Machine and Deep Learning Methods for Predicting Immune Checkpoint Blockade Response

Danliang Ho HO.DANLIANG@U.NUS.EDU
NUS Graduate School, Integrative Sciences and Engineering Programme, National University of Singapore

Mehul Motani MOTANI@NUS.EDU.SG
College of Design and Engineering, Department of Electrical and Computer Engineering, Integrative Sciences and Engineering Programme, The N.1 Institute for Health, The Institute for Digital Medicine (WisDM), The Institute of Data Science, National University of Singapore

Abstract

Immune checkpoint blockade (ICB) therapy has improved treatment options in various cancer malignancies and holds promise for increasing the overall survival of treated patients. However, only a small proportion of patients benefit from ICB treatment. Furthermore, ICB therapy has been known to induce adverse autoimmunity reactions in certain patients. These two reasons motivate the clinical need to identify factors that predict a patient's response to ICB treatment. In our study, we developed several machine and deep learning-based models to predict response to ICB treatment, using a real-world tabular dataset across sixteen cancer types. We showed that our best model `CB16`, which is based on gradient boosting, outperforms all-known published results for this task, with sensitivity and specificity scores of 80.6% and 78.8% respectively. Our model also offers insights to clinical interpretability through the use of the SHAP explanation framework, which are consistent with known important predictors. Next, in order to see if deep learning can improve performance, we propose a methodology for the design of deep neural networks that addresses the lack of spatial and temporal structure in tabular data. Our approach is based on a combination of learning ordered representations and ensembling techniques. We show that, for the ICB prediction problem, current SOTA deep-learning architectures such as TabNet and TabTransformer do not perform well while our method achieves good performance. Our method achieves an F1 score 12.4 percentage points beyond that of TabTransformer, and sensitivity and specificity scores of 77.3% and 62.2% respectively. Through our work, we hope to improve the task of predicting ICB response, and contribute towards the creation of high-performance and interpretable AI models for real-world tabular data.

Keywords: immunotherapy, deep learning, tabular data

1. Introduction

Immunotherapy is a promising treatment option for cancer due to its ability to selectively target tumour cells by activating components within the patient's own immune system. Within the list of immunotherapy agents, immune checkpoint blockade (ICB) drugs remove constraints on the reactivity of the immune system to allow more effective targeting of tumour cells and have been shown in various studies to exhibit substantial clinical benefit across multiple cancer

types (Ye et al., 2022). In particular, ICB treatment has been shown in various clinical trials to improve overall survival across tumor types as compared to other forms of treatment such as chemotherapy or molecular therapy (Pons-Tostivint et al., 2019). Furthermore, it has the potential to achieve durable response in patients with recurrent and metastatic cancers, which are typically considered incurable, where other treatments fail (Murciano-Goroff et al., 2020).

Unfortunately, studies have shown that the objective response rate is only around 20% (Kim et al., 2020; Shen et al., 2020). Given its low cost-effectiveness as well as the potential of experiencing side effects from auto-immunity (Bajwa et al., 2019; Darvin et al., 2018), there exists a strong clinical need to identify patients who will respond to this treatment option. With the popularity of machine learning (ML) and deep learning (DL) in recent years, it is unsurprising that these techniques have been applied to address the problem of prognosticating response to ICB (Lu et al., 2020; Chowell et al., 2021; Abuhelwa et al., 2021). As far as we know, the best performing model, known as `RF16` and based on the random forest algorithm, achieved sensitivity and specificity scores of 76.7% and 74.2% respectively (Chowell et al., 2021).

In our work, we developed several ML and DL models to predict response to ICB. Our main contributions are three-fold:

1. We propose a model based on gradient boosting, which we refer to as `CB16`, that outperforms `RF16`, in a head-to-head comparison based on the same dataset and the same train-test split. Our model uses 10-fold fewer trees yet maintains a performance improvement over `RF16` in all calculated metrics (accuracy, sensitivity, precision, specificity, F1-score) by around 3 to 5 percentage-points.
2. Our approach offers enhanced insights to clinical interpretability for the ICB response prediction problem. We not only show overall feature importance scores, like the authors of `RF16`, we also highlight the direction of association, and demonstrate how features contribute to the prediction for individual patients, using Shapley additive explanations (SHAP) for post-hoc explanations.
3. Lastly, we describe our strategies to tackle the challenge of modelling heterogeneous tabular data using DL methods (see Section 2.2). Our methodology is based on learning ordered representations from tabular data; we also explore current ideas from representation learning and ensembling. Practical insights from our work could be a valuable resource for researchers interested in applying DL models to tabular data without spatial and temporal structure.

2. Related Work

2.1. Predictive models for ICB

The task of predicting response efficacy from ICB is not new and several biomarkers have been developed for ICB patient stratification. One example is tumour mutation burden (TMB) - various studies have established a relationship between high TMB as well as improved responsiveness to ICB treatment (Zheng, 2022; Goodman et al., 2017). However the predictive power of individual markers remains limited; this is usually attributed to the complexity of factors involved in ICB response that cannot be modelled by any single predictor (Jiang et al., 2021).

Machine learning is a viable solution due to its ability to combine heterogeneous features non-linearly. For example, Benzekry et al. (2021) and Chen et al. (2021) developed predictive models for ICB response in metastatic non-small cell lung cancer and triple negative breast cancer respectively, with reasonable performance (0.58 sensitivity and 0.78 specificity for the former, 0.76 AUROC for the latter). Recent years have

seen a shift towards response prediction for multiple cancer-types concurrently. For example, Lapuente-Santana et al. (2021) downloaded tumour RNA-seq data across 18 solid cancers from The Cancer Genome Atlas database, to derive tumour molecular environment signatures and create predictive models. The downside of this approach is its requirement for RNA and proteomics sequencing data, which might not be readily available in a clinical context.

To our knowledge, one of the most promising studies is that published by Chowell et al. (2021), where the authors trained a random forest model named `RF16` on the MSK-IMPACT dataset consisting of 1479 patients across 16 different cancers. Their model utilised 16 predictive features extracted from clinical and genomic data including clinically-validated biomarkers such as TMB. However, the model is likely not optimal for the problem - the complexity of `RF16` (1000 trees and a maximum depth of 8 for a relatively small dataset) suggests that there could be inefficiency in terms of computational overhead from the excess trees. As part of our work, we seek to create more optimised models with lower run-time complexity, while maintaining or even exceeding the performance of `RF16`.

2.2. Deep learning for tabular data

Deep learning methods have demonstrated remarkable success with data types such as images (He et al., 2016; Nandhini Abirami et al., 2021; Shorten and Khoshgoftaar, 2019), text (Brown et al., 2020; Devlin et al., 2019; Khan et al., 2021) and audio (Oord et al., 2016; Purwins et al., 2019; Zhao et al., 2019); however their application to tabular data remains challenging. Various comparisons based on real-world competitions (Kaggle, 2019) and research studies (Borisov et al., 2022) have shown that ensemble decision trees (DTs) such as XGBoost and Random Forest still outperform deep neural networks (DNNs) on real-world tabular datasets. Reasons include: 1) Unlike image or text data where spatial/semantic relationships can be represented and modelled inherently, tabular data lack structure that can be exploited by DNNs. Tabular DNNs have to deal with a lack of inductive bias as the order of tabular features do not encode position information. 2) The heterogeneity of tabular data, which includes dense numerical features and sparse categorical features, creates difficulties in modelling implicitly. This is unlike DTs where both sparse and dense features are modelled in a similar fashion by virtue of splitting attributes based on a threshold (Borisov et al., 2022).

