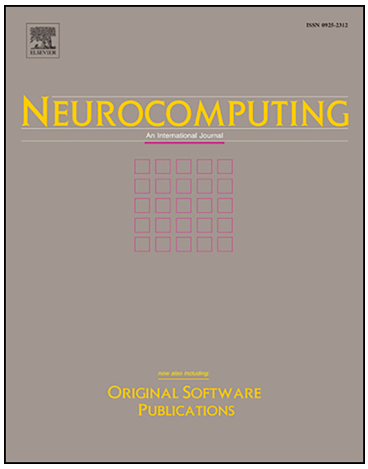


Tabular implicit deep neural networks ensembles for the prediction of pathogenic genetic variants in mendelian diseases

F. Stacchietti, M. Nicolini, L. Chimirri, P.N. Robinson, E. Casiraghi, G. Valentini

PII: S0925-2312(26)02080-1
DOI: https://doi.org/10.1016/j.neucom.2026.134682
Reference: NEUCOM 134682

To appear in: *Neurocomputing*

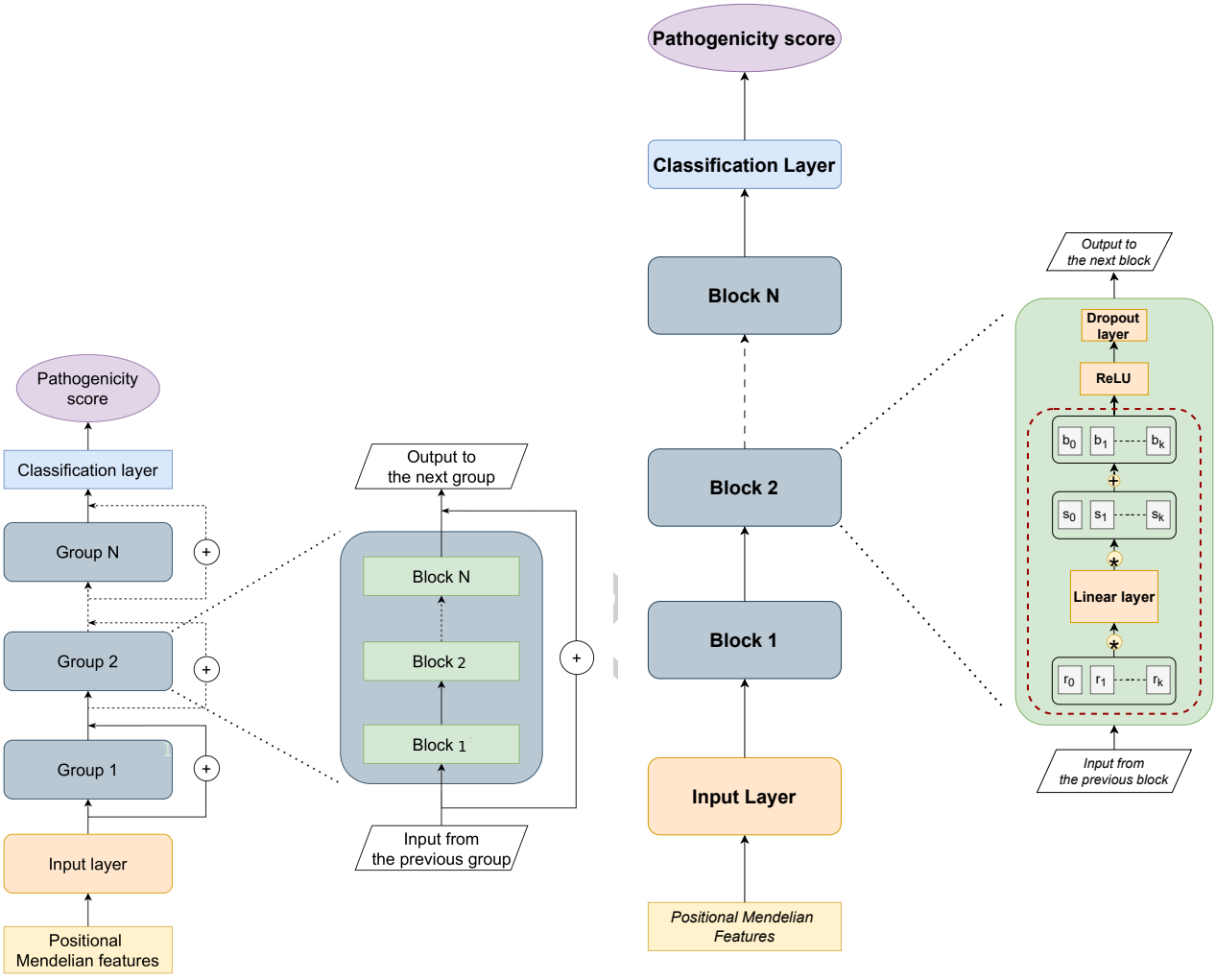
Received Date: 23 January 2026
Revised Date: 17 June 2026
Accepted Date: 29 July 2026

Please cite this article as: Stacchietti F, Nicolini M, Chimirri L, Robinson PN, Casiraghi E, Valentini G, Tabular implicit deep neural networks ensembles for the prediction of pathogenic genetic variants in mendelian diseases, *Neurocomputing* (2026), doi: https://doi.org/10.1016/j.neucom.2026.134682.

Graphical Abstract

Tabular implicit deep neural networks ensembles for the prediction of pathogenic genetic variants in Mendelian diseases

F. Stacchietti, M. Nicolini, L. Chimirri, P.N. Robinson, E.Casiraghi, G. Valentini



Left: *T-ResNet and Right:* TIDE-Var deep neural models for the prediction of pathogenic genetic variants in the non coding human genome.

Highlights

Tabular implicit deep neural networks ensembles for the prediction of pathogenic genetic variants in Mendelian diseases

F. Stacchietti, M. Nicolini, L. Chimirri, P.N. Robinson, E.Casiraghi, G. Valentini

- We propose two modular deep neural network models that outperform state of the art models and achieve comparable results with XGBoost for the prediction of pathogenic genetic variants in non coding regions of the human genome, a challenging problem from both a machine learning and a precision medicine perspective.
- Our first model (*ResNet*) uses a modular residual architecture to simplify model selection and tuning, while residual connections mitigate vanishing gradients.
- Our second model (*TIDE-Var*) features an ensemble whose base learners are jointly trained with shared parameters, while adapter matrices induce base-learner diversity, improving accuracy and at the same time reducing the empirical space and time complexity.
- Ablation experiments isolate the contribution of key design choices and clarify the models' learning behaviour.

Tabular implicit deep neural networks ensembles for the prediction of pathogenic genetic variants in Mendelian diseases⋆

F. Stacchietti[a,*], M. Nicolini[a], L. Chimirri[b], P.N. Robinson[b], E.Casiraghi[a,c,d] and G. Valentini[a,c,**]

[a]AnacletoLab - Diparitmento di Informatica, Universitá degli Studi di Milano, Via Celoria 18, Milano, 20133, Italy
[b]Berlin Institute of Health at Charité – Universitätsmedizin Berlin, 10117 Berlin, Germany
[c]ELLIS - European Lab for Learning and Intelligent Systems, Milan Unit, Via Celoria 18, Milan, 20133, Italy
[d]Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory, Via Celoria 18, Berkeley, CA, United States

ARTICLE INFO

Keywords:
Deep modular neural networks
Ensembles of neural networks
Pathogenic variant prediction in non-coding genome
Mendelian genetic diseases

ABSTRACT

Mendelian genetic diseases comprise approximately 10,000 described disorders, yet the genetic basis is known only for about half of them, and a molecular diagnosis often remains difficult or unresolved. In this context, machine learning methods play a key role. However, identifying pathogenic variants in non-coding regions of the human genome is particularly challenging, as they are vastly outnumbered by neutral variants. This extreme imbalance causes standard machine learning approaches to exhibit a strong predictive bias towards the majority class, significantly limiting their sensitivity.

Building on recent advances in imbalance-aware and ensemble learning methods, we propose two novel deep learning models for predicting pathogenic non-coding variants in Mendelian diseases.

The first model, *T-ResNet* (Tabular Residual Neural Network), adopts a modular architecture with residual connections, along with a mini-batch balancing strategy to address class imbalance. This design simplifies hyperparameter optimization while mitigating vanishing-gradient effects. The second model, *TIDE-Var* (Tabular Implicit Deep neural network Ensembles for Variant prediction), leverages the TabM and BatchEnsemble models to build an implicit ensemble of deep neural networks trained jointly by minimizing a common objective function, and partially sharing learning parameters. Learner-specific adapter parameters promote diversity among the implicit base learners while keeping the ensemble computationally efficient.

Genome-wide experiments show that *T-ResNet* achieves an average Area Under the Precision-Recall Curve (AUPRC) of 0.630 ± 0.033, comparable to that of *HyperSMURF*, a state-of-the-art method. In contrast, *TIDE-Var* yields significantly better results than *HyperSMURF* (AUPRC $= 0.714 \pm 0.025$) and slightly better than *XGBoost*, one of the top-methods for the classification of tabular data. Ablation studies confirm that regularized joint ensemble learning with partially shared and base-learner specific adapter learning parameters are key factors in achieving high predictive performance, essential to improve the diagnostic yield for patients with rare genetic diseases.

1. Introduction

The integration of machine learning, large-scale population genome sequencing projects, and modern whole-genome sequencing technologies [1, 2] plays a central role in personalized medicine, particularly for detecting rare and common variants associated with genetic diseases and cancer [3, 4]. Indeed, identifying pathogenic genetic variants underlying hereditary disorders remains a fundamental objective of precision medicine [5]. Despite recent advances, molecular diagnoses remain elusive for 50% to 80% of patients presenting to genetic clinics [6], and only about half of the 10,000 rare Mendelian diseases have an established phenotype-causing mutation [7, 8].

While protein-coding variants have been extensively characterized [9, 10], pathogenic variants located in non-coding genomic regions, although strongly implicated in Mendelian disorders, remain comparatively poorly understood [11]. Genome-wide association studies have shown that single-nucleotide variants (SNVs) associated with Mendelian diseases map predominantly to non-coding regions, including promoters, enhancers, untranslated regions (UTRs), and RNA genes [12, 13].

