-Level-Nonuniformity Gray-Level-Variance High-Gray-Level_emphasis Large-Dependence-Emphasis Large-Dependence-High-Gray-Level-Emphasis Large-Dependence-Low-Gray-Level-Emphasis Low-Gray-Level-Emphasis Small-Dependence-Emphasis Small-Dependence-High-Gray-Level-Emphasis Small-Dependence-Low-Gray-Level-Emphasis ● GLRLM features: Gray-Level-Nonuniformity Gray-Level-Nonuniformity-Normalized Gray-Level-Variance High-Gray-Level-Run-Emphasis Long-Run-Emphasis Long-Run-High-Gray-Level-Emphasis Long-Run-Low-Gray-Level-Emphasis Low-Gray-Level-Run-Emphasis Run-Entropy Run-Length-Nonuniformity Run-Length-Nonuniformity-Normalized Run-Percentage Run-Variance Short-Run-Emphasis Short-Run-High-Gray-Level-Emphasis Short-Run-Low-Gray-Level-Emphasis ● GLSZM features: Gray-Level-Nonuniformity Gray-Level-Nonuniformity-Normalized Gray-Level-Variance High-Gray-Level-Zone-Emphasis Large-Area-Emphasis Large-Area-High-Gray-Level-Emphasis Large-Area-Low-Gray-Level-Emphasis Low-Gray-Level-Zone-Emphasis Size-Zone-Nonuniformity Size-Zone-Nonuniformity-Normalized Small-Area-Emphasis Small-Area-High-Gray-Level-Emphasis Small-Area-Low-Gray-Level-Emphasis Zone-Entropy Zone-Percentage Zone-Variance	● Shape features: Elongation Flatness Least-Axis-Length Major-Axis-Length Maximum-2D-Diameter-Column Maximum-2D-Diameter-Row Maximum-2D-Diameter-Slice Maximum-3D-Diameter Mesh-VolumeMinor-Axis-Length Sphericity Surface-Area Surface-Volume-Ratio Voxel-Volume

Appendix 2 – summary of research using radiomic features to predict local recurrence of lung cancer

Table 6 – Details of studies predicting local recurrence of lung cancer following SABR treatment using CT radiomic features.

Author	Clinical features	Radiomic features	Participants	Delineation method	Analysis method	Results
Afshar et al (2020)	4	18 hand-crafted PET and CT features based on Oikonomou et al (2018)	132 participants with early stage lung cancer (N0M0) treated with SABR over 4.5 years. Treatment not expanded upon in the paper.	Manual by thoracic radiologist and in-house software	Radiomic features analysed with Cox PHM and Kaplan-Meier analysis, compared to parallel CNN model	No predictors of LR in radiomics models, LR prediction in CNN model not significant (c-index 0.375).
Dissaux et al (2020)	7	92 PET, 92 CT features	87 participants with NSCLC stage I–II who underwent SABR at 4 institutions over 5 years. Treatment was 48–60 Gy in 3–8 fractions.	Fuzzy locally adaptive Bayesian method in PET, manually in CT	Images harmonised with ComBat. Spearman rank correlation, Cox regression models, Kaplan-Meier curves with log-rank test.	Univariate prediction (AUC > 0.7): CT flatness, CT shade, elongation. PET Information Correlation 2 (IC2) and PET texture strength. Multivariate prediction: IC2 and CT flatness model had 92% accuracy. IC2 and texture strength 91% accuracy.
Kakino et al (2020)	9	944 CT features via PyRadiomics	573 participants with early- stage lung cancer who underwent SABR across 11 institutions and 10 years. Treatment exceeded 100 Gy of the BED.	Manual	Adaptive LASSO, Random Survival Forest, Gray's test for statistical significance between high/low risk groups, bootstrapping.	Combined clinical/radiomic prediction model not statistically significant for predicting LR (c-index 0.61). 11 radiomics features prognostic for LR in feature selection: firstorder_Range, gldm_SmallDependenceLowGreyLevelEmphasis, gldm_DependenceVariance, glcm_Idmn (log sigma 0.5mm and 2.5mm), Elongation, gldm_DependenceEntropy, glcm_ClusterShade (wavelet LH and HL), gldm_LargeDependenceHighGreyLevelEmphasis, glcm_MCC
Lafata et al (2019)	0	43 CT features via PyRadiomics	70 participants, stage 1 who underwent SABR treated at Duke University between 2007 and 2014. Treatment was a mean dose of 51 Gy, hypofractionation scheme.	Manual by experience physician and physicist	Welch's t-test, singular value decomposition, LASSO.	Univariate predictors: GLCM_Homogeneity2, Long-Run-High-Grey-Level-Emphasis
Li et al (2017)	24	219 CT features via Definens	92 participants with stage I or IIA who underwent SABR over 4.5 years. Standard treatment was 50 Gy in 5 fractions.	Automatic with Definiens Developer	ICC, Cox PHM, Harrell's c-index, 10-fold cross validation	Univariate predictors of loco-regional recurrence: long axis diameter, short axis*longest diameter, short axis, volume in cm, av-dist-COG-to-border, min-dist-COG-to-border, volume-pxl, AvgGLN
Pyka et al. (2015)	1	Hand-crafted CT and PET features	45 sequential participants with T1 or T2 (NOMO) NSCLC who underwent SABR. Treatment was 24–45Gy in 3–5 fractions.	Automatic with InterView Fusion	ROC, Kaplan-Meier curves, log-rank test, Cox regression	Univariate CT: MTV, tumour size Univariate PET: NGTDM and GLCM entropy, correlation.

