Pfam Protein Sequence Classification

Oscar Windrath-Carr

Dataset Analysis

The dataset under consideration consists of protein sequences categorized into various Pfam families. Upon examination, it was observed that 4,858 Pfam families are exclusive to the training dataset, while 13,071 Pfam families are common across the train, dev and test sets. The exclusive families, due to their lack of representation in both the dev and test datasets, were excluded from further analysis and model training.

An analysis of the family size distributions across the three datasets (Figure 1) reveals that all sets exhibit significant class imbalance. Specifically, a large proportion of families are represented by a relatively small number of sequence examples, with a few families being highly represented, while others contain only a limited number of sequences. This inherent imbalance highlights the challenge of training a model to adequately generalize across the full range of families. In order to mitigate this issue and streamline the learning process, the datasets were restricted to the 100 most populous families within the training dataset. This step not only reduces the computational complexity of the problem but also aligns with future models' ability to learn from more frequent and diverse family examples.

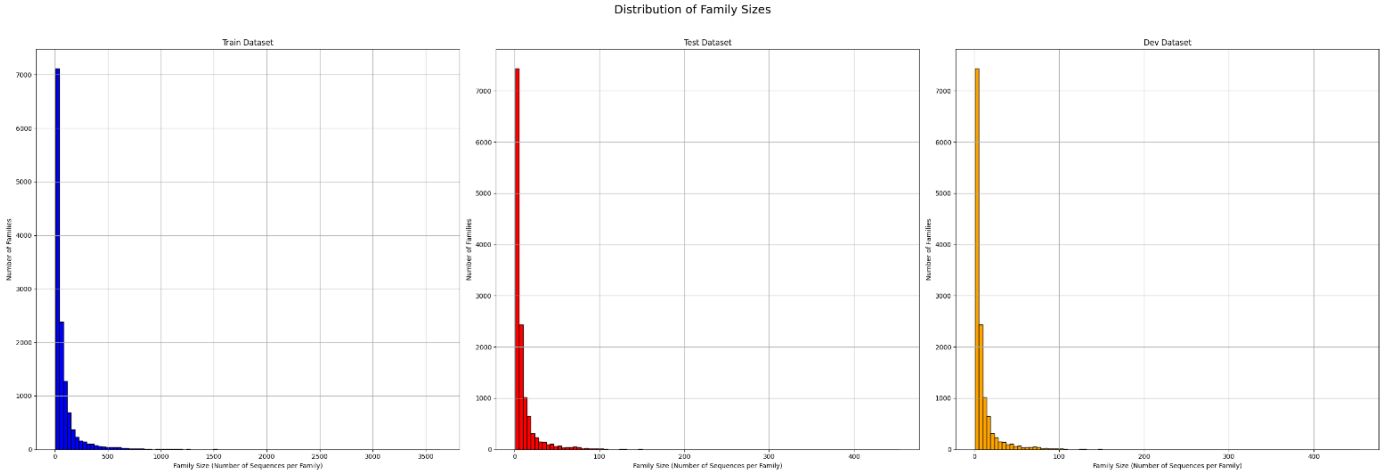

Figure 1: Distribution of family sizes for each dataset from left to right, train, test and dev.

Furthermore, the amino acid frequency distribution (Figure 2) reveals that certain amino acids, such as X, U, and B, are notably underrepresented, while O and Z are entirely absent. These findings highlight the variability in amino acid composition across protein families, which may influence model performance and generalization.

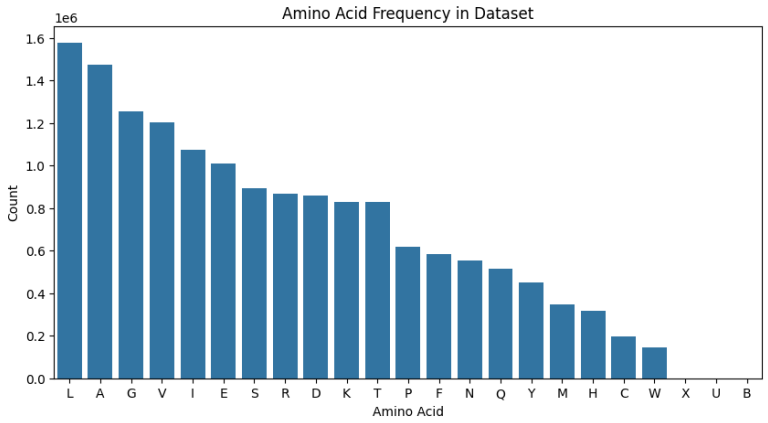

Figure 2: Amino Acid Frequency in all reduced datasets combined.