Given the importance of detecting pathogenic variants across the human genome, several machine learning methods have been developed for this task, typically leveraging conservation, genetic, and epigenetic features associated with each variant [14]. One of the first widely adopted approaches, CADD, integrated diverse annotations into a unified score using an ensemble of support vector machines [15], while the current version incorporates a convolutional neural network and a protein language model to evaluate the deleteriousness of variants genome-wide [16]. Other relevant machine learning efforts include multiple-kernel integrative approaches [17], linear models coupled with probabilistic models of molecular evolution [18], and unsupervised learning strategies designed to compensate for insufficient annotations [19]. Deep learning approaches have also been applied, though primarily for predicting

⋆This document is the results of the research project funded by the CN3 NextGenerationEU

federico.stacchietti@unimi.it (F. Stacchietti); leonardo.chimirri@bih-charite.de (L. Chimirri); peter.robinson@bih-charite.de (P.N. Robinson); elena.casiraghi@unimi.it (E.Casiraghi); valentini@di.unimi.it (G. Valentini)
https://casiraghi.di.unimi.it (E.Casiraghi); https://valentini.di.unimi.it (G. Valentini)
ORCID(s): 0000-0001-0000-0000 (F. Stacchietti); 0000-0001-0000-0000 (M. Nicolini); 0000-0002-6912-8518 (L. Chimirri); 0000-0001-0000-0000 (P.N. Robinson); 0000-0003-2024-7572 (E.Casiraghi); 0000-0002-5694-3919 (G. Valentini)

pathogenic variants in coding regions [20] or more generally for identifying deleterious variants without focusing specifically on non-coding variants involved in Mendelian diseases [21]. Related efforts such as deep learning and gkm-SVM models aim to predict the regulatory impact of sequence variation [22, 23, 24, 25].

A major challenge in pathogenic variant prediction, especially in the non-coding genome, is the extreme imbalance between pathogenic and neutral variants. Only a few hundred pathogenic variants are available for training, compared with millions of neutral variants distributed throughout the genome [26]. Such severe imbalance significantly hampers machine learning performance, as conventional algorithms tend to favor the predominant negative neutral class [27].

To address this issue, several studies have demonstrated the benefit of imbalance-aware ensemble strategies for improving prediction performance [28, 29]. State-of-the-art results for non-coding pathogenic variant prediction have been achieved by hyperSMURF [30], which combines undersampling of the abundant neutral variants with oversampling of the minority pathogenic class, and employs hyper-ensembles of random forests to enhance predictive accuracy. HyperSMURF also serves as the machine-learning core of Genomiser, a leading diagnostic tool for Mendelian disorders [26]. Another method based on XGBoost [31] showed the effectiveness of ensemble methods in the prediction of pathogenic non coding variants [32]. The centrality of ensemble methods is underscored not only in non-coding variant prediction, but also in coding-variant predictors such as Maverick, an ensemble of eight neural networks which uses a transformer as its primary building block [33].

Motivated by the advances introduced through imbalance-aware methods and ensemble learning, this work investigates whether deep neural architectures specifically designed for imbalanced tabular data can achieve competitive results for the prediction of pathogenic variants in the non-coding human genome.

The genomic, epigenomic, and conservation features used for this task form inherently tabular datasets, which pose unique challenges due to feature heterogeneity, limited positive examples, and extreme class imbalance. While traditional models such as random forests are naturally suited to heterogeneous tabular data, recent advances have highlighted the potential of deep learning architectures tailored for tabular domains [34].

Building on these developments, we introduce *T-ResNet*, a modular tabular deep neural network (DNN) architecture composed of repeated block-shaped modules having the same layer dimensionality. This modularity enables efficient hierarchical hyperparameter tuning and architectural optimization. To mitigate class imbalance, we incorporate specialized mini-batch balancing strategies during training.

Furthermore, inspired by the recently proposed TabM model [35] and BatchEnsemble models [36], we present *TIDE-Var*, an ensemble of modular neural networks that share subsets of their parameters. This parameter sharing substantially reduces the computational footprint of the ensemble while maintaining strong predictive performance by jointly minimizing the same cross-entropy objective across the base neural networks.

A genome-wide empirical evaluation shows that *TIDE-Var* is competitive with state-of-the-art methods, while ablation studies reveal its main learning characteristics.

T-ResNet preliminary results have been presented in a recent international conference [37]. In this paper we extended the experimental comparison of *T-ResNet* and introduced *TIDE-Var*, a new method based on TabM for pathogenic variant prediction.

2. Methods

We designed two deep "block-shaped" neural models for their relatively simplicity and modularity that make them easily tunable with standard model selection procedures.

The first model, *T-ResNet*, extends the residual-network paradigm originally developed for convolutional neural networks [38] to tabular genomic data. T-ResNet is composed of groups of fully connected blocks linked by residual connections, and uses mini-batch balancing to improve learning of pathogenic variants, which are strongly underrepresented in the available data.

The second model, *TIDE-Var*, builds on the recently proposed TabM model [35]. It trains an implicit ensemble of deep neural networks that share their weight parameters, while learner-specific adapter parameters introduce diversity among them. The main advantage of this approach is that multiple neural predictors can be optimized jointly within a single parameter-efficient architecture, reducing the computational cost of conventional deep ensembles and optimizing the ensemble as a whole rather than each base learner independently.

Both models process a tabular dataset $D = \{\mathbf{x}^i, y^i\}_{i=1}^n$ of genomic variants, where n denotes the sample size, $\mathbf{x}^i \in \mathbb{R}^m$ represents an m-dimensional feature vector of genomic attributes, and $y^i \in \{0, 1\}$ indicates variant pathogenicity. All features are standardized before being provided as input to the models.

2.1. *T-ResNet*: Tabular Residual Network

T-ResNet employs a streamlined modular architecture with residual connections to mitigate vanishing-gradient effects in deeper architectures and a small number of hyperparameters to simplify model selection.

Given an input vector $\mathbf{x} \in \mathbb{R}^m$ representing the genomic features associated with a variant, the input layer first maps $\mathbf{x}$ to a hidden representation $\mathbf{z}_0 \in \mathbb{R}^d$ through a linear transformation:

$$\mathbf{z}_0 = Input\_layer(\mathbf{x}) = \mathbf{x}^T \mathbf{W}_{\text{in}} + \mathbf{b}_{\text{in}}, \tag{1}$$

where $\mathbf{W}_{\text{in}} \in \mathbb{R}^{m \times d}$ and $\mathbf{b}_{\text{in}} \in \mathbb{R}^d$ are learned parameters.

The network is composed of repeated *blocks*, each consisting of a linear layer followed by a ReLU activation, which introduces non-linearity, and dropout regularization. Each

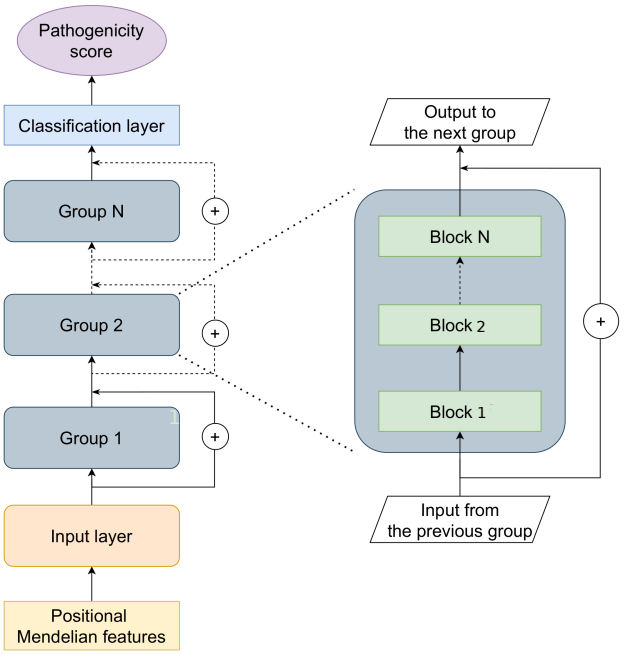


Figure 1: High-level scheme of the *T-ResNet model*.

block maps an input vector $\mathbf{z} \in \mathbb{R}^d$ to a new representation in $\mathbb{R}^d$:

$$Block(z) = Dropout\Big(ReLU\left(\mathbf{z}^T \mathbf{W}_{\text{block}} + \mathbf{b}_{\text{block}}\right)\Big), \quad (2)$$

where $\mathbf{W}_{\text{block}} \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_{\text{block}} \in \mathbb{R}^d$. All blocks have the same hidden dimension and dropout rate (Fig. 1).

A fixed number of sequential blocks is then collected into a higher-level module, called a *group*, which includes a residual connection to mitigate vanishing-gradient effects. If $F_{\text{group}}(\mathbf{z})$ denotes the output obtained by applying all blocks in the group to an input vector $\mathbf{z} \in \mathbb{R}^d$, the residual connection computes:

$$Group(\mathbf{z}) = F_{\text{group}}(\mathbf{z}) + \mathbf{z}. \quad (3)$$

In other words, the input of each group is added to the output of the sequence of blocks within that group. This design preserves the modular structure of the network and provides direct pathways for gradient propagation, helping to mitigate vanishing-gradient effects.

The complete *T-ResNet* architecture (Fig. 1) is obtained by sequentially applying N groups, all composed of the same number of blocks with constant hidden dimension d. If $\mathbf{z}_{\ell-1}$ is the hidden representation produced by the $(\ell-1)$-th group, then the hidden representation computed by the ℓ-th group is:

$$\mathbf{z}_\ell = Group_\ell(\mathbf{z}_{\ell-1}), \qquad \ell = 1, \dots, N. \quad (4)$$

The output $\mathbf{z}_N$ of the N-th group is then passed to an output layer, implemented as a classification head consisting of a single linear layer that maps $\mathbb{R}^d$ to $\mathbb{R}^2$. This layer produces two logits, one per class, which are converted into class probabilities through a softmax function.

Keeping the hidden dimension d constant across all blocks and using the same number of blocks per group ensures architectural modularity and reduces the number of architectural hyperparameters for model selection. Fig. 1 illustrates the overall architecture, where residual connections span entire groups. For example, an architecture with 4 groups and 3 blocks per group contains 12 blocks and 4 residual connections.

2.2. *TIDE-Var: Tabular Implicit Deep neural network Ensembles for VARiant prediction*

Ensembles of learning machines [39] have been successfully applied to the prediction of pathogenic variants [40, 30], and ensembles of deep neural networks have also obtained competitive results on different types of tabular data [41, 42].

The main drawback of deep neural ensembles lies in their computational complexity, since training is repeated for each base neural network in the ensemble. Moreover, independently optimizing each base learner may yield strong individual models, but does not explicitly optimize the ensemble as a whole.

To overcome these limitations, we introduce the **T**abular **I**mplicit **D**eep neural network **E**nsemble for **Var**iant prediction (*TIDE-Var*). TIDE-Var is a deep ensemble architecture that leverages the parameter-efficient BatchEnsemble framework [36]. It trains an ensemble of deep neural networks implicitly by sharing the main weight matrices across base learners, thereby reducing training costs and enabling joint optimization of the ensemble parameters. Diversity among the base learners is introduced through *adapter matrices* [35], which assign its own additional parameters to each implicit base learner, while keeping the main weight matrices shared across the ensemble.

In this way, TIDE-Var captures the two key principles of ensemble learning [39]: (i) accuracy, since the implicit base learners are trained together by minimizing the same loss function, so that the model is optimized for the ensemble prediction rather than for independently trained predictors; and (ii) diversity, introduced by the adapter matrices, which assign different specific parameters to each implicit base learner.