Despite its difficulty, deep learning on tabular datasets is nevertheless a problem worthy of exploration. Besides a potential performance boost beyond that achievable by classical techniques, deep learning on tabular data also allows us to exploit current DNN techniques such as representation learning and generative modelling. It also opens the door for end-to-end learning and integration with multiple data modalities including image and text (Arik and Pfister, 2020; Bahri et al., 2022). To this end, various studies have published DNN architectures built for tabular data. `TabNet`, perhaps one of the most well-known methods, utilises sequential attention to perform instance-wise feature selection during each decision step (Arik and Pfister, 2020). `NODE` is another DNN architecture based on the idea behind ensembled DTs, but is made fully differentiable by virtue of the entmax transformation and soft splits (Popov et al., 2019). `TabTransformer` is based on self-attention to map categorical features into continuous embeddings, which are subsequently fed into an MLP layer for processing and classification (Huang et al., 2020). Nonetheless, systemic comparisons

have demonstrated that non of these models consistently outperform classical ML methods on real-world tabular datasets of different sizes (Borisov et al., 2022). Our work represents a significant departure from this existing line of research, in that we directly address the problem of a lack of inductive bias, by learning structure from the unordered tabular inputs. We also explore current ideas from contrastive learning as well as ensembling approaches.

3. Dataset description

Our dataset was acquired from information released by Chowell et al. (2021) and consists of 1,479 patients across 16 different cancer types, from Memorial Sloan Kettering Cancer Center. All patient data were de-identified by the study authors. Out of this cohort, 409 responded to immunotherapy and 1070 did not, based on criteria outlined in Response Evaluation Criteria in Solid Tumours (RECIST) (Eisenhauer et al., 2009). The dataset contains 16 different feature types that are a mixture of genomic variables (e.g., tumour molecular burden and fraction of copy number alteration), clinical variables (e.g., cancer type and immunotherapy drug agent) and demographic variables (e.g., sex, age and BMI). We include a description of patient characteristics in Appendix A. To process the data into a suitable format for model input, we one-hot encoded all categorical variables and scaled numerical features via min-max scaling. We performed an additional step of normalisation via logarithmic transformation prior to scaling, for numerical features that were highly skewed.

4. Predicting ICB response via ML

4.1. Model development & evaluation

We utilised the same train-test split as that specified by Chowell et al. (2021) for model training and evaluation. We investigate 5 models, namely: Logistic regression (`LR`), Random Forest (`RF`), Support Vector Machine (`SVM`), XGBoost (`XGB`), CatBoost (`CB16`). Models were trained using the Python packages sklearn (Pedregosa et al., 2011), xgboost (Chen and Guestrin, 2016) and catboost (Dorogush et al., 2018). We report parameters and training details under Appendix B.

Our choice of performance metrics follows that of Chowell *et al.*: Accuracy, Sensitivity (or recall), Positive Predictive Value (PPV, or precision), F1 score and Specificity. We further include AUPRC for comparisons with other models in the ML space. We note that F1 and AUPRC are suitable metrics for imbalanced dataset such as ours. We benchmark our model performance on two classifiers developed by Chowell et al. (2021) on pan-cancer patients: `RF16` and a logistic regression (`LR`) baseline. We optimise firstly for high F1-score to account for the imbalanced data, and secondly for sensitivity, as in the immunotherapy setting, there is greater clinical utility in a model that identifies more patients that will benefit from the treatment (reduce false negatives), compared to excluding patients who would otherwise not have benefited (reduce false positives).

4.2. Model performance

Table 1 shows the performance of our models, against benchmarks from Chowell et al. (2021). In general all our models outperform the benchmark classifiers in most calculated metrics. `CB16` significantly outperforms Chowell *et al*'s best model `RF16`, topping the scoreboard in terms of F1 score and sensitivity, and is among the top 3 positions for the other metrics. Our `RF` also demonstrates strong performance, not only significantly surpassing `RRF16`, but also achieving first or second best performance for all met-

Table 1: Comparing performance of our models versus Chowell et al. (2021)'s. Scores are reported as average (95% CI) of 100 runs. NR: Not Reported. Best and second-best scores are bolded and underlined respectively.

Model	Accuracy	Sensitivity	PPV	F1	Specificity	AUPRC
Benchmarks from Chowell et al. (2021)						
RF16	74.9 (NR)	76.7 (NR)	56.6 (NR)	65.1 (NR)	74.2 (NR)	NR (NR)
LR	71.9 (NR)	56.7 (NR)	53.7 (NR)	55.2 (NR)	78.5 (NR)	NR (NR)
Our models						
LR	73.4 (71.5, 75.3)	68.0 (65.3, 70.7)	55.4 (52.9, 58.0)	60.9 (58.3, 63.4)	75.7 (73.9, 77.5)	65.8 (63.1, 68.4)
SVM	77.6 (76.1, 79.1)	71.3 (69.2, 73.5)	61.5 (59.2, 63.7)	65.9 (63.8, 68.0)	**80.3** (78.9, 81.7)	66.4 (64.3, 68.6)
RF	**79.5** (78.3, 80.8)	78.1 (76.7, 79.6)	**63.5** (61.7, 65.3)	<u>69.9</u> (68.3, 71.4)	<u>80.1</u> (78.7, 81.5)	**70.9** (68.9, 72.9)
XGB	78.5 (77.0, 80.0)	<u>78.9</u> (76.9, 80.9)	61.7 (59.7, 63.8)	69.1 (67.1, 71.0)	78.4 (76.8, 79.9)	<u>70.7</u> (68.7, 72.8)
CB16	<u>79.3</u> (78.0, 80.6)	**80.6** (78.8, 82.4)	<u>62.5</u> (60.8, 64.3)	**70.3** (68.6, 71.9)	78.8 (77.5, 80.1)	69.8 (67.8, 71.9)

rics except sensitivity. We prefer CB16 to RF as the precision-recall curves show that there is a slight advantage towards precision at high recall thresholds, and this is clinically relevant for our setting (see Appendix C). The rest of our models also demonstrate strong performance compared to the benchmarks, with XGB closely following the performance of CB16 and RF.

