scientific reports

Check for updates

OPEN

# Prediction and design of thermostable proteins with a desired melting temperature

Purva Tijare[1,2], Nishant Kumar[1,2] & Gajendra P. S. Raghava[1]✉

**The stability of proteins at higher temperatures is crucial for their functionality, which is measured by their melting temperature (Tm). The Tm is the temperature at which 50% of the protein loses its native structure and activity. Existing methods for predicting Tm have two major limitations: first, they are often trained on redundant proteins, and second, they do not allow users to design proteins with the desired Tm. To address these limitations, we developed a regression method for predicting the Tm value of proteins using 17,312 non-redundant proteins, where no two proteins are more than 40% similar. We used 80% of the data for training and testing and the remaining 20% for validation. Initially, we developed a machine learning model using standard features from protein sequences. Our best model, developed using Shannon entropy for all residues, achieved the highest Pearson correlation of 0.80 with an $R^2$ of 0.63 between the predicted and actual Tm of proteins on the validation dataset. Next, we fine-tuned large language models (e.g., ProtBert, ProtGPT2, ProtT5) on our training dataset and generated embeddings. These embeddings have been used to develop machine learning models. Our best model, developed using ProtBert embeddings, achieved a maximum correlation of 0.89 with an $R^2$ of 0.80 on the validation dataset. Finally, we developed an ensemble method that combines standard protein features and embeddings. One of the aims of the study is to assist the scientific community in the design of targeted melting temperatures. Our standalone software can be used to screen thermostable proteins at the genome level. We demonstrated the application of PPTstab in identifying thermostable proteins in different organisms. We created a user-friendly web server, and a Python package for predicting and designing thermostable proteins is available at https://webs.iiitd.edu.in/raghava/pptstab, https://github.com/raghavagps/pptstab.**

**Keywords** Melting temperature, Prediction, Machine learning, Embeddings, Protein language models, Thermostable proteins

**Abbreviations**

| | |
|---|---|
| ANN | Artificial neural network |
| BFD | Big fantastic database |
| CV | Cross-validation |
| ET | Extra trees regressor |
| LGBM | Light gradient boosting machine |
| LLM | Large language model |
| MAE | Mean average error |
| MLM | Masked language model |
| MLP | Multi-layer perceptron |
| MMS | Min–max scaler |
| MSE | Mean squared error |
| NuSVR | Nu support vector regression |
| PLM | Protein language model |
| PCC | Pearson correlation coefficient |
| $R^2$ | Coefficient of determination |
| ReLU | Rectified linear unit |
| RMSE | Root mean squared error |

[1]Department of Computational Biology, Indraprastha Institute of Information Technology, Okhla Industrial Estate, Phase III (Near Govind Puri Metro Station), Office: A-302 (R&D Block), New Delhi 110020, India. [2]Purva Tijare and Nishant Kumar contributed equally to this work. ✉email: raghava@iiitd.ac.in

SVR Support vector regression
Tm Melting temperature

Proteins are highly versatile organic molecules essential to living organisms, playing a pivotal role in regulating key bodily functions and their ability to carry out the desired functions heavily depends on their thermal stability. The thermal stability of proteins is commonly characterized by their melting temperature (Tm). The melting temperature is a good indicator of the thermal stability of proteins. The Tm is a good indicator of the thermal stability of proteins. Thermostable proteins have applications in medical research and therapy, acting as stable frameworks for drug delivery systems, enzyme optimization, diagnostic tests, and therapeutic treatments. An understanding of factors governing stability is important for the design of stable proteins[1–4]. Industrial production and pharmaceutical development have broad applications and use of thermostable proteins[5].

Additionally, thermostable proteins have applications in various fields, such as medical research and therapy[4,6]. The experimental techniques for identifying protein Tm involve advanced methods such as Mass Spectrometry-based Thermal Proteome Profiling (TPP), Fourier Transform Infrared Spectroscopy, Circular Dichroism, and Differential Scanning Calorimetry[3,7]. Due to cost and complexity, these experimental techniques cannot be used to screen thermostable proteins at the genome scale. There is a need to develop in silico methods that can be used to screen thermostable proteins in protein databases.

In the past, several computational methods have been developed to predict the Tm of proteins, most of them trained on the redundant datasets shown in Table 1. These methods may fail on proteins that do not have high similarity with proteins in the training dataset. Thus, there is a need to develop a method on a non-redundant dataset of proteins using standard bioinformatics protocols. In this study, we obtained the dataset from the DeepSTABp, which contains 35,114 protein sequences[5]. We applied the CD-hit at 40% to remove the redundant sequences and got 17,312 non-redundant proteins, with no two protein sequences having more than 40% similarity among them. The data has been used for training, testing, and evaluating our models, which were developed using various machine learning, large language models, and deep learning algorithms. To generate features of proteins, we used standard software Pfeature and large language models. After all the models were trained and tested using a ten-fold CV, the final model was tested on an independent or validation dataset. We developed a web-based platform called PPTstab for users to predict the Tm of proteins. The design module of PPTstab generates all possible variants of a protein and its Tm so that users can select a variant with the desired Tm. In addition, standalone software has been developed to screen thermostable proteins at the genome-scale.

## Results

The result section has five different categories, including (i) Data analysis of thermostable proteins, (ii) Machine learning methods, (iii) Hybrid approaches, (iv) Web server, and (v) Application. The complete workflow of the study is illustrated in Fig. 1, and the details of the following subsections can be found below.

### Data analysis

We performed compositional and correlation analysis on the main dataset to understand the relationship between Tm values and the composition of residues.

*Composition analysis of proteins*

Ponnuswamy et al. tested the relationship between the amino acid composition and thermal stability of globular proteins[19]. To analyze this relationship, we divided the proteins into two groups: proteins with Tm > 50 °C and proteins with Tm < 50 °C. The average amino acid composition was calculated for each residue in both protein groups, as shown in Fig. 2. From Fig. 2, we can see that in thermophilic proteins, Leucine (L), Alanine (A),

| Method | Year | Methodology |
|---|---|---|
| ProtStab[8] | 2019 | Gradient boosting of regression trees algorithm with 100 features from PROFEAT, ProtDCal, and ProtParam (Regressor) |
| iStable 2.0[9] | 2020 | Weka and XGBoost for classification and regression models with 10-fold cross-validation (Classifier & Regressor) |
| SCMTPP[10] | 2021 | A predictor based on support vector machines for the identification and characterization of thermophilic proteins utilizing estimated propensity scores of dipeptides (Classifier) |
| ProtStab2[11] | 2022 | LightGBM algorithm with 6395 Features; including protr, ProtDCal, and ProtParam descriptors (Regressor) |
| TMPpred[12] | 2022 | SVM-based thermophilic protein predictor that uses ANOVA on an 188-dimensional feature set to classify thermophilic and non-thermophilic proteins (Classifier) |
| SAPPHIRE[13] | 2022 | A predictor based on a stacking model has been developed to effectively identify thermophilic proteins (Classifier) |
| DeepTP[14] | 2023 | The CNN combined with a Bi-LSTM model is utilized to extract features with long-range dependencies, which are subsequently weighted through a self-attention mechanism and predicted using a Multi-Layer Perceptron (Classifier) |
| BertThermo[15] | 2023 | Classification of thermophilic proteins using BERT-bfd features and feature engineering methods like SMOTE, LGBM and LR as a classifier (Classifier) |
| DeepSTABp[16] | 2023 | Transformer based PLM for sequence embeddings generation and used as features with MLP as predictor (Regressor) |
| DeepTM[5] | 2023 | A prediction model utilizing Graph Convolutional Neural Networks, Self-Attention Networks, and Multi-Layer Perceptrons, designed for sequence lengths of fewer than 1028 (Regressor) |
| ProLaTherm[17] | 2023 | PLM-based thermophilicity predictor using ProtT5-XL-UniRef50 encoder using sequence information (Classifier) |
| TemStaPro[18] | 2024 | A binary classifier with embeddings generated from ESM-2 and ProtT5-XL protein language models (Classifier) |

