# Towards thermophilic protein stability prediction: A comprehensive study of machine learning approaches

**Xin Wang**

School of Computing, Australian National University, 108 North Road, Action, ACT 2601, Australia

Xin.Wang1@anu.edu.au

**Abstract.** Thermophilic proteins are critical in biology due to their enhanced thermal stability. Different machine learning approaches have been applied to estimate protein thermal stability. Current study categorizes the previous research into classification and regression tasks and explores the impact of different data representations, including tabular, sequence, and graph data on evaluation performance. Pipelines for prediction using different representations are thoroughly described. Current challenges, such as insufficient and imbalanced datasets, are addressed with potential solutions, such as transfer learning and re-sampling methods. Additionally, model interpretability, discussing various approaches to obtain model explanations and highlighting that some explanations are inconsistent have also been included in current study. Such comprehensive overview provides insights into existing methodologies and suggests potential research directions and improvements.

**Keywords:** The paper must have at least three keywords. Protein Thermal Stability, Machine Learning, Thermophilic Proteins, Protein Sequence.

## 1. Introduction

The applications of thermally stable proteins span various industries, including drug design, biological techniques, and food chemistry [1]. Proteins' ability to stay structurally and functionally intact at high temperatures is essential for these applications and determines their usability in various contexts [2]. The Melting Temperature (Tm) represents protein thermal stability, signifying the temperature where 50% of the proteins lose their natural structure and activity. In some researches proteins are classified as thermophilic or non-thermophilic based on their thermal stability [3-12]. In summary, calculating or predicting the thermal stability of proteins is a significant task in bioinformatics and industry.

Before the widespread application of deep learning techniques, experimental approaches were commonly used to evaluate the thermal stability of proteins. For example, the Thermal Proteome Profiling (TPP) method performs mass spectrometry on the supernatant of proteins at different temperatures and plots a melting curve to determine the Tm [2]. However, such experimental methods have several drawbacks: they are labour-intensive, require high material and equipment costs, and are time-consuming, making it nearly impossible to determine the thermal stability of a large number of proteins. Additionally, experimental methods can introduce errors during sample preparation, temperature control, and other steps. Moreover, not all proteins are suitable for experimental methods [2]. In summary, experimental approaches are not ideal for measuring the thermal stability of proteins.

On the other hand, computational methods, such as statistical or machine-learning-based approaches, stand out for their lower cost and acceptable computational time. Machine learning methods are widely used for their ability to learn from data and provide accurate results. These models typically approach this prediction as either a classification or regression task. Regression provides specific prediction values, while classification categorizes proteins into different groups, such as high/low stability or temperature intervals. There are three types of machine learning techniques for this task: feature-based models (e.g., Multi-Layer Perceptron), sequence-based models (e.g., Recurrent Neural Networks), graph-based models (e.g., Graph Neural Networks). An illustration of how different representations are generated and used is shown in Figure 1. In the upcoming sections, we will explore state-of-the-art research with different data representations in detail, provide suggestions on the applications of different models, and propose possible future research directions.