4.3. Model characteristics

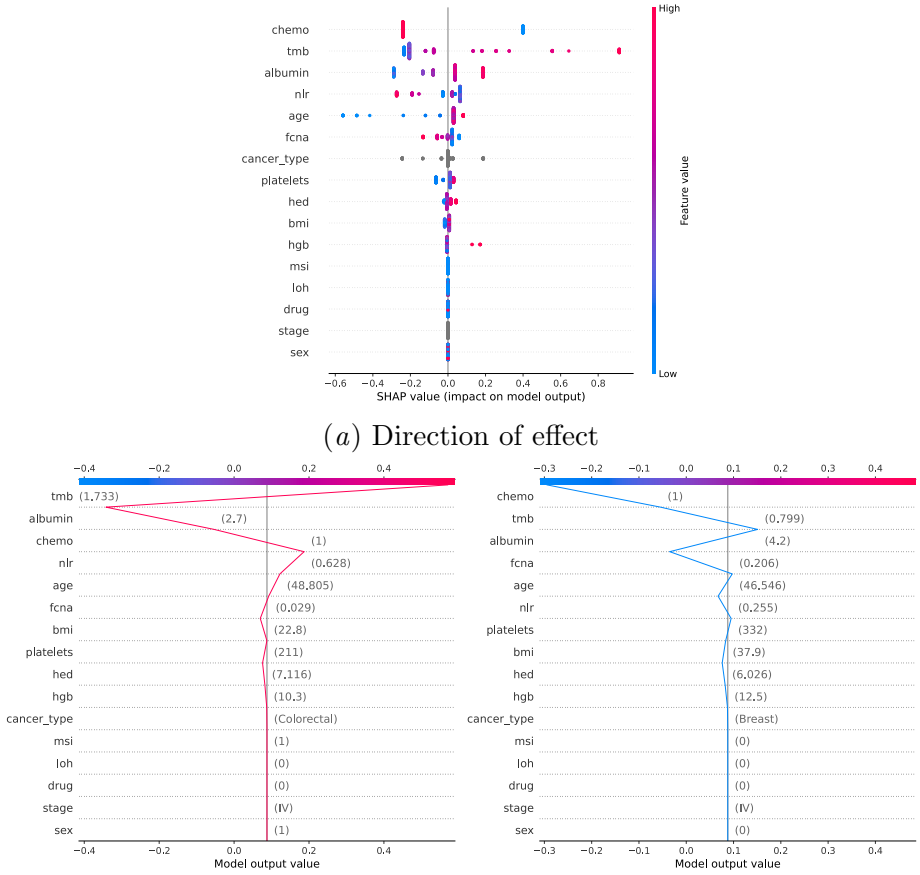
We highlight the following characteristics of our best model CB16:

- Our model, CB16 is less complex than both RF16 and our own RF model. RF16 has a run-time complexity more than 100X greater than CB16, and 11X greater than RF[1]. This means that our model is simpler and potentially generalises better to unseen patient cohorts.
- Unlike RF16 which uses inherent feature importance scores, we employed Shapley Additive Explanations (SHAP) to provide additional insights into model predictions from CB16. We posit that SHAP offers greater clinical utility compared to RF16's method, because: 1) SHAP explanations award direction of effect, and 2) SHAP generates localised explanations tailored for specific patients. We provide further details in 4.4.

[1]Calculated as $O(TD)$ where T is number of trees, D is tree depth. RF16;RF;CB16: T=1000;281;73; and D=8;3;1, respectively)

4.4. Model explanations using SHAP

Beyond importance scores: Similar to RF16, our model makes the decision process transparent and provides reassurance that the model is making reasonable predictions (see Appendix D). However, our model explanations go beyond feature importance scores by providing a direction of effect. In Figure 1a, we observe positive and negative relationships between each predictor with the outcome variable, as well as the magnitude of its contribution towards the overall prediction. For example, both high TMB and blood marker levels (albumin, HGB and platelet counts) were shown to be significantly predictive of response to



(*a*) Direction of effect

(*b*) Predict response

(*c*) Predict non-response

Figure 1: SHAP visualisations for `CB16`. SHAP values are organised by features and ordered by decreasing importance. Feature values are represented along a red-blue colour spectrum (red - high; blue - low), which signify push towards prediction of response (right) /non-response (left) respectively.

ICB, while the converse was true for non-response. These identified relationships may be easier to interpret under the context of biological and clinical knowledge as compared to simply overall feature importance scores traditionally output by DT-based models.

Instance-wise explanations: Figure 1b and Figure 1c shows instance-wise explanation plots that offer fine-grained details into the decision process of `CB16`, by delineating the path of decisions taken to arrive at a prediction for each patient. Rather than outputting an average score per feature like `RF16`, our explanations provide a precise description of how individual variables contribute in unique ways to create the patient's condition. This allows us to analyse specific combinations of features at the patient level or any user-specified level such as disease type. This type of analysis could offer further insights into the biology, and prove useful to clinicians and researchers interested in selecting potentially novel prognostic features or generate hypotheses about how features could be causally related.

4.5. Encouraging clinical adoption

Despite usage of explanation frameworks like SHAP to improve model interpretability, we have yet to see widespread adoption of ML in the clinic. Through talks with clinicians, we understand that it partly lies in differences in our approach to understanding machine learning topics, leading to varying degrees of trust in ML models. For example, a computer scientist may understand SHAP from a viewpoint of mathematical robustness and therefore trusts that it explains feature importance adequately. However, the inner workings of SHAP may matter less to a clinician, who sees SHAP as not so different from other models that output feature importance. Thus, we believe that to gain trust with clinicians, it is integral for us to go beyond SHAP, and explain our model through the lenses of clinicians' understanding.

Here, we attempt a first-principles approach to explaining `CB16`, using a concept that clinicians are familiar with. Tumour mutation burden (TMB) is known to be a strong predictor (Wang et al., 2021), and clinicians may prefer to use it as a simple yet effective way to predict response to ICB. First, we create a model `TMB-only` that uses only TMB, and show that the performance is significantly worse than `CB16`, with F1 score decreasing 24 percentage points to 46% (see Appendix E). Thus, overly simplistic models do not do well.

Secondly, we demonstrate that we could see boosting models such as `CB16` as error-correcting, as the modelling process adds new predictors to learn from the errors of previous predictors. We fit a CatBoost model `TMB-EC` to improve upon `TMB-only` significantly (F1 score 60%, see Appendix E), to a score much closer to our original model. We note that `TMB-EC` works similarly to `CB16`, except it starts from a simpler baseline that uses only TMB. We also note that error-corrective models such as `CB16` are useful in that they are able to learn from previous mistakes (in themselves, or other models), and thus often perform better than non-error correcting ones.

5. Deep learning for tabular data

Having studied the performance of classical ML models, we now turn our attention to developing DL models to improve model performance. The primary challenge is that tabular data, which is predominant in healthcare applications, lacks inductive biases that can be easily exploited by deep neural networks. In this section, we propose DL models that address this issue and we test them on the task of predicting ICB response.

5.1. Proposed model architectures

In this section we develop DL models based on the following approaches:
1. We create and learn structure from tabular data, by leveraging on learnt spatial projections from tabular inputs, which can be processed by networks such as CNNs using appropriate inductive bias.
2. We learn good representations of tabular data using autoencoders that employ denoising and contrastive objectives.
3. We ensemble our models to reduce variance and mitigate overfitting.

Model training and evaluation details are described in Appendix G.

5.1.1. Learning structure from tabular data using CNNs

We attempt to introduce structure into tabular data by proposing the following approach inspired by (Kaggle, 2020). We first increase feature dimension using a fully-connected network (FCN), followed by reshaping to groups of features, where each group will be considered an "image". We consider

these high-dimensional features as different aspects of the original features, which can then be combined non-linearly using a CNN model. The model learns the correct spatial order of these feature aspects as the FCN learns weights that determine how to project features in a manner that will allow the CNN to extract local patterns from each "image".

Specifically, we propose a model architecture as follows:

- A wide FCN with hidden size 4096
- A reshaping layer to reshape data into dimensions (256x16)
- A CNN architecture with N blocks, each block consisting of a 1DCNN with 512 filters, kernel size 3, batch normalisation, dropout at a rate of 0.1, residual connections between blocks, and max pooling between layers
- A classifier layer using a linear FCN

A schematic illustration is shown in Figure 4 under Appendix F. We thereafter refer to this architecture as `tabular-cnn`.