Appendix 3

Table 7 – R packages used for data analysis

R packages	References
`> ggpubr`	[31]
`> glmnet`	[42]
`> randomForest`	
`> tidymodels`	
`> tidyverse`	[32]
`> workflows`	[33]
`> ranger`	[34]
`> survival`	[35]
`> survminer`	[36]
`> intsurv`	[37]
`> randomForestSRC`	[38]
`> pec`	[39]
`> cmprsk`	[40]

Appendix 4 – Technical method

First, the radiomic features need to be normalised to aggregate the scales by converting all values to a z-score. This can be done with the following code:

```
dataframe <- scale(dataframe)
```

In a Spearman’s rank correlation test, varying levels of correlation significance are reported, usually above 0.7–0.9. Spearman’s rank correlation test calculates the correlation between the ranks of x and y in the following equation, where rho is the Spearman correlation efficient, x’=rank(x) and y’=rank(y):

Equation 1 – Spearman’s rho equation

$$rho = \frac{= \sum(x' - mx')(y'i - my')}{\sqrt{\sum(x' - mx')2\sum(y' - my')^2}}$$

Spearman’s rank correlation for ≥ 0.9 can be performed with the following code for a numeric data frame:

```
> library(ggpubr)[31]
> spearmandf <- cor(x = dataframe, method = c("spearman")
> correlationdf <- subset(spearmandf[,]> = 0.9)
```

The adaptive LASSO is an iteration of LASSO with oracle properties that work by weighting the following equation to counteract known biases in regular LASSO:

Equation 2 – adaptive LASSO

$$\hat{\boldsymbol{\beta}}^{*(n)} = \arg\min_{\boldsymbol{\beta}} \left\| \mathbf{y} - \sum_{j=1}^{p} \mathbf{x}_j \beta_j \right\|^2 + \lambda_n \sum_{j=1}^{p} \hat{w}_j |\beta_j|.$$

where β = the constant coefficient, X = the covariate matrix, w = a known weight vector.