**Table 1**. List of available methods with description.

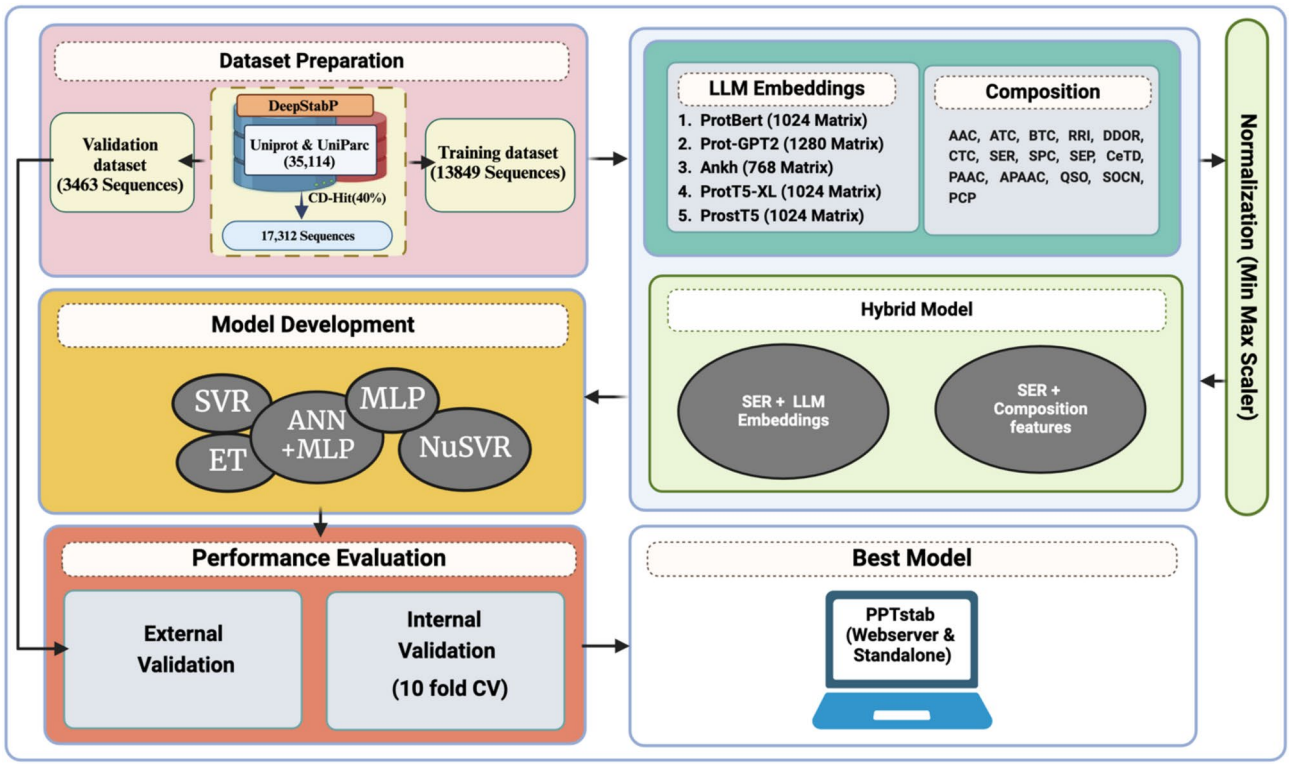
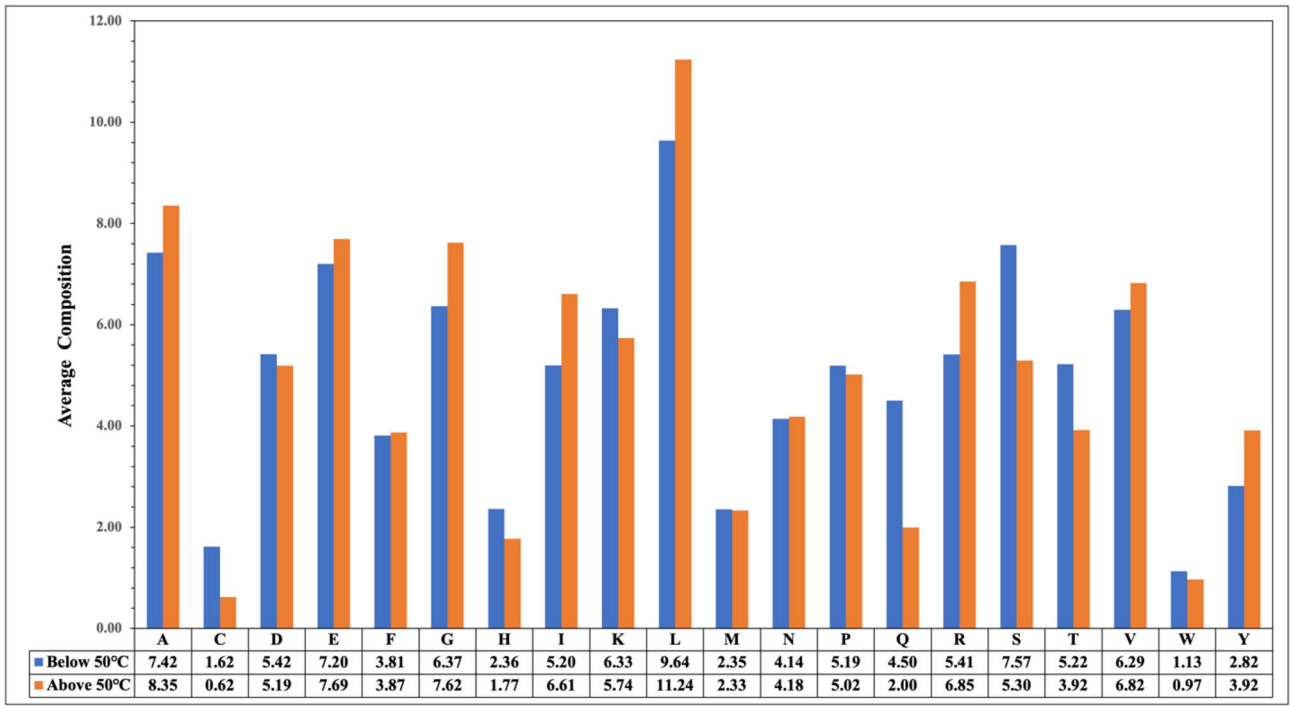



**Fig. 1**. Overview of the study's complete workflow.




| | A | C | D | E | F | G | H | I | K | L | M | N | P | Q | R | S | T | V | W | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Below 50°C | 7.42 | 1.62 | 5.42 | 7.20 | 3.81 | 6.37 | 2.36 | 5.20 | 6.33 | 9.64 | 2.35 | 4.14 | 5.19 | 4.50 | 5.41 | 7.57 | 5.22 | 6.29 | 1.13 | 2.82 |
| Above 50°C | 8.35 | 0.62 | 5.19 | 7.69 | 3.87 | 7.62 | 1.77 | 6.61 | 5.74 | 11.24 | 2.33 | 4.18 | 5.02 | 2.00 | 6.85 | 5.30 | 3.92 | 6.82 | 0.97 | 3.92 |

**Fig. 2**. Average percent amino acid composition of thermostable proteins.

| Name and description of descriptor | Training | | Validation | |
|---|---|---|---|---|
| | PCC | $R^2$ | PCC | $R^2$ |
| AAC (amino acid composition) | 0.81 | 0.65 | 0.77 | 0.51 |
| ATC (atomic composition) | 0.53 | 0.28 | 0.45 | 0.17 |
| BTC (bond composition) | 0.29 | 0.08 | 0.26 | 0.07 |
| PCP (physico-chemical properties based composition) | 0.78 | 0.60 | 0.68 | 0.45 |
| RRI (residue repeat information) | 0.49 | 0.24 | 0.25 | 0.07 |
| DDR (distance distribution of residues) | 0.74 | 0.54 | 0.64 | 0.40 |
| SER (Shannon entropy for all residues) | 0.80 | 0.64 | 0.80 | 0.63 |
| SEP (Shannon entropy based on physico-chemical properties) | 0.50 | 0.25 | 0.47 | 0.22 |
| CTC (conjoint triad calculation) | 0.63 | 0.38 | 0.66 | 0.42 |
| PAAC (pseudo amino acid composition) | 0.81 | 0.64 | 0.78 | 0.59 |
| APAAC (amphiphilic pseudo amino acid composition) | 0.81 | 0.65 | 0.79 | 0.59 |
| QSO (quasi-sequence order) | 0.80 | 0.64 | 0.63 | 0.32 |
| SOCN (sequence order coupling number) | 0.34 | 0.11 | 0.32 | 0.10 |
| CETD (composition enhanced transition distribution) | 0.74 | 0.54 | 0.65 | 0.07 |
| SPC (Shannon entropy at property level) | 0.74 | 0.55 | 0.71 | 0.49 |

