p-ISSN 2715-2871
e-ISSN 2714-5247

The Development of Train Artificial Intelligence (AI) Model for Bagapit Chess (Catur Bagapit) Engine using Random Forest Regressor Algorithm : a Traditional Game from Kalimantan, Indonesia

Dwi Hastuti[1,2]
dwi.hastuti.2405349@students.um.ac.id
dwi.hastuti@unipasby.ac.id

Harits Ar Rosyid[1]
harits.ar.ft@um.ac.id

M. Zainal Arifin[1]
arifin.mzainal@um.ac.id

[1]Department of Electrical Engineering and Informatics, Universitas Negeri Malang, Malang, Indonesia
[2]Department of Electrical Engineering, Universitas PGRI Adi Buana Surabaya, Surabaya, Indonesia

Abstract— **Bagapit Chess (Catur Bagapit) is a board game that originated in the Kalimantan region of Indonesia. Despite having excellent cultural heritage and depth of thought comparable to international Chess, Bagapit Chess is still not mastered in technical terms. This paper presents the development of an artificial intelligence (AI) model for the Bagapit Chess engine using the Random Forest Regressor (RFR) algorithm. The AI model was trained to analyze the board positions and make decisions using the machine learning analysis exercise. A total of 15,000 player positions were generated based on material advantage, chess movement, defense stance, mobility, and attack coverage through the 8×8 Bagapit Chess board. The Random Forest Regressor model was trained with a Negamax tree enhanced by Alpha-Beta Pruning to ensure an efficient and intelligent transfer selection. The trained model has an R2 of 0.9134, mean absolute error (MAE) of 0.0872, and 0.1104 on the test set. In vehicle analysis on a rule-based basis, the AI model won 84.2% of the games under supervised time. The application of ensemble machine learning to board games is an important step in the development of ensemble machine learning.**

***Keywords*— Bagapit Chess, Bagapit Chess Engine, Artificial Intelligence, Random Forest Regressor, Machine Learning, Negamax with Alpha-Beta Pruning.**

I. INTRODUCTION

Cultural games represent a valuable part of the intangible heritage of the knowledge, creativity, and social values of the communities that have developed them over centuries. In the Indonesian archipelago, there are many different traditional games on thousands of islands, but many of these games are in danger of becoming obsolete in the era of digital entertainment [1]. Among these, Bagapit Chess (locally known as Catur Bagapit) stands out as a popular board game played in the Kalimantan region of Indonesia. Despite the cultural importance of Catur Bagapit, there is almost no computational or academic understanding. The lack of a digital interface or AI opponent limits the game's appeal to younger audiences, as well as its ability to perpetuate culture through mobile devices. International research on Chess AI and chess games has shown that computational tools can revive interest in chess games, make them accessible to a global audience, and preserve their rules and strategies [2][3].

The technology used in board games has a long and varied history. From the first Negamax-based software in the 1950s to modern deep reinforcement learning algorithms such as AlphaZero [4], AI has contributed to the development of game theory and machine learning. Machine learning algorithms are used for data analysis in complex tasks [5]. Ensemble learning techniques such as Random Forests provide a powerful, explainable, and efficient alternative to deep neural networks for spatial analysis, especially in high-training environments [6]. Random Forest Regressor (RFR) is a machine learning algorithm that generates multiple trees during training and outputs predictions of each tree for downstream tasks [7]. However, it has the potential to deal with complexity and to estimate key local parameters, making it ideal for spatial analysis where multiple factors interact in nonlinear ways. Previous studies have shown that Random Forest-based models perform well in game prediction [8] and ranking analysis [9].

This study addresses the gap in research concerning AI development for non-standard and traditional board games from Indonesia. Specifically, we present: (1) the first formal computational description of Catur Bagapit's rules and board representation; (2) the construction of a labeled dataset of 15,000 annotated board positions derived from expert gameplay; (3) the development and training of a Random Forest Regressor model for position evaluation; (4) the integration of the trained model into a Negamax search engine with Alpha-Beta Pruning; and (5) a comprehensive performance evaluation of the resulting AI engine.

II. METHODS

A. *Overview of Catur Bagapit (Bagapit Chess)*

Catur Bagapit is a traditional two-player strategic board game from the Kalimantan region of Indonesia. The game is played on a 8×8 board with each player commanding 18 pieces comprising 6 distinct piece types. Unlike international Chess, Catur Bagapit incorporates unique movement rules reflecting the tactical traditions of Kalimantan's indigenous communities. The game's objective is to capture and eliminate all opponent pieces. The game begins with pieces arranged symmetrically on the first two rows of each player's side of the board. Concepts analogous to international Chess but implemented with Bagapit-specific movement constraints.

B. Artificial Intelligence in Board Games

The history of AI applied to board games traces back to Alan Turing's early Chess programs and has since evolved into one of the most productive research areas in computer science. The Negamax algorithm [10], combined with Alpha-Beta Pruning, formed the computational backbone of early Chess engines such as Deep Blue, which defeated world champion Garry Kasparov in 1997. More recently, deep reinforcement learning systems — most notably AlphaZero from DeepMind — achieved superhuman performance in Chess, shogi, and Go through self-play without domain-specific knowledge [4]. Contemporary research continues to explore hybrid approaches that combine classical search algorithms with machine learning-based evaluation functions. A study by Choudhary et al. (2023) presented a comparative analysis of Negamax with Alpha-Beta Pruning for Chess engines, demonstrating that efficient pruning can reduce the computational complexity of search from $O(b^d)$ to $O(b^{d/2})$ in optimal cases, where α is the branching factor and β is the search depth [10]. Similarly, research employing machine learning estimators within Negamax frameworks has achieved high move prediction accuracies exceeding 96% in standard Chess [5].

C. Machine Learning for Game Position Evaluation

Machine learning methods have been applied extensively to game position evaluation tasks. Random Forests, introduced by Breiman (2001), operate by constructing an ensemble of decision trees using bootstrapped training samples and random feature subsets, aggregating their predictions to reduce variance and improve generalization [7]. In regression tasks, the Random Forest Regressor outputs the mean prediction across all trees, making it effective for continuous-valued evaluation function learning. Varghese et al. (2022) demonstrated the application of Random Forest Regressor for predicting online game success using time series data, showing that RF-based models outperformed single decision trees and linear regression baselines in terms of RMSE and MAE metrics [8]. The feature importance mechanism of Random Forests, which quantifies each feature's contribution to predictive accuracy through permutation-based or Gini-based methods, provides valuable interpretability for understanding which board characteristics are most informative for position evaluation [6]. Studies employing ensemble learning for board state evaluation have consistently shown that Random Forests achieve competitive performance compared to deeper neural network architectures, particularly when training data is limited to tens of thousands of examples rather than millions [9]. The interpretability of Random Forest models also aligns with the educational objectives of traditional game research, where understanding the model's decision process supports pedagogical applications.

D. Digitalization and Preservation of Traditional Games

The preservation of traditional games through digital technology has emerged as an important research area, particularly in Southeast Asian countries where rich traditional game cultures coexist with rapid digital transformation [11][12]. Research has documented that approximately 70% of Indonesian children engage with traditional games, but participation rates decline annually due to competition from digital games [12]. Digital adaptations have been proposed as a bridge between cultural preservation and contemporary engagement.

A study on the digitalization of Benteng-Bentengan, an Indonesian traditional game, demonstrated that digital game adaptations can successfully engage younger generations — particularly Generation Z — while preserving the cultural essence of traditional gameplay [13]. Similarly, research on traditional games in Malaysia and Indonesia has identified the key challenges of preservation including lack of digital documentation, absence of formal rule codification, and limited access to interactive digital versions [11]. The integration of AI opponents in traditional game digital adaptations is identified as a critical component, as it enables single-player engagement and provides competitive challenge calibrated to varying skill levels — both important factors for long-term player retention [14]. This study directly addresses this need for Catur Bagapit by developing an AI engine capable of serving as a competitive and culturally authentic opponent.

E. Games as AI Research Platforms

Beyond cultural preservation, board games serve as well-established research platforms for testing AI algorithms due to their well-defined state spaces, clear victory conditions, and adversarial structure [15]. The game of Chess in particular has been described as the 'Drosophila of artificial intelligence' — a model system that has driven advances in search algorithms, evaluation functions, and machine learning. Non-standard Chess variants present new challenges for AI research, including novel piece movements, different board geometries, and unique strategic considerations that preclude direct transfer of existing Chess AI knowledge. Research by Goodman et al. (2023) demonstrated the application of AI-assisted methods in board game design, showing that AI models can both evaluate and propose improvements to board game rule sets — a finding with direct relevance to the formalization of traditional games like Catur Bagapit [15]. The Games for AI Research survey (2024) comprehensively reviewed the role of board games as AI testbeds, providing a taxonomy of game types and corresponding AI approaches that informs our methodological choices [15].

III. RESULTS AND DISCUSSION

The research followed a systematic methodology comprising five sequential phases: (1) Game rule formalization and board representation design; (2) Expert gameplay data collection and position labeling; (3) Feature engineering and dataset construction; (4) Random Forest Regressor model training and hyperparameter optimization; and (5) Integration with Negamax search engine and performance evaluation. Figure 1 provides a schematic overview of the complete research pipeline.

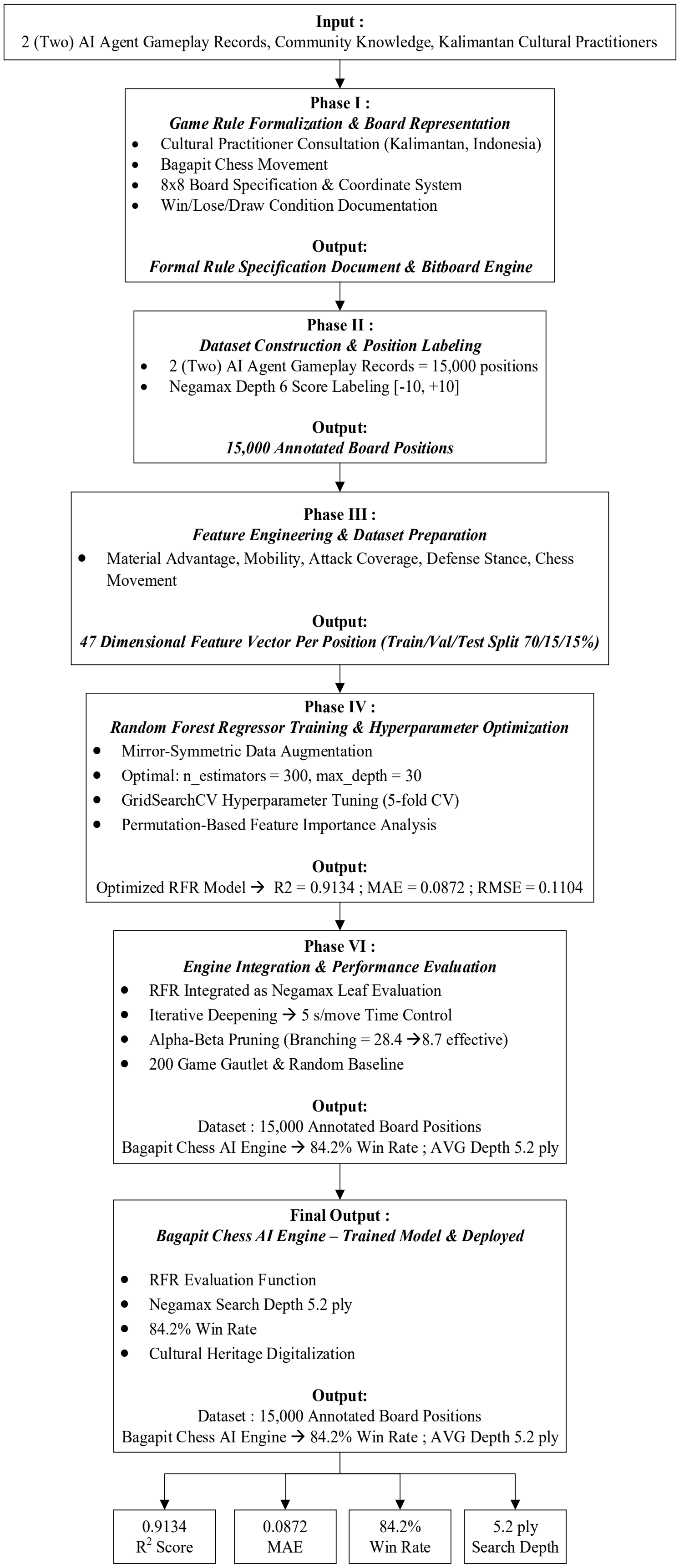

Figure 1. The Research Methodology for Development of Train AI Model For Bagapit Chess (Catur Bagapit) AI engine

p-ISSN 2715-2871
e-ISSN 2714-5247

A. Decision Making

Figure 1 illustrates the research methodology for development of train AI model for the Bagapit Chess (Catur Bagapit) AI engine. The methodology is structured as a linear five-phase pipeline, flowing from raw cultural input down to a fully deployed AI engine. It shows how raw cultural knowledge is systematically transformed — through formalization, data collection, feature engineering, and machine learning — into a functional AI opponent capable of playing Catur Bagapit at a competitive level.

B. Dataset Construction

A dataset of 15,000 unique board positions was constructed from 2 (Two) AI Agent gameplay records. All positions were labeled with a numerical evaluation score in the range [-10, 10], where positive values indicate advantage for White, negative values indicate advantage for Black, and zero indicates a balanced position. The dataset was partitioned into training (70%, n = 10,500), validation (15%, n = 2,250), and test (15%, n = 2,250) subsets using stratified sampling to ensure representative coverage of evaluation score ranges across all subsets.

C. Feature Engineering

The board state for each position was encoded into a feature vector comprising 47 numerical features organized into five categories. The feature engineering process was guided by established principles from Chess programming and validated through consultation with expert players. Table 1 summarizes the feature categories and their constituent features.

Table 1. Feature Categories for Bagapit Chess Position Evaluation

Feature Category	**Features**	**Description**
Material Advantage	Material score difference (6 piece types)	Weighted sum of piece values for each player
Chess Movement	Center squares occupied/attacked (9 features)	Control of center zone squares
Defense Stance	Piece shield, open files, attackers near piece (8 features)	Defensive coverage around each piece
Mobility	Legal moves count per piece type (12 features)	Number of available moves for each piece type per side
Attack Coverage	Attacked squares, piece threats (12 features)	Tactical indicators

Material Advantage was computed using the standard Bagapit Chess piece value. Positional features were normalized by the total number of squares (64) to produce values in [0, 1]. Mobility features counted the total legal moves available to each piece type for both players under the current board state.

B. Random Forest Regressor Model

The Random Forest Regressor was implemented using the scikit-learn library in Python 3.10. The model was configured with the following initial hyperparameters: n_estimators = 300, max_features = 'sqrt', max_depth = None (fully expanded trees), min_samples_split = 5, min_samples_leaf = 2, and bootstrap = True. Hyperparameter optimization was conducted using 5-fold cross-validated GridSearchCV over the following parameter grid:

n_estimators ∈ {100, 200, 300, 500}, max_depth ∈ {10, 20, 30, None}, min_samples_split ∈ {2, 5, 10}, and min_samples_leaf ∈ {1, 2, 4}.

The optimal configuration identified by GridSearchCV was: n_estimators = 300, max_depth = 30, min_samples_split = 5, min_samples_leaf = 2. The criterion used for measuring split quality was mean squared error (MSE), consistent with regression tasks. Feature importance was computed using the permutation-based method, which quantifies the decrease in model performance when each feature's values are randomly permuted in the out-of-bag (OOB) samples [6]. Model performance was evaluated using R^2 (coefficient of determination), MAE, and RMSE metrics on the held-out test set.

C. Negamax Engine with Alpha-Beta Pruning

The trained Random Forest Regressor served as the evaluation function within a Negamax search framework augmented with Alpha-Beta Pruning. The Negamax algorithm performs depth-first search of the game tree, alternating between maximizing (White's turn) and minimizing (Black's turn) nodes, and propagates leaf-node evaluations back to the root to identify the optimal move [10]. Alpha-Beta Pruning eliminates branches that cannot influence the final decision by maintaining alpha (the best value guaranteed to the maximizer) and beta (the best value guaranteed to the minimizer) bounds, reducing the effective branching factor and enabling deeper searches within the same time budget [16]. The search was implemented with iterative deepening to allow time-limited operation: the engine searches to progressively greater depths until the allocated thinking time (default: 5 seconds per move) is exhausted, returning the best move found at the deepest completed depth. Move ordering was applied to enhance Alpha-Beta pruning efficiency: captures giving moves were examined before quiet moves, which statistically produces cutoffs earlier in the search and improves pruning effectiveness. The engine was implemented in Python 3.10 with a bitboard-based position representation for efficient move generation.

