Towards Interpretable Deep Neural Networks for Tabular Data

Khawla Elhadri
Marburg University
khawla.elhadri@uni-marburg.de

Jörg Schlötterer
Marburg University
joerg.schloetterer@uni-marburg.de

Christin Seifert
Marburg University
christin.seifert@uni-marburg.de

Abstract

Tabular data is the foundation of many applications in fields such as finance and healthcare. Although DNNs tailored for tabular data achieve competitive predictive performance, they are blackboxes with little interpretability. We introduce XNNTab, a neural architecture that uses a sparse autoencoder (SAE) to learn a dictionary of monosemantic features within the latent space used for prediction. Using an automated method, we assign human-interpretable semantics to these features. This allows us to represent predictions as linear combinations of semantically meaningful components. Empirical evaluations demonstrate that XNNTab attains performance comparable to that of state-of-the-art, black-box neural models and classical machine learning approaches while being fully interpretable.

1 Introduction

Tabular data is the most common type of data in a wide range of industries, including advertising, finance and healthcare. While deep neural networks (DNNs) achieved major advances in computer vision, their performance on tabular data remained below that of Gradient Boosted Decision Trees (GBDTs) [Chen and Guestrin, 2016; Khan *et al.*, 2022]. To improve the performance of DNNs on tabular data, several deep learning methods have been developed with specialized architectures that take into account the unique traits of tabular data (mixture of categorical, ordinal and continuous features) in order to be on par [Huang *et al.*, 2020] or even outperform GBDTs [Chen *et al.*, 2024]. While these methods address the limitations of DNNs in terms of performance, DNNs still remain blackboxes, whose decisions are hard to communicate to end users [Schwalbe and Finzel, 2023].

In this paper, we address the interpretability limitation of existing DNNs and present XNNTab, a deep neural network with a sparse autoencoder (SAE) component. The SAE learns a dictionary of monosemantic, sparse features that are used for outcome prediction. We take inspiration from the success of SAEs in learning interpretable features from LLM's internal activations [Huben *et al.*, 2023; Yun *et al.*, 2021]. We use an automated method to assign human-understandable semantics to the learned monosemantic dictionary features. Our method is illustrated in Figure 1 and described in Section 3. We show that XNNTab outperforms interpretable models and has similar performance to classical blackbox models. Additionally, XNNTab is fully interpretable: the final prediction is a simple linear combination of the dictionary features, which have easily accessible human-understandable semantics.

Preprint.

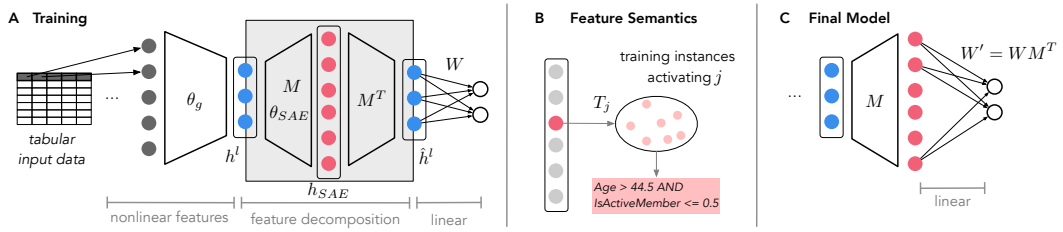


Figure 1: Architecture overview. **A.** An MLP learns nonlinear features, which are decomposed into monosemantic dictionary features using an SAE. **B.** Dictionary features are assigned a human-interpretable meaning by learning rules for the subset of training instances that highly activate a specific feature. **C.** Predictions are linear combinations of monosemantic features by combining the linear model components (the decoder, M^T, and the linear layer of the MLP, W).

2 Related Work

Gradient boosted decision trees (GBDTs) [Chen and Guestrin, 2016; Khan *et al.*, 2022] have shown state-of-the art performance on tabular data [Grinsztajn *et al.*, 2022]. To close this performance gap to GBDTs, neural architectures specifically designed for tabular data have been proposed, such as attention-based architectures [Gorishniy *et al.*, 2021; Somepalli *et al.*, 2022; Yan *et al.*, 2023] or retrieval-augmented models [Somepalli *et al.*, 2022; Gorishniy *et al.*, 2023]. Additional techniques include tailored regularizations [Jeffares *et al.*, 2023], deep ensembles [Gorishniy *et al.*, 2024], attentive feature selection strategies [Arik and Pfister, 2021], categorical feature embedding [Huang *et al.*, 2020] and reconstruction of binned feature indices [Lee *et al.*, 2024].

