Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 807–812
DOI: 10.15439/2025F4532
ISSN 2300-5963 ACSIS, Vol. 43

A Multi-Stage Framework for Chess Puzzle Difficulty Prediction

Ling Cen
Neurospark Lab
Singapore
cen.ling@neurosparklab.tech

Jiahao Cen
University College Dublin
Ireland
cen.jiahao@ucdconnect.ie

Malin Song
Anhui University of Finance and Economics
China
songml@aufe.edu.cn

Zhuliang Yu
South China University of Technology
China
zlyu@scut.edu.cn

***Abstract*—Accurately estimating the difficulty of the chess puzzle is important for adaptive training systems, personalized recommendations, and large-scale content curation. Unlike engine evaluations optimized for perfect play, this task involves modeling human-perceived solving difficulty, typically expressed by Glicko-2 ratings. We present a multi-stage framework developed for the FedCSIS 2025 Challenge. The method trains four rating-banded neural regression models in different Elo ranges to capture localized difficulty patterns and reduce bias from unbalanced data. Their predictions are combined with statistical attributes, including success probabilities, failure distributions, and solution length, through a feature-based regression stage to improve cross-range generalization. A final calibration step adjusts the output to statistically plausible rating levels, mitigating systematic prediction biases without adding computational complexity. An additional mask selection procedure was explored as part of the competition extension to identify 10% of the puzzles that are most likely to benefit from the refined evaluation. The proposed solution ranked 5th on the public leaderboard and 6th in the final standings. These results demonstrate that a lightweight and interpretable regression pipeline can achieve competitive precision in modeling human-perceived chess puzzle difficulty.**

***Index Terms*—Chess puzzle difficulty prediction, Multi-stage framework, Regression, Calibration, Mask selection**

I. Introduction

PREDICTING the difficulty of chess puzzles is a challenging but important problem that integrates machine learning, human cognition, and game theory. The difficulty of the puzzle on platforms such as Lichess[1] is expressed as a Glicko-2 rating[2] that dynamically updates based on the success or failure of the players. Automating this rating prediction accelerates puzzle curation, enables personalized recommendations for players of varying skill levels, and provides insight into which tactical or strategic motifs challenge human solvers the most. This is also the goal of the FedCSIS 2025 Challenge [1] organized on the KnowledgePit platform[3].

A. Related Work

Research on puzzle difficulty prediction has advanced considerably in recent years. Early approaches used handcrafted features with classical machine learning: Björkqvist [2] modeled puzzle "puzzlingness" via positional and tactical indicators, and Rafaralahy [3] applied pairwise learning-to-rank to capture ordinal relations between puzzles. With the rise of deep learning, sequence-based and representation-learning methods became prominent. Ruta et al. [4] applied convolutional neural networks (CNNs) to predict puzzle ratings directly from move sequences, Miłosz and Kapusta [5] proposed Transformer-based models treating puzzles as sequences, and Omori and Tadepalli [6] developed a CNN-LSTM model incorporating moves and timing to jointly estimate puzzle and player ratings.

The IEEE BigData Cup 2024 [7] demonstrated the success of hybrid pipelines that integrate statistical features and learned representations. Woodruff et al. [8] trained neural models with rating-based features, while Schütt et al. [9] introduced a human-problem-solving-inspired architecture combining move distribution analysis and cognitive heuristics.

B. Motivation and Contributions

Despite these advances, two key challenges remain. First, models trained across broad rating ranges often exhibit systematic biases, overestimating simple puzzles and underestimating long tactical sequences due to global error optimization. Second, while models trained on different rating distributions provide complementary perspectives, effectively integrating them to improve generalization is non-trivial.

To address these challenges, we propose a multi-stage framework that first trains four independent neural models on separate rating ranges to capture localized difficulty patterns and reduce bias caused by imbalanced data. Their predictions are then combined with statistical attributes, including success probabilities, failure distributions, and solution length, in a feature-based regression stage to improve cross-range generalization. A final calibration step adjusts the outputs toward statistically plausible rating levels, reducing systematic prediction biases without increasing computational complexity. Additionally, the framework was extended with a mask selection procedure to identify the 10% of puzzles most likely to benefit from refined evaluation, as required in the competition extension.

Our method ranked 5th on the public leaderboard and 6th in the final standings of the FedCSIS 2025 Challenge, demonstrating that a lightweight, interpretable pipeline can achieve competitive accuracy in modeling human-perceived chess puzzle difficulty.