An adaptive LASSO can be implemented in R by first performing a regular LASSO and then scaling the X matrix [43]. The model can be assessed with k-fold cross-validation to find the optimal lambda that minimises the test mean squared error (MSE).

```
> library(glmnet)
> x <- data.matrix(dataframe)
> y <- dataframe$responsevariable
> n <- nrow(x)

># standardise data
> ymean <- mean(y)
> y <- y - mean(y)
> xmean <- colMeans(x)
> xnorm <- sqrt(n-1)*apply(x,2,sd)
> x <- scale(x, center = xmean, scale = xnorm)

> #fit ordinary least squares
> lm.fit <-lm(y~x)
> beta.init <-coef(lm,fit)[-1] #exclude 0 intercept

> #calculate weights
> w<-abs(beta.init)
> x2 <-scale(x, center = FALSE, scale = 1/w)

># fit adaptive lasso
> adaplasso <- cv.glmnet(x2, y, family = "gaussian", alpha = 1, standardize =
FALSE, nfolds=10)
> best_lambda <- adaplasso$lambda.min # minimizes test MSE
> best_model <- predict(adaplasso, x2, type="coefficients", s="best_lambda")[-1]

> # calculate estimates
> best_model <- best_model * w / xnorm # back to original scale
> best_model <- matrix(best_model, nrow = 1)
> xmean <- matrix(xmean, nrow = 10)
> b0 <- apply(best_model, 1, function(a) ymean - a %*% xmean) # intercept
> coef <- cbind(b0, best_model)
> coef
```

6.1 Random Forest Model

A random forest machine learning model uses hundreds of decision trees to average the results and predict the most likely outcome.

The random forest model consists of a number of hyperparameters that can be optimised to improve the performance of the model. One of the ways to do this is through a grid search method, wherein the operator defines a range of values and runs tests on how these improve the model, through comparing the out-of-bag (OOB) error or a different metric of model fit such as the AUC.

The machine learning model used in this paper used a 70/30 training/testing data split. This can be implemented in R, as is shown in the following example that also includes a grid search to optimise the m_{try} , sample size and minimum node size hyperparameters:

```
> #partitioning data into training/testing set for random forest
> set.seed(9999)
> index <- sample(1:nrow(dataframe),0.7*nrow(dataframe))

> datatrain = dataframe[index,] #create the training dataset
> datatest = dataframe[-index,] #create the testing dataset

> library(randomForest)
> bestmtry <- tuneRF(datatrain, datatrain$predictor, stepFactor= 1.5, improve=1e-5, ntree=500)
```

The following is a grid search conducted in R to optimise random forest hyperparameters:

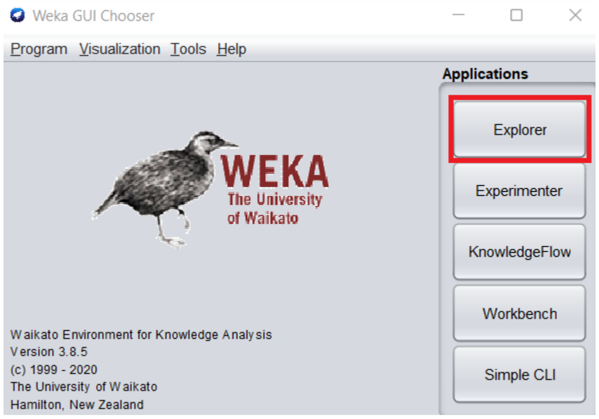
```
> library(tidymodels)
> library(tidyverse)
> library(workflows)
> library(ranger)

> #2 packages are used for different steps in building this model. They use
different shorthand for the hyperparameters. min_n = min.node.size, trees =
num.trees, mtry = mtry.

> #create cross-validation object from training data
> train_cv <- vfold_cv(datatrain)

> #define recipe and include preprocessing
> rad_recipe <- recipe(outcome~predictors, data = dataframe)%>%
+               step_normalize(all_numeric())%>%
+               step_impute_knn(all_predictors()

> #apply the recipe to the training data and extract pre-processed dataset
> datatrainpreproc <- rad_recipe%>%
+             prep(datatrain)%>%
+             juice()

> #specify the model
>rfmodel <- rand_forest(mode = "classification",
+               mtry = tune(),
+               trees = tune(),
+               min_n = tune(),%>%
+               set_engine("ranger", importance = "impurity") #set importance to
visualise feature importance

> #build workflow
> rf_workflow <- workflow()%>%
+            add_recipe(rad_recipe)%>%
+            add_model(rfmodel)

#test values in grid and analyse outputs
> rfgrid <- expand.grid(mtry = c(3,4,5,6), trees = c(500, 1000, 1500, 2000),
min_n = c(1:10, by=1)
> rf_tune <- rf_workflow%>%
+            tune_grid(resamples = traincv)
+            grid = rfgrid
+            metrics = metric_set(accuracy,roc_auc)
>rf_tune%>%
              collect_metrics()

> #requires analysis of standard error for error minimisation
> best_hyperparameters_search <- select_best(rf_tune, metric = "roc_auc",
maximize = TRUE)
> best_hyperparameters_search

> #run random forest with optimal parameters
> rf <- ranger(formula = outcome~predictor, data = datatrain, num.trees = a,
mtry = b, min.node.size = d,  sample.fraction = 1, splitrule = "extratrees",
importance = "impurity")
```

```
#repeat model 100 times to improve estimate of error
> OOB_RMSE <- vector(mode = "numeric", length = 100)
> for(i in seq_along(OOB_RMSE)){
+      optimal_rf < -ranger(
+          formula  = outcome~predictor,
+          data = datatrain,
+          num.trees = a,
+          mtry = b,
+          min.node.size = d,
+          sample.fraction = 1,
+          splitrule = "extratrees",
+          importance = "impurity"
+      )
+   }
> # printing optimal_rf will give best OOB error

#run random forest model against test group
> testpredictions<-predict(optimal_rf, datatest)
> # print testpredictions for binary classification outcomes

#assess feature importance in model
>rf$importance
```

6.2 Multilayer Perceptron

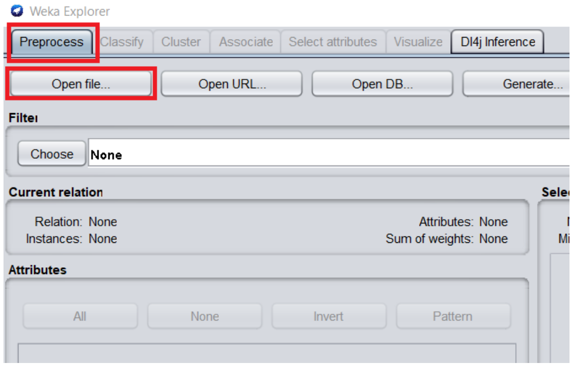
The following steps to create an MLP can be performed in Weka's GUI. The Dl4MlpClassifier package [28] is an extension that can be downloaded. The trained MLP described in this paper is available on the author's GitHub (github.com/allijan45), and the reader is encouraged to try it with their own radiomic data.

1. Open the explorer application.



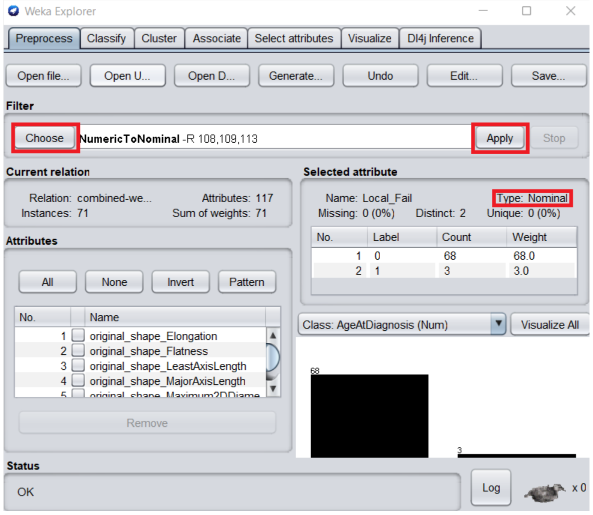
Waikato Environment for Knowledge Analysis
Version 3.8.5
(c) 1999 - 2020
The University of Waikato
Hamilton, New Zealand

2. In the "Preprocess" tab: click "Open file."



Filter: None
Relation: None
Instances: None
Attributes: None
Sum of weights: None

3. Ensure all of the attributes are of the correct data type (e.g. numeric and nominal). These can be changed with the filter option.