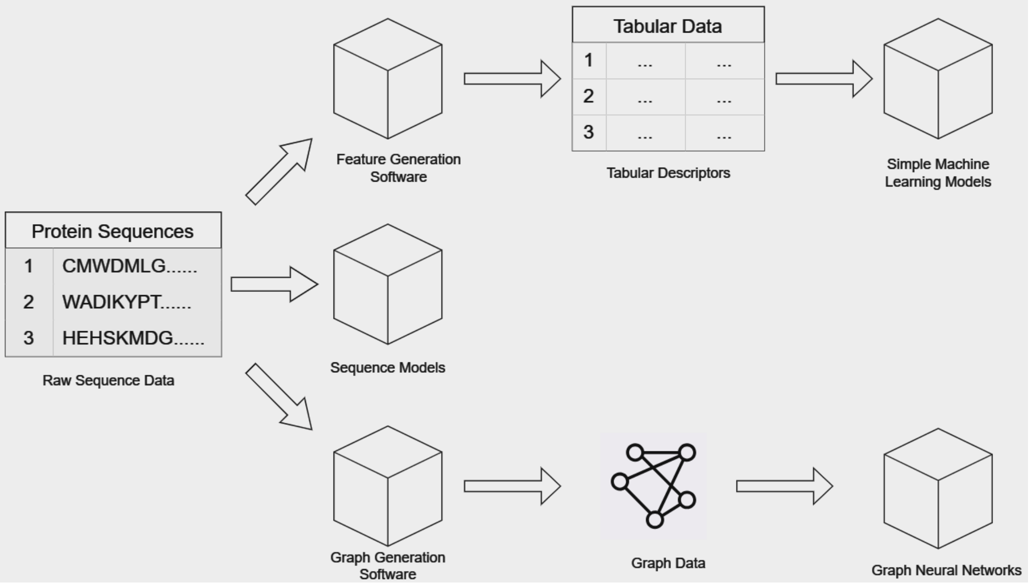

**Figure 1.** Illustration of how different representations are generated and utilized for different models. Sequence models directly use raw sequence data, tabular and graph data require generation software.

## 2. Tasks and datasets

### *2.1. Regression and classification tasks*

Research efforts in predicting protein thermal stability can be grouped into two major methods. Some studies treat this as a classification task, where proteins are classified as thermophilic or non-thermophilic [3-12]. Other studies use regression models to predict the Tm value of proteins [1-2,13]. Classification tasks categorize proteins based on their Optimal Growth Temperature (OGT), while regression tasks directly predict the Melting Temperature (Tm) of proteins. Some studies found Tm and OGT are closely related [14,15]. Tm directly indicates thermal stability, whereas OGT represents thermal stability through the hypothesis that species with higher OGT have proteins with better thermal stability. The threshold of OGT that determines thermophilic classification varies in different studies. Some studies use 50°C as the threshold, which is not an objective threshold [11]. Although OGT can indicate thermal stability, the relationship is indirect and subject to variance. Instead of using a single threshold, Lin and Chen classified proteins with OGT > 60°C as thermophilic and OGT < 30°C as non-thermophilic [16]. This standard addresses the indirect relationship but also ignores proteins with an OGT between 30°C and 60°C. Most subsequent classification studies follow this criterion or directly use Lin and Chen’s labelled dataset [3-5,7-9,11].

Generally, regression tasks are more difficult and considered superior to classification tasks because they provide precise prediction values. In recent years, with more sophisticated feature extraction and machine learning techniques, regression tasks with different data representations have begun to emerge [1,2,13]. Another difference is that these tasks are evaluated by different metrics. In machine learning, regression tasks are assessed with the similarity of model predictions and actually values, by metrics such as R-squared and Pearson Correlation Coefficient (PCC). Classification tasks are assessed based on how accurately the model classifies instances using metrics such as Accuracy, Precision, Recall, and ROC-AUC. Besides predicting thermal stability, some research focuses on predicting changes in stability with protein mutations (amino acid variants) [17]. This study focuses on the direct prediction of thermal stability.

### *2.2. Datasets for model training*

*2.2.1. Dataset status.* Supervised models, which are machine learning models that predict target values, are trained on pre-labelled datasets. Most regression tasks are trained on the Meltome atlas dataset, which contains the Tm values of 48,000 proteins from 13 species, with Tm values spanning from 30°C to 90°C [18]. Another Tm dataset used in the ProTstab model, the initial version of ProTstab2, contains 3,520 proteins from humans and bacteria [13,19]. For classification tasks, the situation is more complicated. Lin and Chen's dataset is widely used in many studies; it contains 915 thermophilic proteins and 793 non-thermophilic proteins [16]. Some subsequent studies directly use this dataset or combine it with other datasets [5-9]. Meanwhile, some researchers collect their own datasets, which are then used in later studies. For example, iThermo collected a dataset of 1,442 thermophilic proteins and 1,366 non-thermophilic proteins, later used in the BertThermo model [4,11]. Furthermore, some studies focus on creating a protein thermal dataset only, such as the ProtDataTherm database [20]. However, current classification research prefers using datasets that have already been validated and tested in other researches. For fair model comparisons, researchers test their models and other state-of-the-art models with independent blind-test datasets, often retrieved from unrelated research [12]. In summary, the current situation of protein thermal stability databases is quite complicated.