In more detail, given an input sample $\mathbf{x} = (x_1, \dots, x_m) \in \mathbb{R}^m$ representing a variant with m numerical feature values x_i, $i = 1, \dots, m$, TIDE-Var first embeds the numerical features and replicates the resulting embedded representation across k parallel streams, one for each implicit base learner. The replicated representation is then processed by a sequence of BatchEnsemble-based blocks and finally mapped to k predictions that are aggregated by averaging.

The architecture, schematized in Fig. 2, is composed of four main components: (1) An input embedding layer, which maps the input sample to an embedded representation replicated across the k implicit learners; (2) BatchEnsemble layers, which share the main weight matrices while

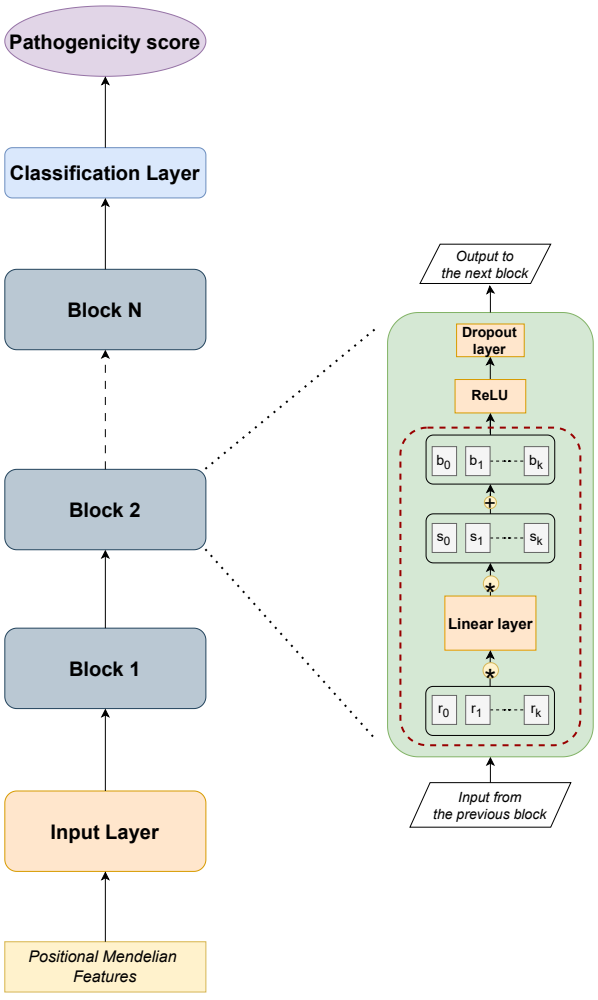


Figure 2: High-level scheme of the *TIDE-Var* model. Left: overall modular architecture. Right: structure of a single block. The dotted red line encloses the BatchEnsemble layer, and the asterisk denotes pointwise multiplication. The main linear transformation is shared across the k implicit base learners, while learner-specific adapter parameters introduce diversity.

using learner-specific adapter parameters; (3) Stacked non-linear blocks built from BatchEnsemble layers, ReLU activations, and dropout; and (4) Learner-specific classification heads followed by ensemble aggregation. Each module is described below.

1. Input embedding layer. The input layer first embeds numerical features using either piecewise linear encodings (PLE) based on binning or periodic activation functions [43]. These embedding strategies have been shown to improve the predictive performance of deep learning models for tabular data [44].

In more detail, given the input sample $\mathbf{x} = (x_1, \dots, x_m)$, let ϕ denote the selected numerical embedding operator, either PLE or periodic activation embedding. The operator ϕ embeds each numerical feature x_i into a latent representation $\phi(x_i) \in \mathbb{R}^{d_i}$, with $d_i \geq 1$. The concatenation of all feature embeddings produces the embedded input representation:

$$\mathbf{z}_0 = \phi(x_1) \| \cdots \| \phi(x_m) \in \mathbb{R}^d, \qquad d = \sum_{i=1}^{m} d_i, \tag{5}$$

where $\|$ denotes vector concatenation.

Given k implicit learners, TIDE-Var replicates the embedded representation k times, storing the replicas in the rows of the matrix $\mathbf{Z}_0$:

$$\mathbf{Z}_0 = Input\_layer(\mathbf{x}) = \begin{bmatrix} \mathbf{z}_0 \\ \vdots \\ \mathbf{z}_0 \end{bmatrix} \in \mathbb{R}^{k \times d}. \tag{6}$$

Each row of $\mathbf{Z}_0$ corresponds to one implicit base learner and is given as input to the first BatchEnsemble block.

2. BatchEnsemble layer. Given a hidden representation $\mathbf{Z} \in \mathbb{R}^{k \times d}$, the BatchEnsemble layer computes:

$$BE(\mathbf{Z}) = \mathbf{S} \odot \big((\mathbf{Z} \odot \mathbf{R})\mathbf{W}\big) + \mathbf{B}, \tag{7}$$

where $\odot$ denotes pointwise multiplication, $\mathbf{W} \in \mathbb{R}^{d \times d}$ is the weight matrix shared across all k base learners, and $\mathbf{R}, \mathbf{S}, \mathbf{B} \in \mathbb{R}^{k \times d}$ are learner-specific adapter matrices.

In more detail, for the i-th implicit base learner, the transformation is:

$$BE(\mathbf{z}_i) = \mathbf{s}_i \odot \big((\mathbf{z}_i \odot \mathbf{r}_i)\mathbf{W}\big) + \mathbf{b}_i, \tag{8}$$

where $\mathbf{z}_i$, $\mathbf{r}_i$, $\mathbf{s}_i$, and $\mathbf{b}_i \in \mathbb{R}^d$ denote the i-th rows of $\mathbf{Z}$, $\mathbf{R}$, $\mathbf{S}$, and of the bias matrix $\mathbf{B}$, respectively. Thus, all base learners share the same matrix $\mathbf{W}$, while their computations are differentiated by the learner-specific adapter parameters.

Following TabM [35], we use a specific initialization scheme for the multiplicative adapters $\mathbf{R}$ and $\mathbf{S}$: in the first BatchEnsemble layer, they are initialized with random ± 1 values, whereas in subsequent layers all their entries are initialized to 1. The additive adapter $\mathbf{B}$ is treated as a trainable learner-specific bias and optimized by backpropagation together with the shared matrix $\mathbf{W}$ and the multiplicative adapters $\mathbf{R}$ and $\mathbf{S}$.

3. Block structure. A TIDE-Var block applies a BatchEnsemble layer followed by a ReLU activation and dropout regularization:

$$Block(\mathbf{Z}) = Dropout\Big(\mathrm{ReLU}\big(BE(\mathbf{Z})\big)\Big). \tag{9}$$

The BatchEnsemble blocks are stacked sequentially. If $\mathbf{Z}_{\ell-1}$ is the hidden representation computed after the $(\ell-1)$-th block, then the output of the ℓ-th block is:

$$\mathbf{Z}_\ell = Block_\ell(\mathbf{Z}_{\ell-1}), \qquad \ell = 1, \dots, N, \tag{10}$$

where N is the number of stacked TIDE-Var blocks. The final hidden representation is therefore $\mathbf{Z}_N \in \mathbb{R}^{k \times d}$.

4. Classification layer and ensemble aggregation. The final hidden representation $\mathbf{Z}_N$ is passed to k learner-specific linear classification heads.

For each implicit base learner i, the corresponding classifier learns a linear transformation:

$$\mathbf{Y}_i = \mathbf{z}_{N,i}\mathbf{W}_i^{out} + \mathbf{b}_i^{out}, \qquad \mathbf{W}_i^{out} \in \mathbb{R}^{d\times c}, \quad \mathbf{b}_i^{out} \in \mathbb{R}^c, \tag{11}$$

where $\mathbf{z}_{N,i}$ is the i-th row of $\mathbf{Z}_N$, $\mathbf{W}_i^{out}$ and $\mathbf{b}_i^{out}$ are, respectively, the weight matrix and the bias term individually learned by each base learner and $\mathbf{Y}_i = (Y_{i1}, \dots, Y_{ic}) \in \mathbb{R}^c$ contains the c logits for the i-th base learner, where c is the number of classes. In our setting the two classes represent pathogenic vs neutral (non-pathogenic) classes.

Next, a softmax independently normalizes each of the c logits of each base learner:

$$\hat{Y}_{ij} = \mathrm{softmax}\big(\mathbf{z}_{N,i}\mathbf{W}_i^{out} + \mathbf{b}_i^{out}\big)_j = \frac{\exp(Y_{ij})}{\sum_{p=1}^{c}\exp(Y_{ip})}, \tag{12}$$
$$j = 1, \dots, c, \quad i = 1, \dots, k.$$

The ensemble prediction is then obtained by averaging the class probabilities across the k implicit base learners:

$$P(C = j) = \frac{1}{k}\sum_{i=1}^{k}\hat{Y}_{ij}. \tag{13}$$

The predicted class is then:

$$\hat{C} = \arg\max_j P(C = j). \tag{14}$$

Overall, the TIDE-Var computation can be summarized as:

$$\mathbf{Z}_0 = Input\_layer(\mathbf{x}), \tag{15}$$
$$\mathbf{Z}_\ell = Block_\ell(\mathbf{Z}_{\ell-1}), \qquad \ell = 1, \dots, N, \tag{16}$$
$$P(C = j) = \frac{1}{k}\sum_{i=1}^{k}\mathrm{softmax}\big(\mathbf{z}_{N,i}\mathbf{W}_i^{out} + \mathbf{b}_i^{out}\big)_j. \tag{17}$$

All *TIDE-Var* parameters are learned by minimizing the cross-entropy objective function.

In summary, *TIDE-Var* adapts the parameter-sharing mechanism of BatchEnsemble and the TabM strategy of training multiple implicit predictors within a single model to pathogenic non-coding variant prediction. In this architecture, the shared matrix $\mathbf{W}$ reduces the computational and memory cost of the ensemble, while the learner-specific adapter matrices $\mathbf{R}$, $\mathbf{S}$, and $\mathbf{B}$ differentiate the implicit base learners.

Building on these efficient components, *TIDE-Var* introduces a modular block-based architecture for tabular variant features, in which BatchEnsemble layers are combined with nonlinear activations and dropout regularization. This design promotes expressive nonlinear feature transformations and improves robustness against overfitting. Finally, ensemble-level inference is performed by averaging the predictions of all implicit base learners, yielding a single pathogenicity score for each variant.

2.3. Addressing class imbalance with large batches and mini-batch balancing

To address the extreme class imbalance of the dataset, with a negative-to-positive ratio of approximately 10^4, we considered two complementary training strategies. On one hand, we used large mini-batches, which increase the probability that each batch contains at least one positive example. On the other hand, we implemented an imbalance-aware mini-batch sampling strategy, in which the proportion of positive and negative examples in each batch is explicitly controlled.

