JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 10 Issue 01
2025

Research Article
https://doi.org/10.56979/1001/2025/1128

Harnessing Deep Learning and Language Models for Protein Function Prediction: A CAFA5-Based Study

Ali Haider[1], Jamal Shah[1], Musadaq Mansoor[2], and Omar Bin Samin[1]

[1]School of Computer Sciences & IT, Institute of Management Sciences, Peshawar, Pakistan.
[2]School of CS, Pak-Austria Fachhochschule Institute of Applied Sciences and Technology, Haripur, Pakistan.
*Corresponding Auhtor: Ali Haider. Email: ali.haider@imsciences.edu.pk

Received: September 12, 2025 **Accepted:** November 21, 2025

Abstract: Recent advancements in deep learning have brought remarkable progress in the area of predicting the protein functions from its amino acid sequences. These sequences play a crucial role in accelerating drug evaluation and uncovering how cells work. This research investigated several deep models for predicting protein functions, which include Bi-LSTM coupled with an attention mechanism, Gated Recurrent Unit, Long Short-Term Memory, Deep Neural Networks, and Bidirectional LSTM. This research used the CAFA5 dataset along with the T5 embedding, which is created from this dataset, to test these DL models for the multi-label protein functions prediction task. The researchers used state-of-the-art matrices to measure the performance of these models, which includes ROC-AUC, Hamming loss AUC, and binary accuracy. The analysis demonstrates the Bi-LSTM paired with attention mechanism and DNN models outperformed the baseline traditional RNN models in both minimizing loss and accuracy. With an outstanding ROC-AUC score of 0.9239 and consistent prediction reliability, the Bi-LSTM plus Attention model performed well. This research showed that combining DL models with integrated attention layers produces more scalable and accurate results for predicting protein functions. Showing their usefulness in practical bioinformatics tasks.

Keywords: Deep Learning; CAFA5; T5 Embedding; LSTM; GRU; Bi-LSTM; Attention Mechanism

1. Introduction

Bioinformatics is an interdisciplinary branch that incorporates knowledge of computer science, biology, and mathematics in the study of biological systems through computational approaches. Among others, one of the most important goals is to understand how protein sequences relate to their biological functions. Although in many cases, a protein's three-dimensional structure underlies its function, the experimental determination of structure and function is still time-consuming, resource-intensive, and impractical for the volumes of sequence data increasing day by day through high-throughput sequencing technologies. In this regard, computational approaches that predict protein function directly from sequence information have become critical for large-scale biological studies, genome annotation, and downstream applications involving drug discovery, precision medicine, and systems biology. The availability of huge protein databases, such as Swiss-Prot and UniProt, has further encouraged the development of automated and accurate computational models that can assign functional labels in an efficient and consistent manner.

Currently, deep learning is one of the most powerful families of methods for the prediction of protein function based on sequence. Unlike the traditional techniques that are primarily dependent on sequence alignment, handcrafted features, or homology-based inference, a deep learning model can learn the relevant features of raw amino acid sequences automatically. The importance of the latter ability emerges when working with proteins either with no closely related homologs or belonging to rarely studied families. RNNs have played a significant role in this area due to their capability of processing sequential

data and capturing the contextual information about the residues over long sequences. Among RNN variants, LSTMs and GRUs have garnered particular attention thanks to their extraordinary capabilities to model long-range dependencies while coping with issues such as exploding/vanishing gradients. Moreover, the literature has already reported that LSTM- and GRU-based models are capable of effectively modeling complex residue patterns and conserved motifs, thus leading to an increase in prediction accuracy over several GO categories [1–3].

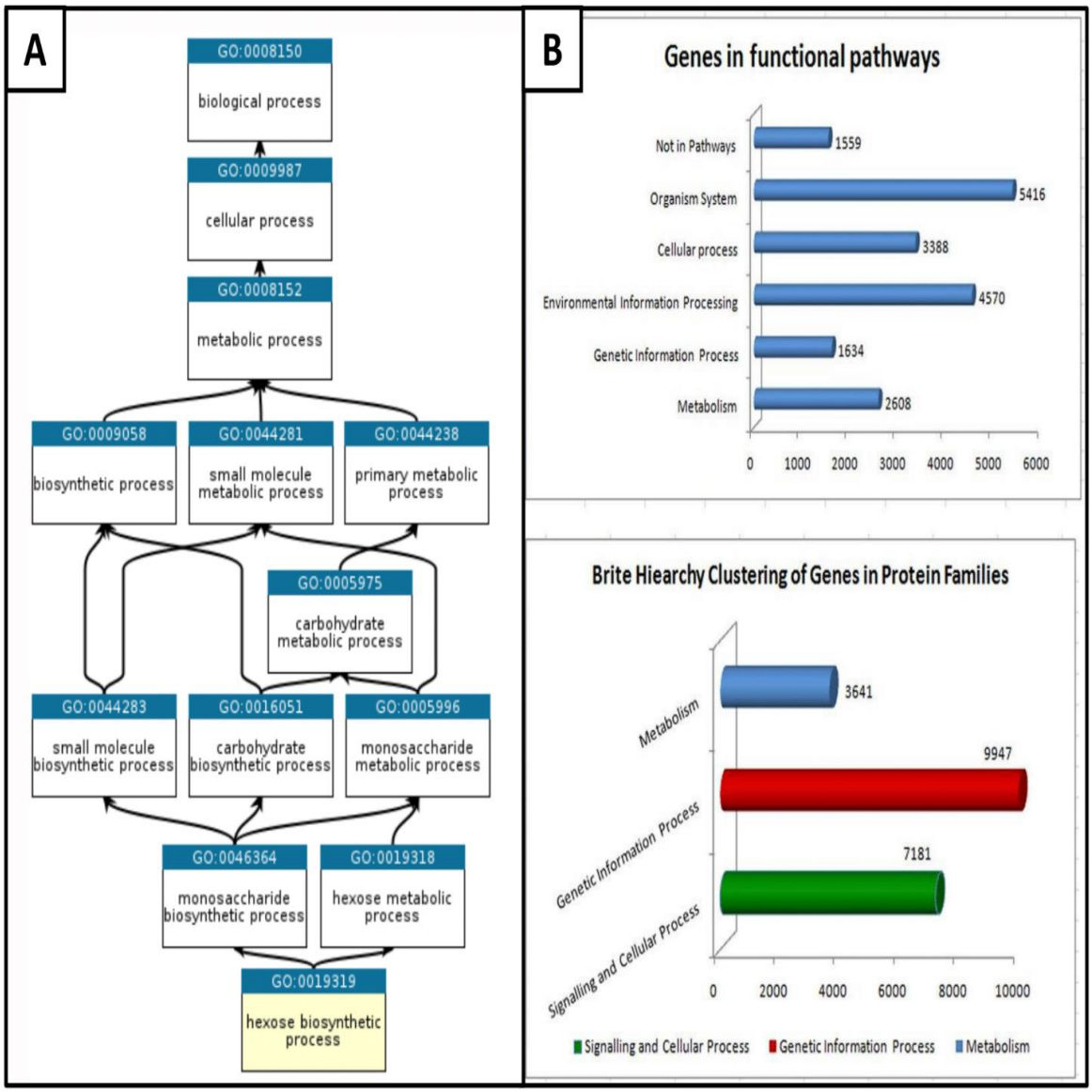

Figure 1. SEQ Figure * ARABIC 1 (A) Simplified Gene Ontology (GO) hierarchy representing three main categories: Biological Process, Molecular Function, and Cellular Component [9] **(B)** Gene distribution across KEGG biological pathways, illustrating functional diversity and annotation complexity [10]

In parallel, CNNs have been explored as an alternative approach. CNNs capture local patterns in a hierarchical way, enabling the modeling of biologically meaningful motifs, evolutionary blocks, or localized structural signals from sequences. These properties make CNNs particularly useful in applications where function is determined significantly by short-range but high-impact patterns. Very recently, attention-based architectures have gained interest due to their superior capability of modeling non-local interactions compared to both CNNs and RNNs. Attention mechanisms allow for assigning different importance weights to residues regardless of their sequential distance, thus better capturing long-range interactions influencing protein function. Among those, transformers have shown outstanding

performance in a variety of sequence modeling tasks including protein function prediction and large-scale protein language modeling [4–6]. Despite such progress, a number of challenges remain in the field. First, many existing studies evaluate only a single model or a narrow selection of architectures, which makes it hard to generalize conclusions as to which deep learning approaches work best for different data conditions. For example, some works focus on RNN-based models exclusively, while others consider CNNs or attention-based networks alone. This fragmented landscape of evaluations limits the ability of researchers to understand trade-offs in prediction accuracy, computational efficiency, and their ability to model long-range dependencies and generalization across various GO labels or protein families. Another challenge involves the diversity and complexity of the Gene Ontology system itself: proteins often have multiple overlapping functions, making the prediction task inherently multi-label. The evaluation of models across such a multi-faceted ontology requires rigorous, consistent large-scale benchmarking. Large-scale, community-wide challenges such as the Critical Assessment of Function Annotation have established the importance of rigorous assessment using high-quality curated data sets. In such competitions, deep learning approaches often achieve state-of-the-art performance. However, a systematic, side-by-side comparative study in which different models are benchmarked against the same dataset using an identical preprocessing pipeline and identical evaluation metrics is still missing in the literature. Until this kind of comparative analysis is carried out, it is difficult to know whether improvements reported from one study to the next are the result of the model architecture, or rather of dataset preparation, differences in training procedures, or differences in the functional categories under prediction. To these ends, the paper presents the systematic, comparative analysis of several deep learning architectures for protein function prediction, including LSTM, GRU, bi-LSTM, DNN, and attention-enhanced bi-LSTM. We evaluate each architecture in several GO categories using the CAFA5 benchmark dataset derived from Swiss-Prot, with key performance measures that include binary accuracy, Hamming loss, ROC-AUC, and computational efficiency. Emphasizing the use of a consistent embedding method and identical training settings for all tested models, this study provides a controlled and fair comparison that points out the strengths and weaknesses of each architecture within the context of scalable protein functional annotation. Overall, the present work contributes to the field by performing a unified and rigorous analysis of state-of-the-art, widely used deep learning architectures and identifies the model with the highest potential to carry out correct large-scale sequence-based function predictions. New insights from this head-to-head comparison will be able to support future developments in protein annotation pipelines, enhance interpretability and reliability of computational predictions, and guide model selection for practical applications in bioinformatics and computational biology.

The paper is laid out like this: Section 2 reviews existing research and methods in detail. Section 3 explains the approach, focusing on preparing the dataset and building the model. Section 4 includes the experimentation, and section 5 discusses the outcomes and suggests areas to explore later.

2. Literature Review

Mansoor et al. [15] presented a new model called GOGAN, short for Gene Ontology GAN that helps to improve the protein function predictions by using a large number of protein sequence unlabeled data. The study aims to address the issues and limitations of the traditional methods that require large amounts of labeled data, which isn't easy to collect. A special artificial intelligence approach is used by the GOGAN model called Generative Adversarial Network, which is also known as GAN for short. It generates protein sequences and identifies important features on its own. When compared against the other techniques, the proposed model GOGAN improved the accuracy of protein function predictions, although one particular drawback of this approach is that if the generator produces unrealistic and low-quality sequences, it may damage the accuracy of the overall predictions.

A novel tool Propagation of Affinity and Domain Architecture, also known as PANDA, for prediction functions of the proteins on the basis of protein sequences using Gene Ontology was presented by Wang et al. [16]. The main objective of this proposed approach is to improve the accuracy of predicting the protein functions by combining profile alignment and Bayesian computational techniques to analyze the domain architecture of proteins. By narrowing down the gene ontology terms and grouping those using statistical importance, the proposed tool does a better job than the current baseline predictors. However, the downside of this proposed PANDA tool depends upon the toughness and quality of protein databases. If

the databases include incorrect domain details and missing data, the predictions may be biased and inaccurate.

Belper and Berger [17] employed a bidirectional LSTM architecture to model protein sequences with the aim of learning similarity relationships directly from sequential data. Instead of introducing the Bi-LSTM model itself, their work uses this architecture to capture contextual residue information in both forward and backward directions, allowing for improved sequence-based comparison. The authors have used the SCOP database for training their approach to learn representations indicative of underlying structural relationships. A key element of their study is a soft symmetric alignment mechanism that aligns sequence embeddings with consideration of the correspondence of residues. Their approach results in competitive performance compared to traditional sequence-based comparison methods. However, because the SCOP database is mainly composed of single-domain proteins, the model will have limited generalizability to multidomain or structurally diverse proteins, which is a potential drawback of the approach.

Rives et al. [13] introduce a large-scale protein language model that uses unsupervised learning. They trained this model, called ESM-1b, with the UniRef50 and UniRef datasets. These datasets include 250 million protein sequences and 86 billion amino acids. Using a transformer-based deep neural network, the model aims to learn biological properties from sequence data. It finds links in evolution and structure without requiring any guidance. Its predictions on long-range residue contacts, secondary structure, and mutational effects outperform those made by LSTM models. One crucial problem with this approach is that it depends upon the variety and quality of data. If the dataset is biased or lacks protein types, the model's results can be more biased against the rare protein families and can be less accurate.

Pakhrin et al. [18] focused on tools like RaptorX and AlphaFold2, which have made big achievements in protein structure prediction, to discuss how DL is helping to improve the structure prediction of proteins. The researchers explain how methods like contact map prediction and multiple sequence alignments increase the accuracy by utilizing the data from CASP competitions. During CASP14, AlphaFold2 showed its exceptional ability to predict the protein structures with a median GDT-TS score of 92.4. This study reveals that how DL is evolving and changing the bioinformatics fields suggests its better potential paths to explore structure prediction of proteins. One potential limitation of this study was that they only focused on RaptorX and AlphaFold2 techniques and did not cover all other techniques used in protein structure predictions.

A new deep generative technique was designed by Qiao et al. [19] to predict protein-ligand complex structures. This model uses protein sequences and ligand molecular graphs as input. They compared its performance against well-known tools like AlphaFold2 by analyzing datasets from the Protein Data Bank and PDBBind2020. NeuralPLexer achieved impressive results showing advanced performance. It scored an average TM-score of 0.93 for predicting protein structures and boosted ligand pose accuracy by as much as 78% over existing approaches. This study showcases how data-driven models can help understand the dynamic connections between proteins and ligands, opening doors to advances in drug creation and enzyme development. However, one big challenge is the dependence on high-quality and diverse training data. If the datasets fail to include a wide range of protein-ligand interactions, the model's ability to generalize might be limited.

Yang et al. [20] used deep reinforcement learning to tackle protein structure prediction. They applied a deep Q-network combined with a long short-term memory setup to the hydrophobic-polar model. This model simplifies protein folding by treating proteins as chains of hydrophobic and polar amino acids laid out on a grid. Its core idea focuses on increasing contacts between hydrophobic amino acids. They tested their approach with benchmark sequences from Istrail's team using lengths such as 20, 24, 25, 36, 48, and 50. For these, they hit top energies, like -10 in the 20mer-B sequence and -21 in the 50mer.Their method demonstrated how deep reinforcement learning may explore the structure space and find the best potential configurations. It makes the PSP progress faster. Although one drawback of this study is that the HP model design is simpler, which causes it to not account for real protein folding, such as secondary structure prediction or solvent effects,

Espitia et al. [21] take on the task of predicting protein structures through deep reinforcement learning while working with the 3D hydrophobic-polar (HP) model. They introduce two new models. The first uses a hybrid reservoir method designed to manage chains with a maximum of 36 residues. The second involves

an LSTM model equipped with multi-headed attention aimed at handling longer chains. To evaluate these methods, they use sequence data provided by Istrail's group and benchmark it against the best-known energy scores, like -11 for the 20mer and -55 for the 60mer. Their results reveal that the hybrid model finishes training 25% quicker but still finds the best conformations. The LSTM model, on the other hand, captures long-range dependencies better and enhances both the efficiency and accuracy of protein folding predictions during training. A limitation they acknowledge is the use of the simplified HP model, which does not completely represent the real-world complexity of protein folding, like solvent interactions or post-translational changes.

Panou et al. [22] introduce DeepFoldit, a model built with deep reinforcement learning to improve protein structure predictions through the Foldit platform. They train their model using data from 40 proteins in the Protein Data Bank. Their focus is on small, unfolded proteins. DeepFoldit uses a Q-learning approach with experience replay. It shows notable improvements in scoring. Early results show better performance on both the proteins used during training and new test proteins. The research demonstrates the value of blending simple user interfaces with advanced deep learning strategies to create more effective protein folding methods. However, the study relies on a small protein dataset of 40 examples. This limited size might not capture enough of the structural variety needed to work well on more complex and diverse proteins found in real-world cases.

The studies reviewed here show several weak points that affect how generalizable or accurate their predictions are. These models often rely on the quality and completeness of training datasets or databases. Examples include protein domain annotations [16], the SCOP database [17], UniRef datasets [15], or protein-ligand interaction data [19]. These dependencies can create biases or make them less useful for rare protein families or structural types. Using simple models like the hydrophobic-polar (HP) [20, 21] approach does not capture the complexity of real protein folding. It misses things like how proteins interact with solvents or go through changes after they're made. Some studies also rely on benchmark datasets, like the PDB [18], or small datasets [22] that fail to reflect the variety seen in actual protein structures. On top of that, certain models such as GOGAN [15], depend on the quality of the sequences they produce. Others stick to specific tools like AlphaFold2 and RaptorX [18], which limits how much they explore the whole field of protein structure prediction.

3. Methodology

The proposed method explains how experiments were set up to predict protein functions with deep learning. It covers steps like data preparation, how protein sequences are represented, the model designs used, training settings, and how results are measured. Figure 2 shows the entire process.

3.1. Data Preprocessing and Representation

The protein sequences in FASTA format were gathered and standardized as an initial step, prior to feature extraction. In order to generate rich numerical representations suitable for deep-learning models, we employed contextual embeddings from Rost Lab's T5 protein language model [23]. These pretrained embeddings, which capture biochemical properties, evolutionary context, and sequence semantics, were obtained from publicly released resources provided by Sergei Fironov [24]. Each protein sequence was then represented as a fixed-length 1024-dimensional embedding vector and saved as a NumPy array. Corresponding protein identifiers were maintained in a separate indexing file to ensure consistent mapping between sequences, embeddings, and annotations.

3.2. Data Splitting

Our complete dataset consisted of 142,246 protein embeddings along with their Gene Ontology (GO) annotations. Since the complete GO repository contains more than 40,000 terms, direct modeling of the full label space would introduce extreme sparsity and generally decrease the reliability of supervised learning. Hence, in order to retain biological relevance and at the same time reduce noise and computational complexity, we selected the top 1,500 most frequent GO terms. This filtering strategy preserved the majority of meaningful functional diversity present in the dataset and guaranteed that each of the selected GO terms had enough representation for robust model training. The final curated embeddings and their corresponding reduced annotation set formed the input for training, validation, and testing of all deep-learning models used in our study.

The dataset was split into training, validation, and test sets, respectively, with the purpose of unbiased evaluation. A random split was carried out: 80%, 10%, and 10% are used for training, model optimization during validation, and testing on unseen data, respectively, as presented in Figure 3. Although stratification is generally encouraged to maintain class distributions, it has challenging properties to use in a high-dimensional multi-label splitting problem due to the huge number of labels (1,500 GO terms) and sparsity of many rare classes. However, we monitored label distributions and ensured that all major classes appeared adequately in every split. Given the big size of the dataset, 142,246 proteins, even random splitting is a quite reasonable approximation of the overall distribution. Advanced multi-label stratification chniques could be considered in future works to further avoid potential effects of class imbalance and to enhance the robustness of the developed models.

3.3. Techniques

This research evaluates different deep learning models to predict protein functions with recurrent neural network (RNN) variants. It examines how well several architectures classify protein sequences based on their expected biological roles. The goal is to find the model that captures protein sequence-function connections most accurately. The study focuses on these RNN-based models: LSTM [25, 26], GRU [27, 28], Bi-LSTM [29, 30], DNN [31, 32], and the Bi-LSTM with an attention mechanism [33, 34]. The architecture of each RNN model used in this study is shown in Figure 5 and the architecture of DNN and Bi-LSTM with Attention Mechanism model is shown in Figure 6.

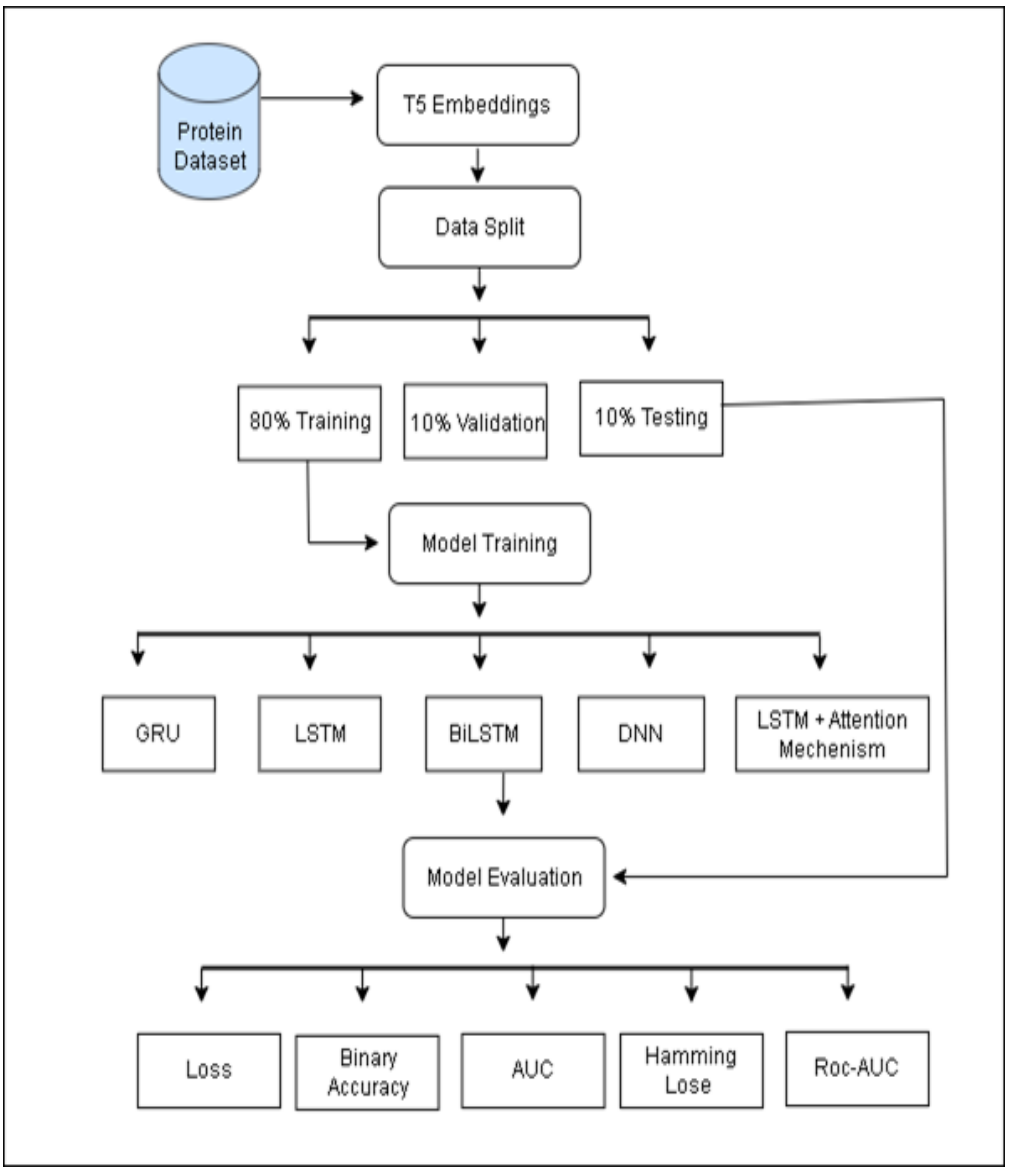

Figure 2. Methodology Workflow

Figure 3. Data Split

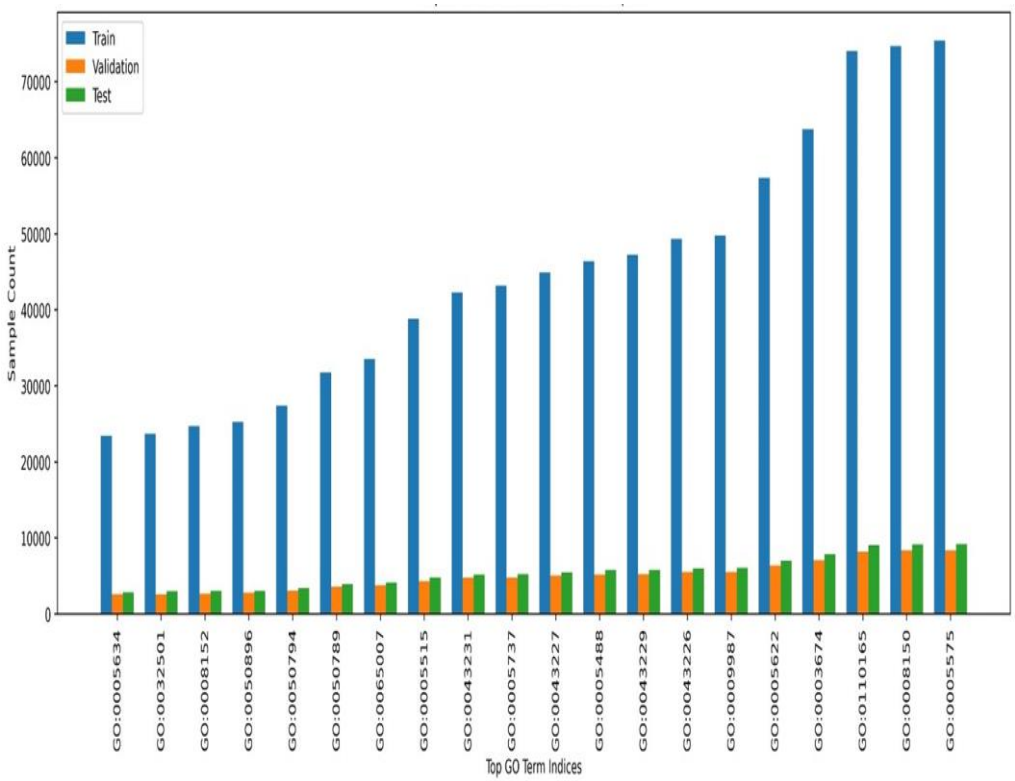

Figure 4. Top 20 Label Distributions across Splits

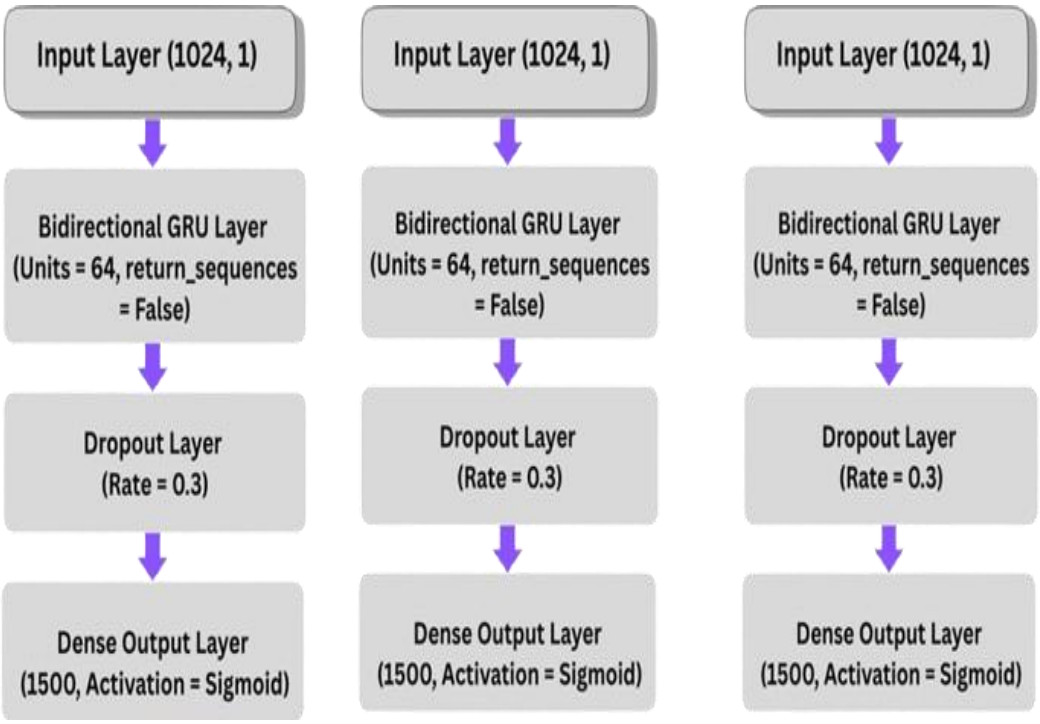

Figure 5. Illustration of different RNN architectures including LSTM, GRU, and BiLSTM

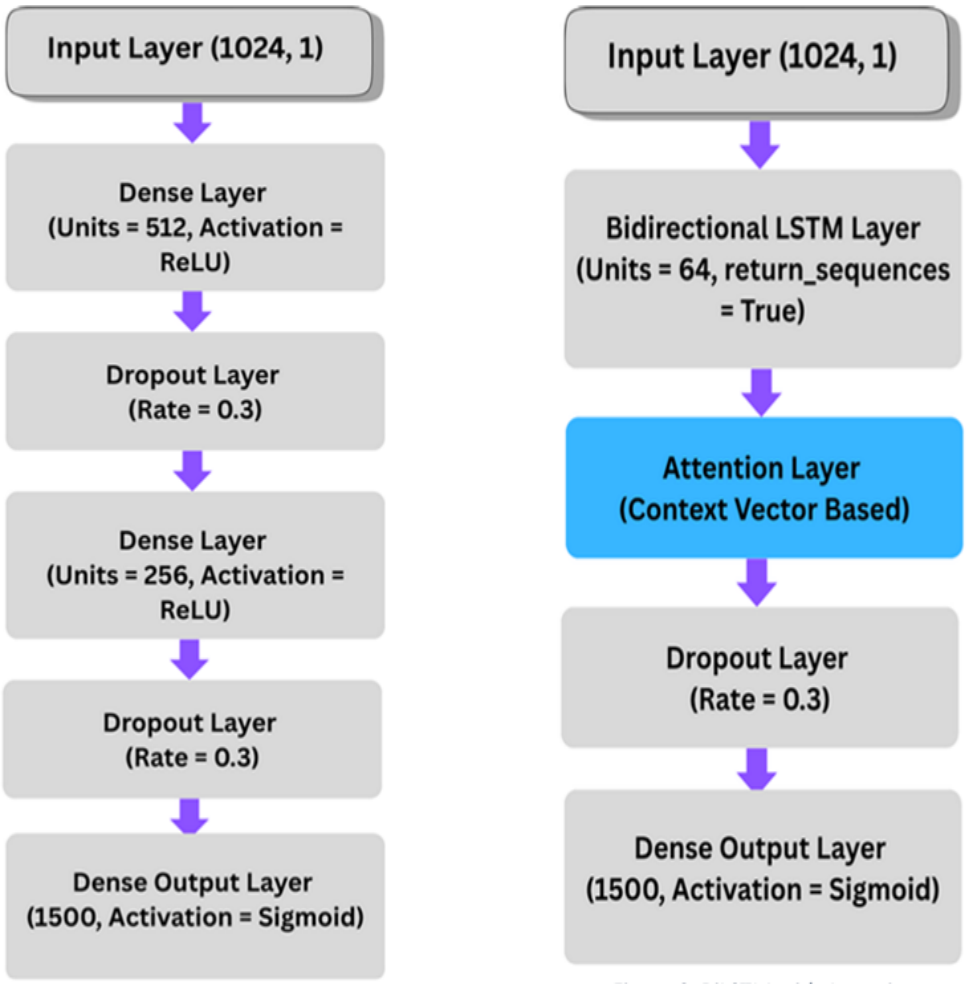

Figure 6. DNN and Bi-LSTM model architectures with attention mechanism

In this study, all deep learning models were implemented using TensorFlow/Keras and trained on 1024-dimensional T5 protein embeddings. The architectures of the recurrent models included one bidirectional layer in LSTM, Bi-LSTM, and GRU models, respectively, each with 64 units in each direction, followed by a dropout of 0.3 to prevent overfitting, and a dense output layer with sigmoid activation for multi-label classification across the top 1,500 Gene Ontology (GO) terms. DNNs were built as three fully connected layers composed of 512, 256, and 256 units, respectively, using ReLU activations, dropout of 0.3, and L2 regularization. An attention layer was used to deal with long-range dependencies in protein sequences by the Bi-LSTM models. One-dimensional convolutional layers with 64 filters, kernel sizes ranging from 3 to 5, followed by max pooling and attention mechanisms including both a custom attention and multi-head attention with four heads and key dimension of 16, were used in CNN variants. A more advanced DNN architecture has also been explored: it is based on fully connected layers of size 1024, 512, and 256 units with batch normalization and dropout combined with an output attention mechanism in order to model the label correlations. All models were trained using Adam or AdamW optimizer with binary cross-entropy loss, a batch size of 512, and early stopping with patience of 2–3 epochs, which restores the best weights. Evaluation metrics included binary accuracy and AUC-ROC to measure the performance of the classification models. The dataset was split into random training, validation, and test sets according to the ratio of 80–10–10 and was not stratified, hence having a sufficient representation of the multi-label space. These comprehensive specifications guarantee that all models are fully reproducible; they also provide a clear understanding of the architecture, hyperparameters, and training strategy used for the comparative analysis of protein function prediction.

3.4. Evaluation Metrics

We utilized a range of standard performance metrics on our models, such as the F1 score, recall, accuracy, and precision. These metrics yield a good understanding of the performance of the models in classification. Binary accuracy is a widely used method of assessing binary and multi-label classification problems. Hamming loss is one of the popularly used measuring metrics for multi-label classification problems. A typical metric of measuring binary and multi-label classification issues is area under receiver operating characteristic curve, or AUC-ROC. The most widely utilized loss function for binary and multi-label classification issues is binary cross-entropy, or BCE. Large scale datasets such as IMDB, when tuned, have a good implementation of ROC-AUC. It plots false positive against true positive with varying thresholds to see how well a model can separate classes. The formulas are as follows:

$$Binary\ Accuracy = \frac{1}{N}\sum_{i=1}^{N} 1(\widehat{y_i} = y_i) \quad (1)$$

where $\boldsymbol{N}$ denotes the number of samples, $\boldsymbol{y_i}$ is the true label, and $\boldsymbol{\widehat{y}_i}$ is the predicted label. Hamming loss captures the fraction of labels incorrectly predicted and is defined as:

$$Hamming\ Loss = \frac{1}{N \cdot L}\sum_{i=1}^{N}\sum_{j=1}^{L} 1\left(y_{ij} \neq \hat{y}_{ij}\right) \quad (2)$$

where L is the number of labels. The AUC evaluates the discriminative ability of the models across thresholds and is given by:

$$AUC = \int_0^1 TPR\left(FPR^{-1}(x)\right) dx \quad (3)$$

Similarly, ROC-AUC measures the area under the receiver operating characteristic curve:

$$ROC - AUC = \int_0^1 TPR(FPR)\, d(FPR) \quad (4)$$

Because this is a multi-label setting, all models produce sigmoid outputs for each label; a fixed threshold of 0.5 was used to binarize predictions, as customary in CAFA scoring. This makes the results comparable across different models and maintains the interpretability of binary accuracy, Hamming loss, and other measures. Together, these measures enable a thorough assessment of both the overall correctness of predictions and the ability to discover rare or functionally important labels, hence providing informative insights relevant for large-scale protein function prediction.

4. Experimentation

Our experiments ran on a laptop featuring a 7th generation Intel Core i7 chip, 12 GB RAM, and a 2.0 GHz processor. We also used Kaggle without a GPU accelerator to boost processing capabilities with GPU support. This arrangement benefited data processing speed and model training leading to better outcomes that were both accurate and dependable.

4.1. Dataset Description

This work utilizes the dataset from the Critical Assessment of Functional Annotation (CAFA 5) challenge [35] for computational prediction of protein functions. Proteins were selected with protein sequences in FASTA format, each assigned a unique UniProt accession ID. Functional annotations for these proteins are provided separately in a TSV file with the relevant GO term ID and an aspect: Biological Process (BP), Molecular Function (MF), or Cellular Component (CC) [36]. Obsolete GO terms were removed using GO Consortium metadata. The GO hierarchy was propagated according to the "true path rule," i.e., each protein was made to inherit all ancestor terms of its annotated GO terms. Negative labels were defined as those that did not have an annotation after hierarchy expansion. This provided a clear division between positive and negative examples. After curation, it filtered the dataset to the top 1,500 most frequent GO terms, ending up with over 142,000 protein sequences, making this a highly multi-label classification problem. The distribution of these Gene Ontology terms among the top 1,500 labels is shown in Figure 7.

5. Results and Discussion

Table 1. Performance Metrics of Deep Learning Models

Model	Loss	Binary Accuracy	AUC	Hamming Loss	ROC-AUC
GRU	0.0781	0.9767	0.8317	0.0233	0.8348
LSTM	0.0781	0.9771	0.8317	0.0229	0.8348
Bi-LSTM	0.0781	0.9755	0.8317	0.0245	0.8348
DNN	0.0638	0.9757	0.9180	0.0243	0.9239

Bi-LSTM + Attention	0.0638	0.9759	0.9180	0.0241	0.9239

Table 1 shows how five deep learning models performed on predicting multiple protein functions at the same time. The Deep Neural Network (DNN) and Bi-LSTM with Attention had the best loss value of 0.0638 doing better than simpler RNN models like GRU, LSTM, and Bi-LSTM, which had a loss of 0.0781.

Looking at binary accuracy, the LSTM model came out on top with 0.9771 followed by Bi-LSTM with Attention at 0.9759, which put up a close fight. DNN and Bi-LSTM with Attention showed much better scores in AUC and ROC-AUC hitting 0.9180 and 0.9239, while GRU, LSTM, and Bi-LSTM lagged behind with 0.8317 and 0.8348. This shows they had stronger ability to tell differences in data. When measuring Hamming Loss, a way to see prediction errors, Bi-LSTM with Attention had 0.0241 and LSTM had the lowest at 0.0229. This highlights that attention-based models lowered mistakes in predictions.

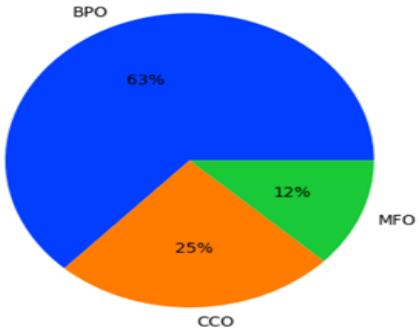

Figure 7. Distribution of Gene Ontology Terms in the First 1500 Labels

Results in Table 1 show that while all five deep learning models report high performance in predicting multiple protein functions, differences with biological consequences can be observed. The fact that DNN and Bi-LSTM with Attention improved the AUC by 0.02–0.04 over their simpler RNN variants suggests that the architectures are better at distinguishing subtle functional signals across large datasets of proteins, thus improving the reliability of annotation pipelines. Moreover, improvements in recall are important to reduce false negatives, which allow for the discovery of previously unannotated or novel proteins. In particular, the superior performance by Bi-LSTM with Attention reflects its modeling of long-range dependencies in protein sequences that capture biologically relevant interactions between distant residues influencing functional activity. Besides, the competitive performance by deep feedforward networks using T5 embeddings underlines the importance of informative sequence representations in capturing functional patterns beyond sequential modeling alone. These results together point out that an attention mechanism and deeper architecture bring a considerable advantage in modeling complex relationships inherent in protein sequences and offer practical value for large-scale functional annotation and prioritization of candidate proteins for experimental validation.

While the binary accuracy values for the tested models are relatively high and vary little among them, the differences in AUC are more significant. This is because in highly imbalanced multi-label data, where most labels are negative, correct predictions of the dominant class may inflate the overall accuracy. In contrast, AUC captures how well the model can distinguish between positive and negative labels across all threshold values, making it more sensitive to modeling improvements regarding the identification of infrequent annotations. Therefore, models like the Bi-LSTM with Attention and DNN show meaningfully higher AUCs compared to simpler RNNs with similar binary accuracy. These results highlight the importance of multiple measures when evaluating models, especially in multi-label protein function prediction, where most labels might be infrequent but of critical biological importance.

These findings highlight how adding attention layers or deeper structures gave these models an edge with tough protein sequence tasks. The experimental data shows that all the tested models perform well in predicting protein functions, but deeper networks and attention-based architectures achieve better results. Models like LSTM and GRU, which are traditional RNN systems, delivered solid binary accuracy with low Hamming loss. However, they fell short compared to DNNs and Bi-LSTM paired with attention when measured by AUC and ROC-AUC. This highlights how sequential models have a hard time handling

long-range connections in protein sequences, which are crucial to identifying function. On the other hand, attention mechanisms shine by focusing on key residues tied to function letting them interpret protein sequences in a more detailed way. Deep feedforward networks trained with meaningful embeddings can compete with sequential techniques, as demonstrated by the similar performance of DNNs and Bi- LSTM + Attention models. Additionally, the study demonstrates that the use of T5-based embedding improves the generalization performance of models across different Gene Ontology terms.

Several limitations should be acknowledged. First, the models were trained for only five epochs, which may have constrained their ability to fully converge and capture intricate sequence-function relationships. Second, the hardware limitations restricted Hyperparameters optimization and the use of larger batch sizes or deeper models. Third, although practical for dimensionality reduction, the choice to limit the dataset to the top 1,500 GO terms would have left out biologically significant, albeit less common, annotations. Furthermore, the study only used one embedding representation (T5); ensemble embedding or domain-specific language model refinement could be useful for future models. Finally, the biological interpretability of the predictions was not addressed, even though the models were tested using conventional metrics.

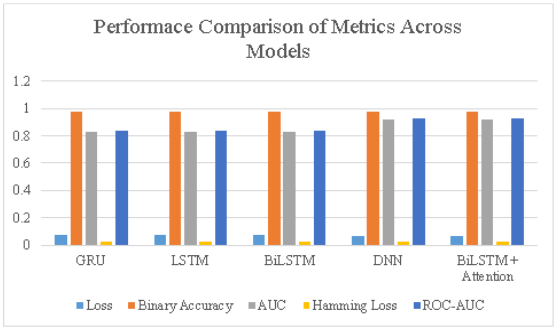

Figure 8. Performance comparison of metrics across models.

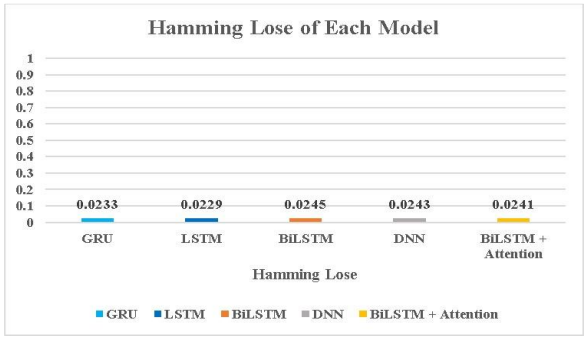

Figure 9. Hamming loss of each model.

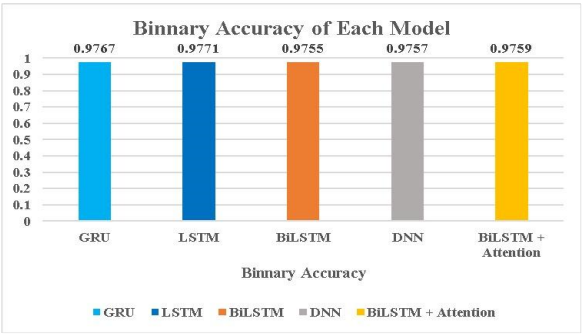

Figure 10. Binary accuracy of each model

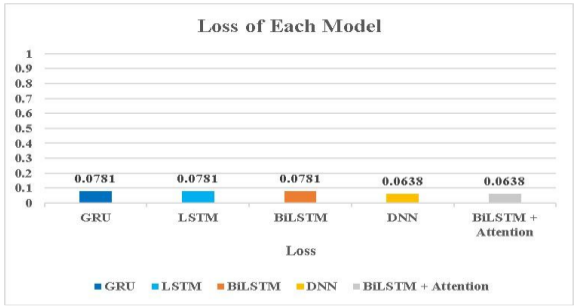

Figure 11. Loss of each model

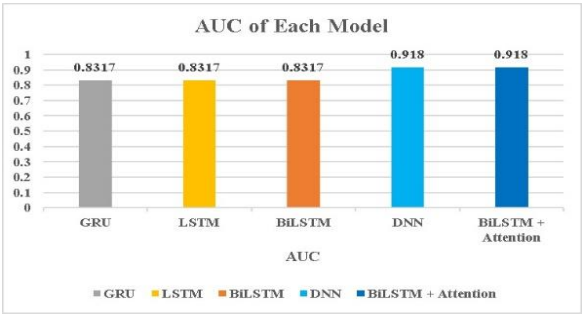

Figure 12. AUC of each model

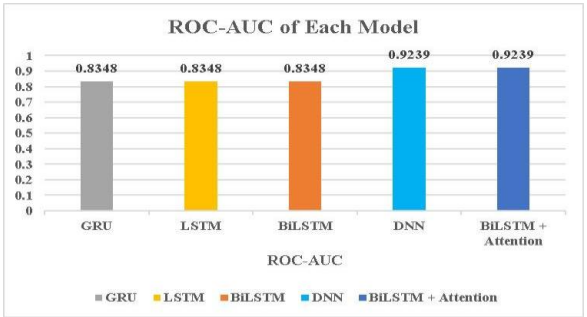

Figure 13. ROC-AUC of each model

6. Conclusions

This research shows that deep learning models DNNs and attention-equipped Bi-LSTM, can estimate protein functions using complex sequence embedding. High ROC-AUC values and better loss metrics highlight how deep architectures can unravel intricate links between sequences and their functions. Researchers should concentrate on training models with more epochs and exploring better designs to push the field further. Using transformer models like BERT or ESM bringing in multi-modal inputs such as structural or evolutionary data, and applying methods to connect labels can help boost results. Adding explainability tools might also make predictions easier to understand and could bring computational findings closer to real-world experiments.

References

1. Le NQK, Yapp EKY, Yeh HY, Et-gru: using multi-layer gated recurrent units to identify electron transport proteins, BMC bioinformatics 20:1–12, 2019.
2. Schuster M, Paliwal KK, Bidirectional recurrent neural networks, IEEE transactions on Signal Processing 45(11): 2673–2681, 1997.
3. Liu J, Tang X, Guan X, Grain protein function prediction based on self-attention mechanism and bidirectional lstm, Briefings in Bioinformatics 24(1): bbac493, 2023.
4. LeCun Y, Bengio Y, Hinton G, Deep learning, nature 521(7553):436–444, 2015.
5. Kulmanov M, Khan MA, Hoehndorf R, Deepgo: predicting protein functions from sequence and interactions using a deep ontology-aware classifier, Bioinformatics 34(4):660–668, 2018.
6. Tang M, Wu L, Yu X, Chu Z, Jin S, Liu J, Prediction of protein–protein interaction sites based on stratified attentional mechanisms, Frontiers in Genetics 12:784863, 2021.
7. Koehler Leman J, Szczerbiak P, Renfrew PD, Gligorijevic V, Berenberg D, VatanenT, Taylor BC, Chandler C, Janssen S, Pataki A, et al., Sequence-structure-function relationships in the microbial protein universe, Nature communications 14(1):2351, 2023.
8. Bertoline LM, Lima AN, Krieger JE, Teixeira SK, Before and after alphafold2: An overview of protein structure prediction, Frontiers in bioinformatics 3:1120370, 2023.
9. Wang W, Shuai Y, Zeng M, Fan W, Li M, Dpfunc: accurately predicting protein function via deep learning with domain-guided structure information, Nature Communications 16(1):70, 2025.
10. Mienye ID, Swart TG, Obaido G, Recurrent neural networks: A comprehensive review of architectures, variants, and applications, Information 15(9):517, 2024.
11. Boadu F, Lee A, Cheng J, Deep learning methods for protein function prediction, Proteomics 25(1-2):2300471, 2025.
12. Chen C, Chen X, Morehead A, Wu T, Cheng J, 3d-equivariant graph neural networks for protein model quality assessment, Bioinformatics 39(1): btad030, 2023.
13. Omar M, Ur Rehman H, Samin OB, Alazab M, Politano G, Benso A, Capgan: Text-to-image synthesis using capsule gans, Information 14(10), 2023. doi:10.3390/ info14100552, URL https://www.mdpi.com/2078-2489/14/10/552.
14. Samin OB, Omar M, Mansoor M, Capplant: a capsule network-based framework for plant disease classification, PeerJ Computer Science 7:e752, 2021. doi:10.7717/peerj-cs.752, URL https://doi.org/10.7717/peerj-cs.752.
15. Ontology documentation, Gene Ontology Website, 2025, URL https://geneontology.org/docs/ontology-documentation, accessed: 2025-06-15.
16. ResearchGate Contributor, Gene Categorization into KEGG Pathways, 2025, accessed: 2025-06-15.
17. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, ˇZ´ıdek A, Potapenko A, et al., Highly accurate protein structure prediction with alphafold, nature 596(7873):583–589, 2021.
18. Liu J, Wu T, Guo Z, Hou J, Cheng J, Improving protein tertiary structure prediction by deep learning and distance prediction in casp14, Proteins: Structure, Function, and Bioinformatics 90(1):58–72, 2022.
19. Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, Guo D, Ott M, Zitnick CL, Ma J, et al., biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences, Proceedings of the National Academy of Sciences 118(15): e2016239118, 2021.
20. Samin OB, Algeelani NAA, Bathich A, Adil GM, Qadus A, Amin A, Malicious agricultural iot traffic detection and classification: a comparative study of ml classifiers, Journal of Advances in Information Technology 14(4):811–820, 2023.
21. Mansoor M, Nauman M, Ur Rehman H, Benso A, Gene ontology gan (gogan): a novel architecture for protein function prediction, Soft Computing pp. 1–15, 2022.
22. Wang Z, Zhao C, Wang Y, Sun Z, Wang N, Panda: protein function prediction using domain architecture and affinity propagation, Scientific reports 8(1):3484, 2018.
23. Bepler T, Berger B, learning protein sequence embeddings using information from structure, arXiv preprint arXiv:190208661, 2019.
24. Pakhrin SC, Shrestha B, Adhikari B, Kc DB, Deep learning-based advances in protein structure prediction, international journal of molecular sciences 22(11):5553, 2021.
25. Qiao Z, Nie W, Vahdat A, Miller III TF, Anandkumar A, State-specific protein ligand complex structure prediction with a multiscale deep generative model, Nature Machine Intelligence 6(2):195–208, 2024.
26. Yang K, Huang H, Vandans O, Murali A, Tian F, Yap RH, Dai L, applying deep reinforcement learning to the hp model for protein structure prediction, Physica A: Statistical Mechanics and its Applications 609:128395, 2023.

27. Espitia G, Pang YT, Gumbart JC, Protein structure prediction in the 3d hp model using deep reinforcement learning, arXiv preprint arXiv:241220329, 2024.
28. Panou DN, Reczko M, Deepfoldit–a deep reinforcement learning neural network folding proteins, arXiv preprint arXiv:201103442, 2020.
29. M. M. Abualhaj, S. N. Al-Khatib, A. A. Abu-Shareha, O. Almomani, H. Al-Mimi, A. Al-Allawee, M. Sh. Daoud, and M. Anbar, "Spam detection boosted by firefly-based feature selection and optimized," Int. J. Adv. Soft Comput. Appl., vol. 17, no. 3, pp. 1–19, Nov. 2025.
30. Elnaggar A, Heinzinger M, Dallago C, Rehawi G, Wang Y, Jones L, Gibbs T, Feher T, Angerer C, Steinegger M, et al., Prottrans: Toward understanding the language of life through self-supervised learning, IEEE transactions on pattern analysis and machine intelligence 44(10):7112–7127, 2021.
31. Fironov S, T5 Protein Embed-dings, https://www.kaggle.com/datasets/sergeifironov/t5embeds , 2023, dataset retrieved June 5, 2025 from Kaggle.
32. Masadeh, R., Almomani, O., Zaqebah, A., Masadeh, S., Alshqurat, K. et al. (2025). Narwhal Optimizer: A Nature-Inspired Optimization Algorithm for Solving Complex Optimization Problems. Computers, Materials & Continua, 85(2), 3709–3737. https://doi.org/10.32604/cmc.2025.066797
33. Choong ACH, Lee NK, Evaluation of convolutionary neural networks modeling of dna sequences using ordinal versus one-hot encoding method, 2017 International Conference on Computer and Drone Applications (IConDA), IEEE, pp. 60–65, 2017.
34. Y. Sanjalawe, S. Fraihat, S. Al-E'mari, M. M. Abualhaj, S. Makhadmeh, and E. Alzubi, "Smart load balancing in cloud computing: Integrating feature selection with advanced deep learning models," PLOS One, vol. 20, no. 9, Sep. 2025, Art. no. e0329765. DOI: 10.1371/journal.pone.0329765.
35. Hochreiter S, Schmidhuber J, long short-term memory, Neural computation 9(8):1735–1780, 1997.
36. Elhaj-Abdou ME, El-Dib H, El-Helw A, El-Habrouk M, Deep cnn lstm go: protein function prediction from amino-acid sequences, Computational Biology and Chemistry 95:107584, 2021.
37. Cho K, Van Merri¨enboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y, learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:14061078, 2014.
38. Radivojac P, Clark W, Oron T, Schnoes A, Wittkop T, Sokolov A, Graim K, Funk C, Verspoor K, Ben-Hur A, Pandey G, Yunes J, Talwalkar A, Repo S, Souza M, Piovesan D, Casadio R, Cheng J, Friedberg I, A large-scale evaluation of computational protein function prediction, Nature Methods 10:221–227, 2013. doi:10.1038/nmeth.2340.
39. Consortium TGO, the gene ontology resource: enriching a gold mine, Nucleic Acids Research 49(D1): D325–D334, 2020. doi:10.1093/nar/gkaa1113, URL https://doi.org/10.1093/nar/gkaa1113.