Recent methods focus not only on performance, but also aim to include interpretability. ProtoGate [Jiang *et al.*, 2024] is an ante-hoc interpretable architecture for tabular data, which learns prototypes of classes and predicts new instances based on their similarity to class prototypes. InterpreTabNet [Si *et al.*, 2024], a variant of TabNet [Arik and Pfister, 2021] adds post-hoc explanations by learning sparse feature attribution masks and using LLMs to interpret the learned features from the masks. Similar to ProtoGate and different from InterpetTabNet, XNNTab is intrinsically interpretable, i.e., it uses interpretable features directly for prediction. While ProtoGate relies on class prototypes, XNNTab learns relevant feature combinations that describe parts of instances, similar to part-prototype models [Elhadri *et al.*, 2025].

3 Method

XNNTab consists of a standard neural blackbox (e.g., an MLP) to learn nonlinear latent features, and a sparse autoencoder (SAE) [Huben *et al.*, 2023] to decompose this latent polysemantic feature space into monosemantic features (cf. Figure 1, A). The monosemantic features are in a latent space with unknown (but unique) semantics. To assign meaning to each feature, we learn rules that describe the subset of the training samples that highly activate each feature (cf. Figure 1, B). Finally, we combine the last linear layers into a single linear layer by multiplying their weight matrices (cf. Figure 1, C).

3.1 Architecture

The base blackbox model is an MLP with multiple hidden layers and a softmax output. We denote the penultimate layer as l, and the representations of this layer as h_l. The MLP learns a function $\hat{y} = f(x)$. $f(x)$ can be decomposed into a function g learning the hidden representations h_l and a linear model h predicting the target based on h_l, i.e., $\hat{y} = h(h_l) = h(g(x))$. $g(\cdot)$ is characterized by the parameters θ_g, and $h(\cdot)$ is characterized by the weight matrix W.

We adopt the SAE architecture introduced by Huben *et al.* [2023]. The SAE consists of an input layer, a hidden layer, and an output layer. The weights of the encoder and decoder are shared, and given by the matrices M and M^T, respectively. The encoder has an additional bias vector b, the decoder has not [Huben *et al.*, 2023]. We set the size of the hidden layer as $d_{hid} = R \cdot d_{in}$ with d_{in} being the dimension of the MLP's layer l. R is a hyperparameter that controls the size of the feature

dictionary. The input to the SAE are the hidden representations of the blackbox MLP h_l. The output of the SAE is given by $\hat{h}_l = M^T\Big(ReLU(Mh_l + b)\Big)$. By design, $\hat{h}_l$ is a sparse linear combination of (latent) dictionary features [Huben *et al.*, 2023].

3.2 Training

XNNTab is trained in four steps: i) training of the MLP's representation, ii) training of the SAE, and iii) finetuning of the decision layer, iv) combining linear components.

Step 1: MLP training. The MLP's parameters $\theta_{\text{MLP}}= (\theta_g, W)$ are trained using cross-entropy loss and L1 regularization. The SAE is not present in this step.

Step 2: SAE Training. To learn a dictionary of monosemantic features we follow the training procedure by Huben *et al.* [2023]. The SAE is trained to reconstruct the hidden activations h_l of training samples x. The SAE uses a reconstructing loss and a sparsity loss on its hidden activations h_{SAE} to learn sparse dictionary features.

Step 3: Finetuning the decision layer. We freeze the parameters θ_g of the MLP and θ_{SAE} of the SAE and finetune the weights W in the decision layer with the same loss as in step 1 to make predictions from the reconstructions learned by the SAE $\hat{h}_l$.

Step 4: Aggregating linear components. Both, the decoder part of the SAE and the final decision layer of the MLP are linear layers, and can be combined into one layer. Therefore, we directly connect the hidden layer of the SAE to the output and set the weights of this layer to $W' = WM^T$.[1]

3.3 Learning Representation Semantics

The hidden representations of the SAE represent monosemantic dictionary features in a latent space, but their semantics are unknown. To assign human-understandable semantics to those features, we follow a similar idea as in [Huben *et al.*, 2023] for tabular data instead of text (cf. Figure 1, B). For each dimension in the latent representation of the SAE $j \in \{1, ...d_{hid}\}$, we identify the subset of training samples T_j which highly active this feature, i.e., whose activations are above a threshold t. We then use a rule-based classifier to learn simple decision rules describing this subset.