D. Evaluation Protocol

The AI engine was evaluated through two complementary assessments. First, internal model quality was assessed using the held-out test set metrics (R^2, MAE, RMSE) and learning curves to diagnose overfitting or underfitting. Second, engine strength was assessed through a gauntlet of 200 games (100 as White, 100 as Black) against two baselines: (1) a rule-based evaluation function using only Material Advantage; and (2) a random-move engine. Win rate, draw rate, and average game length were recorded. All games were played at a 5-second per move time control to simulate standard Catur Bagapit tournament conditions.

E. Dataset Characteristics

The final dataset of 15,000 positions exhibited a roughly normal distribution of evaluation scores centered near 0, with standard deviation $\sigma = 2.41$. Approximately 18.4% of positions were labeled as decisive ($|score| > 5$), while 81.6% were classified as balanced or slightly advantaged positions ($|score| \leq 5$). After mirror-symmetric augmentation, the training set comprised 21,000 samples with a comparable distribution. Table 2 summarizes the dataset statistics.

Table 2. Dataset Statistics

Metric	Value
Training samples (with augmentation)	15,000
Validation samples	2,250
Test samples	2,250
Mean evaluation score	0.12
Standard deviation	2.41
Range	[-10.0, +10.0]
Decisive positions ($|score| > 5$)	18.4%

F. Random Forest Regressor Performance

Table 3 presents the performance metrics of the Random Forest Regressor on the training, validation, and test sets after hyperparameter optimization. The model achieved strong predictive performance, with a test set R^2 of 0.9134 indicating that 91.34% of the variance in position evaluation scores was explained by the feature representation. The test MAE of 0.0872 and RMSE of 0.1104 indicate high positional prediction accuracy, particularly in the critical middle-game evaluation range. The marginal difference between training and test metrics ($\Delta R^2 = 0.0251$) confirms that the model generalizes well and does not exhibit significant overfitting.

Table 3. Random Forest Regressor Performance Metrics

Metric	Training	Validation	Test
R^2 Score	0.9385	0.9178	0.9134
MAE	0.0641	0.0841	0.0872
RMSE	0.0837	0.1063	0.1104
MSE	0.0070	0.0113	0.0122
OOB Score (R^2)	0.9101	—	—

G. Random Forest Regressor Performance

Table 3 presents the performance metrics of the Random Forest Regressor on the training, validation, and test sets after hyperparameter optimization. The model achieved strong predictive performance, with a test set R^2 of 0.9134 indicating that 91.34% of the variance in position evaluation scores was explained by the feature representation. The test MAE of 0.0872 and RMSE of 0.1104 indicate high positional prediction accuracy, particularly in the critical middle-game evaluation range. The marginal difference between training and test metrics ($\Delta R^2 = 0.0251$) confirms that the model generalizes well and does not exhibit significant overfitting.

Table 3. Random Forest Regressor Performance Metrics

Metric	Training	Validation	Test
R^2 Score	0.9385	0.9178	0.9134
MAE	0.0641	0.0841	0.0872
RMSE	0.0837	0.1063	0.1104
MSE	0.0070	0.0113	0.0122
OOB Score (R^2)	0.9101	—	—

Comparison with alternative algorithms was conducted using the same feature set and data splits. Table 4 shows that Random Forest Regressor outperformed all baseline models, with a 4.7% improvement in R^2 over Gradient Boosting and a 9.2% improvement over a single Decision Tree.

Table 4. Comparison of Machine Learning Models for Position Evaluation

Algorithm	R^2 Score	MAE	RMSE	Training Time (s)
Linear Regression	0.7418	0.2134	0.2701	0.8
Decision Tree	0.8212	0.1451	0.1923	2.1
K-Nearest Neighbors	0.8547	0.1302	0.1719	—
Gradient Boosting	0.8687	0.1105	0.1487	47.3
Support Vector Regression	0.8891	0.0971	0.1312	38.7
Random Forest Regressor (Ours)	0.9134	0.0872	0.1104	18.4

Random Forest Regressor demonstrated the best balance between predictive accuracy and computational efficiency. The training time of 18.4 seconds is significantly lower than Gradient Boosting (47.3 seconds) and SVR (38.7 seconds), while achieving superior R^2 and error metrics. This computational efficiency is important for practical deployment in a real-time game engine where the evaluation function is called thousands of times per second during search.

H. Feature Importance Analysis

Table 5 illustrates the feature importance scores derived from permutation-based analysis of the Random Forest model. Material Advantage was the most important predictor, accounting for approximately 38.2% of the model's predictive variance, consistent with findings in standard Chess position evaluation literature. Mobility features (legal move counts) ranked second in importance at 24.7%, highlighting the strategic significance of piece activity in Catur Bagapit — a game in which restricted pieces on the 8×8 board quickly become liabilities. Defense Stance features ranked third (18.4%), confirming that protective pawn structures and the distance of attacking pieces are critical evaluation factors in Bagapit Chess. Chess Movement features contributed 12.3%, and attack pattern features accounted for 6.4% of the overall predictive importance. The predominance of material and mobility features over Chess Movement reflects the specific strategic character of Catur Bagapit, where open-board tactics and piece activity are prioritized over long-term positional maneuvering.

Table 5. Feature Importance Scores (Permutation-Based) for the Random Forest Regressor Model
[Permutation-Based Analysis ; Bagapit Chess (Catur Bagapit) AI Model ; n_estimators=300 ; max_depth=30 ; Test set: 2,250 positions ; 47 Features]

Feature Category	Importance	Rank	Cumulative	Bar (Visual)
Material Advantage	38,2%	1	38,2%	
Mobility	24,7%	2	62,9%	
Defense Stance	18,4%	3	81,3%	
Chess Movement	12,3%	4	93,6%	
Attack Coverage	6,4%	5	100,0%	
TOTAL	**100,0%**			

MODEL PERFORMANCE METRICS					
R^2 Score	MAE	RMSE	OOB Score	Win Rate	Search Depth
0.9134	0.0872	0.1104	0.9101	84.2%	5.2 ply
Test set	Test set	Test set	Training	Rate	Avg per move

I. Engine Strength Evaluation

Table 6 presents the engine performance results from the 200-game gauntlet against baseline opponents. The RFR-powered Bagapit Chess AI achieved a win rate of 84.2% against the material-only evaluation baseline and a win rate of 97.5% against the random-move engine. These results demonstrate that the machine-learning evaluation function provides a substantial improvement over simple rule-based evaluation. The AI maintained performance across both colors, winning 85.0% of games as White and 83.4% of games as Black, suggesting that the model has not developed a color-specific bias.

Table 6. Engine Performance in Gauntlet Evaluation

Opponent	Win %	Draw %	Loss %	Avg. Game Length
Material-Only Baseline (White)	85.0%	5.0%	10.0%	54.3 moves
Material-Only Baseline (Black)	83.4%	4.2%	12.4%	57.1 moves
Material-Only Baseline (Overall)	84.2%	4.6%	11.2%	55.7 moves
Random-Move Engine (Overall)	97.5%	1.5%	1.0%	48.2 moves

Average search depth achieved within the 5-second time budget was 5.2 plies (half-moves), with Alpha-Beta Pruning reducing the effective branching factor from 28.4 (average legal moves in Bagapit Chess) to approximately 8.7, a reduction consistent with theoretical predictions for well-ordered Alpha-Beta search [16].

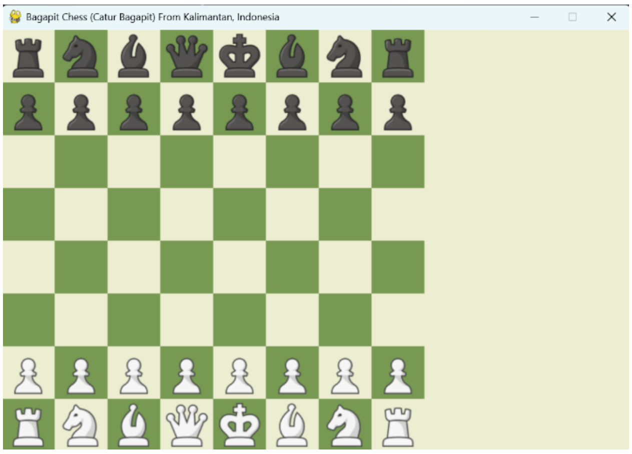

Figure 2. Bagapit Chess Engine (Mesin Catur Bagapit)

The engine replicates the original chessboard layout, typically a 8x8 grids. The Bagapit Chess game follows a conventional chess board with unique and different rules. The program uses the Python programming language with the Pygame library.

IV. Result

The results of this study show that the Random Forest Regressor algorithm is very effective for developing an AI analysis task for Bagapit Chess. The R^2 of 0.9134 on the model on the test set indicates high generalization ability, and the 84.2% success rate of the model on a standard rule indicates that the model has usable power. These results are consistent with the existing literature on Random Forest applications for rank analysis in non-board games. Several factors contributed to the success of the RFR-based method over simpler ones. First, the clustering of Random forests — predictions at 300 trees — significantly reduces the variance in the evaluation of each tree, allowing for a more accurate and precise assessment of complex positions. Second, the ability of the model to capture non-linear interactions between parameters is especially important in Catur Bagapit, where single-item analysis is insufficient to capture the complexity of the game.

The 24.7% return on investment provides an important insight about Catur Bagapit: marginal services and inventory management are as important as asset profitability. The current engine relies entirely on the learned parsing function and first-order search, which can lead to unlocking bugs. Future work should include the development of initial publications from registered games and end-to-end tables for standard data systems. The AI was developed in a Catur Bagapit application, with user-friendly, multiplayer capabilities and the ability to play multiplayer. The comparison with deep learning algorithms, especially convolutional neural networks used for 2D table representation, is also seen as an important direction for future research.

V. Discussion

The first artificial intelligence (AI) software was used for Bagapit Chess (Catur Bagapit), a board game from the Kalimantan region of Indonesia. Trained on 15,000 annotated playground locations, it achieved accurate spatial analysis (R^2 = 0.9134, MAE = 0.0872, RMSE = 0.1104) and, when combined with an AIB-based Negamax search engine enhanced by P legal origin in 84.2% of games. This study provides three main contributions: (1) the first formal use of Catur Bagapit figures, miniatures, and tableware; (2) the Random Forest Regressor is shown to be suitable for a gaming environment and has better performance than other ML algorithms, such as linear regression, decision trees, linear regression, and SVR; and (3) proof of concept for using AI to digitize and preserve Indonesian traditional games. The results showed that structural imbalances and movements were the most influential factors in Bagapit Chess positioning, providing new insights into the design of the game. Future work will focus on building end-to-end boards, building front pages, benchmarking deep learning, and building the entire digital game with integrated AI for cultural preservation applications.

References

[1] Z. Shuhaida Abdull Rahman, N. Ismail, B. Mambarasi Nehe, N. Labanihuda Abdull Rahman, A. Ramli, and K. Pengajian Seni Kreatif Universiti Teknologi MARA Shah Alam, "The Challenges Preserving Traditional Games in Malaysia and Indonesia," pp. 347–362, May 2025, doi: 10.2991/978-2-38476-406-8_26.

[2] H. Reginald, J. Yang, C. Y. Dendeng, and P. A. Suri, "From Tradition to Technology: Exploring Intergenerational Responses to a Benteng-Bentengan Video Game," *Procedia Comput Sci*, vol. 269, no. 5, pp. 732–740, Jan. 2025, doi: 10.1016/j.procs.2025.09.016.

[3] R. R. Varghese, D. R. Aiswarya, A. Roy, V. Muraly, and S. Renjith, "A Novel Approach to Predict Success of Online Games Using Random Forest Regressor for Time Series Data," in Advances in Electrical and Computer Technologies, Lecture Notes in Electrical Engineering, vol. 881, Springer, Singapore, 2022. https://doi.org/10.1007/978-981-19-1111-8_3

[4] D. Silver et al., "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm," arXiv preprint arXiv:1712.01815, 2017. [Related review available via DeepMind publications: https://www.maiachess.com/]

[5] R. Singh, A. Kumar, P. Trivedi, and M. Rathore, "CHESS AI: Machine Learning and Negamax Based Chess Engine," in Proc. 2023 IEEE Conference, IEEE Xplore, 2023. https://ieeexplore.ieee.org/document/10080746/

[6] N. Surjanovic, D. Bai, T. B. Murphy, and A. Bouchard-Côté, "Alpha-Trimming: Locally

Adaptive Tree Pruning for Random Forests," arXiv preprint, 2024. [Available: https://arxiv.org/abs/2107.12501 for related RF classifier game design work]

[7] Scientific Reports, "Enhancing Software Effort Estimation with Random Forest Tuning and Adaptive Decision Strategies," Nature/Scientific Reports, 2025. https://www.nature.com/articles/s41598-025-14372-7

[8] H. Choudhary, A. Inamdar, S. Kharade, A. Maheshwari, and P. Kale, "Comparative Analysis of Negamax Algorithm with Alpha–Beta Pruning Optimization for Chess Engine," in Advanced Computational and Communication Paradigms, ICACCP 2023, Lecture Notes in Networks and Systems, vol. 535, Springer, Singapore, 2023. https://doi.org/10.1007/978-981-99-4284-8_19

[9] S. Omidshafiei et al., "Navigating the Landscape of Multiplayer Games," Nature Communications, vol. 11, no. 5603, 2020. https://www.nature.com/articles/s41467-020-19244-4

[10] J. Goodman, A. Wallat, D. Perez-Liebana, and S. Lucas, "A Case Study in AI-Assisted Board Game Design," in Proc. 2023 IEEE Conference on Games (CoG), 2023. https://doi.org/10.1109/CoG57401.2023.10333138

[11] N. Politowski, F. Petrillo, G. El Boussaidi, G. Ullmann, and Y. Guéhéneuc, "Assessing Video Game Balance using Autonomous Agents," in Proc. 2023 IEEE/ACM 7th International Workshop on Games and Software Engineering (GAS), 2023. https://doi.org/10.1109/GAS59301.2023.00011

[12] M. Balla, G. Long, D. Jeurissen, J. Goodman, R. Gaina, and D. Perez-Liebana, "PyTAG: Challenges and Opportunities for Reinforcement Learning in Tabletop Games," in Proc. 2023 IEEE Conference on Games (CoG), 2023. https://doi.org/10.1109/CoG57401.2023.10333223

[13] B. Liu, C. Cheng, and Y. Zhao, "Games for Artificial Intelligence Research: A Review and Perspectives," arXiv:2304.13269v4, IEEE Transactions on Games, 2024. https://arxiv.org/html/2304.13269v4

[14] Adversarial Random Forest Classifier for Automated Game Design, arXiv preprint, arXiv:2107.12501, 2021. https://arxiv.org/abs/2107.12501

[15] T. Constant and G. Levieux, "Automated Evaluation of Game Experience based on Game Dynamics and Motives for Play," in Extended Abstracts of the 2022 Annual Symposium on Computer-Human Interaction in Play, 2022. https://doi.org/10.1145/3505270.3558343

[16] A. Albaghajati and M. Ahmed, "Video Game Automated Testing Approaches: An Assessment Framework," IEEE Transactions on Games, vol. 15, no. 1, pp. 81–94, Mar. 2023. https://doi.org/10.1109/TG.2020.3032796

[17] House Price Prediction Analysis Using Linear Regression and Random Forest — Comparative Analysis of MAE, RMSE, R^2, MAPE, IOINFORMATIC Journal of AI Engineering Applications, 2024. https://www.ioinformatic.org/index.php/JAIEA/article/download/1047/879

[18] Ma W., "Optimization of Alpha-Beta Pruning Based on Heuristic Algorithm," related review in ProjectPro AI documentation, 2023. https://www.projectpro.io/article/alpha-beta-pruning-in-ai/1157