*2.2.2. Insufficient data.* The performance of machine learning models is determined by the amount of training data available, as larger datasets improve the model's generalization ability. Large protein language models (pLMs), like ESM and ProtTrans, undergo training using a vast dataset comprising millions of protein sequences [21-23]. However, current protein thermal databases are quite limited in size. ProthermDB, a thermodynamic database for proteins and mutants, contains 120,000 records, but fewer than 10,000 of them include Tm values [24]. As noted in Section 2.2.1, the current Tm database contains approximately tens of thousands of entries, while the thermophilic proteins database contains several hundred to a few thousand entries.

Experimental methods are not ideal for expanding datasets due to high material and time costs, and they are not universally suitable for all proteins [1-5,10,12,13]. Transfer learning offers a promising solution by learning protein representations from other related tasks and then using these extracted representations as input features for thermal prediction. Semi-supervised learning involves learning protein representations from unlabeled data by creating tasks like predicting pseudo-labels, followed by labelled downstream tasks. Both approaches can address the issue of limited dataset size. For example, BertThermo pre-trained a BERT model on an unlabeled dataset and fine-tuned it on labelled data [4,25]. TemStaPro used the pLM ProtTrans for generating representations [12,23].

*2.2.3. Imbalanced data.* There is an imbalance in the natural occurrence of thermophilic and non-thermophilic proteins, with non-thermophilic proteins being more prevalent [3]. For example, in the dataset collected for DeepTM, 83% of protein Tm values range from 40°C to 60°C, while only 0.4% are lower than 30°C [1]. This type of imbalanced data leads to biased models that perform better with instances close to the major instances in the feature space. In DeepTM, their model performs better in the temperature range of 50°C to 60°C than 60°C to 80°C, where there are 4,785 instances in the first interval and 2,005 in the second. This imbalance problem can be addressed with two main approaches: re-sampling and data augmentation.

Re-sampling is a data preprocessing technique specifically designed for imbalanced data and can be divided into two categories: undersampling, which reduces the majority data instances, and oversampling, which increases the minority data instances. SMOTE, or Synthetic Minority Over-sampling Technique, is a widely-used method that generates new instances between minority instances in the feature space [26]. BertThermo applied the SMOTE oversampling technique to their imbalanced dataset with 1,443 negative and 1,366 positive instances [4]. On the other hand, DeepTP applied an undersampling technique to randomly remove mesophilic proteins to achieve a balanced dataset [3]. However, considering the issue of insufficient data, oversampling might be a more suitable approach.

Data augmentation is a technique originally designed to address insufficient data, includes techniques such as “uniform sampling” (US) and “extended uniform sampling” (EUS) introduced by Jung et al [2]. These techniques modify the Tm values of proteins from a uniform distribution, with the distribution's bounds corresponding to the Tm range of the respective organism. However, there is a trade-off: while applying such a method can address the problem of imbalanced data by ensuring the sampled data follows a uniform distribution, the Tm values in the training set are not ground truth values but randomly sampled values. Using such a dataset with synthetic target values could potentially lead to poor performance and lower interpretability. Nevertheless, as per Lang et al.'s research, they succeeded in attaining top-tier prediction accuracy.

Moreover, since there are more thermophilic proteins than non-thermophilic ones in nature, models are likely to encounter applications in industry where they need to make predictions on imbalanced datasets. Therefore, model performance on imbalanced test datasets should be evaluated [3]. For example, DeepTP constructed an imbalanced test dataset with 30 thermophilic proteins and 1,800 non-thermophilic proteins [3].

### *2.3. Protein sequence pre-processing*

After acquiring the raw dataset, several pre-processing steps are necessary before forming the training, validation, and test datasets. These steps are as follows:

(1) Remove duplicates, especially when the dataset is created by merging multiple existing datasets [1, 9].

(2) Remove protein sequences that are not annotated or manually reviewed [5,7,11].

(3) Remove sequences with non-standard or ambiguous amino acid residues [1,6,7,11,13] (e.g., ‘B’, ’U’, ’X’, ’Z’).

(4) Remove sequences that are part of another protein or contain other protein fragments [3,5,7,11].

(5) Remove sequences whose target value is obtained from prediction or homology instead of experiments [5,9,11].

(6) Remove sequences with extreme target values or sequence lengths [1,3].

(7) Apply the CD-HIT method. CD-HIT is a clustering method initially introduced by Li et al. in 2001, which represents a set of similar sequences with a single sequence, thereby removing highly similar sequences [27]. This step is crucial because similar sequences can affect the model similarly to imbalanced data [1,3,5-7,10,11].