5.1.2. Learning good representations using autoencoders

Real-world datasets are noisy and we consider the use of autoencoders to learn noise-robust representations. We re-interpret `tabular-cnn` as an autoencoder and pre-train the model to learn representations via either (a) a denoising objective or (b) a contrastive objective. Schematic illustrations of both models are shown in Figure 5 under Appendix F. We refer to the denoising and contrastive autoencoder as `denoising-ae` and `contrastive-ae` respectively.

To train both autoencoders, we first create a noisy view of the input data as follows. For each patient i and feature j, we generate a new training sample $\tilde{x}_{i,j}$, by swapping data with some probability p from $x_{k,j}$, where k is a randomly sampled patient from the same class and $k \neq i$.

To construct `denoising-ae`, we use `tabular-cnn` as an encoder and consider the flattened layer as the bottleneck. The decoder is a series of transposed convolution layers meant to upsample the data and project back to the original input dimensions. The model receives as input a noisy view $\tilde{x}$ and attempts to reconstruct the original input data x by minimising the reconstruction loss, given as:

$$\mathcal{L}_{\text{denoising}} = -\sum_{i=1}^{m}[\mathcal{L}_{\text{MAE}}(\text{model}(\tilde{\text{x}}_\text{i}, \text{x}_\text{i})] \quad (1)$$

To construct `contrastive-ae`, we create separate prongs of projection and convolution layers for clean and noisy data. We encourage the model to learn a contrastive objective that pushes the latent representation of two data views from the same patient close, while pushing data points from different patients apart. This is achieved via minimizing the InfoNCE loss function (Oord et al., 2019), which is defined as:

$$\mathcal{L}_{\text{contrastive}} = -\sum_{i=1}^{m} \frac{\exp(z_i \cdot z_i'/\tau)}{\sum_{k=1}^{m} \exp(z_i \cdot z_k'/\tau)} \quad (2)$$

where z_i and z_i' are the latent representations of views from the same patient where we maximise similarity, z_k' is a representation from any other patient j within the same batch where $i \neq j$, m is the total number of patients and τ is a temperature parameter that rescales logit scores before applying the Softmax function and encourages better learning (Guo et al., 2017).

For the prediction task, we finetune the model using all training samples. For the `denoising-ae`, we strip off the decoder and consider the bottleneck layer as the latent representation. We connect a fully-connected layer with linear activation to the latent representation. We freeze the encoder weights, and only train the weights of the fully-connected layer on the noisy training data. We subsequently pass a noisy version

of the test dataset through the model to get final outputs. For the `constrastive-ae`, we do the same except that we take only the encoder outputs corresponding to the noisy view as the latent representation.

5.1.3. Ensemble modelling

Ensembling models is a known technique to reduce variance and overfitting (Ganaie et al., 2022), and we considered this method to improve the performance of our deep neural networks. We stacked 5 `tabular-cnn` models together using a bagging approach, where each model was trained on a random subset of training data. Model predictions were aggregated using majority voting. We refer to this model as `ensemble-4096`, which represents the fact that the FCN following the input layer is of size 4096 (see Figure 4). To test the effect of reducing model parameters, we performed experiments that reduced the feature dimension projection size by half (`ensemble-2048`), quarter (`ensemble-1024`) and one-eighth (`ensemble-512`).

5.1.4. Baselines

We create competitive baselines for our DL models that utilise state-of-the-art (SOTA) architecture to process tabular data, namely: TabNet (Arik and Pfister, 2020), TabTransformer (Huang et al., 2020), and NODE (Popov et al., 2019). Details of hyperparameters are available in Appendix H.

In addition, we also run a baseline for our ensemble models, by ensembling 100 simple MLPs, each with a single hidden layer and 10 nodes. This achieves a similar effect as an ablation study where we examine only the effect of 1 variable (base model `tabular-cnn` or MLP) while keeping the rest constant.

5.2. Performance comparisons

Table 2 shows the results of our proposed DL models, based on `tabular-cnn`, `denoising-ae` and `contrastive-ae`. Note that the ensemble models use `tabular-cnn`. We exclude results of `node` from our analysis due to large performance variation across runs.

We note that all our models achieve a good balance between sensitivity and specificity, and they all surpassed the performance of baseline models significantly in terms of the F1 score. In particular, the effect of learning structure from tabular inputs using `tabular-cnn` already contributes 12.4 percentage-point improvement to the F1 score compared to the best baseline `tabtransformer`. While we note that the AUPRC favours `tabtransformer` slightly, again inspection of the precision-recall curves demonstrate that `tabular-cnn` achieves consistently good precision levels across most sensitivity thresholds while `tabtransformer` favours high precision only at low sensitivity (see Appendix I). After ensembling, PPV improves significantly while slightly compromising sensitivity, resulting in scores that top the charts for both F1 score and AUPRC. Furthermore, when ensembled, our model can accomodate up to 8X reduction of parameters with no significant loss in performance. Nonetheless, our best model `ensemble-512` still exhibits a performance gap of around 10 percentage-point difference in F1 score when comparing against our best classical ML model `CB16`.

Interestingly, representation learning resulted in a drop in AUPRC which is indicative of lowered discriminatory performance, and did not give significant gains in the other metrics. Between the two autoencoders, `denoising-ae` performed slightly better than `contrastive-ae` by around 1 to 2 percentage-points in all scores. We discuss possible reasons in Section 6.

Table 2: Comparing performance of various DL architecture. Metrics are reported as average (95% CI) of at least 10 model initialisations. Best scores are bolded; second-best scores are underlined.

Model	Accuracy	Sensitivity	PPV	F1	Specificity	AUPRC
Single models						
`tabular-cnn`	66.8 (64.7, 68.9)	**77.3** (75.8, 78.9)	47.6 (45.8, 49.5)	58.9 (57.3, 60.4)	62.2 (59.1, 65.2)	52.6 (49.6, 55.6)
`denoising-ae`	67.7 (64.5, 70.9)	<u>76.3</u> (74.2, 78.4)	48.5 (45.2, 51.8)	59.2 (56.7, 61.7)	63.9 (59.1, 68.6)	39.8 (35.1, 44.6)
`contrastive-ae`	66.3 (64.6, 68.0)	74.3 (70.7, 77.9)	46.9 (45.1, 48.7)	57.4 (56.2, 58.5)	62.8 (59.1, 66.4)	37.1 (33.0, 41.2)
Ensemble models						
`ensemble-4096`	72.1 (71.0, 73.3)	67.8 (66.1, 69.5)	53.5 (51.8, 55.2)	<u>59.8</u> (58.8, 60.7)	74.0 (72.0, 76.1)	**53.7** (52.3, 55.0)
`ensemble-2048`	71.0 (69.0, 73.0)	63.9 (59.1, 68.7)	52.5 (49.9, 55.1)	57.3 (55.8, 58.8)	74.1 (69.6, 78.7)	50.5 (49.1, 51.8)
`ensemble-1024`	72.3 (70.2, 74.5)	67.6 (63.2, 71.9)	54.2 (51.3, 57.0)	<u>59.8</u> (58.0, 61.7)	74.4 (70.3, 78.6)	52.2 (50.3, 54.0)
`ensemble-512`	**72.9** (71.7, 74.2)	67.0 (63.9, 70.1)	<u>54.8</u> (52.8, 56.8)	**60.2** (58.9, 61.5)	75.6 (72.9, 78.2)	<u>53.4</u> (51.7, 55.2)
Baseline models						
`tabnet`	69.9 (69.4, 70.4)	5.8 (3.2, 8.4)	50.1 (30.9, 69.4)	10.2 (5.9, 14.6)	**98.1** (97.3, 98.9)	44.1 (39.6, 48.7)
`tabtransformer`	<u>72.3</u> (71.1, 73.5)	39.4 (37.5, 41.4)	**57.1** (53.1, 61.0)	46.5 (44.9, 48.1)	<u>86.7</u> (84.8, 88.6)	52.9 (51.9, 54.0)
`node`	71.3 (69.5, 73.0)	56.2 (34.7, 77.8)	47.0 (35.0, 59.1)	48.3 (30.2, 66.3)	77.9 (69.6, 86.1)	50.8 (43.1, 58.4)
`100-mlps`	50.3 (49.0, 51.6)	48.4 (46.5, 50.4)	30.3 (29.2, 31.5)	37.3 (35.9, 38.7)	51.1 (49.6, 52.7)	31.0 (29.7, 32.2)

