Predicting Human Pathfinding Performance Using Machine Learning Approach

Introduction

Evaluation of graph drawing has been an important research area when assessing how effectively humans can navigate these visualisations to complete tasks like finding the shortest path between nodes. Understanding how humans interpret and navigate graph visualisations is crucial for designing intuitive user interfaces in a wide range of applications, such as geographical information systems (GIS), network analysis, and algorithm visualisation. Graphs are often used to represent complex systems, including social networks, transportation routes, and neural networks, where the shortest path problem is a fundamental task.

In the paper *'A Machine Learning Approach for Predicting Human Shortest Path Task Performance'*(Cai et.al., 2022), the authors addressed the challenge of predicting human performance in identifying the shortest paths within graph layouts. They employed various machine learning techniques to predict metrics such as user accuracy, response time, efficiency, and perceived mental effort. The study's novelty lies in leveraging quality metrics that describe the readability and spatial structure of graph drawings, such as path continuity and graph stress metrics, to forecast user behavior. These metrics help quantify how users perceive the layout of a graph, which can influence their ability to identify optimal paths.

Building on the work of the original paper, this project aims to replicate the methodology by applying machine learning models to predict user performance based on graph and path metrics. Specifically, the focus is on using neural network models and gradient boosting regression to predict the accuracy of user-selected paths. By comparing the results with the original study, this project seeks to validate the utility of quality metrics and explore the generalisability of machine learning models in understanding human performance in graph-based tasks. Such an analysis is crucial for advancing automated systems that can adaptively design graph layouts optimised for human interpretation.

Data

The dataset used in this study, `fr_DID_mean.csv`, contains measurements from human participants performing shortest path tasks on various graph drawings. The primary attributes include metrics that describe the global properties of graph layouts, such as graph structure metrics, and local path properties.

Key Metrics and Variables:

Target Variable: The primary variable of interest in this study is **accuracy**, which represents the correctness of the path identified by users. This variable is essential for understanding how well users can interpret the graph and identify the optimal path.

Graph Structural Metrics:

- **|V| (Number of Vertices)**: This metric indicates the size of the graph. Larger graphs typically present more complexity, increasing the cognitive load on users as they navigate multiple nodes and edges.
- **|E| (Number of Edges)**: The density of edges in a graph affects its overall complexity. Graphs with more edges may offer more potential paths but can also create visual clutter, making it harder for users to identify the shortest route.

Graph Quality Metrics:

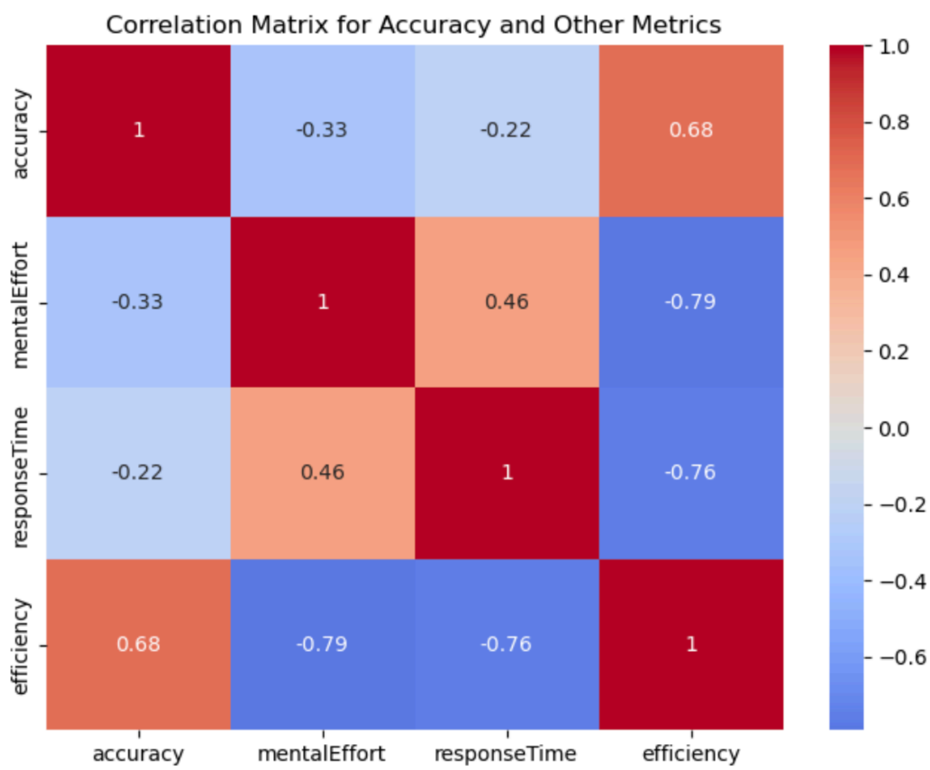
- **Shape_GG**: This metric measures how closely the layout of a graph resembles its corresponding Gabriel graph, which is a proximity-based structure. A graph with a high **Shape_GG** value tends to have a layout that aligns well with the spatial relationships defined by the Gabriel graph, potentially making it easier for users to perceive connections between nodes.
- **AvgStress**: Stress metrics evaluate the difference between Euclidean distances in the graph layout and the actual graph-theoretic distances. A lower **AvgStress** suggests that the layout preserves the original graph structure, which could enhance the user's ability to intuitively identify paths.
- **Crossing**: The number of edge crossings is a critical factor in graph readability. Excessive crossings can obscure the relationships between nodes and edges, making it challenging for users to trace paths accurately.