4 Experiments

Datasets and Evaluation Metrics. We evaluate on two benchmark datasets for structured data, Adult (ADULT) [Vanschoren *et al.*, 2014] and Churn Modelling (CHURN)[2]. ADULT contains 15 features describing approx. 50k instances. CHURN contains 14 features and 10k instances. In both cases, the task is binary classification. For evaluation, we use 5-fold-cross-validation. The training/validation/test proportion of the datasets for each split are 65%/15%/20%. We report accuracy and macro F1 on the test set, averaged across all folds.

Models and Parameters. We manually define the neural architecture for the MLP and SAE components of XNNTab. The base MLP has 3 layers with (100, 64, 32) neurons, the SAE $2\times32 = 64$ (R=2) neurons in the hidden layer. See Appendix A.2 for details on all hyperparameters.

Representation Semantics. To assign human-understandable semantics to the dictionary features, we employ Skope-rules[3]. For each dictionary feature f_j, we retrieve a set T_j of instances for which the activation of neuron j is above the threshold $t = 0.75$. We assign $label = 1$ to instances in T_j and $label = 0$ to the remaining instances. The Skope-rules classifier learns a set of decision rules describing instances of the positive class. We set the precision hyperparameter of Skope-rules to 1 and keep the rest of the hyperparameter to the default settings. For each T_j, we keep the rule with the highest coverage (recall).

Results. Table 1 shows that, on both datasets, XNNTab outperforms interpretable models. On ADULT, XNNTab is comparable to MLP and slightly below the performance of XGBoost and Rando Forest. On CHURN, XNNTab is on par with all baseline blackbox models including XGBoost.

[1] $\hat{h}_l = M^T h_{SAE}$, and $\hat{y} = W\hat{h}_l = WM^T h_{SAE}$

[2] https://www.kaggle.com/datasets/shrutimechlearn/churn-modelling

[3] https://github.com/scikit-learn-contrib/skope-rules. Accessed July 2025

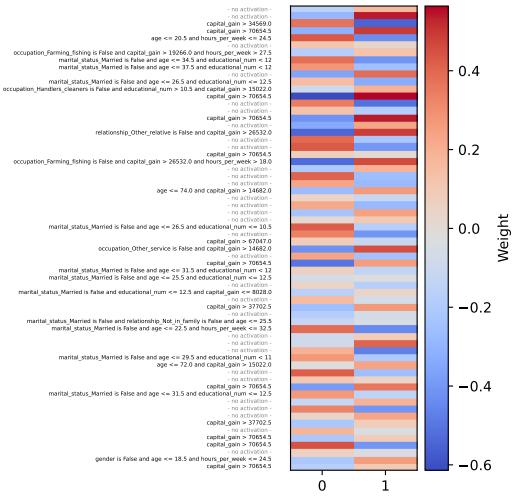
These results show that our model introduces interpretability to DNNs with very little compromise on performance. A selection of rules extracted for the dictionary features on ADULT are shown in Table 2 (full list in Appendix B.1, for CHURN in Appendix B.2). The learned rules provide a global explanation of the model's behavior (Figure 2 left), since the predictions are simply a linear combination of the learned features (rules). Each weight in this linear combination directly indicates whether the presence of the corresponding features increases or lowers the probability of a class, and to which extent. Figure 2 (right) shows that on average 30 out of the 64 features that we learn for ADULT are active above the chosen threshold. That means, one local explanation requires us to inspect 30 rules to understand the model's prediction. The number of active features on average slightly increases for CHURN, but not the complexity of the rules (average rule length remains 2). While the number of active rules is high, several rules are redundant and could be pruned. A pruning mechanism could improve the interpretabiliy of the explanations, and while not in the scope of this paper, it is part of our future work.

Table 1: Performance of blackbox (✕) and fully interpretable (✓) models on two benchmark datasets. Best values are marked bold, best values for interpretable models bold italic.