Filter

NumericToNominal -R 108,109,113

Current relation

Relation: combined-we... Attributes: 117
Instances: 71 Sum of weights: 71

Selected attribute

Name: Local_Fail Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	0	68	68.0
2	1	3	3.0

Attributes

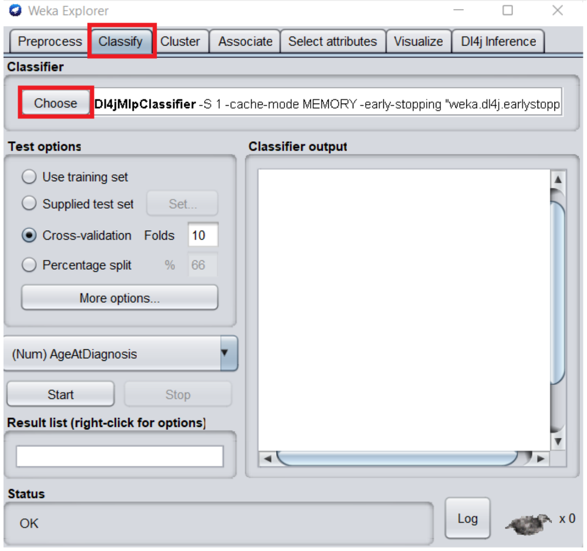
No.	Name
1	original_shape_Elongation
2	original_shape_Flatness
3	original_shape_LeastAxisLength
4	original_shape_MajorAxisLength
5	original_shape_Maximum2DDiame

Class: AgeAtDiagnosis (Num)

Status

OK

4. In the "Classify" tab: choose the desired classifier. The Dl4jMlp Classifier is one option for building an MLP.



Classifier

Dl4jMlpClassifier -S 1 -cache-mode MEMORY -early-stopping "weka.dl4j.earlystopp

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Num) AgeAtDiagnosis

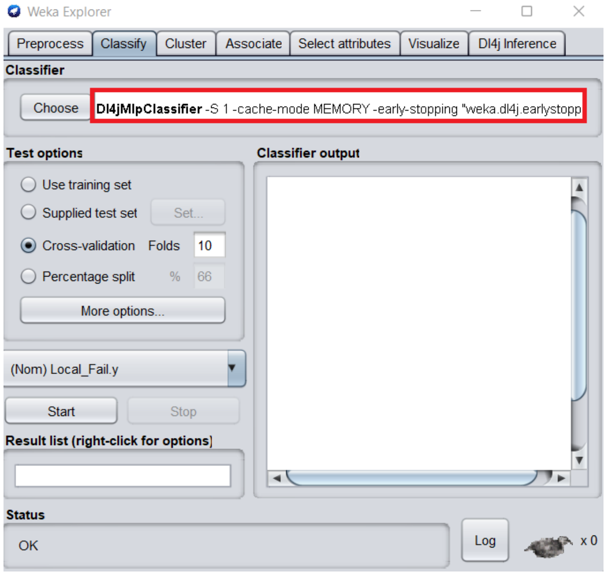
Classifier output

Result list (right-click for options)

Status

OK

5. Changes to the model can be made by clicking the classifier name, e.g. changing filters and adding a cost-sensitive classifier.