## 3. Tabular representation and models

A typical pipeline for tabular-feature-based protein thermal stability prediction includes feature generation, feature selection, model training, and testing.

### *3.1. Feature generation*

Predicting the thermal stability of proteins using properties that are easy to compute is one of the most classic methods. The extracted properties are presented in a tabular data format. This data representation integrates well with simple machine learning methods, such as Multi-Layer Perceptron. There are mainly two types of features for proteins: Physicochemical properties (e.g., hydrophobicity and polarity), which need to be calculated using specialized software. Or sequence-based features that describe the composition of protein sequences, such as AAC (Amino Acid Composition), which can be computed by software or manually.

The automatic generation of protein properties is a well-researched area, with numerous mature software tools, programming language packages, and web services developed for this purpose. For example, the iFeature Python package can generate 53 types of feature descriptors [28]. The R package Protr generated 6,295 different descriptors for the ProTstab2 model [13,29]. Web services like ProtDCal can compute a multitude of protein features, including 3D structure and physical and chemical parameters [30].

### *3.2. Feature selection*

Although thousands of feature descriptors can be automatically generated, having more features does not necessarily lead to a better model. Irrelevant, noisy, or useless features can force the model to learn incorrect patterns and decrease performance. Highly relevant or similar features might interact and cause overfitting, resulting in the model excelling on the training set but exhibiting poor generalization to new data. Additionally, fewer features result in faster training times. Feature selection methods currently fall into three main categories:

(1) Feature selection methods that allow specifying the number of features: Recursive Feature Elimination (RFE) is a typical feature selection technique that recursively removes the least important features and outputs a feature importance ranking, allowing researchers to manually select the number of features to use [31]. Similarly, Analysis of Variance (ANOVA) computes the ANOVA F-value for each feature (a high F-value indicates significant influence on the target value) and outputs a feature importance ranking [32,11]. The number of top-important features to use can be determined using techniques like cross-validation or treated as a hyperparameter.

(2) Feature selection methods do not specify the number of features: RFECV is a variation of RFE that performs feature elimination during cross-validation, automatically computing the optimal subset of features. Despite its convenience, RFECV does not always guarantee the best performance. For example, the ProTstab2 model achieved better performance with 200 RFE-selected features compared to 1,214 RFECV-selected features [2].

(3) Principal Component Analysis (PCA): PCA is a dimensionality reduction method and is extensively used in diverse machine learning models [8]. Technically, it is not a feature selection method, as it does not calculate important features but instead creates a projection from high-dimensional raw features to low-dimensional new features while retaining the maximum variance. PCA generates new features rather than selecting existing ones. The drawback of PCA is that the new features lack interpretability, as their original meanings are unknown.

### *3.3. Model training*

After generating and selecting features, each protein is encoded as a fixed-length feature vector. Numerous simple machine learning models can utilize these vectors as inputs, including both regression and classification models. For example, TMPpred used a Support Vector Machine (SVM), while Guo et al. employed the LIBSVM model [5,7]. Simple machine learning models are easy and fast to train, allowing researchers to test multiple models to identify the best performance. For example, Feng et al. utilized four simple models while ProTstab2 tested seven models [13,8].

Stacking is an ensemble learning method where simple models are treated as base models, and the outputs of these base models are used as inputs to a new meta-model. For example, the SAPPHIRE model combined 12 groups of features with six simple models, resulting in 72 base models. This 72-dimensional vector was then trained using a Partial Least Squares (PLS) model [10]. The stacking method can significantly improve model performance.

## 4. Sequence representation and models

Simple machine learning models can only accept fixed-length vectors as inputs. However, converting protein sequences into tabular feature descriptors often results in information loss. With advancements in Natural Language Processing (NLP) models, and considering the parallels between protein sequences and natural language, NLP techniques are increasingly used for analyzing protein sequences [2-4,9,12]. Although these models have complex structures, they are easier to use because manual feature selection is not required. A typical pipeline for protein sequence models involves generating amino acid embeddings, creating sequence representations, and then using regression or classification models, often implemented with several fully connected layers.