		ADULT		CHURN	
		F1-Macro	Acc	F1-Macro	Acc
✕	Random Forest	0.799 ± 0.003	0.861 ± 0.002	0.747 ± 0.013	**0.862 ± 0.007**
✕	XGBoost	**0.815 ± 0.002**	**0.869 ± 0.001**	0.750 ± 0.012	0.861 ± 0.007
✕	MLP	0.796 ± 0.002	0.850 ± 0.002	0.751 ± 0.106	0.860 ± 0.006
✓	Logistic Regression	0.782 ± 0.002	0.815 ± 0.002	0.601 ± 0.006	0.808 ± 0.000
✓	Decision Tree	0.787 ± 0.001	0.851 ± 0.002	0.735 ± 0.008	0.849 ± 0.006
✓	XNNTab (ours)	***0.795 ± 0.002***	***0.850 ± 0.002***	**0.752 ± 0.008**	***0.861 ± 0.002***

Table 2: A selection of dictionary features for the ADULT dataset. $|T_j|$ - size of the training subset that strongly activate feature j. Coverage of the rule reported as number of samples and percentage of samples. Table sorted by $|T_j|$.

j	|T_j|	Description	Coverage
40	15294	`marital_status_Married is False and educational_num < 13 and capital_gain <= 8028.0`	11625/0.76
9	11968	`marital_status_Married is False and age <= 37.5 and educational_num < 12`	7271/0.61
8	11634	`marital_status_Married is False and age <= 34.5 and educational_num < 12`	6517/0.56
54	10889	`marital_status_Married is False and age <= 31.5 and educational_num < 13`	5953/0.55
44	7457	`marital_status_Married is False and relationship_Not_in_family is False and age <= 25.5`	3330/0.45
45	3975	`marital_status_Married is False and age <= 22.5 and hours_per_week <= 32.5`	1516/0.38
5	1758	`age <= 20.5 and hours_per_week <= 24.5`	754/0.43
34	1658	`occupation_Other_service is False and capital_gain > 14682.0`	547/0.33
26	1358	`marital_status_Widowedand is False and capital_gain > 14682.0`	546/0.40
50	982	`age <= 72.0 and capital_gain > 15022.0`	532/0.55
63	978	`gender is Female and age <= 18.5 and hours_per_week <= 24.5`	236/0.24
7	290	`occupation_Farming_fishing is False and capital_gain > 19266.0 and hours_per_week > 27.5`	209/0.73
18	226	`relationship_Other_relative is False and capital_gain > 26532.0`	185/0.82
22	225	`occupation_Farming_fishing is False and capital_gain > 26532.0 and hours_per_week > 18.0`	183/0.82
3	212	`capital_gain > 34569.0`	153/0.73



	ADULT	CHURN
Rule Length		
min	1	1
avg	2.18	1.8
max	3	2
Active Features		
min	23	19
avg	30	33
max	37	40

Figure 2: Left: Decision weight matrix W' for ADULT on interpretable features. Right: Statistics on complexity of features on ADULT and CHURN.

Impact of activation threshold t. Figure 3 shows the impact of t on the number of rules that can be extracted per active feature and on the number of instances covered by each rule (recall). The average recall increases with an increasing threshold, as the sets of instances by which features get activated become more focused. This makes it easier for Skope-rules to find a rule that explains most of the instances in the set. However, a focused set limits the data instances a rule could apply to. For some of these sets, Skope-rules fails to extract meaningful rules, i.e., the sets of instances that are activated by extracted rules become empty. In consequence, the amount of active features for which rules can be extracted decrease with increasing threshold t. Choosing the right threshold requires balancing the trade-off between the average rule recall and the fraction of extracted rules per active features.

5 Conclusion

We introduced XNNTab, an interpretable model for tabular data. Our results show, that XNNTab outperforms interpretable models and performs comparably to blackbox architectures. In future work, we plan to investigate pruning of redundant rules and automated methods to identify activation thresholds for an optimal balance between rule recall and fraction of extracted rules.