Sequence length distributions (Figure 3) show that the lengths of protein sequences across the datasets are similarly distributed, with the majority of samples ranging between 50 and 200 amino acids. Given this distribution, no additional sequence length balancing was deemed necessary for the analysis.

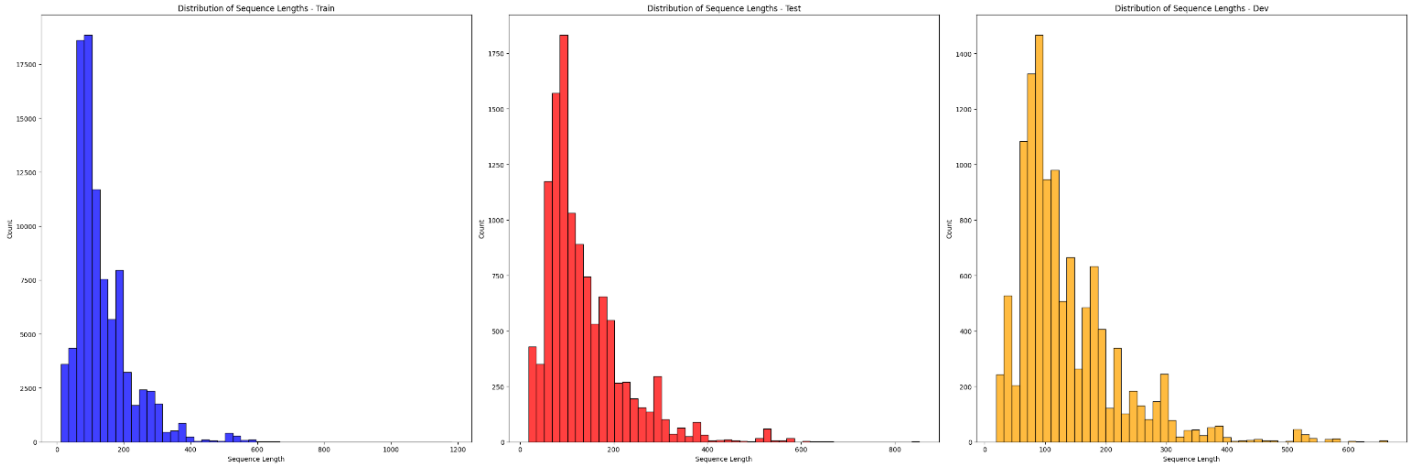

Figure 3: Distribution of sequence lengths for each reduced dataset from left to right, train, test and dev.

Method Explanation

Amino acid sequences were pre-processed to standardise rare amino acids by replacing them with a placeholder ('X'). This ensures consistency and prevents rare amino acids from introducing noise into the models. The sequences were one-hot encoded for the baseline model, where each amino acid is represented as a 20-dimensional binary vector. Due to computational limits, sequences were truncated to a maximum length of 128, balancing training efficiency and model interpretability.

Class labels were encoded using label encoding, converting protein family identifiers into numerical values. This encoding is compatible with both models used and facilitates learning.

The baseline model uses a Convolutional Neural Network (CNN), which is effective for extracting spatial features in sequence data. The architecture consists of:

1. Input Layer: One-hot encoded sequences.
2. Convolutional Layers: These layers extract local patterns from the sequences.

3. Pooling Layer: Max-pooling reduces the feature dimensionality while retaining key information.
4. Flatten Layer: Converts 2D outputs from the pooling layer into a 1D vector for further processing.
5. Dense Layer: Captures complex relationships between features.
6. Softmax Output: Classifies the sequence into one of the 100 protein families.

A CNN was chosen for their success in pattern recognition tasks, as they excel in learning local sequence features and are computationally efficient for this task.

The second model utilizes a transformer-based architecture with the ESM2 tokenizer/model to generate embeddings. The architecture includes:

1. Transformer Blocks: Two blocks with multi-head self-attention mechanisms to capture long-range dependencies in the sequences.
2. Feedforward Network (FFN): Introduces non-linearity after the attention mechanism to model higher-level interactions.
3. Dropout and Normalization Layers: Help stabilize training and prevent overfitting.
4. Pooling Layer: A global average pooling layer reduces sequence length to a single feature vector.
5. Dense Layers: Classifies the sequence based on the learned features.

The ESM2 model was selected for generating embeddings due to its pretraining on large-scale protein data. It effectively captures complex biological patterns and relationships in amino acid sequences, outperforming traditional methods for protein sequence analysis. By using ESM2, we leverage a robust, pretrained model that significantly enhances feature extraction for protein family classification.

Both models employed early stopping, monitoring the accuracy on the validation dataset to avoid overfitting. Training halts when the validation performance plateaus. This method ensures generalizability and avoids the model fitting noise in the training data.

Experiment Description

Two models were evaluated: a baseline CNN using one-hot encoded data and a more complex FFNN with transformer layers trained on protein embeddings. The goal was to assess whether the transformer-based model, leveraging pre-trained embeddings, offers advantages in capturing long-range dependencies and improving classification accuracy compared to the CNN, which focuses on local patterns within the sequences. This is all in the context of time and computational restrictions for this task.

Result Analysis

The CNN-based model achieved a high classification accuracy of 99.32% on the test dataset. As shown in Figure 4, the model's performance was consistent across the majority of classes, with high precision and recall scores. This suggests that convolutional architectures, when trained on sufficiently large and balanced subsets of the dataset, can effectively capture sequence-level patterns and dependencies required for a simplified family classification problem.

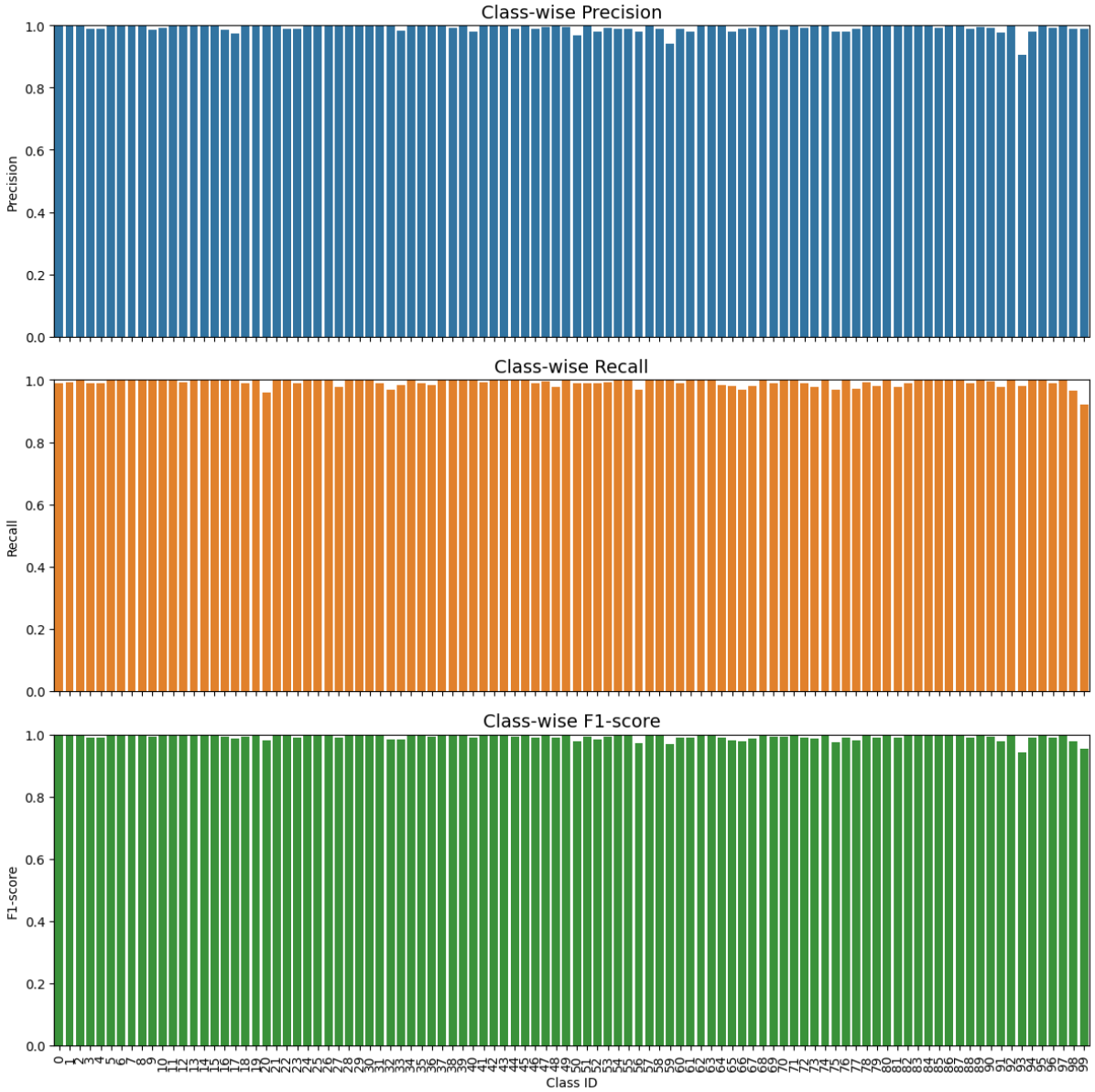

Figure 4: Recall, precision and F1-score per family for CNN predictions on the test dataset.

The Transformer-based FFN model achieved a test accuracy of 92.40%, which is lower than the CNN model. The performance distribution across classes (Figure 5) reveals that while some families were classified with high accuracy, others exhibited significantly lower predictive performance. This suggests that the Transformer-based model may require additional computational resources and hyperparameter tuning to achieve its full potential. The limitations in model complexity and training time likely contributed to the observed performance gap.

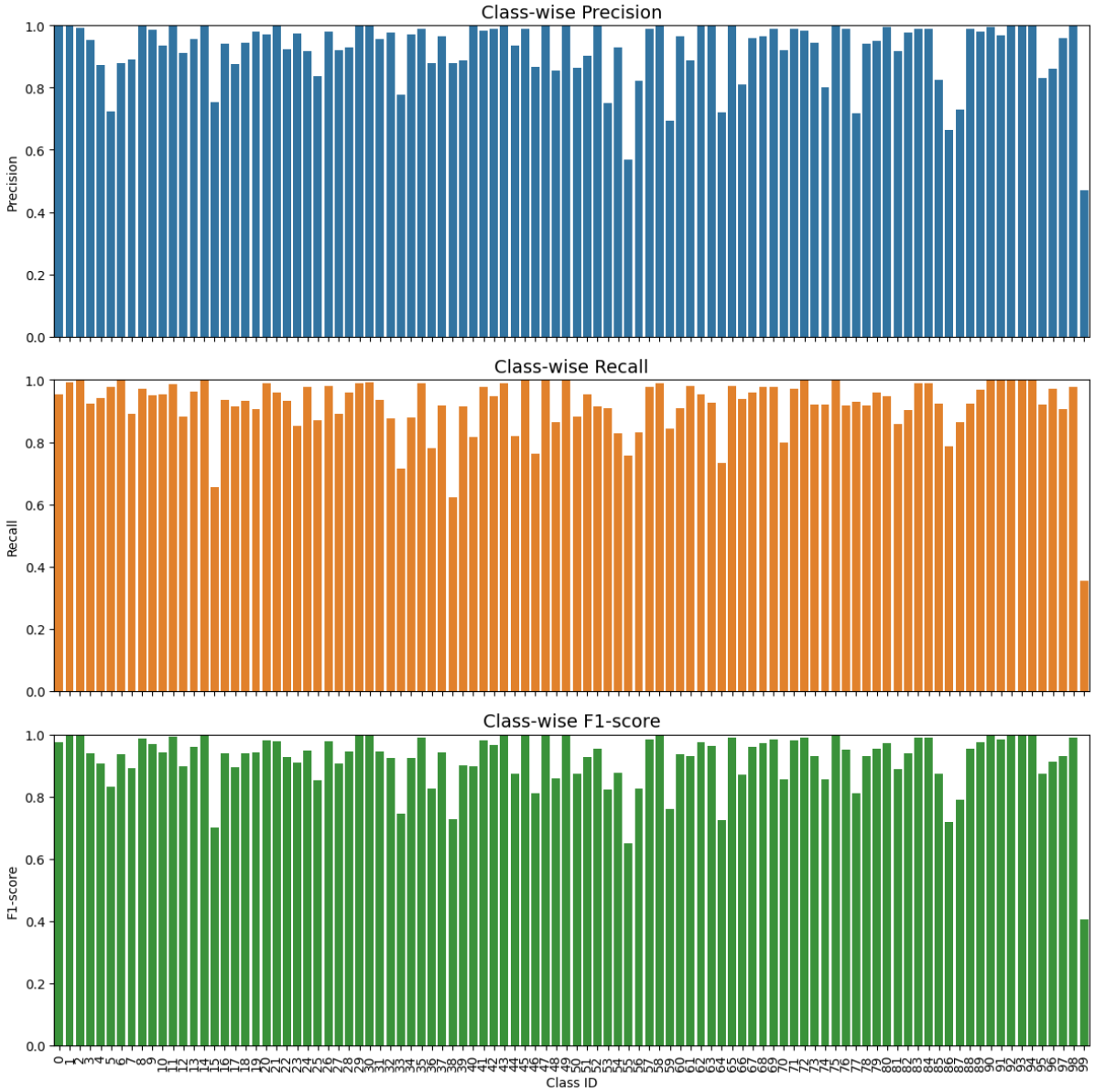

Figure 5: Recall, precision and F1-score per family for the Transformer-based FFN predictions on the test dataset.

Despite the lower accuracy, the Transformer FFN model remains a promising approach, particularly for scaling the classification task to a larger number of families. With increased computational power and extended training time, this architecture is expected to yield better generalization and improved performance across all classes. Given that Transformer models have shown strong performance in other sequence-based tasks, future work could explore further optimizations, including deeper architectures, alternative self-attention mechanisms, and pretraining strategies to enhance performance on the full dataset.