### *4.1. Sequence embeddings*

The first step in protein sequence models is creating amino acid embedding vectors. The simplest method for doing this is one-hot encoding, which creates a sparse vector for each amino acid, where only the index for that amino acid is set to 1 and all others are set to 0. A more popular method is to encode amino acids as integers and use embedding layers to generate the embedding vectors. For example, DeepTP created embedding vectors using encodings based on amino acid composition and physicochemical properties [3]. Most modern models utilize pre-trained embedding models. ProtTrans embeddings are used by DeepSTABp and TemStaPro [2,12,23]. Pre-trained pLM embeddings are generally a better choice since they are trained on large datasets and are easier to use than implementing custom embedding methods.

### *4.2. Sequence models*

In NLP, after generating word embeddings, the sequence is represented by a matrix of embedding vectors. This matrix is then input into complex models like Bidirectional Long Short-Term Memory (BiLSTM) networks or transformers, as well as simpler models like 1-D Convolutional Neural Networks (CNNs), to create a vector representation for each sequence. This is followed by fully connected layers for regression or classification. The DeepTP model follows this pipeline, using CNN and BiLSTM layers for sequence representation [3]. However, most protein thermal stability prediction models obtain sequence representations by simple average pooling, which computes the average of all embedding vectors [2,4,9]. Although these methods achieve good performance, sequence models like BiLSTM are generally a better choice because they can capture relationships between amino acids and generate more informative sequence representations.

In addition to the methods mentioned above, models like TemStaPro directly generate sequence representations from pre-trained language models (pLMs) rather than from amino acid embeddings [12]. Moreover, the BertThermo model uses simple classification models such as Logistic Regression after obtaining the sequence representations [4]. In summary, the optimal method for processing amino acids to obtain a protein representation that enhances performance remains an open question.

The combination of feature descriptors and sequence representations is another promising direction for future advancements. Sequence models may sometimes fail to extract certain meaningful information. For example, the DeepSTABp model combined sequence representation with the Optimal Growth Temperature (OGT) of proteins [2]. Similarly, DeepTP generated 205 features in the same way as ProTstab2 and combined them with sequence representations [13,3]. This type of ensemble model offers better results and interpretability than using sequence models alone [3].

## 5. Graph representation and models

Mathematical graphs are defined as sets of nodes and edges. In the context of proteins, amino acids can be considered as nodes, and interactions between them, such as residue connections, can be considered as edges. Graph Neural Networks (GNNs) are specifically designed to process graph-based datasets, taking the structure of proteins into account. GNNs represent nodes and edges as vectors, update these vectors, and then input them into fully connected layers for regression or classification. Based on different update algorithms, GNNs can be categorized into several types: Graph Convolutional

Networks (GCNs) and Message Passing Neural Networks (MPNNs) differ primarily in their updating mechanisms. GCNs utilize convolutional operations, while MPNNs employ manually defined passing and updating functions. There are also more advanced structures, such as Graph Attention Networks, which apply attention mechanisms to MPNNs. Although GNNs deliver outstanding performance, they are relatively new, and few studies have utilized them. For example, the DeepTM model with a GCN showed superior performance compared to the ProTstab2 model [1,13].

To represent proteins as graphs, nodes and edges must be represented as feature vectors. This process, similar to feature selection for tabular data, requires manual selection and domain knowledge. For example, in the DeepTM model, researchers selected various features such as blocks of amino acid substitution matrices to represent nodes [1]. (In node representation, an amino acid is usually depicted by its relationships with other amino acids.) Similar to tabular feature descriptors, plenty of software are available, such as HHBLITS computes amino acid properties [33]. However, unlike tabular data, where thousands of features are generated, nodes and edges are typically represented with dozens or hundreds of features. Table 1 shows the differences in data size among the three representations. They also differ in compositional level, with one representing amino acids and the other representing entire sequences. Due to the complex model structure and the long protein sequences with many nodes and edges, the training time is significantly longer compared to simple machine learning models. Generally, GNNs are preferred over tabular data models. With the increasing variety of GNNs, further research on different models or node representations remains to be explored.