In the mini-batch balancing strategy, each mini-batch B of size b is constructed by:

- sampling b_+ positive examples *with replacement* from the set of positive variants;
- sampling b_- negative examples *without replacement* from the set of negative variants;

with $b = b_+ + b_-$.

The mini-batch composition is therefore controlled by the negative-to-positive ratio in the mini-batch: $r = \frac{b_-}{b_+}$.

Thus, $r = 1$ corresponds to a balanced mini-batch with the same number of positive and negative examples, whereas larger values of r increase the fraction of negative examples in the batch. Sampling positives with replacement compensates for their limited number, while sampling negatives without replacement promotes broader coverage of the majority class across training. The rationale behind these approaches is to assure that the neural networks can learn from at least some positives sampled in each mini-batch. Indeed if we randomly extract examples according to a uniform distribution from very unbalanced data having small mini-batches, it is very likely that in most mini-batches only negative examples will be sampled, thus biasing the learning process toward negative examples.

2.4. State-of-the-art models

We compared the proposed models to three state-of-the-art models, *hyperSMURF* [30], *XGBoost [32]* and *Combined Annotation Dependent Depletion* (CADD) v1.7 [45].

We chose these methods because CADD is arguably the most widely used and well-known tool for predicting deleterious variants [45]; *hyperSMURF* has reported leading performance relative to other state-of-the-art methods, including Eigen [46], GWAVA [40], and DeepSea [47]; *XGBoost* has been shown to outperform deep learning methods on tabular data [34] and its potential was indeed confirmed by its successful application to the prediction of pathogenic genetic variants [32].

HyperSMURF [30] has been specifically designed to target deleterious non-coding variants under extreme class imbalance by employing a hyper-ensemble of Random Forests (RF). Beyond leveraging an RF ensemble as base learner to enhance the model's generalization capability, hyperSMURF manages high imbalance by partitioning the

dataset, and then training each base RF of the ensemble on a subsampled version of the training data. In these subsets, the abundance of neutral ("negative") variants is reduced through undersampling, while the representation of pathogenic "positive" variants is increased via oversampling techniques. Today, hyperSMURF constitutes the machine learning core module of Genomiser, a widely adopted [48] state-of-the-art tool for the diagnosis of Mendelian genetic diseases [26].

XGBoost [31] is a regularized gradient-boosted tree ensemble, where the base learners are trained to focus their attention on the most difficult examples to learn, thus improving the generalization abilities of the overall ensemble. The method represents the state-of-the-art for the classification of data in tabular format; it mostly outperforms deep learning models on supervised learning tasks, as shown by recent extensive experimental results [34].

CADD [45] has been designed to unbiasedly score deleteriousness of variants in a genome-wide fashion, i.e. for the whole variety of genomic regions and functions across the 3 billion bases making up the human genome. It uses a machine learning model trained on a binary distinction between variants that have arisen and become fixed in human populations since the split between humans and chimpanzees and simulated de-novo variants. The rationale is that the former variants should be overwhelmingly neutral (or, at most, weakly deleterious) by virtue of having survived millions of years of purifying selection, while the latter are free of selective pressure and may thus include both neutral and deleterious alleles. The current CADD version (v. 1.7) is implemented by combining a convolutional neural network and a protein language model [16].

3. Results

3.1. Dataset

We trained and tested our neural models on the dataset published in [26]; it contains SNVs and small (less than 50 nucleotides) insertions/deletions in the non coding regions of the human genome. The dataset comprises $n > 14$ million variants; among them only $n_+ = 406$ are pathogenic, thus resulting in an imbalance between pathogenic (positive) and physiological (negative) variants of about $1 : 35,000$. Each variant is characterized by 26 numerical features that include genomic attributes downloaded from public data bases (UCSC, NCBI and others). The main features represent conservation scores, transcriptional, regulation and epigenomic features (e.g. DNAse features, histone methylation and acetylation, transcription factor binding sites), and other features relevant to assess the potential pathogenicity of the genetic variants (see [26] for more details).

3.2. Experimental Setting

To ensure a fair comparison and unbiased evaluation across all models, we assessed their performance on ten stratified external training/test holdouts, each defined by an 80:20 train:test split.

To balance hyperparameter optimization with computational efficiency, all models were tuned using a grid search strategy within a cross-validation framework applied to the first single holdout (details in subsection 3.3). The optimal hyperparameter configurations identified on this holdout were then fixed and reused for training and testing on the remaining holdouts.

More precisely, the external training set has been subjected to an internal 5-fold stratified cross-validation, and the best hyperparameter configuration was selected by maximizing the mean area under the precision–recall curve (AUPRC).

3.3. Hyper-parameter optimization

A schematic representation of hyper-parameter search space for each model, together with the hyper-parameters values selected by the optimization procedure are reported in Table 1.

Optimization of T-ResNet hyper-parameters. We optimized T-ResNet in a hierarchical way, according to a procedure described in detail in [37]. First, we run preliminary experiments on a smaller subsample of the training data to identify promising ranges of subspace model–training hyper-parameters.

Next, we used the training splits to run an internal grid search and explored all combinations of architectural hyper-parameters, including the number of groups, the number of blocks per group, and the number of units per layer (Table 1).

The grid search allowed us to restrict the search space to the following optimal architectural parameters: number of groups equal to 3 or 5, number of blocks per group equal to 1 or 3, number of units per hidden layer equal to 128. By using the selected architectural hyper-parameters, we run a new grid search, to set the best values for the batch size and mini-batch balancing computed as the ratio of the number of negatives and positive (b_-/b_+), leading to the final selected values listed in Table 1.

Optimization of TIDE-Var hyper-parameters. We followed a hierarchical hyper-parameter optimization similar to that pursued for T-ResNet. Initially, hyperparameter configuration was performed by subsampling the number of negatives in the external training set. More precisely, we explored increasingly sized datasets (of negatives) of $|N_-| = 10^5$, 10^6, and "full" (all negatives). In this way we preliminarily identified promising regions of the hyperparameter space, and then performed a grid search over the selected ranges on the full training set parameters (Table 1).

By using these selected ranges of hyper-parameter values, we optimized by grid-search the architecture configuration, by choosing:

- the number of implicit base-learners k,
- the input embedding dimension d,

- the feature embedding technique applied before the BatchEnsemble blocks, including: (1) periodic activation-based embeddings [43], (2) Piecewise Linear Embeddings (PLE) [43]. Following [35], we used an updated implementation of PLE and constructed the PLE breakpoints through quantile-based binning, applying the same bin edges across all ensemble learners.

After the embedding layer two blocks consisting of 512 neurons each were used. Models were trained by using a dropout of 0.1, AdamW with learning rate equal to $2 \cdot 10^{-3}$, weight decay of $d_w = 3 \cdot 10^{-4}$ and a batch size of 8192.

Optimization of hyperSMURF hyper-parameters. For hyperSMURF we optimized the following hyper-parameters:

- the number of dataset partitions (np), where each partition includes all positive examples and a unique subset of negatives (i.e., a maximum of $n\_/np$ negatives, if $n\_$ is the number of negative examples).
- The oversampling factor for positive examples (fp): if n_+ is the number of positives, the cardinality of the over-sampled positive set, obtained through the SMOTE algorithm [49], is $N_+ = n_+ + (\text{fp} \times n_+)$.
- The negative to positive ratio (r): this parameter controls the number of negatives in each partition. The number of negatives is $N\_ = min(r \times N_+, n\_/np)$.
- The number of trees ($ntree$) in each random forest.

The hyper-parameter search space (Table 1) was chosen based on the hyper-parameter optimization results reported by authors of [30] using the same dataset. All other hyperSMURF and random forest settings were kept at their defaults.

Optimization of XGBoost hyper-parameters. For XGBoost, we optimized the following hyper-parameters:

- The number of boosting rounds ($n\_estimators$), which controls the number of trees added to the ensemble.
- The learning rate ($learning\_rate$), which controls the contribution of each tree to the final prediction.
- The maximum tree depth (max_depth), which controls the complexity of the individual trees.
- The subsampling ratio ($subsample$), which controls the fraction of training examples used to fit each tree.

The hyper-parameter search space (Table 1) was defined as a grid over standard XGBoost hyper-parameters, including the default configuration and alternative values controlling model capacity, learning dynamics, and regularization. Hyper-parameters were selected using the same internal cross-validation protocol adopted for the other models. All other XGBoost settings were kept at their defaults.

Setting of CADD predictions. Since for CADD the precomputed predictions for the whole human genome are fully available, to carry out the computation, we downloaded the file "All possible SNVs of GRCh37/hg19" from https://cadd.bihealth.org/download, containing 3 entries for each genomic position (totaling about 9 billion entries). We extracted the roughly 14M genomic positions present in our dataset, further selecting the one SNV with the maximum raw score for each genomic position.

3.4. Evaluation of the generalization performance

The AUPRC scores were computed on the ten different test sets according to a multiple hold-out procedure (Figure 3). All models except CADD achieved scores ranging from 0.55 to 0.75 across the ten test sets. *HyperSMURF* and *T-ResNet* showed comparable performance, with each model outperforming the other depending on the test set (Figure 3). In contrast, *TIDE-Var* consistently delivered the best results across the ten test sets. F1-scores confirmed that *TIDE-Var* outperformed all other models, while *T-ResNet* achieved better results than *HyperSMURF* on each test set in the multiple hold-out procedure (Figure 4).

TIDE-Var emerged as the best model, with an average AUPRC equal to 0.714 ± 0.025*, while* HyperSMURF *scored* 0.654 ± 0.039*,* T-ResNet 0.630 ± 0.034 *and* XGBoost 0.709 ± 0.021 *(Figure 5a). The AUPRC difference is statistically significant between* TIDE-Var and the other models, except for *XGBoost* (Wilcoxon signed rank test), and a significant difference can be also detected between *T-ResNet and* HyperSMURF (Figure 5b). Also on the basis of the F1-score, the improvement is always statistically significant in favour of *TIDE-Var* (p-value < 0.01), except with *XGBoost* where the difference in favour of *TIDE-Var* is not statistically significant, and for this metric *T-ResNet* achieves significantly better results than *HyperSMURF* (p-value < 0.05).

CADD performed poorly on genetic Mendelian data. Indeed with the first hold-out CADD obtained an AUPRC ≈ 0.01, two order of magnitude less than *TIDE-Var. For this reason, we did not include its results in Fig. 3, 4 and 5.*

CADD's relatively poor performance, possibly consistent with previous findings [50], likely stems from its design goal of providing an unbiased, genome-wide deleteriousness score. In other words, it is intended to assess the potential impact of variants across the full spectrum of genomic regions and functions spanning the 3 billion bases of the human genome, rather than focusing on the specific subset of pathogenic variants underlying Mendelian genetic diseases.

3.5. Characterization of the predictions and of the learning behavior of the models

We first characterized the learning behaviour of the compared models, then we focused on the characterization of TIDE-Var, the best performing model.

