Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 843–848
DOI: 10.15439/2025F2456
ISSN 2300-5963 ACSIS, Vol. 43

Multi-Source Feature Fusion and Neural Embedding for Predicting Chess Puzzle Difficulty

Haitao Xiao*,†, Daiyuan Yu*,†, Xuegang Wen*,†, Le Chen*,†, Kun Fu*,†
* China Mobile Information Technology Co., Ltd., China
† China Mobile Communications Group Co., Ltd., China
Email: {xiaohaitao, yudaiyuan, wenxuegang, chenle, fukun}@chinamobile.com

***Abstract*—Estimating the difficulty of chess puzzles provides a rich testbed for studying human–computer interaction and adaptive learning. Building on recent advances and the FedCSIS 2025 Challenge, we address the task of predicting chess puzzle difficulty ratings using a multi-source representation approach. Our approach integrates pre-trained neural embeddings of board states, solution move sequences, and engine-derived success probabilities. These heterogeneous features are fused via dedicated embedding and projection layers, followed by a multi-layer perceptron regressor. Post-processing calibration and model ensemble further enhance robustness and generalization. Experiments on the FedCSIS 2025 dataset demonstrate that our method effectively leverages both structural and empirical information, achieving strong predictive performance. Our approach achieved fifth place on the final official leaderboard, highlighting the effectiveness of combining neural representations with domain-specific probabilistic features for robust chess puzzle difficulty prediction.**

***Index Terms*—Human-Computer Interaction, Chess Puzzle Difficulty, Multi-Source Feature Fusion, Representation Learning, Ensemble Learning**

I. Introduction

Chess has served not only as a competitive arena, but also as a richly structured and controlled testbed for exploring the foundations of human–computer interaction. From Deep Blue's brute-force victory over Kasparov [1] to AlphaZero's self-taught superhuman play [2], each algorithmic milestone has advanced a deeper goal: understanding how machines can model, anticipate, and ultimately support human cognitive behavior. Contemporary online chess platforms now record every move, reaction time, and mistake across millions of human–machine interactions. These data enable a direct estimation of puzzle difficulty from behavioral patterns. Predicting chess puzzle difficulty thus emerges as a core task that bridges cognitive science, adaptive learning, and recommender system design.

While chess has long served as a model system for cognitive research, it is in the advent of modern online platforms that has enabled large-scale, quantitative analysis. Lichess [1] is a widely used open-source online chess platform that provides millions of user-generated chess puzzles. The difficulty of each puzzle is quantified using rating systems originally designed for human players. The most common systems include the Elo rating [3], which updates a player's rating based on game outcomes, and the Glicko [4] and Glicko-2 [5] systems, which further incorporate rating volatility and adjust more dynamically to player performance. On Lichess, each puzzle receives a Glicko-2 rating that reflects its empirical difficulty for the average user, with accuracy improving as more players attempt the puzzle.

Building on the IEEE BigData 2024 Cup [6], which demonstrated the utility of feature-rich and neural approaches for chess puzzle difficulty prediction [7–12], the FedCSIS 2025 Challenge [13] introduces both a larger dataset and new data modalities. The main objective remains to predict the difficulty rating of a chess puzzle using its initial board state and solution moves. However, in contrast to the first edition, the FedCSIS 2025 Challenge provides 22 precomputed engine-based success probabilities per puzzle, generated by Maia-2 models [14] to simulate human move likelihoods across different player ratings and types of rating. This addition eliminates the need for costly local engine simulations and facilitates more equitable benchmarking of model designs. With a training set of over 4.5 million puzzles, FedCSIS 2025 offers an enriched setting for advancing research in chess puzzle difficulty prediction.