Classifier

Dl4jMlpClassifier -S 1 -cache-mode MEMORY -early-stopping "weka.dl4j.earlystopp

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) Local_Fail.y

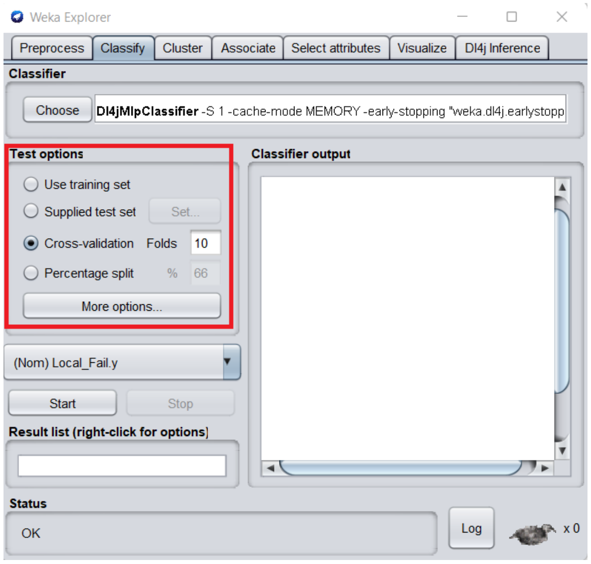
Classifier output

Result list (right-click for options)

Status

OK

6. Decide between training/testing the classifier or using cross-validation.



Classifier

Dl4jMlpClassifier -S 1 -cache-mode MEMORY -early-stopping "weka.dl4j.earlystopp

Test options

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Nom) Local_Fail.y

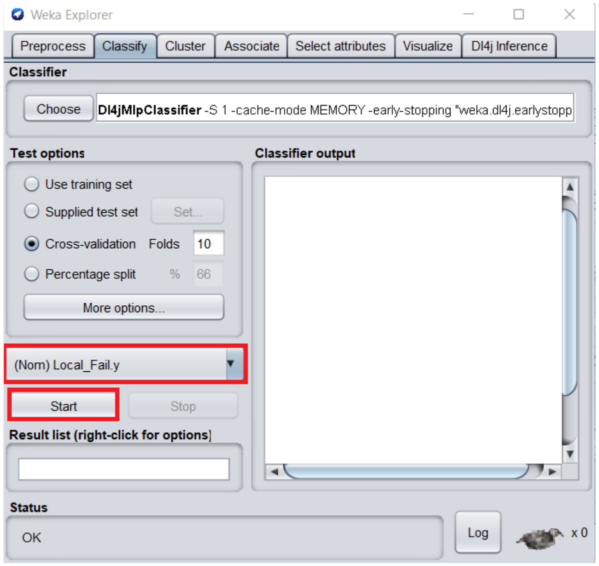
Classifier output

Result list (right-click for options)

Status

OK

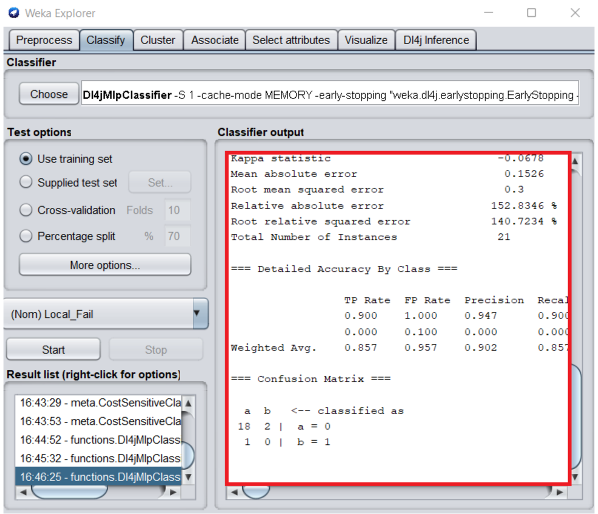
7. Select for the outcome variable and then click start.