Table 1
Hyperparameter search space and selected values for TRes-Net, TIDE-Var, hyperSMURF and XGBoost.

Model	Hyperparameter	Search space	Selected value
TRes-Net	Number of groups	{3, 5, 7, 9}	5
	Number of blocks per group	{1, 2, 3}	3
	Number of units per layer	{128, 256, 512}	128
	Batch size	{1024, 8192, 32768}	32768
	Mini-batch negative-to-positive ratio ($r = N_-/N_+$)	{1, 4, 19}	19
TIDE-Var	Number of implicit MLP (k)	{8, 12, 16, 24, 32, 48}	48
	Input embedding dimension (d_{emb})	{12, 18, 24, 32, 48}	32
	Encoding Strategy	Periodic activation functions PLE target-aware binning PLE quantile-based binning	PLE quantile-based binning
HyperSMURF	Number of partitions (np)	{50, 100, 150}	50
	Oversampling factor (fp)	{2, 5, 10, 15, 20}	5
	Negative-to-positive ratio (r)	{2, 3, 10, 50, 100}	100
	Number of trees ($ntree$)	{10, 50, 100}	50
XGBoost	Number of estimators	{50, 100, 150}	150
	Learning rate	{0.3, 0.02, 0.01}	0.02
	Maximum tree depth	{3, 6}	6
	Subsampling ratio	{0.8, 1.0}	1.0

3.5.1. Comparison of the learning behaviour of the models

TIDE-Var showed the best results in comparison with T-ResNet and *HyperSMURF (Fig. 5). However, even if* TIDE-Var shows slightly better precision/recall and ROC curves (Fig. 6 a and d), *HyperSMURF and* T-ResNet exhibit better sensitivity than *TIDE-Var at any threshold (Fig. 6c). On the contrary, the lower* TIDE-Var sensitivity is counterbalanced by a very high precision, close to 1 for any threshold higher than 0.6 (Fig. 6b), thus resulting in a significantly larger AUPRC (Fig. 5). This suggests that *TIDE-Var could be preferable in general since variants having a large score are confidently and correctly predicted as pathogenic, but* T-ResNet or *HyperSMURF could be the elective choice in a context where sensitivity is the main concern, e.g in an exploratory phase where the interest is in discovering and collecting potential pathogenic candidates, even at the cost of a few extra false positives.*

Even if TIDE-Var *slightly outperforms* XGBoost*, their precision-recall curves are very close (Fig. 6a), as reflected by the absence of a statistically significant difference in their AUPRC performance (Fig. 5). When investigating their performance at a finer detail, we however notice that, while their precision curves are very close (Fig. 6b), sensitivity of* TIDE-Var *is significantly larger for thresholds larger than* 0.4 *(Fig. 6c).*

The different behaviour of TIDE-Var, T-ResNet, HyperSMURF *and* XGBoost *is outlined in Fig. 7, which reports the histograms of the probabilities of the models on the pathogenic (orange) and neutral (blue) variants. All the methods struggle in correctly predicting a subset of pathogenic variants whose features are very close to those of some negative (neutral) examples, resulting in predicted probabilities close to* 0*.*

When focusing on the distribution for pathogenic variants, both TIDE-Var *and* T-ResNet *exhibit more pronounced and consistent peaks near* 1 *compared with* HyperSMURF*.*

However, the largest difference can be observed with the distribution of the probabilities relative to the negative examples (blue bins): while HyperSMURF presents decreasing but large frequencies across all the predicted probabilities (hence showing high frequencies for probabilities larger than 0.5, Fig. 7 c), *T-ResNet and* TIDE-Var show significantly lower frequencies for large probability values (Fig. 7 a and b). In particular *TIDE-Var confines most of its predicted probabilities for negatives to values less than* 0.2*, thus avoiding to generate false positives (Fig. 7 a). A similar behaviour can be also detected in* XGBoost *(Fig. 7 d), even if negatives examples are more spread toward larger predicted probabilities and positive ones are less polarized toward large values, thus explaining the lower sensitivity of the model with respect to* TIDE-Var*.*

Several variants are wrongly predicted and especially some pathogenic variants are wrongly predicted as neutral by all models (Fig. 7). The difficulty of this prediction task is evidenced by the embedded representations of the variants extracted from the last hidden layer preceding the classification layer of T-ResNet and *TIDE-Var. The embeddings were projected to two dimensions with a combination of PCA and UMAP transformations [51], and visualized through a Gaussian Kernel Density Estimation (KDE) heatmap [52].*

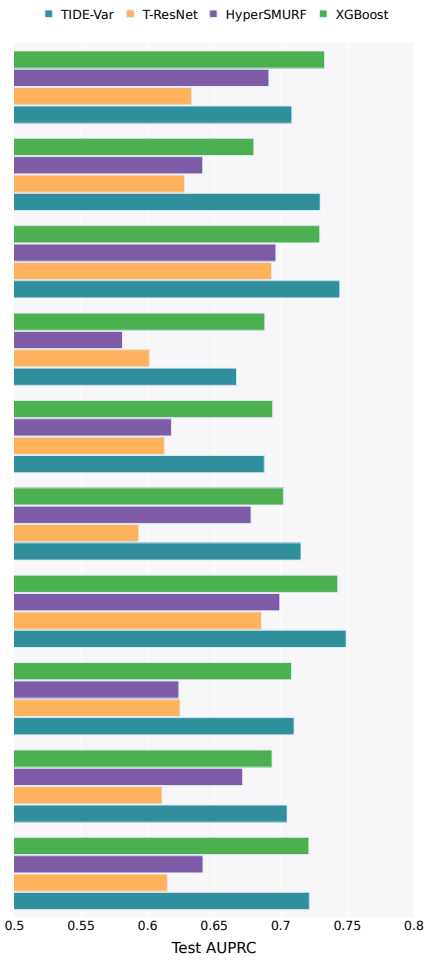


Figure 3: Multiple hold-out comparison: AUPRC results. Each block of four bars represents results on the hold-out test set of *TIDE-Var,* T-ResNet, *HyperSMURF and* XGBoost.

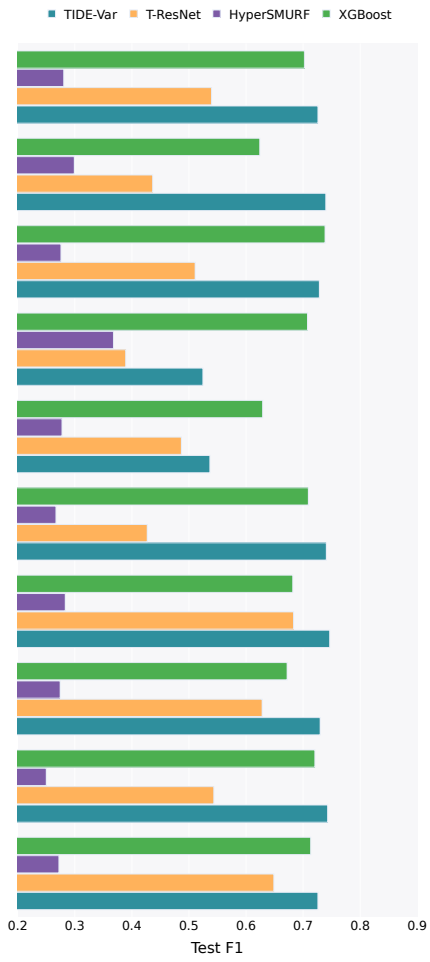


Figure 4: Multiple hold-out comparison: F1 results. Each block of four bars represents results on the hold-out test set of *TIDE-Var,* T-ResNet, *HyperSMURF and* XGBoost.

As shown in Figure 8, positive examples form distinct clusters, but they still overlap substantially with negative examples, indicating that both T-ResNet *and* TIDE-Var *struggle to fully separate pathogenic from neutral variants. Figure 8 presents KDE plots based on embeddings from the test set of one hold-out split, and similar patterns are observed across the other hold-outs as well.*

3.5.2. TIDE-Var learning characterization

We then focused on the learning behaviour and on the main factors underlying the performance of TIDE-Var, *the best performing method in our experimental comparison. We started analyzing the impact of ensembling by reducing the number k of implicit base learners to* 1. *Performance decreased markedly, yielding an average AUPRC of* 0.6374 *vs* 0.7331 *obtained by* TIDE-Var (p-value$<$ 0.01, Wilcoxon signed rank test), highlighting that ensembling is a key contributor to *TIDE-Var's best overall performance.*

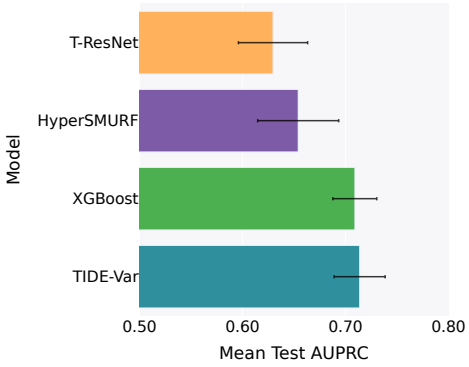
To gain further insight into how the number of implicit base learners k and the embedding dimension d affect overall AUPRC performance, we performed a grid search over several (k, d) *pairs (Fig. 9), by applying an internal 5-fold cross-validation on the training set of the first hold-out used to evaluate the generalization performance of the different models (Section 3.4). The best-performing configuration for TIDE-Var was* $k = 48$ *and* $d = 32$, *achieving an AUPRC of* 0.7331. *However, several other configurations yielded comparable results, and overall we consistently obtained an AUPRC* $>$ 0.70 *across the explored* (k, d) *settings. In particular, the configuration with* $k = 24$ *and* $d = 12$ *(AUPRC* 0.7291*) is statistically indistinguishable from the one with* $k = 48$ *and* $d = 32$, *while requiring about one third of the training time (approximately* 1975 *vs.* 6500 *seconds) and a substantially lower peak GPU memory usage (about* 6.3 *GB vs.* 16.1 *GB). We therefore selected the more efficient* $k = 24$, $d = 12$ *configuration for all reported* TIDE-Var *experiments.*

The marginal effect of the embedding dimension and of the ensemble size confirms that TIDE-Var is not particularly sensitive to these parameters, even if the ensemble size shows a positive trend with respect to the number of implicit base learners (Fig. 10).