[1] https://lichess.org/
[2] https://en.wikipedia.org/wiki/Glicko_rating_system
[3] https://knowledgepit.ai/

The remainder of this paper is organized as follows. Section II briefly describes the challenge, dataset, and evaluation metric. Section III presents the methodology. Section IV discusses the mask selection task. Section V shows xperimental results. Finally, Section VI concludes the paper and outlines future directions.

II. FEDCSIS 2025 CHALLENGE

This competition continues the established task from IEEE Big Data 2024 [7], asking participants to predict the perceived difficulty of chess puzzles based on puzzle configurations and human solving statistics. The objective remains to estimate the Glicko-2 rating assigned to each puzzle, which reflects the likelihood that players of various skill levels can solve it, using only board position and solution moves.

A. Dataset

The official dataset consists of a large annotated training set and a separate unlabeled test set. The training set contains approximately 4.56 million puzzles, each labeled with a human-derived difficulty rating, while the test set includes 2,235 puzzles sharing the same feature structure but without ratings.

Each puzzle is described by multiple feature groups derived from both game records and engine analysis:

- Core identifiers: a unique puzzle ID, board state in Forsyth–Edwards Notation (FEN)[4], and the solution sequence in Portable Game Notation (PGN)[5].
- Human performance indicators (training only): Glicko-2 rating, rating deviation, popularity, and the number of attempts.
- Contextual metadata (training only): puzzle themes, associated game URLs, and opening tags.
- Engine-based success statistics: estimated success probabilities for players at different Elo levels, provided separately for rapid and blitz time controls (10 columns each), approximating human solving likelihood across skill tiers.

Puzzle ratings range from about 400 (trivial tactics) to over 3000 Elo (complex master-level combinations). The distribution is heavily imbalanced, with most puzzles clustered around intermediate difficulty. No official validation set is provided; participants typically constructed small internal validation splits for hyperparameter tuning and model selection.

A training sample (puzzle) of the initial state of the chessboard decoded by FEN and the rating is illustrated in Figure 1.

B. Evaluation Protocol

Predictions are evaluated using the mean squared error (MSE) between the predicted and actual ratings:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2,$$

[4]https://en.wikipedia.org/wiki/ForsythEdwards_Notation
[5]https://en.wikipedia.org/wiki/Portable_Game_Notation

Figure 1. Example of the chess puzzle Rating 1902: initial state of the chessboard decoded by FEN: "r6k/pp2r2p/4Rp1Q/3p4/8/1N1P2R1/PqP2bPP/7K b - - 0 24".

where N is the number of puzzles in the test set. The test set is partitioned internally, with 10% used for provisional public leaderboard updates and the remaining 90% for final ranking. Only the public subset is visible during the competition.

C. Optional Mask Extension

An additional post-evaluation extension allows participants to submit a binary mask marking 10% of test puzzles predicted to have the highest potential benefit from re-evaluation. For these selected puzzles, predictions are replaced by ground-truth ratings, and MSE is recalculated. While this mask-based scoring does not affect the main leaderboard, it serves as an auxiliary measure of a model's ability to recognize cases where its predictions are less reliable.

III. METHODS

Our approach employs a multi-stage pipeline that integrates localized learning of puzzle difficulty with statistical adjustment to correct systematic biases. The design philosophy stems from two key observations. First, the relationship between puzzle features—such as tactical motifs, material configurations, and success probabilities—and human-perceived difficulty is not uniform across the rating spectrum; a single global model tends to bias its predictions toward the overrepresented mid-range puzzles. Second, models trained purely to minimize global mean squared error often fail to respect structural regularities observed in human solving behavior, particularly for very easy or very hard puzzles. We therefore decompose the task into specialized components: localized rating-band models to better learn within-range patterns, a meta-learning step to integrate complementary predictions and additional structural attributes, a calibration step to align outputs with statistical difficulty indicators, and a final averaging procedure to stabilize predictions.

Figure 2 outlines this four-stage pipeline. The first stage trains multiple models specialized for different rating bands to disentangle the heterogeneous difficulty distribution. In the second stage, their predictions are combined with auxiliary

features using tree-based meta-regressors trained on a fixed validation subset. The third stage applies a statistical calibration that shifts the predictions toward historically plausible difficulty levels by exploiting failure distributions and their higher-order characteristics. Finally, the calibrated regressors are averaged to reduce variance and further improve generalization.