Dl4jMlpClassifier -S 1 -cache-mode MEMORY -early-stopping "weka.dl4j.earlystopp

(Nom) Local_Fail.y

8. Assess output. Repeat steps 4–7 as needed to improve the model.



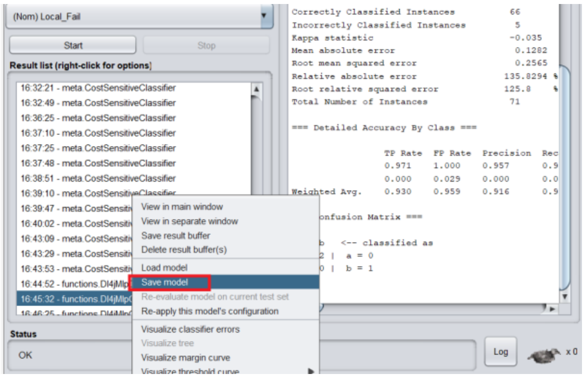
Dl4jMlpClassifier -S 1 -cache-mode MEMORY -early-stopping "weka.dl4j.earlystopping.EarlyStopping -

(Nom) Local_Fail

```
16:43:29 - meta.CostSensitiveCla
16:43:53 - meta.CostSensitiveCla
16:44:52 - functions.Dl4jMlpClass
16:45:32 - functions.Dl4jMlpClass
16:46:25 - functions.Dl4jMlpClass
```

```
Kappa statistic                         -0.0678
Mean absolute error                      0.1526
Root mean squared error                  0.3
Relative absolute error                152.8346 %
Root relative squared error            140.7234 %
Total Number of Instances               21

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recal
                 0.900    1.000    0.947      0.900
                 0.000    0.100    0.000      0.000
Weighted Avg.    0.857    0.957    0.902      0.857

=== Confusion Matrix ===

  a  b   <-- classified as
 18  2 |  a = 0
  1  0 |  b = 1
```

9. Save model by right-clicking on it in the results list

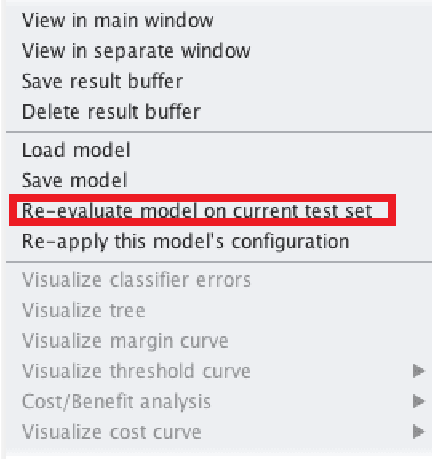


(Nom) Local_Fail

```
16:32:21 - meta.CostSensitiveClassifier
16:32:49 - meta.CostSensitiveClassifier
16:36:25 - meta.CostSensitiveClassifier
16:37:10 - meta.CostSensitiveClassifier
16:37:25 - meta.CostSensitiveClassifier
16:37:48 - meta.CostSensitiveClassifier
16:38:51 - meta.CostSensitiveClassifier
16:39:10 - meta.CostSensitiveClassifier
16:39:47 - meta.CostSensiti
16:40:02 - meta.CostSensiti
16:43:09 - meta.CostSensiti
16:43:29 - meta.CostSensiti
16:43:53 - meta.CostSensiti
16:44:52 - functions.Dl4jMlp
16:45:32 - functions.Dl4jMlp
16:46:25 - functions.Dl4jMlp
```

```
Correctly Classified Instances          66
Incorrectly Classified Instances         5
Kappa statistic                         -0.035
Mean absolute error                      0.1282
Root mean squared error                  0.2565
Relative absolute error                135.8294 %
Root relative squared error            125.8    %
Total Number of Instances               71

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Rec
                 0.971    1.000    0.957      0.9
                 0.000    0.029    0.000      0.0
Weighted Avg.    0.930    0.959    0.916      0.9

 onfusion Matrix ===

 b   <-- classified as
 2 |  a = 0
 0 |  b = 1
```