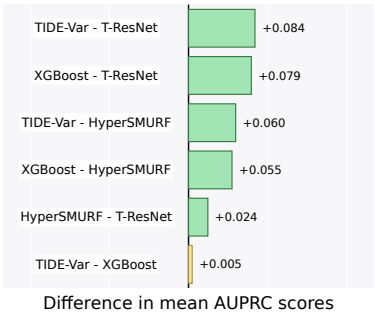
Another important learning characteristic of *TIDE-Var is represented by the fact that the base learners are implicit, in the sense that they are not completely independent, since they share most of their learning parameters and they learn contextually by jointly minimizing the same objective function. To show the impact of this learning strategy we compared* TIDE-Var with a "vanilla" ensemble of base deep neural networks, having the same number of layers and hidden neurons of the implicit *TIDE-Var base learners, but without parameter sharing and with independent base learner training. The final output of the vanilla ensemble is the average of the pathogenic probability computed by each base learner.*

To this end we run a standard deep ensemble having 16 *independently trained neural networks, matching the architecture of a single base neural network but without parameter sharing, and aggregating predictions by probability averaging. Despite the much higher computational cost, this classical ensemble achieved an AUPRC of* 0.5945 *and required around* 69 *GB of peak GPU memory during training, versus the* 0.7291 *AUPRC and* 6.5 *GB required by* TIDE-Var. *Overall, this confirms that* TIDE-Var*'s gains are not only due to ensembling alone, but to its implicit parameter-sharing and joint optimization, which yield both better generalization and far greater efficiency.*

We finally investigated whether mini-batch balancing techniques can significantly improve TIDE-Var *performance. To this end, we compared training* TIDE-Var *with and without mini-batch balancing. While balancing yielded almost perfect training AUPRC (Fig. 11a), it did not introduce a significant advantage during validation, since balanced and unbalanced mini-batch strategies converged to similar AUPRC results (Fig. 11b). This could be surprising, considering the high imbalance of the data, but it can be explained by the relatively large size of the mini-batch (*8192*) that significantly improves the probability of having one or more pathogenic variants in each mini-batch, a mandatory condition to enable learning in highly unbalanced contexts [37]. Moreover, mini-batch balancing implies longer training time. For these reasons, in our comparative experiments we used the unbalanced mini-batch with a large mini-batch size.*



(a) Average AUPRC on test sets



(b) Wilcoxon signed-rank test

Figure 5: Comparison of test AUPRC across models. **(a)** Mean test AUPRC for *T-ResNet*, *HyperSMURF*, *XGBoost*, and *TIDE-Var* over the ten hold-out splits, with error bars showing standard deviation. **(b)** Pairwise differences in mean AUPRC, with bars coloured by the Wilcoxon signed-rank test outcome (green: significant, $p < 0.05$; yellow: not significant, $p \geq 0.05$).

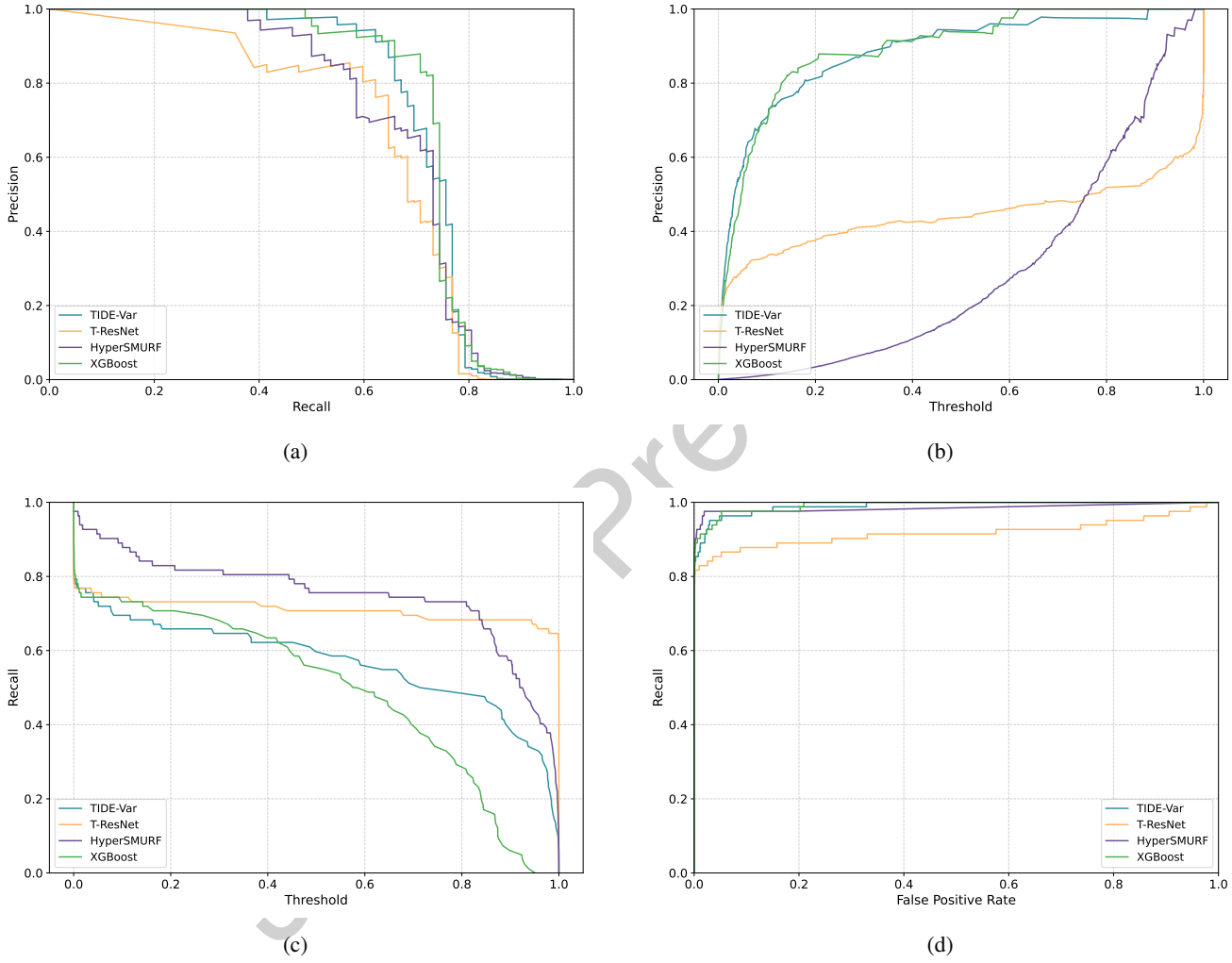
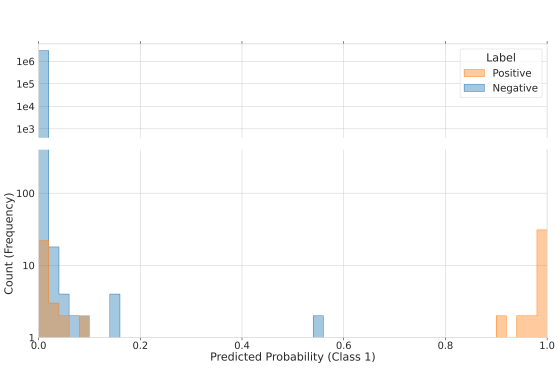
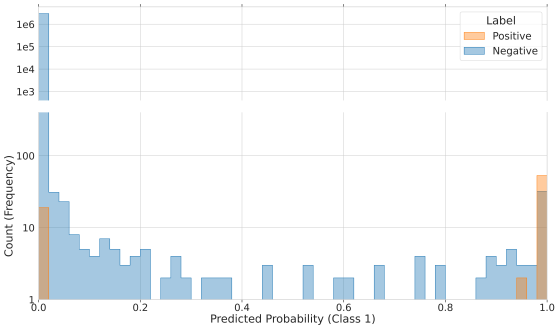


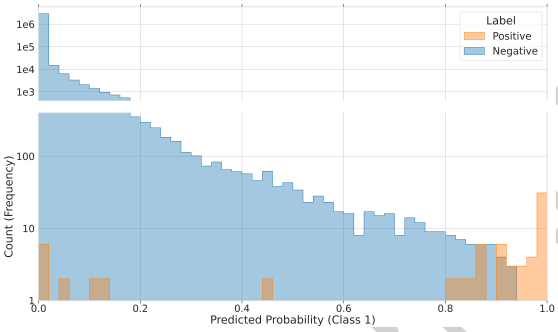
Figure 6: Comparison of TIDE-Var (blue), T-ResNet (orange), HyperSMURF (purple) and XGBoost (green) on the test set of the first hold-out: (a) precision–recall curves, (b) precision at different threshold levels, (c) sensitivity at different threshold levels, (d) ROC curves.



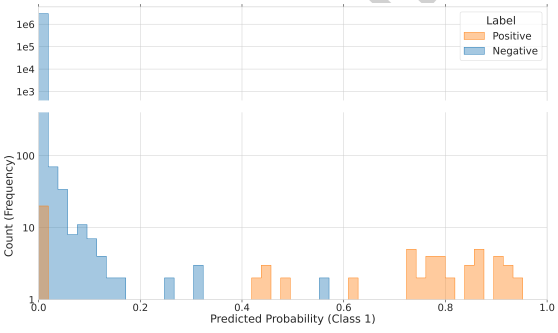
(a) TIDE-Var



(b) T-ResNet

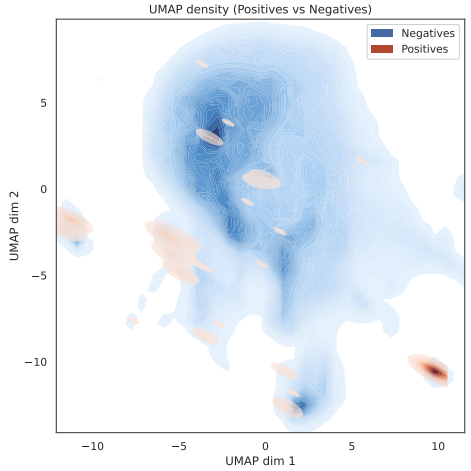


(c) HyperSMURF

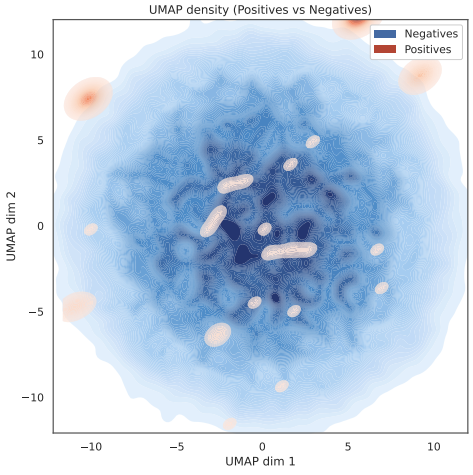


(d) XGBoost

Figure 7: Histograms of the predicted probabilities of the pathogenic (orange) and non-pathogenic (blue) variants for TIDE-Var (a), T-ResNet (b), HyperSMURF (c) and XGBoost (d). X-axis: predicted probabilities; y-axis; frequency of the predictions in log-scale. The representation of the bins having values larger then 200 are broken to allow for the proper visualization of probabilities significantly larger than 0.



(a) T-ResNet



(b) TIDE-Var

Figure 8: KDE heatmap of the test set embeddings projected into two dimensions through UMAP. (a) T-ResNet; (b) TIDE-Var projected embeddings. Positive (pathogenic) variants are represented in red, non pathogenic variants in blue.