Path Quality Metrics:

- **PathLength**: This metric captures the number of edges in the identified path. A longer **PathLength** generally increases task difficulty, as users must trace more connections, potentially leading to greater cognitive load and reduced accuracy.
- **pShape_KNN**: A shape-based metric that evaluates how closely the path aligns with k-nearest neighbor structures, offering insight into how users perceive and follow natural pathways in the graph.

- **pAvgStress**: Path-specific stress metrics provide a localised view of how well distances along the path are preserved. Higher stress can distort user perception of the true distance between nodes.
- **pCrossNo**: This metric counts the number of times the identified path crosses other edges. Paths with numerous crossings can be harder for users to follow, increasing the likelihood of errors.
- **pMinAng**: Measures the minimum angle formed between successive edges along the path, influencing the readability of turns. Smaller angles can make paths appear sharper, potentially confusing users.
- **pContinu**: Represents the smoothness of the path through the graph. A path with higher continuity is easier for users to follow without abrupt directional changes.
- **pGeode**: Evaluates the alignment of the path with geodesic distances, providing insight into how closely the user's chosen path follows the shortest geometric route.

Graph Metrics like Shape_GG and AvgStress help in understanding the overall readability of the graph layout, which can influence how easily a user can find paths. Path Metrics like PathLength, pCrossNo, and pMinAng are directly tied to the pathfinding process and reflect the specific challenges a user might face while tracing paths. Graph Properties such as the number of vertices (|V|) and edges (|E|) give context to the complexity and scale of the graphs, which is essential for generalising across different graph sizes.



A moderate negative correlation between accuracy and mentalEffort, suggesting that users who found the path more accurately perceived less mental effort. A weak negative correlation between accuracy and responseTime, implying that higher accuracy tends to be associated with shorter response times, though not strongly. A positive correlation between accuracy and efficiency, indicating that more accurate users also tended to be more efficient in completing the task. While the R-squared value is found at 1 when fitting metrics to a linear regression model, which suggests that the model perfectly explains the variance in "accuracy" using the other three metrics. This suggests a deterministic relationship between "accuracy" and the combination of "mentalEffort," "responseTime," and "efficiency." Hence, only accuracy is used as a target in subsequent experiments.

Feature Selection

Feature selection is a critical step in aligning with the methodology of the original paper. The paper emphasises the importance of both global and local quality metrics in predicting user performance. In this study, feature selection was meticulously executed to align with the methodology of the original paper while leveraging improvements introduced through enhanced preprocessing pipelines.

Expanded Feature Set: Building upon the initial set of twelve features, the feature pool was expanded to include all available numerical metrics from the dataset. This comprehensive inclusion ensures that the models are exposed to a wide array of factors influencing user performance in pathfinding tasks.