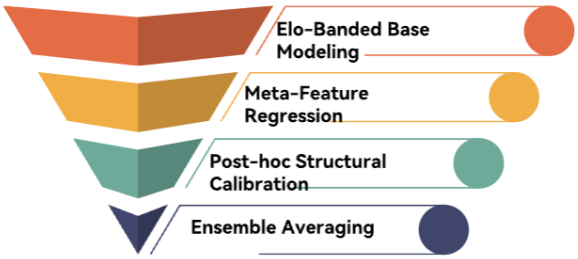

Figure 2. Overview of the proposed prediction pipeline.

A. Rating-Banded Base Models

The first stage employs a rating-banded learning strategy to address heterogeneous solving patterns across Elo levels. Lower-rated puzzles, often simple mate-in-one or basic forks, exhibit solving distributions heavily concentrated in low-skill buckets, whereas higher-rated puzzles show sparse but sharper failure transitions at advanced skill levels. A single global regressor tends to minimize global loss by fitting the over-represented mid-range, which leads to biased underestimation for high-rated puzzles. Inspired by the band-wise modeling idea explored in prior work [8], we explicitly partition the data into four difficulty bands and train separate regressors for each band to focus on localized difficulty dynamics.

Let the global training set be $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the structured feature vector and y_i the true difficulty rating. We define four disjoint subsets:

$$\mathcal{D}_b = \{(x_i, y_i) \in \mathcal{D} \mid L_b \le y_i < U_b\}, \quad b \in \{1, 2, 3, 4\}, \tag{1}$$

with rating intervals:

$$[L_b, U_b) \in \{[0, 1000),\ [1000, 1400),\ [1400, 1700),\ [1700, \infty)\}. \tag{2}$$

This segmentation allocates dedicated modeling capacity to high-rated puzzles ($b = 4$) while reducing mid-range dominance.

Each band b is trained independently using a multi-layer perceptron (MLP) $f_b(\cdot; \theta_b)$ optimized for mean squared error:

$$\mathcal{L}_b(\theta_b) = \frac{1}{|\mathcal{D}_b|} \sum_{(x_i, y_i) \in \mathcal{D}_b} \left(y_i - f_b(x_i; \theta_b)\right)^2. \tag{3}$$

The final band-specific prediction for a puzzle i is:

$$\hat{y}_i^{(b)} = f_b(x_i; \theta_b). \tag{4}$$

The input vector x_i is constructed from all numeric structural features provided in the official dataset preprocessing pipeline, including engine-estimated success probabilities (rapid and blitz modes), their aggregated statistics, failure-related indicators, and a small set of board- and sequence-level descriptors (e.g., material balance and move length). Non-numeric metadata such as puzzle themes or game URLs are excluded from the training features. All continuous features are normalized to $[0, 1]$, while binary indicators (e.g., checkmate markers) are one-hot encoded.

Each f_b is trained with early stopping using 10% of $\mathcal{D}_b$ as a validation set for hyperparameter tuning. Although every model is specialized to its own band, during inference all four models are applied to every puzzle:

$$\hat{\mathbf{y}}_i = [\hat{y}_i^{(1)}, \hat{y}_i^{(2)}, \hat{y}_i^{(3)}, \hat{y}_i^{(4)}], \tag{5}$$

producing complementary perspectives that will be integrated in the subsequent meta-learning stage.

B. Meta-Feature Regression

The second stage integrates the outputs of the banded base models with additional statistical descriptors through a meta-learning framework. Let

$$\mathbf{p}_i = [p_i^{(1)}, p_i^{(2)}, p_i^{(3)}, p_i^{(4)}] \tag{6}$$

denote the predictions of the four base models for puzzle i, where each element corresponds to a specific Elo-banded model. The statistical aggregates are expressed as:

$$\mu_i = \frac{1}{4} \sum_{m=1}^{4} p_i^{(m)}, \quad \sigma_i^2 = \frac{1}{4} \sum_{m=1}^{4} (p_i^{(m)} - \mu_i)^2, \tag{7}$$

$$p_{\max,i} = \max_m (p_i^{(m)}), \quad p_{\min,i} = \min_m (p_i^{(m)}), \tag{8}$$

$$\Delta_{mn,i} = |p_i^{(m)} - p_i^{(n)}| \quad (m \neq n). \tag{9}$$

These aggregates highlight disagreement patterns, which are known to be strong signals for meta-regression.