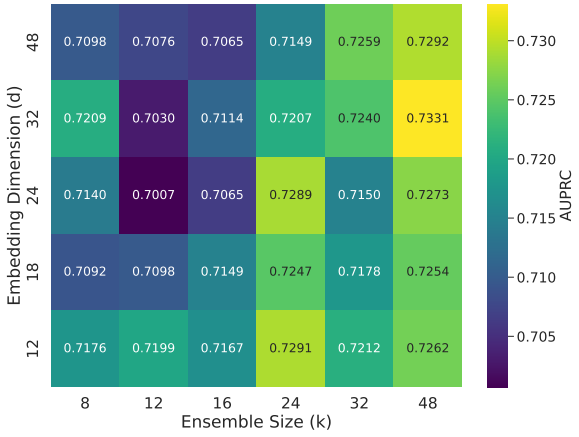


Figure 9: Heatmap of AUPRC scores for the *TIDE-Var model for different values of the ensemble size (x-axis) and of the embedding dimension (y-axis).*

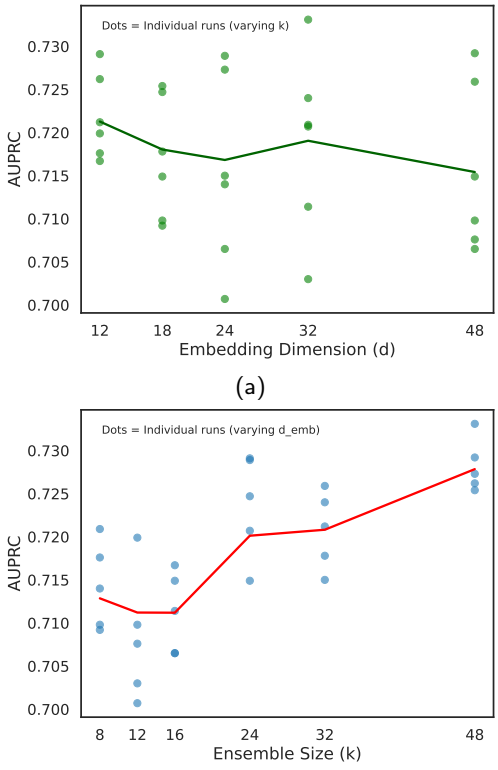


Figure 10: Marginal effect of the embedding dimension and of the number of implicit base learners for the *TIDE-Var model. (a) Marginal effect of the embedding dimension. The green broken line represents the average AUPRC as a function of the embedding size; dots the AUPRC values obtained by varying the k hyperparameter. (b) Marginal effect of the number of implicit base learners. The red broken line is the average AUPRC as a function of the ensemble size, and dots represent AUPRC values obtained by varying the d hyperparameter.*

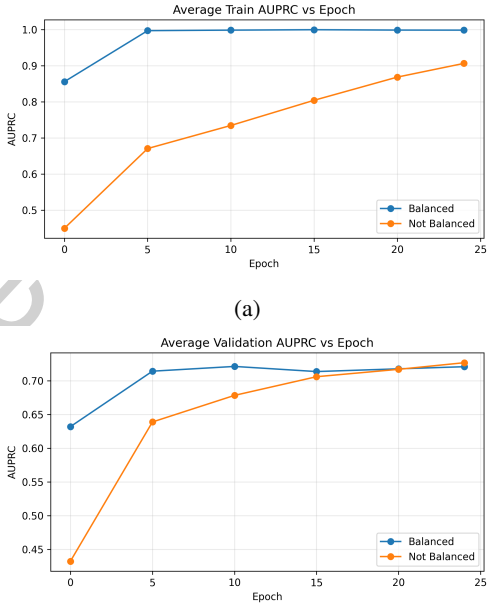


Figure 11: Training and validation AUPRC curves comparison of *TIDE-Var* with and without mini-batch balancing, averaged over the five internal cross-validation folds. (a) Training results; (b) Validation results. X-axis: epochs; y-axis: AUPRC.

4. Conclusions

Identifying pathogenic variants in the non coding genome is an open computational biology problem, representing a significant challenge from both a machine learning and precision medicine standpoint. The severe imbalance between available pathogenic and neutral variants, and recent advances in deep neural network ensembling motivated the design and implementation of two modular deep neural network systems to leverage the characteristics of conservation, genetic and epigenetic features of Mendelian data.

Our first model, by combining a modular architecture, residual connections and balanced mini-batch techniques, achieved performance comparable to HyperSMURF, one of the state-of-the-art methods for the prediction of pathogenic variants in Mendelian diseases.

Our second proposed neural model, *TIDE-Var leverages an ensemble of implicit base neural networks that share parameters and introduces base-learner diversity through adapter matrices. This architecture significantly improved performance with respect to state-of-the-art prediction methods, and also achieved slightly better results than* XGBoost*, even if the difference in favour of* TIDE-Var *is not statistically significant, demonstrating that deep neural networks ensembles can obtain results at least comparable to regularized boosting decision tree ensembles on complex highly imbalanced genetic data in tabular format.*

Ablation studies showed that the implicit ensembling approach (with shared parameters), the contextual learning of base learners, which jointly optimize the same objective function, and the diversity induced by adapter matrices are crucial for improving the prediction of pathogenic variants, without increasing the empirical time complexity.

Even if our results showed that TIDE-Var *achieves significantly more accurate and precise predictions, our analyses highlighted distinct operating profiles among* TIDE-Var*,* T-ResNet*,* HyperSMURF *and* XGBoost*. Indeed,* TIDE-Var *achieves the best precision–recall trade-off, mainly through its very high precision, assigning low scores to neutral variants and strongly reducing the rate of false positives. Conversely,* HyperSMURF *and* T-ResNet *tend to maintain higher sensitivity across thresholds, suggesting that they can be preferable when the priority is to maximize recall. This learning behaviour is valuable in exploratory settings, where capturing as many candidate pathogenic variants as possible is more important than limiting false positives.*

Nevertheless, all models struggle with a subset of pathogenic variants that are hard to distinguish from neutral examples, as indicated by probability histograms and by the embedding-space visualizations, which reveal substantial overlap between the two classes even when pathogenic variants form localized clusters. These observations emphasize that, despite the improvements brought by TIDE-Var*, the prediction task remains intrinsically challenging and partly limited by the similarity between certain pathogenic and neutral variants in feature space.*

To overcome these limitations, which are partly due to the relative low number of features and positive examples used to train our models, we are working to construct a larger set of features to better characterize the genetic and epigenetic properties of the variants, as well to enlarge the number of 'positive' pathogenic variants by collecting them from existing literature.

Another future research direction is to design multimodal ensembles of deep neural networks that integrate the available tabular data (conservation, genetic, and epigenetic features) with sequence context, i.e., nucleotide 'windows' around each variant, to better capture their functional properties. From this standpoint TIDE-Var *could be integrated with another deep learning module for sequence data, e.g. a convolutional neural network or a transformer [53] through a head module to combine the hidden representations provided by the tabular and by the sequence-based models.*

We finally note that TIDE-Var achieved state-of the-art results and could be in perspective used as the machine learning core of Genomiser [26], to improve the diagnosis of Mendelian genetic diseases caused by mutations in the non-coding regions of the human genome.

Acknowledgments

Fundings: National Center for Gene Therapy and Drugs Based on RNA Technology (CN_00000041, NextGeneration EU); National Plan for NRRP Complementary Investments (PNC) in the call for the funding of research initiatives for technologies and innovative trajectories in the health—project n. PNC0000003—AdvaNced Technologies for Human-centrEd Medicine (project acronym: ANTHEM). Computational resources at CINECA for this work have been funded by UNITECH INDACO, which is an HPC project at the University of Milan.

Competing interests

The authors declare no competing interests.

Code Availability

Code publicly available at:
https://github.com/AnacletoLAB/TIDE-Var.

All datasets used in this study are derived from publicly available sources as referenced in the manuscript.

References

[1] A. Fogel, J. Kvedar, Artificial intelligence powers digital medicine, Nature Digital Medicine 1 (2018).

[2] V. Cipriani, L. Vestito, E. Magavern, et al., Rare disease gene association discovery in the 100,000 genomes project, Nature (2025).

[3] D. Adams, C. Eng, Next-generation sequencing to diagnose suspected genetic disorders, N Engl J Med 379 (2018) 1353–62.

[4] H. Nakagawa, M. Fujita, Whole genome sequencing analysis for cancer genomics and precision medicine, Cancer Science 109 (2018) 513–522.

[5] S. Maddirevula, H. Kuwahara, N. Ewida, et al., "analysis of transcript-deleterious variants in mendelian disorders: implications for rna-based diagnostics", Genome Biol 21 (2020).

[6] "100,000 Genomes Project Pilot Investigators" and others, 100,000 genomes pilot on rare-disease diagnosis in health care—preliminary report, N. Engl. J. Med. 285 (2021) 1868–1880.

[7] Amberger, Joanna S and Bocchini, Carol A and Scott, Alan F and Hamosh, Ada, Omim.org: leveraging knowledge across phenotype–gene relationships, Nucleic Acids Research 47 (2018) D1038–D1043.

[8] N. Vasilevsky, S. Toro, N. Matentzoglu, et al., Mondo: integrating disease terminology across communities, Genetics 232 (2026) iyaf215.

[9] P. Kumar, S. Henikoff, P. Ng, Predicting the effects of coding non-synonymous variants on protein function using the sift algorithm., Nat Protoc. 4 (2009) 1073–81.

[10] J. Bendl, M. Musil, J. Stourac, J. Zendulka, J. Damborsky, J.and Brezovsky, Predictsnp2: A unified platform for accurately evaluating snp effects by exploiting the different characteristics of variants in distinct genomic regions, PLOS Computational Biology e100496 (2016).

[11] S. L. Edwards, J. Beesley, J. D. French, A. M. Dunning, Beyond gwass: illuminating the dark road from association to function., American Journal of Human Genetics 93 5 (2013) 779–97.

[12] L. Hindorff, P. Sethupathy, H. Junkins, E. Ramos, J. Mehta, F. Collins, T. Manolio, Potential etiologic and functional implications of genome-wide association loci for human diseases and traits, Proc Natl Acad Sci 106 (2009) 9362–9367.

[13] K. Farh, A. Marson, J. Zhu, et al., Genetic and epigenetic fine mapping of causal autoimmune disease variants, Nature 518 (2015) 337–343.

[14] Rojano, Elena and Seoane, Pedro and Ranea, J and Perkins, James, Regulatory variants: from detection to predicting impact, Briefings in Bioinformatics (2018).

[15] M. Kircher, D. M. Witten, P. Jain, B. J. O'Roak, G. M. Cooper, J. Shendure, A general framework for estimating the relative pathogenicity of human genetic variants., Nat Genet 46 (2014) 310–315.