10. Upload new data for re-evaluation; right-click on the desired model, re-run model, on new data without retraining



6.3 Outcome Prediction

Although many classifiers are able to accurately identify markers that predict within their datasets, publishing these in the absence of validation is often very optimistic. This is usually due to expected limitations that often include a small sample size with limited participant diversity and the likelihood of some methodological errors in image collection or processing. To ensure the quality reporting of radiomic analyses, risk groups should be split around the median or should be reported with continuous risk variables. This then requires improved model performance to accurately differentiate between high- and low-risk groups. This was performed in R as follows:

```
> library(survival)
> library(survminer)
> library (intsurv)
> library(randomForest SRC)

> #Determine high or low risk group: requires data frame with time and event
```

```
variables
> riskgroup <- rfsrc(Surv(time, event)~., dataframe, ntree = a, mtry = b,
importance = TRUE)
> datarisk <- cbind(dataframe, riskgroup$predicted)

> #comparison of high/low risk groups as Kaplan-Meier plot
> datarisk1 <-mutate(datarisk, predicted  =ifelse((predicted >=
median(datarisk$predicted), "1", "0"))
> # if above average chance of event in random survival forest then predicted
column shows high risk (1), otherwise classified as low risk (0)

> datarisk1 <- as.numeric(datarisk1)
> fitrisk <- survfit(Surv(time, event)~predicted, data = datarisk1)
> risk_diff <- survdiff(Surv(time, event)~predicted, data = datarisk1)
> ggsurvplot(fitrisk, data = datarisk1, conf.int = TRUE, pval = TRUE,
legend.labs = c("Low Risk", "High Risk"), legend.title = "Risk Group", xlab =
"Years", ylab = "Probability of Event")
> #output is Kaplan-Meier plot as pictured in Figure 2
```

6.4 Model Evaluation

The MLP was performed with 10-fold cross validation. The significance of separation between the high- and low-risk groups was calculated in R with bootstrapping (10,000 iterations) to determine the difference between groups and the 95% confidence interval.

```
> library(pec)
> library(cmprsk)

> # run Gray's test for competing risk based on cumulative index
>dataframe$event <- as.factor(dataframe$event)
> cuminc(ftime = dataframe$time, fstatus = dataframe$event, group =
dataframe$risk

> #calculate c-index of model
> fit1 <- coxph(Surv(time, event)~predicted, data = datarisk1)
> cIndex(time = fit1$time, event = fit1$event,risk_score =
fit1$linear.predictors)

> #assess confidence intervals with bootstrapping
> #evaluate outcome separation
> #separation of outcomes should show a difference in means between high and low
risk groups
> meanhighrisk <- mean(dataframe$time[dataframe$risk == "High"])
> meanlowrisk <- mean(dataframe$time[dataframe$risk == "Low"])
> meanlowrisk - meanhighrisk #to show superiority the low risk group should have
a higher mean than the high risk group therefore result should be a positive
number

> n.highrisk = x #number of individuals classified as high risk
> n.lowrisk = y #number of individuals classified as low risk
> #if you split around the median n.highrisk = n.lowrisk
> B = 10000 #number of bootstrap resamples

> #create a matrix which samples many results
> set.seed(13573)
> boot.high <- (matrix(sample(dataframe$time[dataframe$risk == "High"], size =
B*n.highrisk, replace=TRUE), nrow = n.highrisk, ncol = B)
> boot.low <- (matrix(sample(dataframe$time[dataframe$ris k== "Low"], size =
B*n.lowrisk, replace=TRUE), nrow = n.lowrisk, ncol = B)

> #calculate the difference in bootstrapped means
> bootdiff<-colMeans(boot.high) - colMeans(boot.low)

> #calculate 95% confidence intervals
> quantile(bootdiff, prob = 0.025)
>quantile(bootdiff, prob = 0.975)
># if the 95% confidence intervals do not cross 0 there is statistically
significant difference between the means
```