**Table 1.** Difference of representation size

| | Data size for a single protein |
|---|---|
| Tabular | Thousands of descriptors. E.g. ProTstab2 used 6395 feature descriptors [13]. |
| Sequence | A sequence of thousands of proteins. E.g. DeepTP limited the maximum sequence length to be 1500 [3]. |
| Graph | Thousands of nodes, dozens or hundreds of features for each node. A contact map whose size is the square of node number. E.g. DeepTM limited the maximum sequence length to be 1028 with 135 features for each node [1]. |

## 6. Model interpretability

Complicated machine learning models are often considered ‘black boxes’ because their internal mechanisms are not well understood. Without insight into the inner workings of these models, users may lack confidence in their predictions, and extracting knowledge from them can be challenging. Model interpretability methods aim to address this problem by providing explanations of how models operate, such as through feature importance and interactions. Although there is no unified formal definition of model interpretability, research often provides explanations in similar ways. In protein thermal stability research, feature importance is extracted during the feature selection phase for tabular representations, offering explanations for model behaviour. For example, ProTstab2 identifies amino acid frequency as the most important feature, with an importance score of 85 [13]. Feng et al. found Lysine to be the most crucial amino acid for thermal stability, aligning with Lin’s findings [8,16]. There are also universal methods that focus on input-output relationships without delving into the inner structure of models, offering interpretations after the model is trained. The SHAP (Shapley Additive Explanations) method evaluates the significance of features using principles from game theory [34]. The SAPPHIRE model used the SHAP method to compute feature importance for their 72 base models [10]. Other methods, such as Permutation, which calculates feature importance by randomly shuffling feature values and assessing performance differences, can also provide structure-agnostic explanations. An illustration of these two methods is shown in Figure 2.

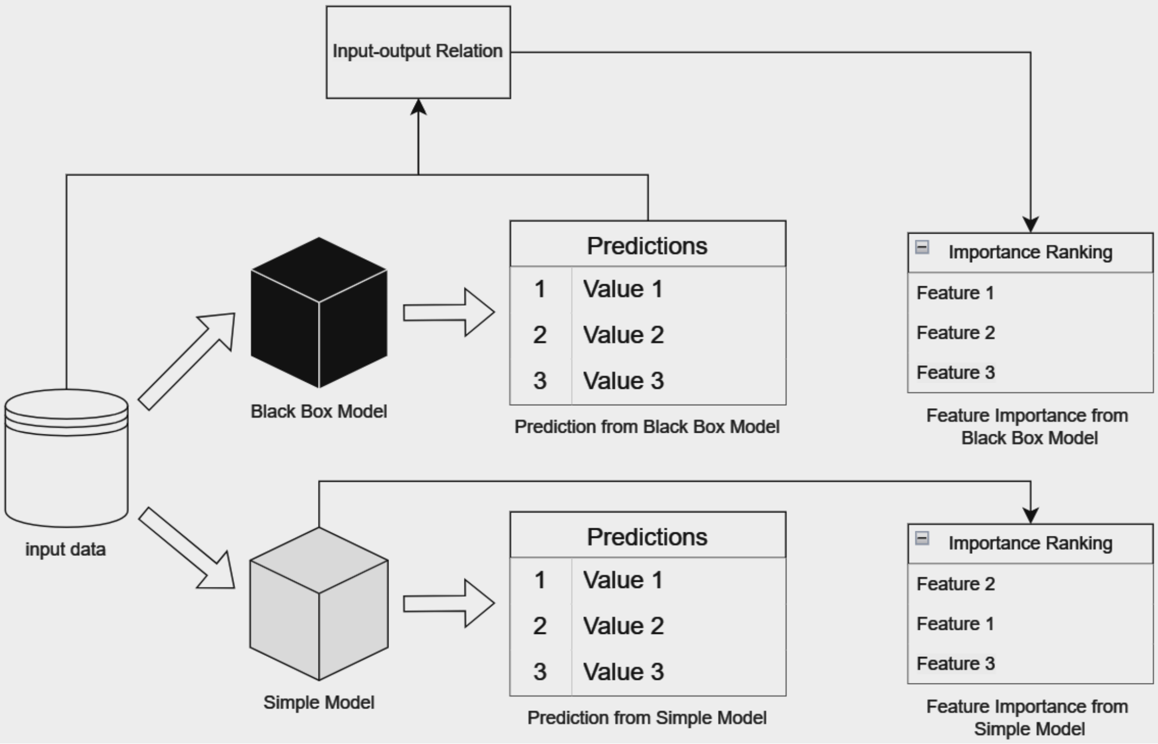

**Figure 2.** An illustration of how model explanations for tabular representations are extracted. For black-box models, feature importance is determined by analysing input-output relationships, while simpler models can inherently provide feature importance.