Structural features are denoted as:

$$s_i = [\mu_i^{\text{succ}},\ p_i^{\text{fail}},\ r_i^{\text{inf}},\ \gamma_i^{\text{fail}}], \tag{10}$$

where μ_i^{succ} is the mean engine-estimated success probability across rating buckets, p_i^{fail} is the aggregated failure probability, r_i^{inf} is the rating bucket with the maximum success-rate gradient (inflection point), and γ_i^{fail} is the skewness of the failure probability distribution. Interaction terms combine base predictions and puzzle-level indicators:

$$\phi_{ik} = p_i^{(m)} \times c_{ik}, \tag{11}$$

where c_{ik} represents puzzle-specific complexity indicators, such as the number of moves or checks. The complete meta-feature vector for puzzle i is:

$$\mathbf{x}_i = [\mathbf{p}_i,\ \mu_i,\ \sigma_i,\ p_{\max,i},\ p_{\min,i},\ \Delta_{mn,i},\ s_i,\ \phi_{ik}]. \tag{12}$$

Three independent gradient boosting regressors are used as meta-learners to map $\mathbf{x}_i$ to the final meta-prediction:

$$\hat{y}_i = f_{\text{meta}}(\mathbf{x}_i; \Theta), \tag{13}$$

where Θ represents the parameters of CatBoost, LightGBM, or XGBoost. The choice of these three algorithms is motivated by their complementary characteristics: CatBoost [10] handles heterogeneous feature distributions effectively through ordered boosting and is robust to overfitting; LightGBM [11] is optimized for large-scale structured data, using histogram-based leaf-wise tree growth to achieve high computational efficiency; XGBoost [12] provides strong generalization by combining second-order optimization with regularization, and often captures non-linear feature interactions overlooked by the other two. Employing these three regressors increases model diversity, which is critical for the final averaging stage to reduce variance and enhance generalization, which We team had used it before in the other competition [13].

All meta-learners are trained using 5-fold cross-validation with out-of-fold base predictions to avoid information leakage. The training objective minimizes mean squared error:

$$\mathcal{L}_{\text{meta}} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 . \tag{14}$$

C. Prediction Calibration

While meta-regression substantially improves the overall consistency of predictions, residual systematic biases persist due to the heterogeneous nature of puzzles. Easy puzzles (e.g., short mate-in-one combinations) are frequently overestimated, whereas deeper tactical sequences or complex positional motifs are often underestimated. To address this, we incorporate a post-hoc calibration stage that aligns predictions with structural patterns inferred from aggregated human performance statistics.

Let the engine-provided success probabilities across J rating buckets for puzzle i be SuccessProb_{ij}. The failure probability for bucket j is:

$$\text{FailProb}_{ij} = 1 - \text{SuccessProb}_{ij}. \tag{15}$$

A primary structural estimate of puzzle difficulty is computed as the failure-probability-weighted average rating:

$$\hat{y}_{\text{struct},i} = \frac{\sum_{j=1}^{J}\text{FailProb}_{ij}\cdot R_j}{\sum_{j=1}^{J}\text{FailProb}_{ij}}, \tag{16}$$

where R_j is the representative rating (Elo) of bucket j. Intuitively, if failures are concentrated in higher buckets, the structural estimate shifts toward a higher rating.

The raw structural estimate does not fully account for sharp transitions in solving probability or asymmetry in its distribution. Two higher-order indicators are introduced:

1) Inflection rating r_i^*: the rating bucket with the steepest drop in solving probability:

$$r_i^* = \arg\max_j \left|\frac{\partial p_{ij}}{\partial R_j}\right|, \tag{17}$$

where p_{ij} is the engine-estimated solving probability for bucket j and R_j the corresponding representative rating.

2) Skewness of failure distribution γ_i: a measure of asymmetry in the failure probability:

$$\gamma_i = \frac{\frac{1}{J}\sum_{j=1}^{J}(q_{ij}-\bar{q}_i)^3}{\left(\frac{1}{J}\sum_{j=1}^{J}(q_{ij}-\bar{q}_i)^2\right)^{3/2}}, \quad \bar{q}_i = \frac{1}{J}\sum_{j=1}^{J} q_{ij}, \tag{18}$$

where $q_{ij} = 1 - p_{ij}$ is the failure probability for bucket j.

The refined structural estimate integrates these two indicators:

$$\hat{y}_{\text{refined},i} = \hat{y}_{\text{struct},i} + \lambda_1\left(r_i^{\text{inf}} - \mu_R\right) + \lambda_2\,\gamma_i^{\text{fail}}, \tag{19}$$

where μ_R is the mean bucket rating, and λ_1, λ_2 are empirical weights controlling the contribution of sharp transitions and asymmetry.

The final calibrated prediction $\tilde{y}_i$ is a convex combination of the meta-regressor output $\hat{y}_i$ and the refined structural estimate:

$$\tilde{y}_i = (1-\alpha_i)\,\hat{y}_i + \alpha_i\,\hat{y}_{\text{refined},i}. \tag{20}$$

The blending weight α_i depends on the estimated structural difficulty, with easier puzzles receiving stronger correction:

$$\alpha_i = \begin{cases} 0.35, & \hat{y}_{\text{struct},i} < 1400, \\ 0.28, & 1400 \le \hat{y}_{\text{struct},i} < 1800, \\ 0.20, & \hat{y}_{\text{struct},i} \ge 1800. \end{cases} \tag{21}$$

This formulation enforces stronger adjustment for trivial puzzles—aligning their predicted ratings closer to historically plausible ranges—while retaining the flexibility of the meta-regressors for complex high-rated puzzles. The segmented weighting mimics the behavior observed in residual-analysis curves from validation, where lower-rated puzzles exhibited significantly larger over-prediction variance.

D. Averaging of Calibrated Models

The final stage combines the calibrated outputs of the three meta-regressors to produce a robust and stable prediction. Let $\tilde{y}_i^{(k)}$ denote the calibrated prediction for puzzle i from the k-th meta-regressor, where $k = 1, 2, 3$. The final predicted difficulty rating is obtained by taking the simple arithmetic mean over these calibrated outputs:

$$\hat{y}_i^{\text{final}} = \frac{1}{K}\sum_{k=1}^{K}\tilde{y}_i^{(k)}, \quad K = 3. \tag{22}$$

The uniform weighting is chosen deliberately instead of learned weights to reduce the risk of overfitting, as the validation subset is relatively small. From a variance-reduction perspective, assuming that the calibrated models exhibit only moderate pairwise error correlations, the variance of the averaged prediction can be expressed as:

$$\text{Var}(\hat{y}_i^{\text{final}}) \approx \frac{1}{K^2}\sum_{k=1}^{K}\text{Var}(\tilde{y}_i^{(k)}) + \frac{2}{K^2}\sum_{m<n}\text{Cov}(\tilde{y}_i^{(m)}, \tilde{y}_i^{(n)}). \tag{23}$$

Because the three calibrated regressors are trained independently and incorporate different feature-interaction mechanisms, their prediction errors are not perfectly correlated.

This diversity directly reduces the ensemble variance and improves stability, which validates the choice of using multiple complementary meta-regressors in Stage 2.

IV. Mask Prediction

The optional mask prediction task was introduced as an extension to evaluate a model's ability to identify the most problematic predictions, i.e., puzzles where replacing the predicted ratings with ground-truth values would yield the largest reduction in the overall evaluation error. This task measures not only prediction accuracy but also the model's capability for uncertainty estimation.

A. Task Definition

For each submitted solution, the organizers defined a binary "perfect mask" indicating the 10% of test puzzles that contributed the most to the prediction error. Formally, the perfect mask for puzzle i is denoted by $M_i \in \{0, 1\}$, and the predicted mask by $\hat{M}_i \in \{0, 1\}$.

The evaluation was based on the model's original Mean Squared Error (MSE) scores:

1) Perfect Score P: the lowest MSE achievable if the 10% worst-predicted puzzles were replaced with ground-truth values;

2) New Score N: the MSE after applying the participant's submitted mask, where only the puzzles marked by $\hat{M}_i = 1$ are replaced with ground-truth values;

3) Final Ranking Criterion:

$$mask_{score} = \frac{N}{P}, \tag{24}$$

with values closer to 1 indicating better mask quality. A ratio of exactly 1 corresponds to perfectly recovering the ideal mask.

Unlike the main difficulty-prediction task, this mask task does not require modifying rating predictions directly; instead, it evaluates how effectively a model can identify the most uncertain or systematically biased cases.

B. Proposed Mask Prediction Strategy

To construct the predicted mask, we developed a structure- and residual-guided selection strategy designed to identify puzzles with high prediction error and high structural confidence. The method proceeds in three steps:

1) Residual scoring: The absolute residual between the calibrated meta-regression prediction $\tilde{y}_i$ and the refined structural estimate $\hat{y}_{\text{refined},i}$ is used to measure disagreement:

$$r_i = |\tilde{y}_i - \hat{y}_{\text{refined},i}|. \tag{25}$$

Puzzles with larger residuals are considered more likely to be mispredicted.

2) Structure-confidence weighting: Puzzles with low structural uncertainty are given higher priority. A confidence weight is defined as:

$$w_i = \frac{1}{1 + \sigma_{p,i}^2}, \tag{26}$$

where $\sigma_{p,i}^2$ is the variance of solving probabilities across rating buckets for puzzle i. A low variance indicates that the structural estimate is more reliable.

3) Weighted residual ranking: A combined score is computed by weighting the residual with structural confidence:

$$s_i = r_i \cdot w_i. \tag{27}$$

Puzzles are ranked in descending order of s_i, and the top 10% are selected as mask targets:

$$\hat{M}_i = \begin{cases} 1, & \text{if } i \text{ is among the top 10\% of } s_i, \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

This strategy is motivated by the hypothesis that systematically biased predictions can be detected as large residuals, and that structural features provide additional reliability signals to avoid over-selecting uncertain cases.

Because the official mask results were not released, we could not directly validate the method against the competition's test set. However, the residual- and structure-guided ranking approach provides a principled framework for identifying predictions with the highest expected impact on score improvement.

This strategy can be generalized beyond the mask task, as it effectively combines error analysis and structural confidence estimation, which are key components for model uncertainty quantification in rating-prediction problems.

Our uncertainty mask ratio is equal to 1.684 which ranks seventh among nine teams that decided to participate in this additional task. And our final score with the mask submitted is approximately equal to 56756 while the final score with the perfect mask is equal to 33709 [1].

V. Experimental Results

This section presents the quantitative evaluation of the proposed four-stage pipeline, including the progressive improvements introduced at each stage and a discussion of their relative contributions to the final performance.

A. Quantitative Summary

Table I summarizes the evolution of the public leaderboard Mean Squared Error (MSE) through different stages of the pipeline. The private test score, which determined the final ranking, is also reported for the final ensemble.

Table I
Performance progression across pipeline stages.

Stage	Public LB MSE (↓)
Best Single Base Model	89,627
Simple Average of Four Base Models	85,908
Stacking (CatBoost meta, raw output)	84,511
After Structural Calibration	67,471
Final Ensemble Averaging	**66,485**
Private Test MSE (Final Ensemble)	**62,567**

Averaging across all four banded models provided modest improvement, indicating that band diversity contributes to generalization even without meta-learning. Stacking with

meta-features yielded further gains, reducing the public MSE by 6% relative to simple averaging. The largest improvement came from the structural calibration step, which lowered the public MSE by 20%. Finally, averaging the three calibrated meta-regressors slightly improved stability and achieved the best overall score.

The final submission achieved a private test MSE of 62,567, ranking 6th among all participating teams.

B. Discussion

Several observations can be drawn from these results:

- **Effectiveness of Band-Wise Base Models:** The best single base model significantly outperformed global baselines, validating the hypothesis that Elo-banded training better captures localized difficulty semantics. Lower-rated bands provided complementary perspectives, which collectively improved performance through averaging.
- **Meta-Feature Integration:** The 6% improvement from stacking demonstrates that disagreement patterns among base models and structural descriptors (e.g., success-probability variance, inflection points) are strong predictive signals. Feature importance analysis from CatBoost indicated that structural features such as mean solving probability and failure skewness ranked among the top predictors.
- **Impact of Structural Calibration:** Post-hoc calibration aligned predictions with statistically plausible difficulty levels derived from aggregated failure distributions. The segmented blending weight α_i played a crucial role, applying stronger correction to low-rated puzzles where residual biases were largest.
- **Ensemble Averaging and Robustness:** The final averaging reduced variance by combining three diverse meta-regressors with moderately correlated errors. This contributed to consistent private-test performance, narrowing the public-private gap compared to earlier stages.
- **Computational Efficiency and Interpretability:** Despite competitive accuracy, the proposed pipeline remains lightweight, interpretable, and suitable for large-scale deployment.