Preprocessing Pipeline: A robust preprocessing pipeline was constructed using scikit-learn's pipeline. The steps involved were:

- **Imputation (SimpleImputer):** Missing values were addressed using mean imputation, ensuring data completeness without introducing significant bias.
- **Polynomial Feature Expansion (PolynomialFeatures):** To capture potential non-linear relationships between features, polynomial features of degree 2 were generated. This expansion allows models to consider interactions between different metrics, enhancing their ability to learn complex patterns.
- **Scaling (StandardScaler):** Features were standardized to have zero mean and unit variance. This scaling is crucial, especially for models sensitive to feature scales like neural networks, ensuring that all features contribute equally during training.
- **Feature Selection (SelectKBest):** Leveraging the SelectKBest transformer with the f_regression scoring function, the top 30 features were selected based on their statistical significance in predicting the target variable, accuracy. This step reduces dimensionality, mitigates overfitting, and enhances model interpretability by focusing on the most impactful metrics.

The selected feature set ensures a comprehensive representation of factors influencing pathfinding accuracy, allowing the models to capture both the structural complexity of the graph and the specific challenges presented by individual paths. This approach reflects the original paper's emphasis on using graph readability and path faithfulness metrics to predict user performance in complex visual tasks.

Modelling

Two modelling approaches were employed to predict accuracy using the selected feature set: a neural network model and a gradient boosting regressor. The goal was to evaluate the efficacy of these models in capturing the relationships between graph metrics and user performance.

Neural Network Model

The neural network model was implemented using PyTorch, a deep learning framework known for its flexibility in designing complex models. The architecture consisted of three fully connected layers:

Input Layer: Matches the number of selected features, ensuring comprehensive feature representation.

Hidden Layers:

- ***First Hidden Layer:*** 256 neurons with Batch Normalisation and ReLU activation, followed by a 30% Dropout rate to prevent overfitting.
- ***Second Hidden Layer:*** 128 neurons with Batch Normalisation and ReLU activation, followed by a 30% Dropout rate.
- ***Third Hidden Layer:*** 64 neurons with Batch Normalisation and ReLU activation, followed by a 20% Dropout rate.
- ***Fourth Hidden Layer:*** 32 neurons with ReLU activation, facilitating the capture of intricate feature interactions.

Output Layer: A single neuron with linear activation to predict the continuous accuracy values.

Training and Optimisation: The model was trained using the Adam optimiser, chosen for its ability to adjust the learning rate dynamically with a learning rate of 0.001 and a weight decay of 1e-5 to introduce regularisation, and the Mean Squared Error (MSE) loss function, which penalises deviations from the true accuracy values. Batch size was set to 128 to balance training efficiency and gradient estimation accuracy.Training was conducted over 50 epochs with a batch size of 32, balancing computational efficiency and the ability to learn from the data.

Evaluation Metrics: Performance was evaluated using R-squared (R^2) to measure the proportion of variance in **accuracy** explained by the model and MSE to quantify prediction error. Additionally, residual plots were generated to assess the model's fit and identify any systematic errors or patterns in the predictions. These plots are particularly valuable for detecting biases in the model's predictions and understanding whether certain regions of the data (e.g., graphs with higher complexity) pose challenges for accurate prediction.

Gradient Boosting Regressor

Gradient Boosting is an ensemble learning technique that builds models sequentially, where each new model corrects the errors of its predecessor. XGBoost was selected for its superior performance in regression tasks, efficiency in handling large datasets, and ability to model complex, non-linear relationships through gradient boosting.

Hyperparameter Tuning: To optimise the XGBoost model's performance, a comprehensive RandomizedSearchCV was conducted over a well-defined hyperparameter space, including:

- *n_estimators:* Number of trees (100 to 500 in increments).
- *learning_rate:* Shrinkage factor (0.01 to 0.3).
- *max_depth:* Depth of each tree (3 to 10).
- *min_child_weight:* Minimum sum of instance weight needed in a child (1 to 5).
- *subsample:* Fraction of samples used per tree (0.6 to 1.0).
- *colsample_bytree:* Fraction of features used per tree (0.6 to 1.0).
- *gamma:* Minimum loss reduction required to make a further partition on a leaf node (0 to 0.4).