**Table 2**. Performance of ANN + MLP regressor on different composition-based features. *R* Arginine, *L* Leucine, *S* Serine, *Q* Glutamine, *lam1* Sequence correlation factor for lambda 1, *T* Threonine, *G* Glycine.

| Composition | Correlation | | | |
|---|---|---|---|---|
| | Positive | | Negative | |
| AAC | R | 0.24 | Q | − 0.25 |
| | L | 0.21 | T | − 0.22 |
| SER | S | 0.25 | R | − 0.22 |
| | Q | 0.30 | G | − 0.19 |
| PAAC | lam1 | 0.26 | Q | − 0.25 |
| | R | 0.24 | T | − 0.22 |
| APAAC | R | 0.24 | Q | − 0.25 |
| | L | 0.21 | T | − 0.22 |

**Table 3**. The top positive and negative correlation in different compositions. $R^2$ Coefficient of determination, *PCC* Pearson correlation coefficient, *MLP* Multi-layer perceptron, *ANN* Artificial neural network.

Glycine (G), and Glutamic Acid (E) are significantly abundant, whereas Serine(S), Lysine(K), Glutamine(Q) and Histidine (H) are mainly found in proteins with Tm < 50 °C.

*Correlation between Tm and descriptors*

We first computed the percent composition of each descriptor, including amino acid composition (AAC), Shannon entropy for all residues (SER), pseudo amino acid composition (PAAC), and amphiphilic pseudo amino acid composition (APAAC), that achieved the maximum performance, as shown in Table 2. Secondly, we computed the correlation between the composition of a residue and Tm for each type of residue, as shown in Supplementary Figs. S1, S2, S3, and S4. The two most highly positive and negative amino acids are shown in Table 3, which lists the top correlations for different compositions.

## Machine learning methods

In this study, we have used various machine learning algorithms to build models for temperature prediction including Extra Trees Regressor (ET), Multi-layer Perceptron (MLP), Nu Support Vector Regression (NuSVR), Light Gradient Boosting Machine (LGBM), Support Vector Regression (SVR), CatBoost, Artificial Neural Network (ANN), and ensemble models such as ANN + MLP regressor. We have developed these models using compositional features, Large Language Models (LLM) embeddings, and hybrid features.

*Model evaluation*

To evaluate the performance of our prediction models, we have analyzed the results obtained from k-fold CV, where k = 10, and external validation following the standard protocols. The evaluation matrices include Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE), Pearson Correlation Coefficient (PCC), and $R^2$ on both training and validation datasets. Supplementary Table S1 represents the CV results showing the mean values, standard deviation (SD), and standard error (SE) for each matric.

*Development of prediction models*

After developing various algorithms using different compositional features, we found that the ensemble model ANN + MLP outperformed all other regression models. Table 2 presents the performance metrics of this ensemble model across different compositions and descriptors. Among 15 different compositions, features like SER, APAAC, PAAC, and AAC emerged as the primary contributors. Notably, the SER feature achieved the highest performance with an $R^2$ value of 0.63 and a Pearson correlation coefficient (PCC) value of 0.80 on the validation dataset. For a comprehensive overview of the results with different regressors, please refer to Supplementary Table S2.

*Composition-based models*

Based on the above observations, we systematically developed a method using the SER composition of proteins as input features, with each feature represented by a 20-dimensional vector corresponding to the 20 different amino acids[20] along with the 'lysate' or 'cell' flag. Various models were trained on a specific dataset containing SER descriptors known as the SER to predict the Tm value as the output. The comprehensive results obtained through ten-fold cross-validation with normalization as preprocessing are summarized in Table 4. However, the SER features alone did not yield good results, as the ensemble model achieved an $R^2$ value of 0.63 and a PCC of 0.80.

*LLM embeddings-based models*

In this study, we developed a prediction model using Protein Language Model (PLM) embeddings, explicitly using the pre-trained Large Language Models (LLM) known as ProtBert. It was used as a static feature encoder without any fine-tuning to derive protein embeddings for input, forming the foundation of our method[21]. These embeddings were combined with categorical flags indicating "lysate" or "cell" conditions, creating a comprehensive dataset of 1026 features to train our predictive model for temperature values. The next step involved training various regressors to enhance performance. ProtBert embeddings significantly outperformed other embeddings derived from models such as Ankh, ProtGPT2, ProtT5-XL-Uniref50, and ProsT5.

Among the models we evaluated, the ensemble model ANN + MLP proved to be the most effective predictor for Tm values, achieving impressive metrics, including an $R^2$ value of 0.80 and a PCC value of 0.89. Additionally, the model demonstrated impressive accuracy with a Mean Absolute Error (MAE) of 3.00, Root Mean Squared Error (RMSE) of 4.11, and Mean Squared Error (MSE) of 0.28, as shown in Table 5. These results show a considerable decrease in error rates compared to prior methods, highlighting our approach's improved generalizability and robustness. Additionally, we explored embeddings from different LLMs without fine-tuning and implemented various machine learning algorithms using these embeddings. The detailed performance results of these models are provided in Supplementary Table S3, showcasing the robustness and superiority of the ProtBert embeddings in our predictive framework.

## Hybrid approaches

In this approach, we developed models by combining various feature combinations as input. Our analysis observed that embeddings from the ProtBert LLM outperformed other composition-based features. Among the composition-based features, SER showed the best performance with a PCC value of 0.80 and an $R^2$ value of 0.63, as shown in Table 2. We combined SER descriptors with different feature sets, including AAC, PAAC, and ProtBert embeddings, to improve performance further. In this approach, we utilized a vector matrix of 1046 features, including SER and ProtBert embeddings and the 'lysate' or 'cell' flag. We have developed models using different regressors, and it was observed that the ensemble model ANN + MLP did not perform well compared to prior approaches, achieving a PCC value of 0.89 and an $R^2$ value of 0.79. The complete SER results are presented in Table 6, while the comprehensive results of other features are provided in Supplementary Table S4. After computing the performance of different feature sets, we found that combining SER features with ProtBert embeddings did not perform better than the ProtBert embeddings model.

| | Training | | | | | Validation | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Regressor | RMSE | MSE | MAE | PCC | $R^2$ | RMSE | MSE | MAE | PCC | $R^2$ |
| ANN + MLP | 9.07 | 0.82 | 6.84 | 0.80 | 0.64 | 9.18 | 0.84 | 13.36 | 0.80 | 0.63 |
| NuSVR | 9.47 | 0.90 | 7.06 | 0.78 | 0.61 | 9.91 | 0.98 | 7.28 | 0.76 | 0.57 |
| LGBM | 9.35 | 0.88 | 6.94 | 0.79 | 0.62 | 9.62 | 0.93 | 7.07 | 0.77 | 0.60 |
| MLP | 9.59 | 0.92 | 7.26 | 0.78 | 0.60 | 9.97 | 0.99 | 7.50 | 0.77 | 0.57 |
| SVR | 9.53 | 0.91 | 7.24 | 0.78 | 0.61 | 9.95 | 0.99 | 7.45 | 0.76 | 0.57 |
| ET | 9.15 | 0.84 | 6.82 | 0.80 | 0.64 | 9.61 | 0.92 | 7.12 | 0.77 | 0.60 |

**Table 4**. Performance of different regressors on SER features/descriptors. $R^2$ Coefficient of determination, *MAE* Mean average error, *SVR* Support vector regression, *MSE* Mean squared error, *ET* Extra Trees Regressor, *RMSE* Root mean squared error, *LGBM* Light gradient boosting machine, *NuSVR* Nu support vector regression, *ANN* Artificial neural network, *PCC* Pearson correlation coefficient, *MLP* Multi-layer perceptron.