6. Discussion

Model performance This paper demonstrates that our approach accurately predicts response to immunotherapeutic drugs, with our best model `CB16` achieving a strong performance of 80.6% sensitivity and 78.8% specificity. Our model surpasses the performance of a previous model `RF16` (Chowell et al., 2021), achieving statistically significant increments on all calculated metrics. While it is difficult to objectively assess whether the improvement is clinically significant, we note that in our patient cohort, this translates to correctly diagnosing an additional 57 responders who might be otherwise overlooked for ICB therapy, and excluding 102 non-responders from a potentially risky treatment. We also note that certain clinical assays in the market also demonstrate 80% sensitivity (Yohe, 2020), demonstrating that our model is potentially clinically viable. Future work should consist of validating our model on other cohorts.

Model intepretability We propose the use of SHAP explanations to enhance transparancy and improve upon end-user trust. Our explanations provide reassurance to the end-user that our model is using reasonable features through analysis of global

SHAP values. It also allows us to interpret the likely direction of effect in the context of biological and clinical knowledge. Furthermore, instance-wise explanations of SHAP provide additional context beyond an average feature importance score by breaking down individual factors for any given patient. This could prove invaluable to clinicians who desire a more personalised approach to patient monitoring based on the unique characteristics of the patient.

Deep learning for tabular data Our work is a prime example of how current deep learning techniques may not be suitable for all kinds of data. Our results are aligned with a large body of literature that demonstrates how current SOTA methods for tabular data may not work well for all datasets, and highlights difficulties in creating DL models that rival the performance of ensembled trees (Shwartz-Ziv and Armon, 2021; Borisov et al., 2022; Gorishniy et al., 2021). Nonetheless, our work proposed several underexplored ideas in the literature that could potentially boost the performance of DL on tabular data, and we described and documented our most promising approaches on a real world noisy dataset. We found that creating structure within data that allowed neural networks to leverage on led to a major performance boost. Furthermore ensembling techniques build upon the performance of such networks by reducing variance of the data, with the caveat that the base model should be sufficiently complex.

The lowered AUPRC scores in the two explored representation learning techniques possibly indicate that the learnt representations were not optimal and impacted the models' discriminatory performance. We believe that it is due to the introduced noise leading to failure in learning well-formed representations that distinguished between the two classes. `denoising-ae` would be more equipped to handle the noise due to its denoising capabilities, explaining its comparatively better performance to `contrastive-ae`. We are interested in improving upon this technique and will look into methods that generate more optimal noise distributions for both autoencoders. In conclusion, our work documents useful starting points for researchers and practitioners interesting in applying ML to noisy real-world tabular data.

Acknowledgments

This research is supported by A*STAR, CISCO Systems (USA) Pte. Ltd and National University of Singapore under its Cisco-NUS Accelerated Digital Economy Corporate Laboratory (Award I21001E0002). Additionally, we would like to thank Dr. Iain Tan and Dr. Raghav Sundar from the local hospitals for useful clinical feedback and interesting discussions on this work.

References

Ahmad Y. Abuhelwa, Ganessan Kichenadasse, et al. Machine Learning for Prediction of Survival Outcomes with Immune-Checkpoint Inhibitors in Urothelial Cancer. *Cancers*, 13, January 2021. URL https://www.mdpi.com/2072-6694/13/9/2001.

Sercan O. Arik and Tomas Pfister. TabNet: Attentive Interpretable Tabular Learning. *arXiv:1908.07442 [cs, stat]*, December 2020. URL http://arxiv.org/abs/1908.07442.

Dara Bahri, Heinrich Jiang, et al. SCARF: Self-Supervised Contrastive Learning using Random Feature Corruption. *arXiv:2106.15147 [cs]*, March 2022. URL http://arxiv.org/abs/2106.15147.

Ravneet Bajwa, Anmol Cheema, et al. Adverse Effects of Immune Checkpoint Inhibitors (Programmed Death-1 Inhibitors and Cytotoxic T-Lymphocyte-Associated Protein-4 Inhibitors): Results of a Retrospective Study. *Journal of Clinical Medicine Research*, 11, April 2019.

Sébastien Benzekry, Mathieu Grangeon, et al. Machine Learning for Prediction of Immunotherapy Efficacy in Non-Small Cell Lung Cancer from Simple Clinical and Biological Data. *Cancers*, 13, December 2021. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8699503/.

Vadim Borisov, Tobias Leemann, et al. Deep Neural Networks and Tabular Data: A Survey. *arXiv:2110.01889 [cs]*, February 2022. URL http://arxiv.org/abs/2110.01889.

Tom Brown, Benjamin Mann, et al. Language Models are Few-Shot Learners. In *2020 Conference on Neural Information Processing Systems*, volume 33, 2020. URL https://papers.nips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, 2016. ISBN 978-1-4503-4232-2. URL http://doi.acm.org/10.1145/2939672.2939785.

Zihao Chen, Maoli Wang, et al. A Machine Learning Model to Predict the Triple Negative Breast Cancer Immune Subtype. *Frontiers in Immunology*, 12, 2021. URL https://www.frontiersin.org/article/10.3389/fimmu.2021.749459.

Diego Chowell, Seong-Keun Yoo, et al. Improved prediction of immune checkpoint blockade efficacy across multiple cancer types. *Nature Biotechnology*, November 2021.

Pramod Darvin, Salman M. Toor, et al. Immune checkpoint inhibitors: recent progress and potential biomarkers. *Experimental & Molecular Medicine*, 50, December 2018. URL http://www.nature.com/articles/s12276-018-0191-1.

Jacob Devlin, Ming-Wei Chang, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. URL http://arxiv.org/abs/1810.04805.

A. V Dorogush, V Ershov, et al. CatBoost: Gradient Boosting with Categorical Features Support. *arXiv:1810.11363 [cs.LG]*, October 2018. URL https://arxiv.org/abs/1810.11363.