Different data representations provide varying levels of explanation. Feature-descriptor-based models offer explanations at the level of protein properties, while graph-based models provide insights at the amino acid level. Sequence-based models are more challenging to interpret, as protein sequences are less intuitive for humans compared to feature properties. Although these models can indicate the importance of tokens and calculate the percentage of amino acids at specific token positions, their interpretations are generally more complex than those of other representations. Additionally, explanations can sometimes be inconsistent. For example, while Guo et al. and Feng et al. identify lysine as the most important amino acid for thermal stability, the iThermo model suggests that tyrosine is the most important [5,8,11]. Such inconsistencies may arise from differences in databases, models, or explanation methods. Defining clear model explanations and achieving consistent interpretations remain significant challenges for future research.

## 7. Conclusion

With the growing popularity of machine learning methods, protein thermal prediction has emerged as a well-studied research area in recent years. This study has addressed the complexities and challenges associated with the availability and retrieval of suitable datasets for model training focusing on three different representations of proteins—tabular, sequence, and graph—and discussed their respective similarities, differences, and working pipelines. Tabular representation is well-researched and supported by various available software tools. Sequence representation offers convenience and ease of use. Graph representation demonstrates outstanding performance and presents significant potential for further research. Additionally, the functionality of model explanations in thermal prediction models has also been addressed highlighting the challenges associated with interpreting these models. Overall, this study provides valuable insights into the current methodologies, identifies key challenges, and highlights future research opportunities in the field of protein thermal prediction.

## References

[1] Li M, Wang H, Yang Z, Zhang L, and Zhu Y 2023 DeepTM: A deep learning algorithm for prediction of melting temperature of thermophilic proteins directly from sequences *Comput. Struct. Biotechnol. J.* 21 5544–60

[2] Jung F, Frey K, Zimmer D, and Mühlhaus T 2023 DeepSTABp: a deep learning approach for the prediction of thermal protein stability *Int. J. Mol. Sci.* 24 7444

[3] Zhao J, Yan W, and Yang Y 2023 DeepTP: a deep learning model for thermophilic protein prediction *Int. J. Mol. Sci.* 24 2217

[4] Pei H, Li J, Ma S, Jiang J, Li M, Zou Q, and Lv Z 2023 Identification of thermophilic proteins based on sequence-based bidirectional representations from transformer-embedding features *Appl. Sci.* 13 2858

[5] Guo Z, Wang P, Liu Z, and Zhao Y 2020 Discrimination of thermophilic proteins and non-thermophilic proteins using feature dimension reduction *Front. Bioeng. Biotechnol.* 8 584807

[6] Charoenkwan P, Chotpatiwetchkul W, Lee VS, Nantasenamat C, and Shoombuatong W 2021 A novel sequence-based predictor for identifying and characterizing thermophilic proteins using estimated propensity scores of dipeptides *Sci. Rep.* 11 23782

[7] Meng C, Ju Y, and Shi H 2022 TMPpred: A support vector machine-based thermophilic protein identifier *Anal. Biochem.* 645 114625

[8] Feng C, Ma Z, Yang D, Li X, Zhang J, and Li Y 2020 A method for prediction of thermophilic protein based on reduced amino acids and mixed features *Front. Bioeng. Biotechnol.* 8 285

[9] Haselbeck F, John M, Zhang Y, Pirnay J, Fuenzalida-Werner JP, Costa RD, and Grimm DG 2023 Superior protein thermophilicity prediction with protein language model embeddings *NAR Genom. Bioinform.* 5 lqad087

[10] Charoenkwan P, Schaduangrat N, Moni MA, Manavalan B, and Shoombuatong W 2022 SAPPHIRE: A stacking-based ensemble learning framework for accurate prediction of thermophilic proteins *Comput. Biol. Med.* 146 105704

[11] Ahmed Z, Zulfiqar H, Khan AA, Gul I, Dao FY, Zhang ZY, et al 2022 iThermo: a sequence-based model for identifying thermophilic proteins using a multi-feature fusion strategy *Front. Microbiol.* 13 790063