[16] Schubach, Max and Maass, Thorben and Nazaretyan, Lusiné and Röner, Sebastian and Kircher, Martin, CADD v1.7: using protein language models, regulatory CNNs and other nucleotide-level scores to improve genome-wide variant predictions, Nucleic Acids Research 52 (2024) D1143–D1154.

[17] Shihab, Hashem and Rogers, Mark and Gough, Julian and Mort, Matthew and Cooper, David and Day, Ian and Gaunt, Tom and Campbell, Colin, An integrative approach to predicting the functional effects of non-coding and coding sequence variation, Bioinformatics 31 (2015) 1536–1543.

[18] Y.-F. Huang, B. Gulko, A. Siepel, Fast, scalable prediction of deleterious noncoding variants from functional and population genomic data, Nature Genetics 49 (2017) 618–624.

[19] I. Ionita-Laza, K. McCallum, B. Xu, J. D. Buxbaum, A spectral approach integrating functional genomic annotations for coding and noncoding variants, Nat Genet 48 (2016) 214–20.

[20] L. Sundaram, H. Gao, S. Padigepati, et al., Predicting the clinical impact of human mutation with deep neural networks, Nat Genet. 50 (2018) 1161–1170.

[21] Quang, Daniel and Chen, Yifei and Xie, Xiaohui, DANN: a deep learning approach for annotating the pathogenicity of genetic variants, Bioinformatics 31 (2015) 761–763.

[22] Zhou, J. and Theesfeld, C.L. and Yao, K. and Chen, K.M. and Wong, A.K. and Troyanskaya, O.G., Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk, Nature Genetics 50 (2018) 1171–1179.

[23] D. Lee, D. U. Gorkin, M. Baker, B. J. Strober, A. L. Asoni, A. S. McCallion, M. A. Beer, A method to predict the impact of regulatory variants from DNA sequence, Nature genetics 47 (2015) 955.

[24] K. Jaganathan, N. Ersaro, G. Novakovsky, et al., Predicting expression-altering promoter mutations with deep learning, Science (2025) eads7373.

[25] J. Linder, D. Srivastava, H. Yuan, et al., Predicting RNA-seq coverage from DNA sequence as a unifying model of gene regulation, Nat Genet 57 (2025) 949–961.

[26] D. Smedley, M. Schubach, J. O. Jacobsen, S. Köhler, T. Zemojtel, M. Spielmann, M. Jäger, H. Hochheiser, N. L. Washington, J. A. McMurry, M. A. Haendel, C. J. Mungall, S. E. Lewis, T. Groza, G. Valentini, P. N. Robinson, A Whole-Genome Analysis Framework for Effective Identification of Pathogenic Regulatory Variants in Mendelian Disease, The American Journal of Human Genetics 99 (2016) 595–606.

[27] H. He, E. Garcia, et al., Learning from imbalanced data, Knowledge and Data Engineering, IEEE Transactions on 21 (2009) 1263–1284.

[28] G. R. S. Ritchie, I. Dunham, E. Zeggini, P. Flicek, Functional annotation of noncoding sequence variants., Nat Methods 11 (2014) 294–296.

[29] A. Petrini, M. Mesiti, M. Schubach, M. Frasca, D. Danis, M. Re, G. Grossi, L. Cappelletti, T. Castrignanò, P. Robinson, G. Valentini, parSMURF, a high-performance computing tool for the genome-wide detection of pathogenic variants, GigaScience 9 (2020) giaa052.

[30] Schubach, Max and Re, Matteo and Robinson, Peter N. and Valentini, Giorgio, Imbalance-Aware Machine Learning for Predicting Rare and Common Disease-Associated Non-Coding Variants, Scientific Reports 7 (2017) 2959.

[31] T. Chen, 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, Association for Computing Machinery, New York, NY, USA, 2016, p. 785–794. doi:10.1145/2939672.2939785.

[32] B. Caron, Y. Luo, A. Rausell, Ncboost classifies pathogenic non-coding variants in mendelian diseases through supervised learning on purifying selection signals in humans, Genome Biology 20 (2019) 32.

[33] M. Danzi, M. Dohrn, S. Fazal, et al., Deep structured learning for variant prioritization in Mendelian diseases, Nat Commun 14 (2023).

[34] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, G. Kasneci, Deep neural networks and tabular data: A survey, IEEE Transactions on Neural Networks and Learning Systems 35 (2024) 7499–7519.

[35] Y. Gorishniy, A. Kotelnikov, A. Babenko, Tabm: Advancing tabular deep learning with parameter-efficient ensembling, in: The Thirteenth International Conference on Learning Representations (ICLR 2025), 2025.

[36] Wen, Y and Tran, D and Ba, J, Batchensemble an alternative approach to efficient ensemble and lifelong learning, in: Eighth International Conference on Learning Representations (ICLR 2020), OpenReview.net, 2020.

[37] F. Stacchietti, M. Nicolini, L. Chimirri, P. Robinson, E. Casiraghi, G. Valentini, Modular deep neural networks with residual connections for predicting the pathogenicity of genetic variants in non coding genomic regions, in: Advances in Computational Intelligence - 18th International Work-Conference on Artificial Neural Networks, IWANN 2025, A Coruña, Spain, June 16–18, 2025, Proceedings, Lecture Notes in Computer Science, Springer, 2025.

[38] K. He, X. Zhang, S. Ren, Sun.J., Deep residual learning for image recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 770–778.

[39] G. Valentini, F. Masulli, Ensembles of learning machines, in: M. Marinaro, R. Tagliaferri (Eds.), Neural Nets, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 3–20.

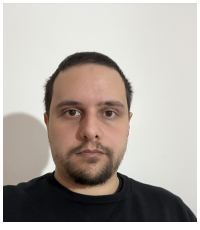
[40] G. Ritchie, I. Dunham, E. Zeggini, et al., Functional annotation of noncoding sequence variants, Nat Methods 11 (2014) 294–296.

[41] Ganaie, MA and Hu, M and Malik, AK and Tanveer, M and Suganthan, PN, Ensemble deep learning: A review, Engineering Applications of Artificial Intelligence 115 (2022) 105151.

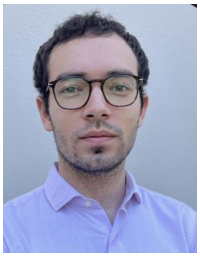
[42] Allen-Zhu, Zeyuan and Li, Yuanzhi, Towards Understanding Ensemble, Knowledge Distillation and Self-Distillation in Deep Learning, in: ICLR, OpenReview.net, 2023. URL: http://dblp.uni-trier.de/db/conf/iclr/iclr2023.html#Allen-ZhuL23.

[43] Y. Gorishniy, I. Rubachev, A. Babenko, On embeddings for numerical features in tabular deep learning, Advances in Neural Information Processing Systems 35 (2022) 24991–25004.

[44] Y. Gorishniy, A. Kotelnikov, A. Babenko, TabM: Advancing Tabular Deep Learning With Parameter-Efficient Ensembling, in: The Thirteenth International Conference on Learning Representations (ICLR), 2025.
[45] P. Rentzsch, D. Witten, G. Cooper, J. Shendure, M. Kircher, Cadd: predicting the deleteriousness of variants throughout the human genome, Nucleic Acids Res. 47 (2019) D886–D894.
[46] I. Ionita-Laza, K. McCallum, B. Xu, et al., A spectral approach integrating functional genomic annotations for coding and noncoding variants, Nat Genet 48 (2016) 214–220.
[47] J. Zhou, O. Troyanskaya, Predicting effects of noncoding variants with deep learning–based sequence model, Nat Methods 12 (2015) 931–934.
[48] I. Cooperstein, S. Marwaha, A. Ward, et al., An optimized variant prioritization process for rare disease diagnostics: recommendations for Exomiser and Genomiser, Genome Medicine 17 (2025) 127.
[49] N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research 6 (2002) 321–357.
[50] Schubach, Max and Nazaretyan, Lusiné and Kircher, Martin, The Regulatory Mendelian Mutation score for GRCh38, GigaScience 12 (2023) giad024.
[51] L. McInnes, J. Healy, N. Saul, L. Großberger, Umap: Uniform manifold approximation and projection, Journal of Open Source Software 3 (2018) 861.
[52] J. Herr, Fast Accurate Gaussian Kernel Density Estimation, in: IEEE Visualization Conference (VIS), 2021, pp. 11–15. doi:10.1109/VIS49827.2021.9623323.
[53] H. Dalla-Torre, L. Gonzalez, J. Mendoza-Revilla, et al., Nucleotide Transformer: building and evaluating robust foundation models for human genomics, Nat Methods 22 (2025) 287–297.



Federico Stacchietti is a Research Fellow at AnacletoLab (University of Milan). He earned an MSc in Computer Science and studied mechanistic interpretability of biomedical LLMs. He develops interpretable neural models for rare-disease variant pathogenicity under extreme class imbalance and sequence-based interaction prediction using pretrained RNA language models. He has authored papers in international journals and conference proceedings and participates in international research projects.



Marco Nicolini is a PhD student in Computer Science at the University of Milan (UNIMI). Research activity: machine learning for biomolecular sequences, including protein and RNA language models, and interaction prediction. He has published in international journals and conference proceedings and is involved in international collaborations.

Leonardo Chimirri is a Postdoctoral researcher in Medical Computer Science and AI at the Berlin Institute of Health at Charité. During his PhD in theoretical particle physics, he developed strong expertise in mathematical modeling, numerical methods, and large-scale computational inference. He later transitioned to applied research in bioinformatics and medical AI, focusing on rare genetic diseases using large language models, biomedical ontologies, phenotype data standardization, Bayesian networks, and tabular- and sequence-based machine learning.

Peter N. Robinson is a von Humboldt Professor for Artificial Intelligence at the Berlin Institute of Health at Charité (BIH). His major work has been the development of the Human Phenotype Ontology (HPO), which is now a standard tool used globally to diagnose gene-related diseases. He is author of hundreds of publications in top journals in the field of Bioinformatics, Human Genetics and applications of AI methods to precision medicine, and PI of international research projects funded by the NIH and the European Union.

Elena Casiraghi is a Full Professor of Computer Science at the University of Milan (UNIMI) and member of ELLIS (European Laboratory for Learning and Intelligent Systems). Her research spans bioinformatics, health informatics, explainable AI, and biomedical signal analysis, with applications in personalized and precision medicine. She has authored over 100 peer-reviewed journal articles and more than 80 conference publications and serves as principal investigator on national and European research projects, working in close collaboration with medical and biomedical institutions.

Giorgio Valentini is a Full Professor at the Department of Computer Science, University of Milan (UNIMI). Member of ELLIS, the European Laboratory for Learning and Intelligent Systems. Author of about 200 scientific publications and PI of 15 national and international research projects. His main research activity is about the design and application of Artificial Intelligence and Bioinformatics methods to Molecular Biology, Personalized and Precision Medicine.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: