Enzyme Classification using Transformer-Based Sequence Embeddings

JATEEN JOHARAPURKAR[1], **SAHAR HOOSHMAND**[1]

[1]Department of Computer Science, California State University-Dominguez Hills, Carson, CA, USA

E-MAIL: jjoharapurkar1@toromail.csudh.edu, hooshmand@csudh.edu

Abstract:

Enzyme function classification plays a critical role in understanding biological processes, drug discovery, and protein annotation. This paper presents a computational pipeline that leverages ESM2, a transformer-based protein language model to generate contextual embeddings from raw amino acid sequences. We explore strategies to address class imbalance and evaluate the embeddings on two supervised learning architectures: a MLP and a deeper custom neural network. Our observations demonstrate that the MLP model with oversampling achieves the best performance, achieving a test accuracy of 93.5% and macro F1-score of 91% outperforming deeper architectures and class-weighted loss. Our findings suggest that even embeddings generated from a lightweight transformer combined with effective imbalance handling techniques can provide an efficient solution for enzyme function classification.

Keywords:

Protein sequence; ESM Transformer; Embeddings; Deep Learning; Enzyme Classification; Class Imbalance handling; Multi-Layer Perceptron

1. Introduction

Enzymes play a vital role in almost all biochemical processes by accelerating reactions that are essential for life. The function of an enzyme can be systematically classified using the Enzyme Commission (EC) number [1]. The EC number is a numerical classification scheme for enzymes based on the type of chemical reactions they catalyze. An EC number consists of four digits, where each successive digit provides increasingly specific functional information. [2]. The first level (Level 1) defines the main class of the enzyme, the second digit (Level 2) defines the subclass, and so on until the fourth digit (Level 4). Thus, due to this numbering scheme, the EC number forms a hierarchical structure. At Level 1, there are seven main classes: Oxidoreductases (EC1), Transferases (EC2), Hydrolases (EC3), Lyases (EC4), Isomerases (EC5), Ligases (EC6), and Translocases (EC7). Accurate enzyme classification is vital for understanding metabolic pathways, discovering potential drug targets, and annotating genomic sequences. However, the process of experimentally deciphering enzyme function is often a resource-intensive and time-consuming task. Therefore, computational approaches have emerged as an attractive alternative to manual annotation. Recent advances in deep learning, especially the advent of transformer-based protein language models, offer a promising avenue to classify enzymes. Inspired by transformer architectures from natural language processing (NLP), protein models such as Evolutionary Scale Modeling (ESM) [3] and ProtBERT [4] treat amino acid sequences analogous to text. This has resulted in generation of rich, contextual embeddings that capture both structural and functional protein information directly from the sequences. These embeddings have been applied to a variety of prediction and classification tasks. However, applying transformer models to enzyme classification poses some challenges: many enzyme datasets exhibit class imbalance, with certain EC classes (EC 1 and EC2) have significantly more labeled examples than others (EC6 and EC7). Such imbalances may complicate the training process and result in a model biased towards the majority classes. Secondly, protein sequences frequently surpass typical size limits for standard transformer models, which leads to computationally intensive embeddings.

This paper introduces a computational pipeline leveraging transformer-based sequence embeddings to predict the top-level EC classes of enzymes. To address the challenges mentioned above, we use a lightweight variant of the ESM model (esm2_t6_8m_UR50D) to generate embeddings from protein sequences truncated to the first 512 residues. To counter the problem of class imbalance, we apply oversampling methods to balance the training dataset. We evaluate our approach on a carefully curated dataset of 4,311 enzyme sequences from the UniProt database. Additionally, we conduct comprehensive experiments comparing performance of two different classifier architectures and imbalance techniques: a Multi-Layer Perceptron (MLP) and a custom neural network (DeepNN). The remainder of the paper is structured as follows. Section 2 reviews related work, particularly transformer-based methods and computational enzyme classification. Section 3 provides a detailed description of our methodology, including dataset preparation, embedding generation, and classifier training. Section 4 presents experimental results along with analysis and discussions, while Section 5 concludes by summarizing our findings and outlining directions for future research.

2. Related Work

Since the process of experimentally deciphering enzyme function is often both time-consuming and resource-intensive, several computational approaches have emerged over the years to automate the classification of enzymes based on their EC numbers. Early work by Jensen et al. [5] proposed ProtFun, one of the first systems to perform enzyme function prediction using an ensemble-based approach using artificial neural networks (ANNs). These were trained on biologically relevant hand-engineered features like hydrophobicity, amino acid charge, and secondary structure. While ProtFun generalized well to low-homology proteins, its performance remained limited by quality and completeness of the handcrafted features.

ECPred adopted an ensemble-based methodology, combining three independent predictors: SPMap, BLAST-kNN, and Pepstats-SVM are based on subsequences, sequence similarity, and amino acid physicochemical features, respectively [6]. ECPred first predicts whether a query sequence is an enzyme or a non-enzyme, and if identified as an enzyme, it sequentially predicts its top-level EC class, followed by its subclass, subsubclass, and substrate classes.

DEEPre uilized a hybrid deep neural network that combined convolutional and bidirectional recurrent layers [7]. The raw sequences were encoded using one hot and position-specific scoring matrices (PSSM), and their model was able to predict enzyme functions at all levels of EC.

More recently, Kim et al. [8] utilized transformer architectures to predict EC numbers from amino acid sequences. Their model was trained on over 22 million protein sequences, and they addressed class imbalance using focal loss and demonstrated strong performance across EC classes.

DAttProt is a double-scale attention enzyme class prediction that combined transformer-based self-supervised pretraining with multiscale convolutions [9]. Their double-scale attention mechanism captured both spatial and positional relationships within protein sequences.

Building on these advancements, our work uses the pretrained ESM2 model [3] to generate high-dimensional contextual embeddings directly from truncated raw amino acid sequences. These embeddings are then used for top-level EC classification, while addressing class imbalance through oversampling and class weight techniques.

3. Methodology

3.1. Background: Transformer-Based Protein Language Model (ESM)

Transformer models, originally proposed for natural language processing (NLP), rely on the self-attention mechanism to generate context-aware embeddings of sequential data [10]. Building on this foundation, BERT [11] introduced a bidirectional training approach of transformer encoders, showing that self-attention can effectively capture long-term dependencies within sequences.

The self-attention mechanism computes relationships between different tokens in an input sequence through a scaled dot-product operation. Each input token is projected into three vectors: a query q_i from input to i and a set of keys k_j to generate the output. The final output is derived from the weighted average of values v_i through the attention function. Each input token is transformed by three learnable weight matrices to create the corresponding queries, keys, and values. This attention output is then computed as the weighted sum of the value vectors, with weights determined by the similarity of queries and keys. Mathematically, the attention function is formulated as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

where *Q, K,* and *V* are matrices of queries, keys, and values respectively and d_k represents the dimensionality of the input queries and keys, acting as a mechanism to mitigate the potential computational instability caused by large inputs in the attention function, thus ensuring computational stability.

In this study, we will utilize Evolutionary Scale Modeling (ESM), a transformer-based protein language model pretrained on millions of protein sequences to capture evolutionary and structural information [3]. ESM uses a standard transformer architecture composed of stacked self-attention layers, layer normalization, and feed-forward networks. Its pretraining objective enables it to learn both structural and functional properties from raw sequences, making it effective for enzyme classification.

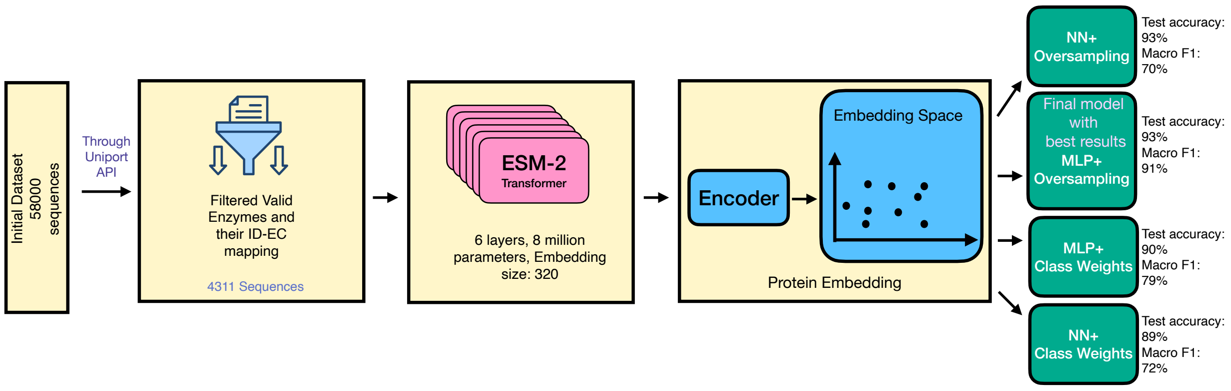


FIGURE 1. Proposed Pipeline

3.2 Proposed Pipeline

The proposed framework leverages the representational power of transformer-based protein language models to classify enzymes according to their top-level EC classes. The pipeline consists of three major stages: data curation and preprocessing, embedding generation using ESM, and multiclass classification. The pipeline begins with the collection of protein sequences from the UniProt database, downloaded in FASTA format. Each sequence is associated with a unique identifier. Using these identifiers, we query the UniProt REST API to retrieve the corresponding enzyme annotations. We focus on predicting the first digit of the EC number, corresponding to the enzyme's primary functional class. Given that the transformer architecture have a quadratic memory complexity with respect to sequence length, we apply an additional pre-processing step where each sequence is truncated to a maximum length of 512 residues.. The truncated sequences are tokenized using the ESM-specific tokenizer. The ESM uses a character-level tokenizer where each amino acid in a protein sequence is treated as a single token. The tokenized sequences are fed then into the pretrained transformer-based language model, specifically the lightweight esm2_t6_8m_UR50D variant of ESM, which consists of six transformer layers and eight million parameters. The model processes the tokenized sequences using multi-layer self attention mechanism to capture residue-residue interactions and generates rich contextual embeddings. These per-residue embeddings are averaged across the sequence length to output a single, fixed-dimensional (320-dimensional) embedding vector which represents each enzyme. Following embedding extraction, we address the class imbalance within our training dataset through two strategies: oversampling and class weights. RandomOverSampler is applied to the training set, increasing the representation of minority enzyme classes by duplicating their instances until we had balanced class proportions. Alternatively, class weights inversely proportional to class frequencies are used to calculate loss to give higher importance to underrepresented classes during training. These embeddings serve as input features for supervised classification models. We evaluate two architectures: a Multi-Layer Perceptron (MLP) and a custom Neural Network, both designed to output a probability distribution over the seven enzyme classes. Class predictions are made by applying a softmax activation to the final logits. The proposed pipeline effectively integrates curated biological sequence data, transformer-based embeddings, and supervised classification techniques to address the enzyme classification task.

4. Experimental Results

In the following section, we detail the experimental setup, performance evaluation, and insights we gained from applying this pipeline to real-world enzyme data.

4.1. Dataset

The dataset used in this study was obtained from the UniProt Knowledgebase [12], initially comprising of approximately 58,000 enzyme sequences. After filtering the entries with valid EC numbers, we obtained a final dataset of 4,311 sequences labeled with their corresponding top-level EC classes. Each enzyme sequence was associated with a unique UniProt ID. The corresponding EC numbers were retrieved via the UniProt REST API. We focused only on the first digit of the EC number, thereby reducing the problem to a seven-class classification. The dataset exhibited significant class imbalance, with

EC 1 and EC 4 having the highest representation, and EC 6 and EC 7 having the lowest. Figure 2 shows the distribution of EC classes in the data. To ensure balanced evaluation, stratified sampling was applied when splitting the dataset into training (80%), validation (10%), testing (10%). Stratification ensured that the class proportions remained consistent across various splits.

4.2. Preprocessing and Embedding Generation

Each protein sequence was first processed to ensure compatibility with transformer models which have quadratic memory complexity with respect to sequence length. The protein sequences collected were truncated to a maximum of 512 amino acids. The processed sequences were then tokenized using the ESM batch converter, which applied a character-level tokenizer. We used the pretrained esm2_t6_8m_UR50D model, consisting of six transform layers and about 8 million parameters. For each sequence, hidden state representations were extracted from layer six of the model, which captured intermediate-level sematic and structural information. These per-residue embeddings were aggregated using mean pooling to produce a single fixed-size 320-dimensional vector representation for each protein. These embeddings serve as input features for all subsequent classification models. To effectively manage GPU memory, we processed the sequences in batches and cleared the memory cache periodically.

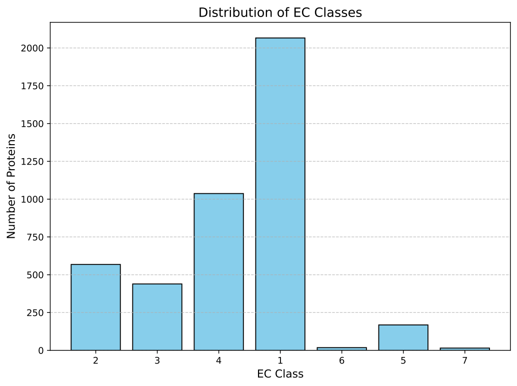


FIGURE 2. EC Class Distribution

4.3. Model Implementation

To evaluate the quality of the generated embeddings for enzyme classification, we designed two supervised learning architectures: a shallow Multi-Layer Perceptron (MLP) and a deeper Neural Network, both of which were implemented using PyTorch [13]. [1] It consisted of two hidden layers with 128 and 64 neurons respectively, each followed by ReLU activations and dropout layers. The DeepNN model comprised of three hidden layers with dimensions 256, 128, 64 respectively to provide greater representation. To address class imbalance, two strategies were explored. In the oversampling approach, we used RandomOverSampler technique to balance the training data by duplicating instances from minority classes. The second approach was computing balanced weights and using them in the CrossEntropyLoss function. This allowed the model to penalize misclassifications of minority classes more heavily. All models were trained using Adam optimizer with an initial learning rate of 0.001 and batch size of 32. The models were trained for up to 100 epochs with early stopping if validation loss failed to improve for 10 consecutive epochs.

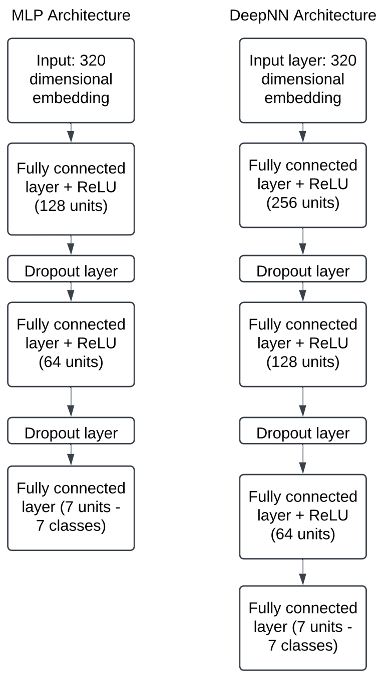

MLP Architecture
Input: 320 dimensional embedding
Fully connected layer + ReLU (128 units)
Dropout layer
Fully connected layer + ReLU (64 units)
Dropout layer
Fully connected layer (7 units - 7 classes)
DeepNN Architecture
Input layer: 320 dimensional embedding
Fully connected layer + ReLU (256 units)
Dropout layer
Fully connected layer + ReLU (128 units)
Dropout layer
Fully connected layer + ReLU (64 units)
Fully connected layer (7 units - 7 classes)

FIGURE 3. Detailed Implementation Architecture

4.4. Results and Discussion

The performance of each model was evaluated on a held-out test set consisting of 432 sequences. In addition to test accuracy, we used several other metrics like precision, recall, F1-

[1] A GitHub repository at Enzyme Classification has been developed, and access can be granted upon request.

score and confusion matrix to provide insight into class-specific behavior. The MLP model trained with oversampling achieved the highest performance achieving a test accuracy of 93% and a macro F1-score of 91%. This model showcased strong generalization across both majority and minority classes, effectively mitigating the class imbalance. As shown in Table 2, majority classes EC 1 and EC 4 achieved very high precision and recall values while minority classes also achieved competitive recall scores.

The DeepNN model trained with random oversampling also reached a similar test accuracy of 93%, however its macro F1-score was relatively lower at 70% as while it performed well on majority classes, it struggled on minority ones. When using class weights, both the MLP and DeepNN exhibited slightly lower performance. The MLP with class weights achieved a test accuracy of 90% and a macro F1-score of 79%, while the DeepNN with class weights achieved a test accuracy of 89% and a macro F1-score of 72%. A comparison of different models and strategies used is shown in Table 1.

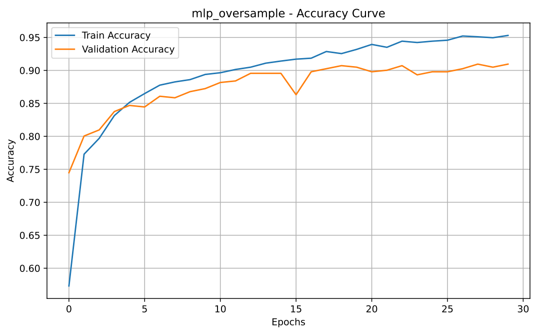
Model	Accuracy	Macro F1-score
MLP + Oversampling	93%	91%
DeepNN + Oversampling	93%	70%
MLP + Class Weights	90%	79%
DeepNN + Class Weights	89%	72%

TABLE 1. Comparison of different models and imbalance handling strategies

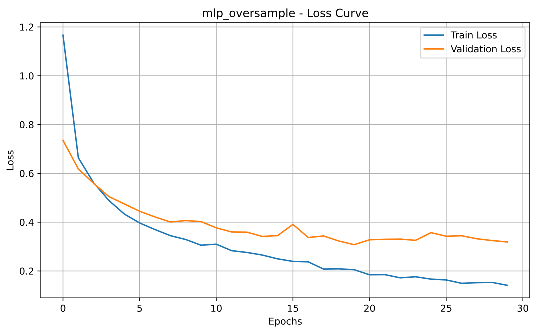
Class	Precision	Recall	F1-score	Support
1	0.98	0.96	0.97	195
2	0.89	0.92	0.91	52
3	0.75	1.00	0.85	47
4	0.97	0.83	0.90	115
5	0.81	0.94	0.87	18
6	1.00	0.75	0.86	4
7	1.00	1.00	1.00	1
Macro Avg	0.91	0.92	0.91	432
Weighted Avg	0.94	0.93	0.93	432

TABLE 2. Classification Metrics for MLP + oversampling model

Overall, random oversampling proved to be a more effective strategy for handling class imbalance. Moreover, the simpler architecture of the MLP seemed to outperform the deeper network. This suggested that the generated ESM embeddings already captured sufficiently rich information. Based on our observations, we selected the MLP model trained with random oversampling as the final model. It's training and validation accuracy and loss curves are presented in Figure 4.



(a) Training and validation accuracy



(b) Training and validation loss

FIGURE 4. Training and validation performance of the MLP model across epochs.

Moreover, Figure 5 provides detailed insights into the MLP model's classification performance with oversampling. The model showcases excellent performance across both majority and minority classes, indicating that the oversampling strategy was effective in mitigating class imbalance.

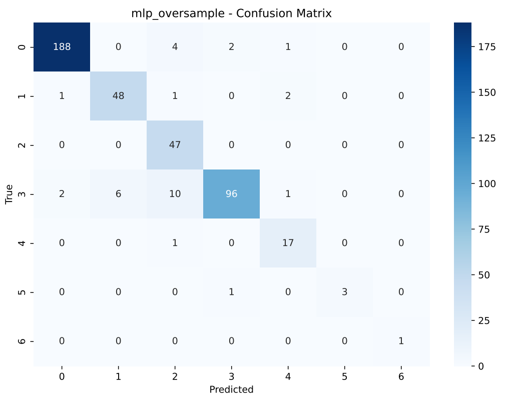


FIGURE 5. Confusion matrix of the model with MLP

5. Conclusions and Future Work

This study presents a pipeline for enzyme classification using embeddings from the pretrained transformer model ESM2. We evaluated the embeddings with multiple classifiers, including an MLP and a custom neural network, and addressed class imbalance using both oversampling and class weighting. Oversampling the minority classes led to better overall performance, with the MLP trained with RandomOverSampler achieving 93.5% test accuracy and strong precision, recall, and F1 scores, even for rare classes. Although the neural network with class weights also performed competitively, it was less effective at handling minority classes. Our findings demonstrate the potential of combining transformer-based protein embeddings with balancing techniques for enzyme function prediction. Future directions include extending classification to full EC hierarchies, exploring larger ESM variants, and applying advanced pooling methods to better capture functionally important residues.

References

[1] A. Cornish-Bowden, "Current iubmb recommendations on enzyme nomenclature and kinetics," *Perspectives in Science*, vol. 1, no. 1-6, pp. 74–87, 2014.

[2] K. A. Khan, S. A. Memon, and H. Naveed, "A hierarchical deep learning based approach for multi-functional enzyme classification," *Protein Science*, vol. 30, no. 9, pp. 1935–1945, 2021.

[3] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, A. dos Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido *et al.*, "Language models of protein sequences at the scale of evolution enable accurate structure prediction," *bioRxiv*, 2022.

[4] N. Brandes, D. Ofer, Y. Peleg, N. Rappoport, and M. Linial, "Proteinbert: a universal deep-learning model of protein sequence and function," *Bioinformatics*, vol. 38, no. 8, pp. 2102–2110, 2022.

[5] L. J. Jensen, R. Gupta, N. Blom, D. Devos, J. Tamames, C. Kesmir, H. Nielsen, H. H. Stærfeldt, K. Rapacki, C. Workman *et al.*, "Prediction of human protein function from post-translational modifications and localization features," *Journal of molecular biology*, vol. 319, no. 5, pp. 1257–1265, 2002.

[6] A. Dalkiran, A. S. Rifaioglu, M. J. Martin, R. Cetin-Atalay, V. Atalay, and T. Doğan, "Ecpred: a tool for the prediction of the enzymatic functions of protein sequences based on the ec nomenclature," *BMC bioinformatics*, vol. 19, pp. 1–13, 2018.

[7] Y. Li, S. Wang, R. Umarov, B. Xie, M. Fan, L. Li, and X. Gao, "Deepre: sequence-based enzyme ec number prediction by deep learning," *Bioinformatics*, vol. 34, no. 5, pp. 760–769, 2018.

[8] G. B. Kim, J. Y. Kim, J. A. Lee, C. J. Norsigian, B. O. Palsson, and S. Y. Lee, "Functional annotation of enzyme-encoding genes using deep learning with transformer layers," *Nature Communications*, vol. 14, no. 1, p. 7370, 2023.

[9] K. Lin, X. Quan, C. Jin, Z. Shi, and J. Yang, "An interpretable double-scale attention model for enzyme protein class prediction based on transformer encoders and multi-scale convolutions," *Frontiers in Genetics*, vol. 13, p. 885627, 2022.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019, pp. 4171–4186.

[12] The UniProt Consortium, "Uniprot: the universal protein knowledgebase in 2025," *Nucleic Acids Research*, vol. 53, no. D1, pp. D609–D617, Jan. 2025, published: 18 November 2024. [Online]. Available: https://doi.org/10.1093/nar/gkae1010

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.