VI. Conclusions

This work presented a structured four-stage framework for predicting chess puzzle difficulty, integrating band-specific modeling, meta-feature regression, statistical calibration, and ensemble averaging. The major findings and contributions can be summarized as follows:

- **Localized base modeling:** Training separate regressors on Elo-banded subsets captured rating-specific solving dynamics and provided complementary perspectives compared to a single global model.
- **Meta-feature integration:** Stacking the base-model predictions with statistical and structural features exploited disagreement patterns and higher-order indicators (e.g., variance, inflection, skewness), improving cross-band generalization.
- **Post-hoc calibration:** Aligning predictions with structural difficulty estimates derived from aggregated failure distributions systematically reduced residual biases, particularly for lower-rated puzzles.
- **Variance-reduced ensembling:** Averaging three independently calibrated regressors improved stability, narrowing the public-private leaderboard gap and achieving a final private test MSE of 62,567, ranking 6th overall.

Furthermore, the framework was extended to the optional mask selection task, where a residual- and structure-guided ranking strategy was proposed to identify the 10% most problematic puzzles. Although official mask results were not released, internal analyses suggest that structural signals are effective not only for improving rating prediction but also for detecting highly uncertain cases, offering potential for future work on confidence-guided puzzle recommendation systems.

References

[1] J. Zyśko, M. Ślęzak, D. Ślęzak, and M. Świechowski, “FedCSIS 2025 knowledgepit.ai Competition: Predicting Chess Puzzle Difficulty Part 2 & A Step Toward Uncertainty Contests,” in *Proc. 20th Conf. Comput. Sci. Intell. Syst. (FedCSIS)*, vol. 43, Polish Inf. Process. Soc., 2025. doi: http://dx.doi.org/10.15439/2025F5937.

[2] S. Björkqvist, "Estimating the Puzzlingness of Chess Puzzles," 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 8370-8376, doi: 10.1109/BigData62323.2024.10825991.

[3] A. Rafaralahy, "Pairwise Learning to Rank for Chess Puzzle Difficulty Prediction," 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 8385-8389, doi: 10.1109/BigData62323.2024.10825356.

[4] D. Ruta, M. Liu and L. Cen, "Moves Based Prediction of Chess Puzzle Difficulty with Convolutional Neural Networks," 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 8390-8395, doi: 10.1109/BigData62323.2024.10825595.

[5] S. Milosz and P. Kapusta, "Predicting Chess Puzzle Difficulty with Transformers," in 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 8377-8384, doi: 10.1109/BigData62323.2024.10825919.

[6] M. Omori and P. Tadepalli, “Estimating Player Ratings and Puzzle Difficulty with CNN-LSTM Models,” arXiv preprint arXiv:2409.11506, 2024.

[7] J. Zyśko, M. Świechowski, S. Stawicki, K. Jagieła, A. Janusz and D. Ślęzak, "IEEE Big Data Cup 2024 Report: Predicting Chess Puzzle Difficulty at KnowledgePit.ai," 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 8423-8429, doi: 10.1109/BigData62323.2024.10825289.

[8] T. Woodruff, O. Filatov and M. Cognetta, "The bread emoji Team's Submission to the IEEE BigData 2024 Cup: Predicting Chess Puzzle Difficulty Challenge," 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 8415-8422, doi: 10.1109/BigData62323.2024.10826037.

[9] A. Schütt, T. Huber and E. André, "Estimating Chess Puzzle Difficulty Without Past Game Records Using a Human Problem-Solving Inspired Neural Network Architecture," 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 8396-8402, doi: 10.1109/BigData62323.2024.10826087.

[10] A. V. Dorogush, V. Ershov, and A. Gulin, “CatBoost: gradient boosting with categorical features support,” in *Proc. Workshop on ML Systems (MLSys)*, 2018.

[11] G. Ke, Q. Meng, T. Finley, et al., “LightGBM: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[12] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proc. ACM SIGKDD*, 2016, pp. 785–794.

[13] M. Liu, L. Cen and D. Ruta, "Gradient Boosting Models for Cybersecurity Threat Detection with Aggregated Time Series Features," *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*, Warsaw, Poland, 2023, pp. 1311-1315, doi: 10.15439/2023F4457.