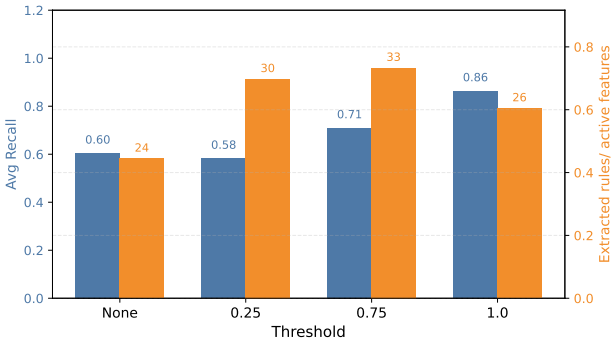


Figure 3: Fraction of features and instances covered by rules to explain ADULT at different feature activation thresholds t. **Orange bars** show the average fraction of rules extracted per active feature with the number of active features on top, **blue bars** show the average recall of the rules in terms of instances.

Funding Declaration

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under project ID 536124560.

References

Sercan Ö. Arik and Tomas Pfister. TabNet: Attentive Interpretable Tabular Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):6679–6687, May 2021.

Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, San Francisco California USA, August 2016. ACM.

Jintai Chen, Jiahuan Yan, Qiyuan Chen, Danny Z. Chen, Jian Wu, and Jimeng Sun. Can a Deep Learning Model be a *Sure Bet* for Tabular Prediction? In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 288–296, Barcelona Spain, August 2024. ACM.

Khawla Elhadri, Tomasz Michalski, Adam Wróbel, Jörg Schlötterer, Bartosz Zieliński, and Christin Seifert. This looks like what? challenges and future research directions for part-prototype models, 2025.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc.

Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. TabR: Tabular Deep Learning Meets Nearest Neighbors. In *The Twelfth International Conference on Learning Representations*, October 2023.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *The Thirteenth International Conference on Learning Representations*, October 2024.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 507–520, Red Hook, NY, USA, November 2022. Curran Associates Inc.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. TabTransformer: Tabular Data Modeling Using Contextual Embeddings, December 2020.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse Autoencoders Find Highly Interpretable Features in Language Models. In *The Twelfth International Conference on Learning Representations*, October 2023.

Alan Jeffares, Tennison Liu, Jonathan Crabbé, Fergus Imrie, and Mihaela van der Schaar. TANGOS: Regularizing Tabular Neural Networks through Gradient Orthogonalization and Specialization. In *The Eleventh International Conference on Learning Representations*, September 2023.

Xiangjian Jiang, Andrei Margeloiu, Nikola Simidjievski, and Mateja Jamnik. ProtoGate: Prototype-based neural networks with global-to-local feature selection for tabular biomedical data. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 21844–21878, Vienna, Austria, July 2024. JMLR.org.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in Vision: A Survey. *ACM Comput. Surv.*, 54(10s):200:1–200:41, September 2022.

Kyungeun Lee, Ye Seul Sim, Hye-Seung Cho, Moonjung Eo, Suhee Yoon, Sanghyu Yoon, and Woohyung Lim. Binning as a pretext task: Improving self-supervised learning in tabular domains. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 26929–26947, Vienna, Austria, July 2024. JMLR.org.

Gesina Schwalbe and Bettina Finzel. A comprehensive taxonomy for explainable artificial intelligence: A systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, 2023.

Jacob Si, Wendy Yusi Cheng, Michael Cooper, and Rahul G. Krishnan. InterpreTabNet: Distilling predictive signals from tabular data by salient feature interpretation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 45353–45405, Vienna, Austria, July 2024. JMLR.org.

Gowthami Somepalli, Avi Schwarzschild, Micah Goldblum, C. Bayan Bruss, and Tom Goldstein. SAINT: Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training. In *NeurIPS 2022 First Table Representation Workshop*, October 2022.

Joaquin Vanschoren, Jan N. van Rijn, Bernd Bischl, and Luis Torgo. Openml: networked science in machine learning. *SIGKDD Explor. Newsl.*, 15(2):49–60, June 2014.

Jiahuan Yan, Jintai Chen, Yixuan Wu, Danny Z. Chen, and Jian Wu. T2G-FORMER: Organizing tabular features into relation graphs promotes heterogeneous feature interaction. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, volume 37 of *AAAI'23/IAAI'23/EAAI'23*, pages 10720–10728. AAAI Press, February 2023.

Zeyu Yun, Yubei Chen, Bruno Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. In Eneko Agirre, Marianna Apidianaki, and Ivan Vulić, editors, *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 1–10, Online, June 2021. Association for Computational Linguistics.