[12] Pudžiuvelytė I, Olechnovič K, Godliauskaite E, Sermokas K, Urbaitis T, Gasiunas G, and Kazlauskas D 2024 TemStaPro: protein thermostability prediction using sequence representations from protein language models *Bioinformatics* 40 btae157

[13] Yang Y, Zhao J, Zeng L, and Vihinen M 2022 ProTstab2 for prediction of protein thermal stabilities *Int. J. Mol. Sci.* 23 10798

[14] Dehouck Y, Folch B, and Rooman M 2008 Revisiting the correlation between proteins' thermoresistance and organisms' thermophilicity Protein *Eng. Des. Sel.* 21 275–78

[15] Gromiha MM, Oobatake M, and Sarai A 1999 Important amino acid properties for enhanced thermostability from mesophilic to thermophilic proteins *Biophys. Chem.* 82 51–67

[16] Lin H, and Chen W 2011 Prediction of thermophilic proteins using feature selection technique *J. Microbiol. Methods* 84 67–70

[17] Montanucci L, Capriotti E, Birolo G, Benevenuta S, Pancotti C, Lal D, and Fariselli P 2022 DDGun: an untrained predictor of protein stability changes upon amino acid variants *Nucleic Acids Res.* 50 W222–27

[18] Jarzab A, Kurzawa N, Hopf T, Moerch M, Zecha J, Leijten N, et al 2020 Meltome atlas—thermal proteome stability across the tree of life *Nat. Methods* 17 495–503

[19] Yang Y, Ding X, Zhu G, Niroula A, Lv Q, and Vihinen M 2019 ProTstab–predictor for cellular protein stability *BMC Genomics* 20 1–9

[20] Pezeshgi Modarres H, Mofrad MR, and Sanati-Nezhad A 2018 ProtDataTherm: A database for thermostability analysis and engineering of proteins *PLoS One* 13 e0191222

[21] Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, et al 2021 Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences *Proc. Natl Acad. Sci. USA* 118 e2016239118

[22] Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, et al 2023 Evolutionary-scale prediction of atomic-level protein structure with a language model *Science* 379 1123–30

[23] Elnaggar A, Heinzinger M, Dallago C, Rehawi G, Wang Y, Jones L, et al 2021 Prottrans: Toward understanding the language of life through self-supervised learning *IEEE Trans. Pattern Anal. Mach. Intell.* 44 7112–27

[24] Nikam R, Kulandaisamy A, Harini K, Sharma D, and Gromiha MM 2021 ProThermDB: thermodynamic database for proteins and mutants revisited after 15 years *Nucleic Acids Res.* 49 D420–24

[25] Devlin J, Chang MW, Lee K, and Toutanova K 2018 Bert: Pre-training of deep bidirectional transformers for language understanding arXiv Preprint arXiv:1810.04805

[26] Chawla NV, Bowyer KW, Hall LO, and Kegelmeyer WP 2002 SMOTE: synthetic minority over-sampling technique *J. Artif. Intell. Res*. 16 321–57

[27] Li W, Jaroszewski L, and Godzik A 2001 Clustering of highly homologous sequences to reduce the size of large protein databases *Bioinformatics* 17 282–83

[28] Xiao N, Cao DS, Zhu MF, and Xu QS 2015 protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences *Bioinformatics* 31 1857–59

[29] Chen Z, Zhao P, Li F, Leier A, Marquez-Lago TT, Wang Y, et al 2018 iFeature: a python package and web server for features extraction and selection from protein and peptide sequences *Bioinformatics* 34 2499–502

[30] Ruiz-Blanco YB, Paz W, Green J, and Marrero-Ponce Y 2015 ProtDCal: A program to compute general-purpose-numerical descriptors for sequences and 3D-structures of proteins *BMC Bioinformatics* 16 1–15

[31] Guyon I, Weston J, Barnhill S, and Vapnik V 2002 Gene selection for cancer classification using support vector machines *Mach. Learn.* 46 389–422

[32] Fisher RA 1936 Design of experiments *Br. Med. J.* 1 554

[33] Steinegger M, Meier M, Mirdita M, Vöhringer H, Haunsberger SJ, and Söding J 2019 HH-suite3 for fast remote homology detection and deep protein annotation *BMC Bioinformatics* 20 1–15

[34] Shapley LS 1953 A value for n-person games Contributions to the Theory of Games II AW Tucker and HW Kuhn