E. A Eisenhauer, P Therasse, et al. New response evaluation criteria in solid tumors: Recist guideline (version 1.1). *European Journal of Cancer*, 45, January 2009.

M. A. Ganaie, Minghui Hu, et al. Ensemble deep learning: A review. *arXiv:2104.02395 [cs]*, March 2022. URL http://arxiv.org/abs/2104.02395.

Aaron M. Goodman, Shumei Kato, et al. Tumor Mutational Burden as an Independent Predictor of Response to Immunotherapy in Diverse Cancers. *Molecular Cancer Therapeutics*, 16, November 2017. URL https://doi.org/10.1158/1535-7163.MCT-17-0386.

Yury Gorishniy, Ivan Rubachev, et al. Revisiting Deep Learning Models for Tabular Data. *arXiv:2106.11959 [cs]*, Novem-

ber 2021. URL http://arxiv.org/abs/2106.11959.

Chuan Guo, Geoff Pleiss, et al. On Calibration of Modern Neural Networks. *arXiv:1706.04599 [cs]*, August 2017. URL http://arxiv.org/abs/1706.04599.

Kaiming He, Xiangyu Zhang, et al. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Xin Huang, Ashish Khetan, et al. TabTransformer: Tabular Data Modeling Using Contextual Embeddings. *arXiv:2012.06678 [cs]*, December 2020. URL http://arxiv.org/abs/2012.06678.

Zedong Jiang, Yao Zhou, et al. A Combination of Biomarkers Predict Response to Immune Checkpoint Blockade Therapy in Non-Small Cell Lung Cancer. *Frontiers in Immunology*, 12, 2021. URL https://www.frontiersin.org/article/10.3389/fimmu.2021.813331.

Kaggle. Historical Data Science Trends on Kaggle, 2019. URL https://kaggle.com/shivamb/data-science-trends-on-kaggle.

Kaggle. Mechanisms of Action (MoA) Prediction, 2020. URL https://www.kaggle.com/c/lish-moa/discussion/202256.

Tauseef Khan, Ram Sarkar, et al. Deep learning approaches to scene text detection: a comprehensive review. *Artificial Intelligence Review*, 54, June 2021. URL https://doi.org/10.1007/s10462-020-09930-6.

Hyera Kim, Minsuk Kwon, et al. Clinical outcomes of immune checkpoint inhibitors for patients with recurrent or metastatic head and neck cancer: real-world data in Korea. *BMC Cancer*, 20, August 2020. URL https://doi.org/10.1186/s12885-020-07214-4.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017. URL http://arxiv.org/abs/1412.6980.

Óscar Lapuente-Santana, Maisa van Genderen, et al. Interpretable systems biomarkers predict response to immune-checkpoint inhibitors. *Patterns*, 2, August 2021. URL https://www.sciencedirect.com/science/article/pii/S2666389921001264.

Tsung-Yi Lin, Priya Goyal, et al. Focal Loss for Dense Object Detection. In *2017 International Conference on Computer Vision*, 2017.

Zhihao Lu, Huan Chen, et al. Prediction of immune checkpoint inhibition with immune oncology-related gene expression in gastrointestinal cancer using a machine learning classifier. *Journal for ImmunoTherapy of Cancer*, 8, August 2020. URL https://jitc.bmj.com/content/8/2/e000631.

Y. Murciano-Goroff, A.B. Warner, et al. The future of cancer immunotherapy: microenvironment-targeting combinations. *Cell Research*, 30:507–519, 2020.

R. Nandhini Abirami, P. M. Durai Raj Vincent, et al. Deep CNN and Deep GAN in Computational Visual Perception-Driven Image Analysis. *Complexity*, 2021, April 2021. URL https://www.hindawi.com/journals/complexity/2021/5541134/.

Aaron van den Oord, Sander Dieleman, et al. WaveNet: A Generative Model for Raw

Audio. *arXiv:1609.03499 [cs]*, September 2016. URL http://arxiv.org/abs/1609.03499.

Aaron van den Oord, Yazhe Li, et al. Representation Learning with Contrastive Predictive Coding. *arXiv:1807.03748 [cs, stat]*, January 2019. URL http://arxiv.org/abs/1807.03748.

Adam Paszke, Sam Gross, et al. Automatic differentiation in PyTorch. In *2017 Conference on Neural Information Processing Systems*, 2017.

F. Pedregosa, G. Varoquaux, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

E. Pons-Tostivint, A. Latouche, et al. Comparative analysis of durable responses on immune checkpoint inhibitors versus other systemic therapies: a pooled analysis of phase iii trials. *JCO Precision Oncology*, 3:1–10, 2019.

Sergei Popov, Stanislav Morozov, et al. Neural Oblivious Decision Ensembles for Deep Learning on Tabular Data. *arXiv:1909.06312 [cs, stat]*, September 2019. URL http://arxiv.org/abs/1909.06312.

Hendrik Purwins, Bo Li, et al. Deep Learning for Audio Signal Processing. *IEEE Journal of Selected Topics in Signal Processing*, 13, May 2019. URL http://arxiv.org/abs/1905.00078.

Yanyun Shen, Yunfeng Chen, et al. Treatment-related adverse events as surrogate to response rate to immune checkpoint blockade. *Medicine*, 99, September 2020. URL https://journals.lww.com/md-journal/fulltext/2020/09110/treatment_related_adverse_events_as_surrogate_to.65.aspx.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6, July 2019. URL https://doi.org/10.1186/s40537-019-0197-0.

Ravid Shwartz-Ziv and Amitai Armon. Tabular Data: Deep Learning is Not All You Need. *arXiv:2106.03253 [cs]*, November 2021. URL http://arxiv.org/abs/2106.03253.

Peipei Wang, Yueyun Chen, et al. Beyond Tumor Mutation Burden: Tumor Neoantigen Burden as a Biomarker for Immunotherapy and Other Types of Therapy. *Frontiers in Oncology*, 11, April 2021. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8117238/.

Youqiong Ye, Yongchang Zhang, et al. Profiling of immune features to predict immunotherapy efficacy. *The Innovation*, 3, January 2022. URL https://www.sciencedirect.com/science/article/pii/S2666675821001193.

S. Yohe. How good are COVID-19 (SARS-CoV-2) diagnostic PCR tests, 2020. URL https://www.cap.org/member-resources/articles/how-good-are-covid-19-sars-cov-2-diagnostic-pcr-tests.

Yuanjun Zhao, Xianjun Xia, et al. Applications of Deep Learning to Audio Generation. *IEEE Circuits and Systems Magazine*, 19, 2019.

Ming Zheng. Tumor mutation burden for predicting immune checkpoint blockade response: the more, the better. *Journal for ImmunoTherapy of Cancer*, 10, January 2022. URL https://jitc.bmj.com/content/10/1/e003087.