| Features | Models | Training | | | | | Validation | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | RMSE | MSE | MAE | PCC | $R^2$ | RMSE | MSE | MAE | PCC | $R^2$ |
| ProtBert | ANN + MLP | 3.97 | 0.26 | 3.00 | 0.90 | 0.81 | 4.11 | 0.28 | 3.00 | 0.89 | 0.80 |
| | CatBoost | 4.46 | 0.33 | 3.32 | 0.87 | 0.76 | 4.58 | 0.35 | 3.34 | 0.86 | 0.75 |
| | NuSVR | 4.60 | 0.35 | 3.38 | 0.86 | 0.75 | 4.67 | 0.36 | 3.36 | 0.86 | 0.74 |
| ProtT5-XL-Uni Ref50 | ANN + MLP | 4.07 | 0.28 | 3.07 | 0.89 | 0.80 | 4.12 | 0.28 | 3.09 | 0.89 | 0.79 |
| | CatBoost | 4.45 | 0.33 | 3.32 | 0.87 | 0.76 | 4.49 | 0.34 | 3.35 | 0.87 | 0.76 |
| | NuSVR | 4.39 | 0.32 | 3.33 | 0.88 | 0.77 | 4.36 | 0.32 | 3.23 | 0.88 | 0.77 |
| Ankh | ANN + MLP | 4.93 | 0.41 | 3.61 | 0.84 | 0.71 | 5.03 | 0.42 | 3.61 | 0.83 | 0.69 |
| | CatBoost | 5.88 | 0.58 | 4.35 | 0.77 | 0.59 | 5.90 | 0.58 | 4.30 | 0.76 | 0.58 |
| | NuSVR | 5.88 | 0.58 | 4.38 | 0.77 | 0.59 | 5.89 | 0.58 | 4.36 | 0.76 | 0.58 |
| ProtGPT2 | ANN + MLP | 4.94 | 0.41 | 3.61 | 0.84 | 0.70 | 5.25 | 0.46 | 3.64 | 0.82 | 0.67 |
| | CatBoost | 5.44 | 0.49 | 3.99 | 0.80 | 0.65 | 5.54 | 0.51 | 4.00 | 0.79 | 0.63 |
| | NuSVR | 5.65 | 0.53 | 4.15 | 0.79 | 0.62 | 5.87 | 0.57 | 4.23 | 0.77 | 0.58 |
| ProstT5 | ANN + MLP | 5.33 | 0.48 | 3.88 | 0.81 | 0.66 | 5.24 | 0.46 | 3.77 | 0.82 | 0.67 |
| | CatBoost | 6.58 | 0.72 | 4.83 | 0.69 | 0.48 | 6.62 | 0.73 | 4.79 | 0.69 | 0.47 |
| | NuSVR | 6.10 | 0.62 | 4.54 | 0.74 | 0.55 | 6.12 | 0.63 | 4.50 | 0.74 | 0.55 |

**Table 5**. Performance of different regressors using various LLM embeddings. $R^2$ Coefficient of determination, *PCC* Pearson correlation coefficient, *MAE* Mean average error, *MSE* Mean squared error, *RMSE* Root mean squared error, *MLP* Multi-layer perceptron, *NuSVR* Nu support vector regression, *ANN* Artificial neural network.

| Models | Training | | | | | Validation | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | RMSE | MSE | MAE | PCC | $R^2$ | RMSE | MSE | MAE | PCC | $R^2$ |
| ANN + MLP | 3.99 | 0.27 | 3.02 | 0.90 | 0.80 | 4.05 | 0.27 | 2.97 | 0.89 | 0.79 |
| NuSVR | 7.67 | 0.59 | 5.63 | 0.86 | 0.75 | 8.06 | 0.65 | 5.81 | 0.86 | 0.72 |
| LGBM | 7.51 | 0.56 | 5.54 | 0.87 | 0.76 | 7.67 | 0.59 | 5.58 | 0.86 | 0.74 |
| MLP | 7.23 | 0.52 | 5.49 | 0.88 | 0.77 | 7.51 | 0.56 | 5.62 | 0.88 | 0.76 |
| SVR | 7.78 | 0.61 | 5.88 | 0.86 | 0.74 | 8.09 | 0.65 | 6.00 | 0.85 | 0.72 |
| ET | 7.46 | 0.56 | 5.50 | 0.87 | 0.76 | 7.59 | 0.58 | 5.51 | 0.87 | 0.75 |

**Table 6**. Performance of combined feature set: SER descriptors and ProtBert embeddings. $R^2$ Coefficient of determination, *PCC* Pearson correlation coefficient, *MAE* Mean average error, *MSE* Mean squared error, *RMSE* Root mean squared error, *ET* Extra Trees Regressor, *SVR* Support vector regression, *MLP* Multi-layer perceptron, *LGBM* Light gradient boosting machine, *NuSVR* Nu support vector regression, *ANN* Artificial neural network.

## Performance of previous methods

It is not feasible to compare our method with all the existing methods listed in Table 1, as some of them are classification-based approaches. We have compared our method with existing regression-based methods, as our method is regression based model. Unfortunately, only web server DeepSTABp is functional rest of them are not available or non-functional. ProtStab2 is an updated version of ProtStab, incorporating a dataset from meltome atlas. DeepSTABp also collects most of the dataset from meltome atlas, which compiles thermo-proteome profiling (TPP) assay datasets from different organisms, including studies involving the heat treatment of cells and protein lysates. It was observed that existing methods do not adhere to standard bioinformatics protocols, where redundant sequences are removed. Thus, its challenging to compare our methods with these methods as our method has been developed on non-redundant dataset where existing methods developed on redundant dataset. Despite our stringent criteria, our method achieved performance comparable to the best-performing model, DeepSTABp, in terms of $R^2$ and PCC, as reported in the literature. As shown in Table 7, our approach performs nearly the same as DeepSTABp and better than the previous method. In addition, our method uses 80% data for training and 20% data for external validation existing method DeepSTABp trained on 90% data and 10% data for external validation. In addition, DeepSTABp need growth temperature of organism of query protein whereas PPT stab have no such requirement. In summary, we have followed high standard to develop and validated our method.

## Web service for the community

To assist the scientific community, we developed an easy-to-use and freely available web server called "PPTstab" and its standalone version. This tool provides the prediction of Tm for protein sequences, using embeddings as features computed by the protein language model named ProtBert. The prediction models used in this

| Metrics | PPTStab | | DeepSTABp | | ProTstab2 | | ProTstab | |
|---|---|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test | Train | Test |
| RMSE | 3.97 | 4.11 | 2.35 | 4.30 | 6.99 | 9.09 | 0.16 | 0.19 |
| MSE | 0.26 | 0.28 | 5.54 | 18.46 | 48.89 | 82.75 | 0.02 | 0.04 |
| MAE | 3.00 | 3.00 | 1.81 | 3.22 | 5.20 | 6.93 | 0.12 | 0.15 |
| PCC | 0.90 | 0.89 | 0.97 | 0.90 | 0.76 | 0.80 | 0.79 | 0.74 |
| $R^2$ | 0.81 | 0.80 | 0.93 | 0.80 | 0.57 | 0.58 | 47.8 | − 8.50 |

**Table 7**. Best performance of previous regression-based methods as reported in their paper. $R^2$ Coefficient of determination, *PCC* Pearson correlation coefficient, *MAE* Mean average error, *MSE* Mean squared error, *RMSE* Root mean squared error.

| Category | Micro organism | Count | 30–40 °C | 40–50 °C | 50–60 °C | 60–70 °C | 70–80 °C | 80–90 °C |
|---|---|---|---|---|---|---|---|---|
| Psychrophile | *Psychrobacter frigidicola* | 2298 | 0.26% | 24.07% | 53.68% | 16.98% | 4.70% | 0.30% |
| Mesophile | *Shewanella oneidensis* | 4068 | 0.32% | 27.24% | 52.19% | 15.71% | 4.40% | 0.15% |
| Thermophile | *Thermus thermophilus* | 2227 | 0.05% | 4.28% | 31.25% | 32.37% | 22.51% | 9.55% |