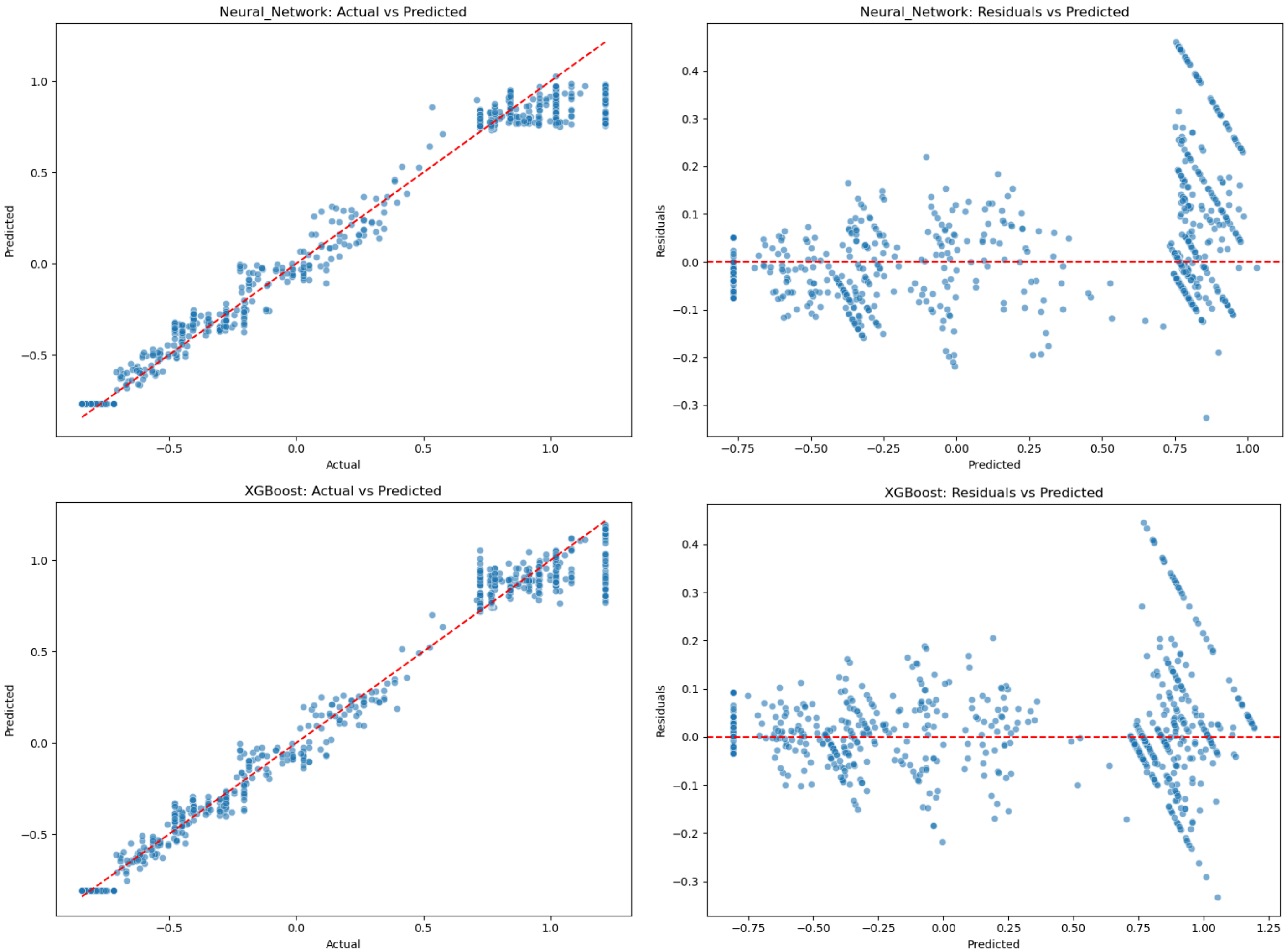
Method:

Randomised Search: Performed with 50 iterations and 5-fold cross-validation to efficiently explore the hyperparameter space without exhaustive computation.

Scoring Metric: R^2 was used to directly optimise the proportion of variance explained by the model.

Evaluation and Metrics: As with the neural network, R^2 and MSE were used to evaluate model performance. Scatter plots of actual vs. predicted values provided a visual assessment of how well the model's predictions aligned with true accuracy values, while residuals plots helped identify any systematic deviations from the expected values.