Appendix A. Description of patient characteristics

Table 3: Characteristics of patients in cohort

Feature	Num patients (n = 1479)	Training set (n = 1184)	Test set (n = 295)
Sex (%)			
Female	668 (45.17)	529 (44.68)	139 (47.12)
Male	811 (54.83)	655 (55.32)	156 (52.88)
Median age (IQR)	64 (55-71)	64 (55-71)	64 (55-72)
Cancer type (%)			
NSCLC	538 (36.38)	430 (36.32)	108 (36.61)
Melanoma	186 (12.58)	149 (12.58)	37 (12.54)
Renal	91 (6.15)	73 (6.17)	18 (6.10)
Bladder	82 (5.54)	66 (5.57)	16 (5.42)
Head and neck	69 (4.67)	55 (4.65)	14 (4.75)
Sarcoma	67 (4.53)	54 (4.56)	13 (4.41)
Endometrial	65 (4.39)	52 (4.39)	13 (4.41)
Gastric	64 (4.33)	51 (4.31)	13 (4.41)
Hepatobiliary	52 (3.52)	42 (3.55)	10 (3.39)
Small cell lung cancer	50 (3.38)	40 (3.38)	10 (3.39)
Colorectal	46 (3.11)	37 (3.13)	9 (3.05)
Esophageal	44 (2.97)	35 (2.96)	9 (3.05)
Pancreatic	35 (2.37)	28 (2.36)	7 (2.37)
Mesothelioma	34 (2.30)	27 (2.28)	7 (2.37)
Ovarian	31 (2.10)	25 (2.11)	6 (2.03)
Breast	25 (1.69)	20 (1.69)	5 (1.69)
Drug class (%)			
PD1/PDL1	1221 (82.56)	969 (81.84)	252 (85.42)
CTLA-4	5 (0.33)	5 (0.42)	0 (0.00)
Combination	253 (17.11)	210 (17.74)	43 (14.58)
ICB response (%)			
Responder	409 (27.65)	319 (26.94)	90 (30.51)
Non-responder	1070 (72.35)	865 (73.06)	205 (69.49)
Chemotherapy prior ICB (%)			
No	463 (31.30)	370 (31.25)	93 (31.53)
Yes	1016 (68.70)	814 (68.75)	202 (68.47)
Stage (%)			
I-III	97 (6.56)	78 (6.59)	19 (6.56)
IV	1382 (93.44)	1106 (93.41)	276 (93.44)

Appendix B. ML model parameters

Hyperparameters were obtained via 5-fold cross-validation on the training dataset, and are as follows:

- **Logistic regression** (`LR`), C=0.61
- **Random forest** (`RF`) criterion='entropy', max depth=3, max features=15, min samples in leaf=6, estimator number=281
- **Support Vector Machine** (`SVM`) gamma='auto', kernel='poly'
- **XGBoost** (`XGB`) estimator number=68, learning rate=0.1, max depth=1, class weight=2.9, subsample=0.7
- **CatBoost** (`CB16`) iterations=73, learning rate=0.122, max depth=1, class weight=2.9, one-hot encoding for all features

We run each model with 10 different initialisations, each initialisation bootstrapped 10 times, to create 100 runs, and calculated test statistics (mean and CI) for each performance metric. We use the default threshold of 0.5 to obtain our model predictions.

Appendix C. Precision-recall curves

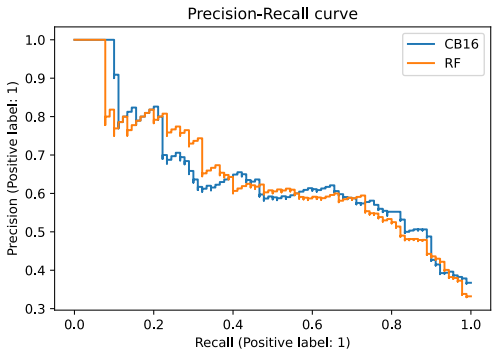
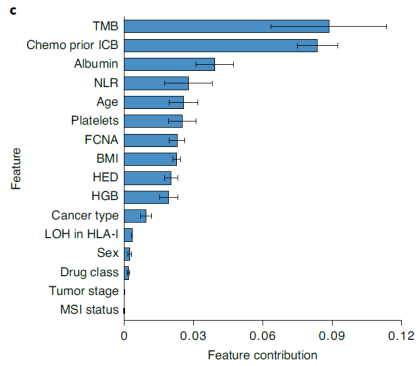


Figure 2: Precision-recall curves of `CB16` versus `RF`

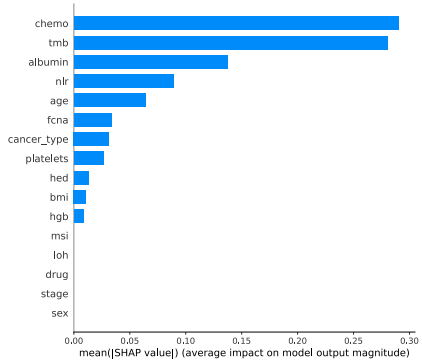
Appendix D. Making reasonable predictions

To determine whether our model was using reasonable features, we performed an exact comparison of overall feature importance from `RF16` and `CB16` in Figure 3a and Figure 3b respectively, and we conclude that both models are offering similar explanations. We note that the order of features is similar in both models: tumour mutation burden (TMB) and chemotherapy his-

tory were both considered most important for prediction, while drug class, cancer stage, microsatellite instability (MSI) status, and genomic predictors such as divergence in human leukocyte antigen (HLA) alleles, were less significant. Furthermore, they correspond to biological and clinical insight that TMB is highly important, as are blood-based markers such as levels of albumin and hemoglobin.



(*a*) `RF16` (theirs)



(*b*) `CB16` (ours)

Figure 3: (a) and (b): Direct comparisons of feature importance between `RF16` (theirs) and `CB16` (ours), both ordered by decreasing importance.

Appendix E. Correcting residual errors

Table 4 shows the results of `TMB-only`, a model that uses tumour mutation burden (TMB) as the sole feature, and `TMB-EC`, an error correction model that fits new predictors on the errors made by `TMB-only` using gradient boosting. Both models used the CatBoost algorithm (Dorogush et al., 2018). `TMB-only` was built by training a Catboost model on the TMB variable. `TMB-EC` was built by leveraging on `TMB-only` predictions as a baseline, and building a new Catboost model based on the errors of the prediction using all features. Hyperparameter details are as follows:

- `TMB-only`: iterations=8, learning rate=0.122, max depth=1, class weight=2.9
- `TMB-EC`: In Stage 1, we train a `TMB-only` model and obtain predictions on the training data. In Stage 2, we build a new CatBoost model on the residual errors of these predictions, and train using the full feature set. Stage 2 hyperparameters are: iterations=65, learning rate=0.122, max depth=1, class weight=2.9, one-hot encoding for all features

Table 4: Results of `TMB-only` versus `EC`

Model	Acc	Sens	PPV	F1	Spec
`TMB-only`	0.685	0.444	0.482	0.462	0.790
`TMB-EC`	0.702	0.744	0.507	0.604	0.683
`CB16`	0.793	0.806	0.625	0.703	0.788