**Table 8**. Categorical distribution of microorganisms with their melting temperature ranges.

study employ an ANN + MLP regressor along with composition features derived from protein sequences for prediction. In contrast, the second model utilizes embeddings calculated via ProtBert, a powerful LLM available at https://github.com/agemagician/ProtTrans and https://huggingface.co/Rostlab/prot_bert. The web server has two modules: (i) Predict and (ii) Design. The "Predict" module allows users to predict the Tm value for their sequence on a larger scale. The "Design" module allows the users to generate all the possible analogs by mutating a single residue at a time for each position in a sequence with their predicted Tm values. This module also allows the users to identify the best mutant with the highest potency of being a thermostable protein. This helps in designing the best thermostable proteins by identifying the most important mutation in the input sequence. This module also calculates some physicochemical properties of input sequences, including amphipathicity, hydropathicity, hydrophilicity, charge, hydrophobicity, molecular weight, net hydrogen content, isoelectric point (pI), side-chain bulk, and steric hindrance. The web server was developed using a responsive HTML template and is compatible with smart devices available at https://webs.iiitd.edu.in/raghava/pptstab/). In addition we developed standalone version which is available from GitHub, PIP site and from our web site (https://webs.iiitd.edu.in/raghava/pptstab/stand.php).

## Application of PPTStab

PPTStab is an advanced tool designed to predict the Tm and help identify thermostable proteins. Thermostable proteins, present in a wide array of organisms from extremophiles thriving in harsh conditions to commonly studied model organisms like bacteria, plants, and mammals, are helpful for various purposes[22,23]. Industrial applications of thermostable proteins include enzyme catalysis in biofuel production, food processing, and pharmaceutical manufacturing, where their ability to withstand temperature fluctuations is crucial. It is also helpful in medical research and therapy, serving as a stable component in drug delivery systems, diagnostic assays, and therapeutic agents. Addressing critical clinical needs, such as developing heat-stable vaccines for global distribution and resilient biologics for targeted disease treatments[24].

Our analysis utilized reviewed proteomes from the Uniprot database, focusing on organisms adapted to extreme temperature environments. This includes microorganisms, such as psychrophiles, mesophiles, and thermophiles, which can thrive at temperatures as low as 0 °C and above 60 °C[25]. Among the species analyzed was *Psychrobacter frigidicola*, a gram-negative, non-motile, aerobic, and osmotolerant bacteria. Used in biotechnological processes like restriction endonucleases, uracil-DNA glycosylases, and producing bioactive metabolites with medical applications[26]. *Shewanella oneidensis* is an electrochemically active mesophilic bacterium with a maximum growth temperature of ~ 35 °C, which has been extensively studied for advancing bioelectrochemistry[27]. Lastly, we analyzed *Thermus thermophilus*, a well-known thermophile used in biotechnological applications such as genetic manipulation, structural genomics, and systems biology[28,29]. The summary of our findings is showcased in Table 8.

Here, the percentages indicate how each microorganism is distributed across different temperature ranges. This table shows that *Psychrobacter frigidicola* has the most protein stability at 50–60 °C, indicating its adaptability beyond cold environments. *Shewanella oneidensis* exhibits broad stability from 40 to 70 °C, reflecting its preference for moderate temperatures. Thermus thermophilus has the highest stability at 60–70 °C, which is typical of thermophiles. This analysis highlights that while each microorganism is adapted to a specific temperature range, its proteins display stability across a broader range, underscoring evolutionary flexibility. The analysis also showcases the utility of PPTStab in identifying thermostable proteomes. Our method shows similar percent of protein in different range of melting temperature in case of Psychrophile and Mesophile. It is important to understand why two class of proteins have similar trends. One possible reason for this is our training dataset, which consists only of Thermophilic proteins or proteins with high melting temperatures. To

develop more generalized models, it would be necessary to include Thermophilic, Psychrophilic, and Mesophilic proteins in the training process. The method proposed in this study is specifically optimized for Thermophilic proteins. In summary, the method proposed in this study is optimized explicitly for thermophilic proteins, making it a valuable tool for potential applications in biomedical and biotechnological fields[30].

## Discussion and conclusion

The cost of experimental methods used for identifying the thermostability of proteins are expensive and time-consuming; therefore, the computational techniques are a better alternative to predict the melting temperature of proteins. It is crucial for understanding their behavior and functionality across diverse biological processes and applications. In this study, we have introduced a method called "PPTstab" developed on a non-redundant dataset, unlike previously available methods developed using redundant datasets. Thus, an accurate and efficient prediction method using primary sequences is needed. We obtained the dataset from a previously developed method named 'DeepSTABp', which mostly took data from the Meltome Atlas study. Then, we applied CD-hit (40%) to sequences to reduce redundancy.

Afterward, we applied various machine learning algorithms such as ET, SVR, LGBM, NuSVR, and MLP regressor to different composition-based features and sequence embeddings from the protein language models, along with the 'lysate' or 'cell' flag. We also computed the performance on individual composition descriptors and different LLM model embeddings and merged the composition with LLM embeddings. Among all the composition-based models, SER demonstrated superior performance. Therefore, we combined the SER features with other features to construct the hybrid model, but merging did not achieve good results, so we excluded it. The ensemble model created using ANN with MLP regressor trained on the ProtBert embeddings outperforms every model, achieving an R2 value of 0.80 and a PCC value of 0.89 on the validation dataset, which consists of a 1026-feature vector. The embedding-based models achieved very high performance with an $R^2$ of 0.80, whereas traditional features like SER performed poorly with an $R^2$ of 0.63. This may be because most of the traditional features of proteins like SER are simple static features. In contrast, LLM embedding captures dataset-specific features where a neural network is trained to compute embeddings. One possible explanation for the underperformance of the combined approach is that the SER feature does not provide any additional useful information beyond what is already captured in the embeddings. Essentially, embeddings already contain all the relevant information, including what is present in traditional features, making the addition of SER features redundant.

Despite the similar performance of Prot-Bert and ProtT5-XL-UniRef50 models in training and testing scenarios, we chose Prot-Bert due to its lower computational cost and faster results. While LLMs have made significant contributions across various domains, deploying them effectively poses numerous challenges. These include high training and maintenance costs, scalability issues, limited causality understanding, short attention spans, restricted transfer learning capabilities, and gaps in non-textual context understanding. Additionally, LLMs struggle with generating long-form text, collaborating effectively, handling ambiguity, incremental learning, and managing unstructured data and input errors. Fine-tuning LLMs for specific tasks is useful but requires huge amounts of memory and computing resources, limiting its access to only a few institutions. Therefore, computational demand remains a significant barrier for fine-tuning LLMs[31]. We have provided a user-friendly web server, PPTstab (https://webs.iiitd.edu.in/raghava/pptstab/), for predicting and designing thermostable proteins. We hope this method will significantly contribute to the scientific community working in this domain.

## Materials and methods

### Main dataset acquisition

We have collected 35,114 protein IDs from the DeepSTABp dataset that was primarily derived from the Meltome Atlas[32], which not only provides a collection of thermal proteome profiling (TPP) datasets derived from a multitude of different organisms but also consists of TPP studies involving the heat treatment of cells and proteins lysates[16]. Their corresponding protein sequences were extracted from the UniProt and UniParc databases; it comprises a wide range of species, including Escherichia coli, Bacillus subtilis, Oleispira antarctica, Thermus thermophilus, Pyrococcus torridus, Geobacillus stearothermophilus, Mus musculus, Homo sapiens (K562 and hepat), Saccharomyces cerevisiae, Caenorhabditis elegans, Danio rerio, Arabidopsis thaliana, and Drosophila melanogaster[16,32,33]. We removed sequences having more than 2500 amino acids and sequences containing non-natural amino acids (e.g., U, Z, O, B, J, X). We employed CD-HIT[34] at 40%, a widely used greedy incremental algorithm, for creating a dataset of non-redundant proteins. We ensured that no two sequences had more than 40% similarity, and this process yielded a collection of 17,312 unique protein sequences. After that, to maintain fairness in model training and evaluation, we partitioned the dataset into an 80:20 ratio and obtained 13,849 training sequences and 3463 validation sequences.