In this study, we propose a multi-source neural representation approach for predicting chess puzzle difficulty. Our approach integrates heterogeneous information from pre-trained board state embeddings, solution move sequences, and a 22-dimensional vector of engine-estimated success probabilities. These complementary features are jointly fused and embedded through dedicated projection layers, followed by a multi-layer perceptron regressor to predict the rating. To further enhance robustness and address distributional shifts, we apply post-processing calibration and ensemble models trained under diverse settings. This design leverages both the structural and empirical dimensions of puzzle difficulty, leading to improved predictive performance and generalization.

The remainder of this paper is organized as follows: Section II analyzes the dataset and outlines the preprocessing pipeline. Section III describes the proposed approach. Section IV details the experimental setup and results. Section V discusses findings, limitations, and potential ways to improve in future work. Section VI concludes the paper.

[1] https://lichess.org

II. Data Analysis

A comprehensive understanding of the dataset structure is essential for designing robust prediction models in this challenge. Table I provides an overview of the principal features included in the competition dataset. Each chess puzzle is uniquely identified by a `PuzzleId` and is characterized by several core components: the board position specified in Forsyth–Edwards Notation (FEN), the solution sequence encoded in Portable Game Notation (PGN), and a 22-dimensional vector of engine-estimated success probabilities (`SuccessProb`), which represents projected solve rates across diverse player rating groups and game types. The `Rating` field denotes the puzzle's Glicko-2 difficulty rating [5], which serves as the primary target variable for model training.

TABLE I
Summary of Features in the Challenge Dataset

Field Name	Description	Type
PuzzleId	Unique identifier	String
FEN	Board position	String
Moves	Solution in PGN	String
SuccessProb	Success probabilities	Float
Rating	Glicko-2 puzzle rating (Target)	Integer
RatingDeviation	Uncertainty in rating	Integer
Popularity	Upvotes minus downvotes	Integer
NbPlays	Number of attempts	Integer
Themes	Puzzle motif tags	String
GameUrl	Lichess game provenance	String
OpeningTags	Opening classification	String

In addition to these core fields, the training data comprises several auxiliary metadata attributes. These include the rating uncertainty (`RatingDeviation`), popularity score (`Popularity`), total number of attempts (`NbPlays`), thematic tags (`Themes`), and optional fields such as the original game URL (`GameUrl`) and opening classification (`OpeningTags`). This rich set of features enables multi-faceted analysis and facilitates the construction of both neural and feature-based models.

TABLE II
Comparison of Training and Test Datasets

Property	Training Set	Test Set
Instances	4,557,000	2,235
Features	PuzzleId, FEN, Moves, SuccessProb, RatingDeviation, Popularity, NbPlays, Themes, GameUrl, OpeningTags	PuzzleId, FEN, Moves, SuccessProb
Target	Rating	-

Table II presents a comparative summary of the training and test datasets used in this challenge. The training set contains 4,557,000 puzzle instances, while the test set contains 2,235 instances. Notably, the test set is restricted to the core features: `PuzzleId`, `FEN`, `Moves`, and `SuccessProb` with the ground-truth `Rating` field hidden to facilitate unbiased model evaluation. Consequently, additional metadata present in the training set, such as `RatingDeviation`, `Popularity`, and `Themes`, must be excluded from the feature set during model development to ensure strict compatibility between training and inference conditions.

The structure of the dataset, characterized by both structural descriptors and empirical engine-based probabilities, enables the exploration of diverse modeling strategies. The inclusion of precomputed success probabilities is particularly noteworthy, as it reduces the computational barrier for participants and provides valuable prior information for puzzle difficulty prediction, especially for those without access to sufficient local computational resources.

Although the dataset provides a comprehensive set of structural and empirical features, ensuring the reliability of the target variable is crucial for robust model training. In particular, the rating uncertainty (`RatingDeviation`) quantifies the confidence of each puzzle's Glicko-2 difficulty estimate and varies substantially across puzzles.

Puzzles exhibiting high rating uncertainty generally correspond to unstable or unreliable difficulty estimates, often resulting from insufficient player attempts or inconsistent solution patterns. To improve data quality and label reliability, all puzzles with a `RatingDeviation` greater than 90 were excluded from the training set. This threshold, which aligns with established practice in prior studies, effectively reduces label noise by filtering out puzzles with highly variable ratings.

After this preprocessing step, the size of the training set decreased from approximately 4.56 million to 3.53 million puzzles. The resulting dataset, comprising high-quality and reliable labels, was used for model training. This preprocessing pipeline enhances the stability of the learning process and ensures that subsequent modeling are grounded in representative data.

III. Methodology

Figure 1 presents an overview of the proposed solution pipeline for chess puzzle difficulty prediction. The overall methodology consists of several key components: multi-source feature embedding, a neural regressor for rating prediction, post-processing calibration, model ensemble, and uncertainty estimation. Each component is described in detail below.

A. Feature Embedding

Our approach integrates three complementary sources of information for each chess puzzle: the board state encoded in Forsyth–Edwards Notation (FEN), the solution move sequence represented in Portable Game Notation (PGN), and a 22-dimensional vector of engine-estimated success probabilities (`SuccessProb`). The FEN string offers a detailed, lossless description of the board configuration, while the PGN sequence captures the temporal progression of solution moves. The `SuccessProb` vector summarizes the predicted probability of a successful solve across various player rating brackets and type of rating (rapid or blitz), as precomputed by the MAIA2 neural engine [14].

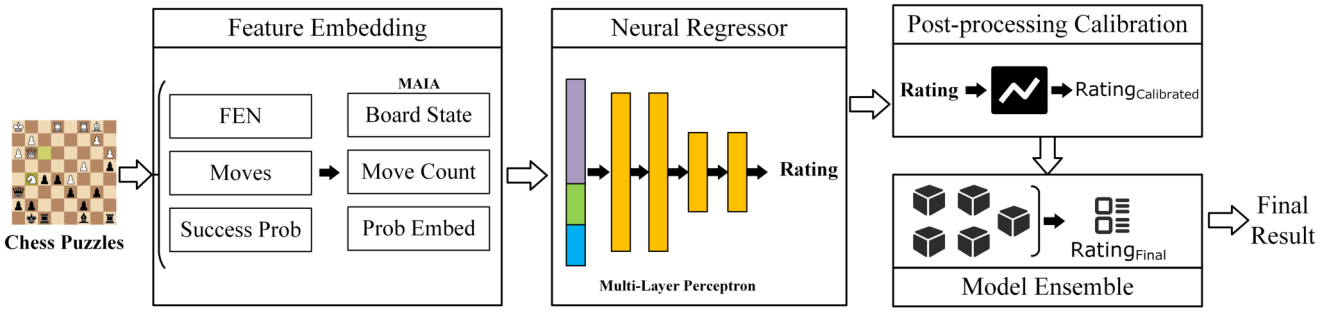

Fig. 1. Overview of proposed approach

To comprehensively capture both the structural and sequential aspects of each puzzle, we generate a sequence of consecutive board states by iteratively applying the solution moves to the initial FEN. Each intermediate position is embedded using a pre-trained Maia neural network, which is specifically trained to emulate human move preferences at defined skill levels [15]. For each board state, we extract a neural embedding from the penultimate hidden layer of the Maia model. These embeddings are mapped into a unified latent space and combined to form a comprehensive representation of the puzzle's solution trajectory.

The solution length is discretized and encoded as a learnable embedding, enabling the model to incorporate information about puzzle complexity. Simultaneously, the `SuccessProb` vector is embedded through a two-layer feedforward network with ReLU activation to produce a dense embedding representing empirical difficulty. Finally, concatenating the aggregated board embedding, the move count embedding, and the success probability embedding yields a unified embedding vector. This multi-source representation integrates structural configuration, sequential solution dynamics, and empirical difficulty priors, providing a rich input for downstream prediction.

B. Neural Regressor

The aggregated feature vector is processed by a multi-layer perceptron regressor designed to capture complex nonlinear relationships between the fused feature embeddings and puzzle difficulty. The architecture comprises a sequence of fully connected layers with progressively decreasing dimensionality, interleaved with ReLU activation functions and dropout regularization to enhance generalization and mitigate overfitting. This deep regression network enables the model to learn intricate mappings from multi-source representations to difficulty ratings. The final output is a single scalar representing the predicted Glicko-2 rating for the given puzzle.

C. Post-processing Calibration

To mitigate the distributional shift between the training set and the competition test data, we employ a simulation-inspired nonlinear rescaling technique [7] to adjust the raw predictions of the neural regressor. This post-processing step compresses prediction values at the distributional extremes while preserving ratings near the empirical mean, thereby reducing the impact of outliers. The calibration function is defined as follows:

$$\hat{r} = \mu - \frac{1+\text{sign}(r-\mu)}{2} \cdot H \cdot \max\left(1, \left|\frac{r-\mu}{D}\right|^4\right) + \frac{1-\text{sign}(r-\mu)}{2} \cdot L \cdot \min\left(1, \left|\frac{r-\mu}{D}\right|^4\right) \quad (1)$$

where $\mu = 1900$ denotes the empirical mean rating, $H = 200$ and $L = 250$ control the scaling magnitude for over- and underestimations respectively, and $D = 1000$ determines the scale at which the rescaling effect saturates. These hyperparameters are selected based on prior studies [7] and further validated through our empirical experience in this challenge.

This calibration strategy aims to improve the alignment between predicted and true rating distributions. By explicitly correcting for known biases in the rating aggregation process, the rescaling aims to make the predicted difficulty ratings more robust.

D. Model Ensemble

To further enhance model robustness and predictive performance, we employ an output-level ensemble strategy. Specifically, multiple base models are trained independently using board embeddings generated from different Maia engine variants, each emulating human play at a distinct ELO rating. The final prediction is computed by averaging the outputs from all base models:

$$\text{Rating}_{\text{final}} = \frac{1}{N}\sum_{i=1}^{N} \text{Model}_i(\text{Input}), \quad (2)$$

where N is the total number of ensembled models. This approach leverages complementary perspectives of models trained on different skill levels and effectively reduces the variance of individual predictions. Ensembling not only mitigates overfitting and model-specific biases but also enhances the model's ability to generalize across puzzles of varying complexity. This strategy is particularly effective in domains where task difficulty spans a wide spectrum and individual models may excel in different sub-regions of the input space.

E. Uncertainty Estimation

To further enhance model reliability and provide interpretable confidence assessments for each prediction, we propose an ensemble-based uncertainty estimation strategy. Specifically, for each test instance, we aggregate the prediction outputs from all base models within the ensemble and compute the standard deviation of these predictions as a proxy for epistemic uncertainty, reflecting the degree of disagreement among ensemble members.

Formally, let $y_i^{(j)}$ denote the prediction for the i-th test sample by the j-th base model in the ensemble, where $j = 1, \ldots, M$. The uncertainty score for the i-th sample is quantified as the standard deviation across the ensemble outputs:

$$\sigma_i = \sqrt{\frac{1}{M} \sum_{j=1}^{M} \left(y_i^{(j)} - \bar{y}_i \right)^2}, \tag{3}$$

where $\bar{y}_i$ represents the mean prediction for the i-th sample. A higher standard deviation σ_i indicates greater predictive uncertainty, as it signals increased model disagreement regarding the sample's difficulty.

In accordance with the challenge requirements, we rank all test samples by their uncertainty scores and flag the top K instances (where K corresponds to 10% of the test set, i.e., $K = 223$ for 2235 samples) using a binary mask. By leveraging the diversity inherent in the ensemble, this strategy systematically identifies predictions with elevated risk of error.

IV. Experiment and Result

A. Experiment Setup

1) Environment: The experimental environment is based on Ubuntu 22.04, equipped with an NVIDIA RTX 3090 GPU (24 GB VRAM). The CPU is an Intel Xeon Gold 6330 operating at 2.00 GHz, complemented by 90 GB of system memory.

2) Toolkit: The implementation is based on Python 3.12, with PyTorch 2.5.1 used for neural network construction and training. CUDA 12.4 is utilized to accelerate GPU computations and enhance overall computational efficiency.

3) Evaluation Metric: The metric for performance evaluation is the Mean Squared Error (MSE), consistent with the official evaluation criterion of the FedCSIS 2025 Challenge.

B. Experiment Result

We conducted a comprehensive set of experiments to evaluate the effectiveness of our approach for chess puzzle difficulty prediction. Table III summarizes model performance on the public leaderboard, measured in terms of MSE. All results correspond to the official public test set provided by the competition platform [16].

As a baseline, we implemented a LightGBM regressor utilizing handcrafted chess features extracted from the board state, move sequence, and basic positional statistics. Feature extraction was performed using the `python-chess` library, which provides utilities for parsing FEN and PGN representations and computing relevant game attributes. This model achieved an MSE of 104703.66, serving as the baseline for subsequent neural network-based approaches.

To investigate the impact of learned representations, we incorporated pre-trained neural board embeddings derived from various Maia engine variants. Employing MAIA-1300 embeddings without the inclusion of organizer-provided success probability vectors resulted in an MSE of 91689.69. Incorporating the success probability features further reduced MSEs for the MAIA-1300, MAIA-1500, and MAIA-1700 models to 82582.37, 85625.50, and 81570.99, respectively, thereby underscoring the complementary value of empirical priors.

To address distributional shift and enhance calibration, we applied a post-processing rescaling procedure to the model outputs, as described in Section III-C. This adjustment led to further improvement, with MSEs decreasing to 76986.22 for MAIA-1300, 78155.64 for MAIA-1500, and 79688.41 for MAIA-1700.

Subsequently, ensemble averaging across the MAIA-1300, MAIA-1500, and MAIA-1700 models yielded an MSE of 75915.97. Our final approach, which ensembles board embeddings from five Maia variants (MAIA-1100, MAIA-1300, MAIA-1500, MAIA-1700, and MAIA-1900) and combines calibrated outputs with success probability features, achieved the best observed performance, attaining an MSE of 67071.66 on the public leaderboard.

TABLE III
Performance Comparison of Different Methods on Public Leaderboard

Method	Public MSE
Handcrafted Features + LightGBM (Baseline)	104703.66
MAIA-1300 (w/o SuccessProb)	91689.69
MAIA-1300	82582.37
MAIA-1500	85625.50
MAIA-1700	81570.99
MAIA-1300 + Post-processing	76986.22
MAIA-1500 + Post-processing	78155.64
MAIA-1700 + Post-processing	79688.41
MAIA-1300/1500/1700 Ensemble	75915.97
MAIA-1100/1300/1500/1700/1900 Ensemble	**67071.66**

The results clearly demonstrate that leveraging pre-trained neural embeddings, incorporating empirical success probabilities, and ensembling the outputs of individually calibrated models together yield substantial improvements over traditional handcrafted feature-based models. The progressive reduction in MSE observed through our stepwise model enhancements highlights the additive value of each component in our solution. Ultimately, our final ensemble approach achieved a marked performance gain relative to classical models. These findings confirm the effectiveness of multi-source feature fusion and neural representation learning for chess puzzle difficulty prediction.

C. Uncertainty Estimation Result

To quantitatively evaluate the effectiveness of our uncertainty estimation strategy, we participated in the additional uncertainty mask task organized as part of the challenge. In this task, each team was required to submit a binary mask identifying the most uncertain 10% of the test puzzles. Let P denote the perfect score, obtained by replacing the predictions for the 10% most erroneous test cases with their ground-truth values, and let N denote the score achieved using the submitted mask. The evaluation metric is defined as:

$$\rho = \frac{N}{P} \tag{4}$$

Our submitted mask achieved a ratio of $\rho = 1.589$, ranking **3rd** among participating teams. On the official leaderboard, this corresponded to a score of approximately 55234, compared with the theoretical lower bound of 34766 under a perfect mask. The results show that our proposed uncertainty estimation strategy provides a reliable means of identifying error-prone cases, with potential for further refinement.

V. Discussion

Our experimental results reveal several insights and limitations that warrant further investigation. While handcrafted features extracted from the board state and move sequences provide a valuable baseline for chess puzzle difficulty prediction, they are inherently limited in their ability to capture the full complexity of positional and sequential information. Models based solely on such features consistently underperform relative to deep neural architectures, likely due to their inability to represent the nuanced dynamics encoded in FEN strings and move sequences, both of which are essential for accurate difficulty modeling.

Nevertheless, prior work and our own attempts indicate that carefully engineered handcrafted features, if effectively integrated into neural network architectures, may offer complementary benefits. However, our straightforward approach to merging these features with neural embeddings did not yield performance improvements, suggesting that more advanced integration strategies, such as attention-based models, may be necessary to unlock the full potential of hybrid feature sets.

Due to computational constraints and time limitations, our current study utilized only five Maia pre-trained model variants, corresponding to targeted ELO levels of 1100, 1300, 1500, 1700, and 1900. Each Maia variant is designed to emulate the play style of human users at its specific rating level, thereby enabling the model to capture a spectrum of human skill profiles. Nonetheless, this approach may not fully represent the diversity of the solver population. Future research should consider integrating all available Maia models, spanning ELO 1100 to 1900, to achieve finer granularity in modeling player abilities. Such comprehensive ensembles could provide a more nuanced simulation of the human skill continuum and further improve prediction robustness.

In summary, our findings underscore the importance of both high-capacity neural architectures and the principled fusion of domain-specific features for robust chess puzzle difficulty prediction. Advancing this field will require continued research into sophisticated feature integration strategies and more comprehensive utilization of human-like engine ensembles, ultimately paving the way for models with enhanced generalization and interpretability.

VI. Conclusion

In this work, we investigated chess puzzle difficulty prediction using a multi-source feature fusion approach based on neural embedding techniques, within the context of the FedCSIS 2025 Challenge. Our approach integrates pre-trained Maia board embeddings, solution move sequences, and success probabilities generated by the MAIA2 engine. These heterogeneous information sources are combined within a unified neural regression model. Results on the official public leaderboard demonstrate that incorporating empirical priors derived from chess engines, neural network-based representations, model ensemble, and post-processing calibration leads to improved performance over traditional feature-based baselines.

In addition, we acknowledge the overarching goal of the knowledgepit competition series [17–20], which is to evaluate the skills of data scientists through carefully designed tasks [16]. Beyond its evaluative role, the knowledgepit provides an excellent platform for testing novel methodologies, comparing approaches under common benchmarks, and advancing the broader data science community.

Acknowledgment

We thank the organizers of the FedCSIS 2025 Challenge for their efforts in hosting this competition and maintaining a fair and engaging environment.

References

[1] M. Campbell, A. Hoane, and F. hsiung Hsu, "Deep blue," *Artificial Intelligence*, vol. 134, no. 1, pp. 57–83, 2002. doi: https://doi.org/10.1016/S0004-3702(01)00129-1. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370201001291

[2] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018. doi: 10.1126/science.aar6404. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.aar6404

[3] A. E. Elo, *The Rating of Chessplayers, Past and Present*. New York: Arco Pub., 1978. ISBN 0668047216 9780668047210

[4] M. E. Glickman, "The glicko system," *Boston University*, vol. 16, no. 8, p. 9, 1995.

[5] ——, "Example of the glicko-2 system," *Boston University*, vol. 28, p. 2012, 2012.

[6] 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," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024. doi: 10.1109/BigData62323.2024.10825289 pp. 8423–8429.

[7] T. Woodruff, O. Filatov, and M. Cognetta, "The bread emoji team's submission to the ieee bigdata 2024 cup: Predicting chess puzzle difficulty challenge," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024. doi: 10.1109/BigData62323.2024.10826037 pp. 8415–8422.

[8] A. Schütt, T. Huber, and E. André, "Estimating chess puzzle difficulty without past game records using a human problem-solving inspired neural network architecture," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024. doi: 10.1109/BigData62323.2024.10826087 pp. 8396–8402.

[9] S. Björkqvist, "Estimating the puzzlingness of chess puzzles," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024. doi: 10.1109/BigData62323.2024.10825991 pp. 8370–8376.

[10] A. Rafaralahy, "Pairwise learning to rank for chess puzzle difficulty prediction," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024. doi: 10.1109/BigData62323.2024.10825356 pp. 8385–8389.

[11] D. Ruta, M. Liu, and L. Cen, "Moves based prediction of chess puzzle difficulty with convolutional neural networks," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024. doi: 10.1109/BigData62323.2024.10825595 pp. 8390–8395.

[12] S. Miłosz and P. Kapusta, "Predicting chess puzzle difficulty with transformers," in *2024 IEEE International Conference on Big Data (BigData)*. IEEE, 2024. doi: 10.1109/BigData62323.2024.10825919 pp. 8377–8384.

[13] 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 *Proceedings of the 20th Conference on Computer Science and Intelligence Systems*, ser. Annals of Computer Science and Information Systems, M. Bolanowski, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 43. Polish Information Processing Society, 2025. doi: 10.15439/2025F5937. [Online]. Available: http://dx.doi.org/10.15439/2025F5937

[14] Z. Tang, D. Jiao, R. McIlroy-Young, J. Kleinberg, S. Sen, and A. Anderson, "Maia-2: A unified model for human-ai alignment in chess," *Advances in Neural Information Processing Systems*, vol. 37, pp. 20 919–20 944, 2024. doi: 10.48550/arXiv.2409.20553

[15] R. McIlroy-Young, S. Sen, J. Kleinberg, and A. Anderson, "Aligning superhuman ai with human behavior: Chess as a model system," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020. doi: 10.1145/3394486.3403219 pp. 1677–1687.

[16] D. Ślęzak, A. Janusz, M. Świechowski, A. Chądzyńska-Krasowska, and J. Kamiński, "Do data scientists dream about their skills' assessment? – transforming a competition platform into an assessment platform," in *2024 IEEE International Conference on Big Data (BigData)*, 2024. doi: 10.1109/BigData62323.2024.10825378 pp. 8403–8414.

[17] A. Janusz, M. Przyborowski, P. Biczyk, and D. Ślęzak, "Network device workload prediction: A data mining challenge at knowledge pit," in *2020 15th Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2020. doi: http://dx.doi.org/10.15439/2020F159 pp. 77–80.

[18] A. Janusz, A. Jamiołkowski, and M. Okulewicz, "Predicting the costs of forwarding contracts: Analysis of data mining competition results," in *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2022. doi: http://dx.doi.org/10.15439/2022F303 pp. 399–402.

[19] M. Czerwiński, M. Michalak, P. Biczyk, B. Adamczyk, D. Iwanicki, I. Kostorz, M. Brzęczek, A. Janusz, M. Hermansa, Ł. Wawrowski *et al.*, "Cybersecurity threat detection in the behavior of iot devices: analysis of data mining competition results," in *2023 18th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2023. doi: http://dx.doi.org/10.15439/2023F3089 pp. 1289–1293.

[20] A. M. Rakićević, P. D. Milošević, I. T. Dragović, A. M. Poledica, M. M. Zukanović, A. Janusz, and D. Ślęzak, "Predicting stock trends using common financial indicators: A summary of fedcsis 2024 data science challenge held on knowledgepit. ai platform," in *2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS)*. IEEE, 2024. doi: http://dx.doi.org/10.15439/2024F7912 pp. 731–737.