Appendix F. DNN architectures

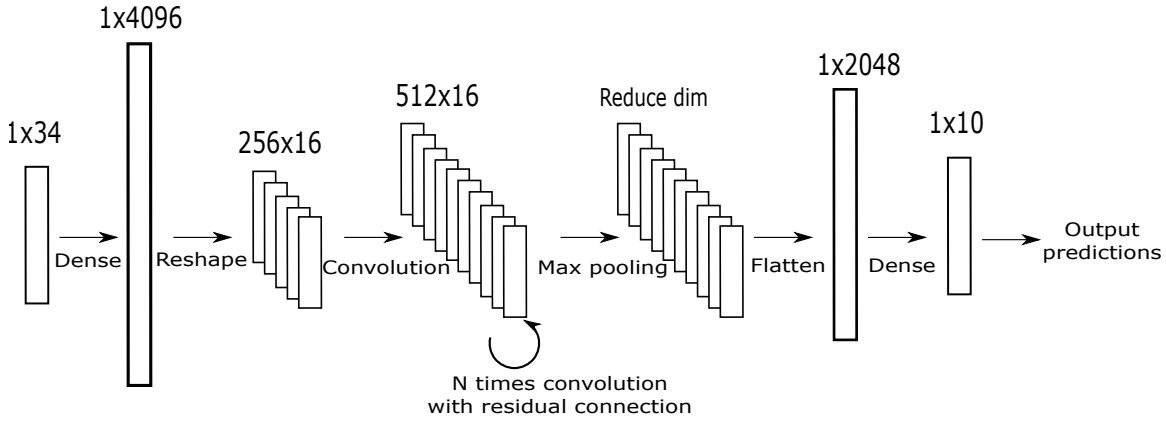


Figure 4: Schematic representation of `tabular-cnn`

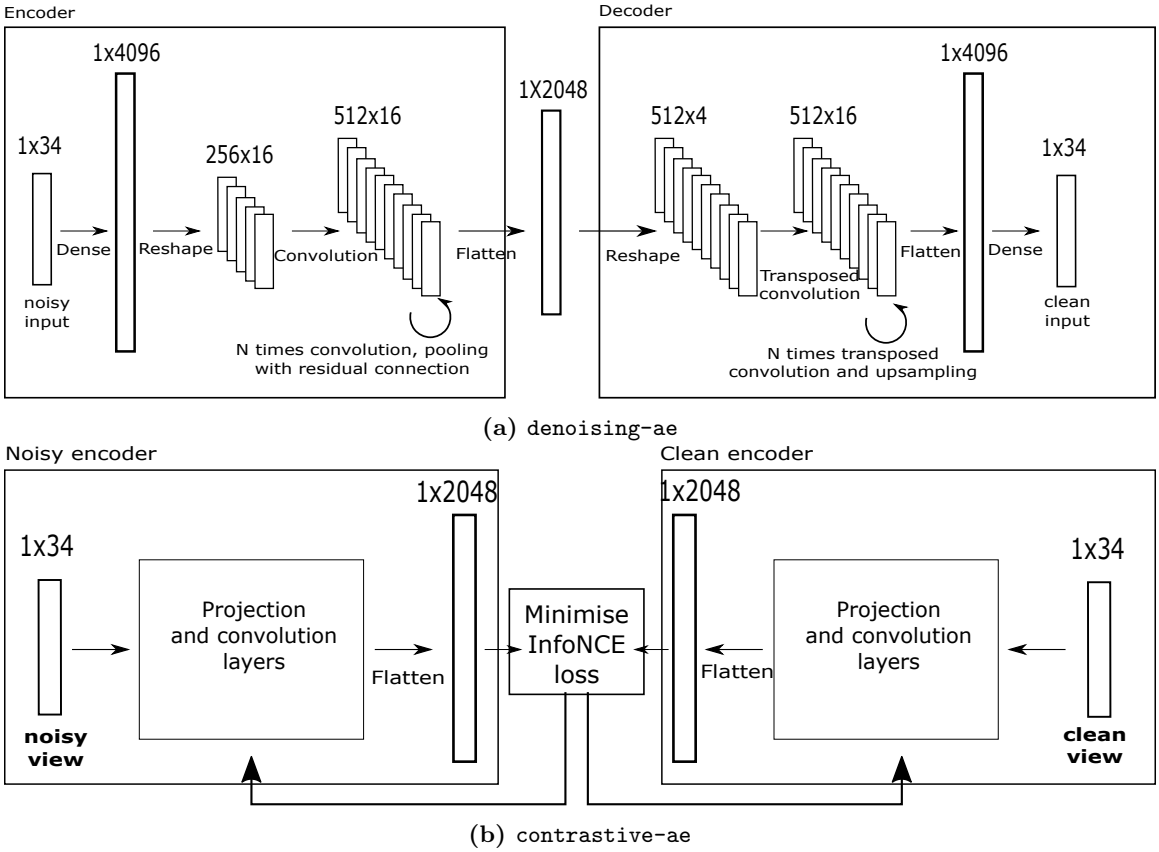


Figure 5: Schematic representation of the pretraining architecture for (a) `denoising-ae` and (b) `contrastive-ae`. The encoder structure of `contrastive-ae` is the same as that of `denoising-ae` and thus not fully illustrated.

Appendix G. DNN training and evaluation

G.1. Model training

All DL models were created using Pytorch (Paszke et al., 2017); each were trained (for autoencoders, finetuned) on the task of predicting immunotherapy response. The training objective was to minimize focal loss (Lin et al., 2017) for all models, with weights set to the inverse of the corresponding class support. We use focal loss as it has been shown to improve performance on classification tasks when the dataset is highly imbalanced. All models were trained for 1000 epochs with early stopping based on best validation loss, and used the Adam optimiser (Kingma and Ba, 2017) with learning rate decay at a rate of 0.1 every 200 epochs.

G.2. Model evaluation

We split the original training data into two parts to create an additional validation dataset, at a ratio of 4:1. All developed models were tuned for best hyperparameters on the validation dataset and evaluated on the test dataset. We trained at least 10 separate models for each model architecture, which differ in terms of initial weights. We report the average for the following performance metrics: Accuracy, Recall (or sensitivity), Precision (or Positive Predictive Value), F1 score, Specificity and AUPRC.

Appendix H. Hyperparameters of baseline models

TabNet We ran `tabnet` using PYTORCH-WIDEDEEP under the following settings:

- Size of embedding dimensions = 32
- Width of attention embedding = 8
- Number of steps = 3
- Number of shared Gated Linear Units (GLU) per step = 2
- Number of independent GLU per step = 2

TabTransformer We ran `tabTranformer` using PYTORCH-WIDEDEEP under the following settings:

- Input dimension = 32
- Number of attention heads = 8
- MLP activation = ReLu
- MLP dropout rate = 0.1
- MLP batchnorm = False

NODE We ran `node` using NODE python package under the following settings:

- Layer dimension = 258
- Number of layers = 1
- Tree depth = 1

All models except `node` were trained to minimise a binary focal loss objective weighted by the support of each class, for at least 50 epochs, while `node` was trained to minimise cross entropy loss for 200 epochs. We used the Adam optimiser, with a learning rate scheduler set at 1e-3, which decayed at a factor of 0.1 every 20 epochs.

Appendix I. Precision-recall curves for DL models

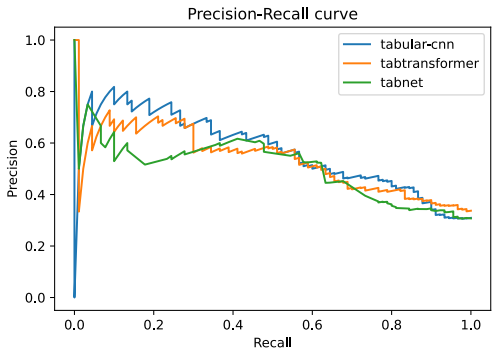


Figure 6: Precision-recall curves of `tabular-cnn` vs. `tabtransformer` & `tabnet`