### Computation of amino acid composition

Amino acid composition, post-translational modifications, protein-protein interaction, and other molecules like ligands can all impact protein thermal stability[6,35]. To determine the abundance of each amino acid in our datasets, we estimated the composition of a single amino acid using Eq. (1). Then, we assessed how common or uncommon certain amino acids are. First, we separated each dataset into two parts, one with $Tm < 50$ and the other with $Tm > 50$. The Pfeature software's amino acid composition module was used to calculate the composition of two sets of sequences.

$$CR_i = \frac{NR_i}{TR} \tag{1}$$

where CRi denotes the composition of residue i; NRi indicates the total count of residues of type i; and TR refers to the overall total of residues.

## Features engineering methods

*Composition-based feature vectors*

We implemented the composition module of Pfeature software[36] to represent the sequence as a numerical vector. It is a library that calculates features from sequences and the structure of the proteins and peptides. It includes five different modules for computing the features: (a) Composition-based, (b) Binary-profile-based, (c) Evolutionary information-based, (d) Structure-based, and (e) Pattern-based. Utilizing Pfeature, we calculated a diverse array of composition-based features, encompassing 15 distinct types, including amino acid composition (AAC), residue repeat information (RRI), distance distribution of residues (DDOR), atomic composition (ATC), bond composition (BTC), composition based on physicochemical properties (PCP), conjoint triad calculation (CTC), composition enhanced transition and distribution (CeTD), Shannon entropy of the entire protein (SEP), Shannon entropy for all residues (SER), Shannon entropy derived from physicochemical properties (SPC), quasi-sequence order (QSO), sequence order coupling number (SOCN), pseudo amino acid composition (PAAC), and amphiphilic pseudo amino acid composition (APAAC). The vector matrix of these descriptors is presented in Table 9.

*Embeddings feature vectors*

We created sequence embeddings utilizing ProtBert, a pre-trained protein language model (PLM) founded on the BERT natural language processing algorithm[37,38]. This model was trained on the Big Fantastic Database (BFD), which comprises more than 2.3 million protein sequences. From the final hidden layers of ProtBert, we obtained 1024-dimensional vectors that serve as the representations of protein sequences, commonly referred to as embeddings. The main distinction between this model and the original Bert model lies in its approach to sequence handling; unlike the original Bert, which utilizes the next-sentence prediction technique, ProtBert treats each sequence as a complete document. It employs a random masking style for up to 15% of amino acids as input. Our methodology involves training through tokenizing sequences composed of uppercase amino acids, with a single space separating each token. The vocabulary consists of 20 distinct tokens, each representing the linear structures of the 20 standard amino acids (A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, W, Y, V), while non-natural amino acids are not included. For optimization, we employed Lamb, a layerwise adaptive large batch optimization technique with a learning rate set at 0.002 and a weight decay parameter of 0.01. This approach yields an embedding vector of 1024 dimensions, situating each sequence within a high-dimensional space to encapsulate the nuanced relationships among amino acids by analyzing the context of their occurrence[39].

Additionally, we extracted embeddings using various LLMs such as Ankh, ProtGPT2, ProtT5-XL-Uniref50, and ProsT5. The Ankh model represents a pioneering general-purpose protein language model developed utilizing Google's TPU-v4, featuring a reduced parameter count of less than 10% for pre-training, under 7% for inference, and below 30% for the embedding dimension[40]. ProtGPT2, conversely, is a decoder-only transformer model that has undergone pre-training on the UniRef50 protein database (version 2021_04) with a causal modeling objective, employing the GPT2 transformer architecture. This model comprises 36 layers and has a dimensionality of 1280, amounting to 738 million parameters[41]. ProtTrans, particularly ProtT5-XL-U50, is derived from the T5-3b model, which is a text-to-text Transfer Transformer (T5) pre-trained in an extensive collection of protein sequences in a self-supervised manner, utilizing raw protein sequences without any human annotations to generate inputs and labels automatically. Unlike the original T5-3b, which used a span denoising objective, ProtT5-XL-UniRef50 employed a BART-like masked language model (MLM) denoising objective[38]. ProsT5, also known as protein structure-sequence T5, is a bilingual protein language model designed to translate

| Type of feature | Vector size |
|---|---|
| AAC | 20 |
| APAAC | 23 |
| ATC | 5 |
| BTC | 4 |
| CeTD | 189 |
| CTC | 343 |
| DDR | 20 |
| PAAC | 21 |
| PCP | 30 |
| QSO | 42 |
| RRI | 20 |
| SEP | 1 |
| SER | 20 |
| SOCN | 2 |
| SPC | 25 |

**Table 9**. Description of the features with their vector matrix extracted using pfeature.

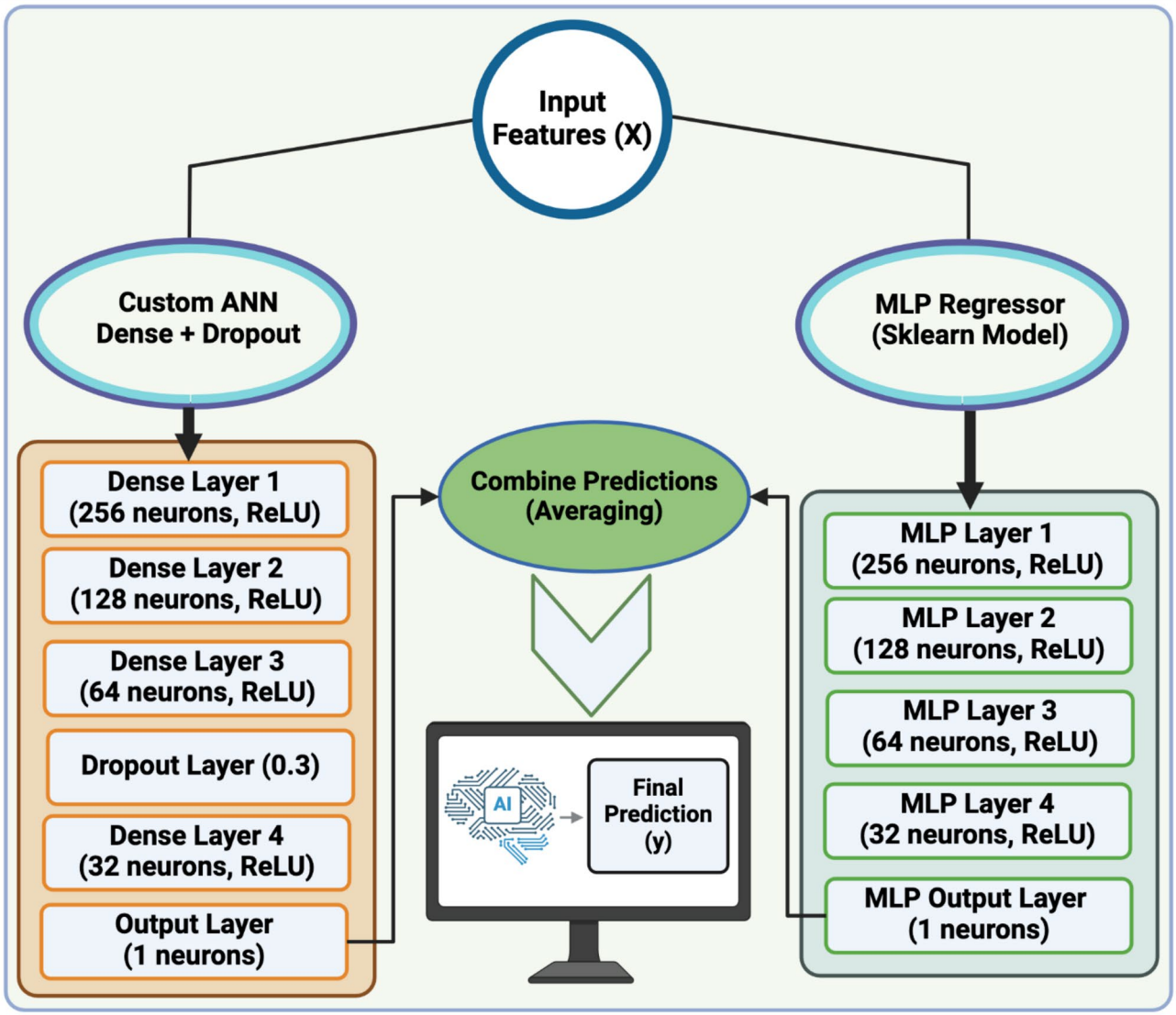



**Fig. 3**. The architecture of ANN + MLP model.

between protein sequences and structures, developed by fine-tuning ProtT5-XL-U50 with a dataset of 17 million proteins and high-quality 3D structure predictions from AlphaFoldDB[42].