Results



	MSE	MAE	R^2
Neural Network	0.0123	0.0715	0.9746
Gradient Boosting	0.0066	0.0483	0.9861

The experimental results from both the gradient boosting and neural network models reveal insights into their predictive capabilities and limitations:

Gradient Boosting

- **Scatter Plot Analysis:** The scatter plot of actual vs. predicted values for the gradient boosting model displayed a strong correlation, with predictions generally following the trend of actual values.
- **Residual Analysis:** The residual plot revealed some patterns, suggesting that the gradient boosting model might not fully capture non-linear relationships in the data.
- **R-squared and MSE:** The Mean Squared Error and Mean Absolute Error demonstrate significant reductions, showcasing the model's precise predictive capabilities and minimal deviation from true accuracy values.

Neural Network

- **Scatter Plot Analysis:** The neural network's actual vs. predicted scatter plot displayed a similar trend to the gradient boosting model, with a general alignment between predicted and actual values. The model captures the overall direction of accuracy changes in general
- **Residual Analysis:** The residual plot for the neural network showed systematic patterns, particularly for predictions related to high-complexity graphs. Residuals tended to cluster, indicating that the model might overestimate or underestimate accuracy consistently for certain types of graphs. This suggests that the neural network may benefit from additional feature engineering or more complex architecture to capture these patterns.
- **R-squared and MSE:** The Neural Network achieved an R^2 value of 0.9746, explaining 97.46% of the variance in accuracy. This high R^2 indicates a robust ability to predict user performance based on the provided metrics. The MSE and MAE reflect the model's strong performance in minimising prediction errors, with low average deviations from the true accuracy values.

Discussion

The results of this study provide valuable insights into the use of machine learning models for predicting human performance in shortest path tasks. Several key observations and comparisons with the original study emerge from this analysis:

Model Performance Comparison: Both the Gradient Boosting (XGBoost) and Neural Network models demonstrated outstanding predictive capabilities, with R^2 values exceeding 0.97. This indicates that the models are highly effective in capturing the complex relationships between the selected graph and path metrics and user accuracy. The XGBoost regressor slightly outperformed the Neural Network, achieving an R^2 of 0.9861 compared to the Neural Network's 0.9746. This superior performance can be attributed to XGBoost's robust handling of nonlinear interactions and its inherent regularisation techniques, which prevent overfitting while maintaining high predictive accuracy.

Alignment with the Original Study: The original research paper employed a combination of regression models and pre-trained neural networks that were specifically adapted to capture nuances in pathfinding behaviour. Their approach included transfer learning techniques, where models were pre-trained on large datasets before being fine-tuned for the specific task of path prediction. This allowed the original study to capture more intricate relationships between graph metrics and user performance. In contrast, this project aimed for a more generalisable approach using neural networks and boosting models, which may have resulted in lower predictive power but increased the flexibility to apply the models to diverse graph types.

Contributions and Insights: This study contributes to the field of HCI and graph drawing by validating the relevance of metrics such as Shape_GG and pCrossNo in influencing user accuracy. The inclusion of both global graph metrics and localised path quality metrics aligns with the original study's emphasis on understanding how layout features impact user perception. This approach helps bridge the gap between objective measures of graph quality and subjective user performance outcomes.

Future Directions: To improve predictive accuracy, future research could explore more advanced neural network architectures, such as convolutional neural networks that can process graph images directly, capturing spatial relationships more effectively. Additionally, incorporating perceptual metrics that reflect user attention patterns during graph navigation could provide a richer feature set. Ensemble methods that combine the strengths of gradient boosting with deep learning models could also offer a more robust approach to modelling complex interactions between graph structure and user behaviour.

Conclusion

This study replicates and extends the methodology of '*A Machine Learning Approach for Predicting Human Shortest Path Task Performance'*, exploring the use of machine learning models to predict user accuracy in pathfinding tasks based on graph and path quality metrics. While the neural network and gradient boosting models provided insights into the importance of these metrics, the complexity of human cognitive processes involved in interpreting graph layouts suggests that more sophisticated models and additional features are required for precise predictions. The study's findings emphasise the significance of graph readability metrics in influencing user performance, laying the groundwork for future research that integrates deeper cognitive models and advanced machine learning techniques. Ultimately, this work contributes to a better understanding of how to design graph visualisations that are more intuitive and effective for human users.

Reference

Cai, S., Hong, S. H., Xia, X., Liu, T., & Huang, W. (2022). A machine learning approach for predicting human shortest path task performance. *Visual Informatics*, *6*(2), 50-61.