## Model development

We started by evaluating multiple algorithms to construct a robust predictor. For this process, we utilized H2O AutoML, a scalable automatic machine learning library, version 3.46.0.4[43]. We applied all available regression algorithms to our training dataset and evaluated their performance using $R^2$ and PCC regression-based metrics. We chose the top 6 performing models from the evaluations, including SVR, ET, NuSVR, LGBM, CatBoost, and MLP. These algorithms were then implemented using the scikit-learn library for Python, while LGBM was implemented in the standard LightGBM implementation[44,45]. Throughout our evaluation, we implemented them using default parameters and aimed to identify the model demonstrating the highest predictive capability for our specific task.

Ultimately, we created a custom ANN model using TensorFlow, consisting of four dense hidden layers with 256, 128, 64, and 32 units, activated by a ReLU (rectified linear unit), followed by an output layer[46]. The model is trained using means squared error loss and the Adam optimizer to prevent overfitting[47]. Furthermore, the custom ANN model and the MLP regressor model were integrated to create a unified predictive framework (Fig. 3), enabling a comprehensive evaluation of their collective effectiveness in prediction.

## Cross validation

In order to train, test, and evaluate our prediction models, we employed K-fold cross-validation and external validation techniques, following the standard protocols mentioned in previous studies[48–50]. The dataset was split into training (80%) and validation (20%) sets, with validation as an independent dataset. For model development,

only the training dataset was used with k = 10, where k−1 folds were used for the training, while the remaining folds were used for testing. The process was repeated iteratively, ensuring that each fold was tested once after training on the remaining K−1 folds. This cross-validation approach comprehensively evaluates the model's performance across different data segments. Additionally, we ensured that no protein from the validation dataset was used in training and testing. Furthermore, no protein in the validation dataset shares 40% or more sequence similarity with any protein in the training dataset[51].

## Normalization

Normalization is a crucial preprocessing technique that ensures all computed features are scaled to a uniform range. This step is crucial as it enhances model stability, accelerates convergence, and improves overall performance by preventing features with larger magnitudes from dominating those with smaller values.

In our approach, we used the Min–Max Scaler (MMS) to transform feature values into a standardized range of [0, 1] or [− 1, 1], depending on the requirements of the machine learning model[52]. This transformation ensures that all input features contribute proportionally to the learning process, optimizing the model's ability to detect meaningful patterns. The Min-Max scaling process is mathematically represented as follows:

$$x' = \frac{x - min\,(x)}{(x) - min\,(x)} \tag{2}$$

where x′ is the normalized value, x is the original feature value, and min(x) and max(x) represent the feature's minimum and maximum values, respectively.

Before training, we normalize the composition-based features in the training dataset using the Min–Max Scaler, which scales value to a range of [0, 1] or [− 1, 1]. Similarly, we apply the same parameters derived from the training sets to the independent datasets to ensure consistency. This process helps maintain uniformity, improves generalization, and enhances the overall performance of our proposed method.

## Performance assessment metrics

We computed five standard evaluation metrics to assess our regression model's effectiveness. These metrics are essential for assessing the predictive capability of regression models across various dimensions.

Coefficient of determination ($R^2$): The proportion of variance in the dependent variable that is predictable from the independent variable is called the coefficients of determination. R2 estimates how close the data is to the regression line. The closer the value is to 1, the better the regression model is.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum_i (y_i - x_i)^2}{\sum_i (y_i - \overline{y})^2} \tag{3}$$

$SS_{tot}$ is the total sum of squares, and $SS_{res}$ is the sum of squares of residuals. yi is the true value, and xi is the prediction.

Mean squared error (MSE): The mean squared error is the average of the squares of the errors. The difference between the estimated and actual values is called the average squared difference.

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - x_i)^2 \tag{4}$$

A sample of N data points is used to generate an N prediction, with the *xi* observed values of the *yi* variable being predicted.

Mean average error (MAE): MAE indicates the error between paired values for predictions and observations.

$$MAE = \frac{\sum_{i=1}^{N} |y_i - x_i|}{N} \tag{5}$$

where yi is the prediction, and xi is the true value.

Root mean squared error (RMSE): The RMSE measures the differences between the predicted and observed values.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - x_i)^2}{N}} \tag{6}$$

The predicted value is yi, and the experimental value is xi.

Pearson correlation coefficient (PCC): The Pearson correlation coefficient quantifies the relationship between two data sets. A normalized covariance measurement is the difference between the product of two variables' standard deviations. The range is − 1 to 1 and is represented in a mathematical way as:

$$PCC = \frac{cov\,(X, Y)}{\sigma_X\,\sigma_Y} = \frac{(E[\,(X - \mu_X)(Y - \mu_y)\,])}{\sigma_X\,\sigma_Y} \tag{7}$$

Here, $\sigma_X$ is the standard deviation of X, $\sigma_Y$ is the standard deviation of Y, $\mu_X$ is the mean of X, $\mu_Y$ is the mean of Y, and E is the expectation[53].

## Data availability

All the datasets used in this study are available at the "PPTstab" web server, https://webs.iiitd.edu.in/raghava/pptstab/data.html.

Received: 16 October 2024; Accepted: 14 April 2025
Published online: 14 May 2025

## References

1. Timr, S., Madern, D. & Sterpone, F. Protein thermal stability. *Prog. Mol. Biol. Transl. Sci.* **170**, 239–272 (2020).
2. Almeida, P. *Proteins (Garland Sci.)*. https://doi.org/10.1201/9780429258817. (2016).
3. Gorania, M., Seker, H. & Haris, P. I. Predicting a protein's melting temperature from its amino acid sequence. *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.* **2010**, 1820–1823 (2010).
4. Kambourova, M. Thermostable enzymes and polysaccharides produced by thermophilic bacteria isolated from Bulgarian hot springs. *Eng. Life Sci.* **18**, 758–767 (2018).
5. Li, M. et al. A deep learning algorithm for prediction of melting temperature of thermophilic proteins directly from sequences. *Comput. Struct. Biotechnol. J.* **21**, 5544–5560 (2023).
6. Bischof, J. C. & He, X. Thermal stability of proteins. *Ann. N. Y. Acad. Sci.* **1066**, 12–33 (2006).
7. Wen-qi, Z. C. Y. Y. U. M.-D. Comparation of three measuring methods for thermodynamic stability of protein. *Anal. Test. Technol. Instrum.* **27**, 252–259 (2021).
8. Yang, Y. et al. ProTstab - predictor for cellular protein stability. *BMC Genom.* **20**, 804 (2019).
9. Chen, C. W., Lin, M. H., Liao, C. C., Chang, H. P. & Chu, Y. W. iStable 2.0: predicting protein thermal stability changes by integrating various characteristic modules. *Comput. Struct. Biotechnol. J.* **18**, 622–630 (2020).
10. Charoenkwan, P., Chotpatiwetchkul, W., Lee, V. S., Nantasenamat, C. & Shoombuatong, W. A novel sequence-based predictor for identifying and characterizing thermophilic proteins using estimated propensity scores of dipeptides. *Sci. Rep.* **11**, 23782 (2021).
11. Yang, Y., Zhao, J., Zeng, L. & Vihinen, M. ProTstab2 for prediction of protein thermal stabilities. *Int. J. Mol. Sci.* **23**, 10798 (2022).
12. Meng, C., Ju, Y., Shi, H. & TMPpred: A support vector machine-based thermophilic protein identifier. *Anal. Biochem.* **645**, 114625 (2022).
13. Charoenkwan, P. et al. A stacking-based ensemble learning framework for accurate prediction of thermophilic proteins. *Comput. Biol. Med.* **146**, 105704 (2022).
14. Zhao, J., Yan, W. & Yang, Y. DeepTP: A deep learning model for thermophilic protein prediction. *Int. J. Mol. Sci.* **24**, 2217 (2023).
15. Pei, H. et al. Identification of thermophilic proteins based on sequence-based bidirectional representations from transformer-embedding features. *Appl. Sci. (Basel)*. **13**, 2858 (2023).
16. Jung, F., Frey, K., Zimmer, D., Mühlhaus, T. & DeepSTABp A deep learning approach for the prediction of thermal protein stability. *Int. J. Mol. Sci.* **24**, (2023).
17. Haselbeck, F. et al. Superior protein thermophilicity prediction with protein Language model embeddings. *NAR Genom. Bioinform.* **5**, lqad087 (2023).
18. Pudžiuvelytė, I. et al. TemStaPro: protein thermostability prediction using sequence representations from protein language models. *Bioinformatics*. **40**, (2024).
19. Ponnuswamy, P. K., Muthusamy, R. & Manavalan, P. Amino acid composition and thermal stability of proteins. *Int. J. Biol. Macromol.* **4**, 186–190 (1982).
20. Raghava, G. P. S. & Han, J. H. Correlation and prediction of gene expression level from amino acid and dipeptide composition of its protein. *BMC Bioinform.* **6**, 59 (2005).
21. Marquet, C. et al. Embeddings from protein language models predict conservation and variant effects. *Hum. Genet.* **141**, 1629–1647 (2022).
22. Dehouck, Y., Folch, B. & Rooman, M. Revisiting the correlation between proteins' thermoresistance and organisms' thermophilicity. *Protein Eng. Des. Sel.* **21**, 275–278 (2008).
23. Cavicchioli, R., Siddiqui, K. S., Andrews, D. & Sowers, K. R. Low-temperature extremophiles and their applications. *Curr. Opin. Biotechnol.* **13**, 253–261 (2002).
24. Zhao, L., Zhao, J., Zhong, K., Tong, A. & Jia, D. Targeted protein degradation: mechanisms, strategies and application. *Signal. Transduct. Target. Ther.* **7**, (2022).
25. García-Descalzo, L., García-López, E. & Cid, C. Comparative proteomic analysis of psychrophilic vs. Mesophilic bacterial species reveals different strategies to achieve temperature adaptation. *Front. Microbiol.* **13**, 841359 (2022).
26. García-López, M. L., Santos, J. A., Otero, A. & Rodríguez-Calleja, J. M. Psychrobacter. In *Encyclopedia of Food Microbiology* 261–268. https://doi.org/10.1016/b978-0-12-384730-0.00285-8 (Elsevier, 2014).
27. Ikeda, S. et al. Shewanella oneidensis MR-1 as a bacterial platform for electro-biotechnology. *Essays Biochem.* **65**, 355–364 (2021).
28. Cava, F., Hidalgo, A. & Berenguer, J. Thermus thermophilus as biological model. *Extremophiles* **13**, 213–231 (2009).
29. Wikipedia contributors. Thermus thermophilus. Wikipedia, The Free Encyclopedia. en.wikipedia.org/w/index.php?title=Thermus_thermophilus&oldid=1228176888 (2024).
30. Rigoldi, F., Donini, S., Redaelli, A., Parisini, E. & Gautieri, A. Review: Engineering of thermostable enzymes for industrial applications. *APL Bioeng.* **2**, 011501 (2018).
31. Hadi, M. U. et al. Large Language Models: A comprehensive survey of its applications, challenges, limitations, and future prospects. https://doi.org/10.36227/techrxiv.23589741.v4 (2023).
32. Jarzab, A. et al. Meltome atlas-thermal proteome stability across the tree of life. *Nat. Methods*. **17**, 495–503 (2020).
33. UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Res.* **49**, D480–D489 (2021).
34. Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150–3152 (2012).
35. Leuenberger, P. et al. Cell-wide analysis of protein thermal unfolding reveals determinants of thermostability. *Science* **355**, (2017).
36. Pande, A. et al. Pfeature: A tool for computing wide range of protein features and Building prediction models. *J. Comput. Biol.* **30**, 204–222 (2023).
37. Devlin, J., Chang, M. W., Lee, K. & Toutanova, K. BERT: Pre-training of deep bidirectional Transformers for Language Understanding. https://doi.org/10.48550/ARXIV.1810.04805 (2018).
38. Elnaggar, A. et al. ProtTrans: toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7112–7127 (2022).
39. Elnaggar, A. et al. ProtTrans: towards cracking the language of life's code through self-supervised learning. *BioRxiv* https://doi.org/10.1101/2020.07.12.199554 (2020).
40. Elnaggar, A. et al. Ankh: Optimized protein language model unlocks general-purpose modelling. https://doi.org/10.48550/ARXIV.2301.06568 (2023).
41. Ferruz, N., Schmidt, S. & Höcker, B. ProtGPT2 is a deep unsupervised Language model for protein design. *Nat. Commun.* **13**, (2022).

42. Heinzinger, M. et al. Bilingual language model for protein sequence and structure. *BioRxiv*. https://doi.org/10.1101/2023.07.23.550085 (2023).
43. Ledell, E. & Poirier, S. H2O AutoML: Scalable Automatic Machine Learning. (2020).
44. Scikit -learn. http://scikit-learn.sourceforge.net.
45. Ke, G. et al. Curran Associates Inc, Red Hook, NY, USA,. LightGBM: a highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)* 3149–3157 (2017).
46. Artificial Neural Networks. https://doi.org/10.1007/978-3-319-09903-3 (Springer International Publishing, 2015).
47. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. https://doi.org/10.48550/ARXIV.1412.6980 (2014).
48. Changli, F. et al. A method for prediction of thermophilic protein based on reduced amino acids and mixed features. *Front. Bioeng. Biotechnol.* **8**, 285 (2020).
49. Li, Y., Middaugh, C. R. & Fang, J. A novel scoring function for discriminating hyperthermophilic and mesophilic proteins with application to predicting relative thermostability of protein mutants. *BMC Bioinform.* **11**, 62 (2010).
50. Pucci, F., Bourgeas, R. & Rooman, M. Predicting protein thermal stability changes upon point mutations using statistical potentials: introducing HoTMuSiC. *Sci. Rep.* **6**, 23257 (2016).
51. Jung, Y. Multiple predicting*K*-fold cross-validation for model selection. *J. Nonparametr. Stat.* **30**, 197–215 (2018).
52. Cabello-Solorzano, K., Ortigosa de Araujo, I., Peña, M., Correia, L. & Tallón-Ballesteros, J. A. The impact of data normalization on the accuracy of machine learning algorithms: A comparative analysis. In *Lecture Notes In Networks and Systems* 344–353. https://doi.org/10.1007/978-3-031-42536-3_33 (Springer Nature Switzerland, 2023).
53. Pearson's correlation coefficient. In *Encyclopedia of Public Health* 1090–1091. https://doi.org/10.1007/978-1-4020-5614-7_2569 (Springer Netherlands, 2008).


## Acknowledgements

Authors are thankful to the University Grants Commission (UGC), and Department of Biotechnology (DBT) for fellowships and financial support, and the Department of Computational Biology, IIITD New Delhi for infrastructure and facilities. We would like to acknowledge that Figures were created using BioRender.com. The current work has been supported by the Department of Biotechnology (DBT) grant BT/PR40158/BTIS/137/24/2021.

## Author contributions

G.P.S.R. collected the dataset. P.T., and N.K. processed the dataset. P.T., and G.P.S.R. implemented the algorithms and developed the prediction models. N.K., P.T., and G.P.S.R. analysed the results. N.K. created the front-end and back-end of the webserver. N.K., P.T., and G.P.S.R. penned the manuscript. G.P.S.R. conceived and coordinated the project. All authors have read and approved the final manuscript.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary Information** The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-98667-9.

**Correspondence** and requests for materials should be addressed to G.P.S.R.