Starlytics: Strategy Analytics Web App for Starcraft II

Adam Wild, Patrick D. Hackett, Irfan Al-Hussaini, Anvesh Reddy Chennuru

Introduction/Motivation

Stacraft II (SC2) is a highly competitive popular real-time strategy game played across the world. Players choose one out of three races to play as: Protoss (P), Terran(T), and Zerg(Z), mining minerals to use as resources to build buildings and armies in real time, with the goal of building armies to destroy an opponent's armies and buildings.

Figure 1. Gameplay photo showing a standard Protoss player's base. This includes a primary building called a Nexus, units mining minerals, defensive structures, and a cluster of army units. Minimap (reproduced in Starlytics App) is shown in bottom left.

Building Artificial Intelligence (AI) for games that can successfully compete with professional players has been a compelling measuring stick for the progress of AI, with Google's DeepMind and Elon Musk's OpenAI leading the field.[1] No AI has come close to defeating professional players in Starcraft II, with in-game variables nearly as dynamic as an actual battlefield.

Our approach is to use machine learning to predict probability of winning during different time points in a game. This can be used as post hoc analysis to help inform player strategy. In the future, these analyses can help improve AI in-game decision-making and move the window forward in the progress of AI making complex decisions.

Data

Our data has been pulled from 36,619 game replays, restricted to games with an MMR (Match-making ratio) above 1000, APM (player Actions Per Minute) above 10, and frames above 10000. This ensures game data is from valid matches with both players of decent skill level and participating.

A normalized and parsed dataset has been provided by Wu, Huikai, and Zang.[2] Game data is collected every 2.57 seconds and includes resource collection rate, army size, building count, unit density, and map features at each point in the game. Total size of dataset is approximately 100 GB.

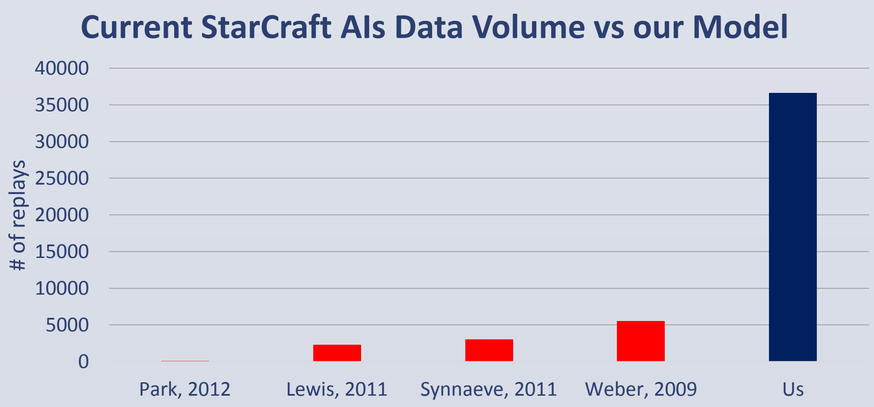

Figure 2. Comparison of our dataset size vs previous publications

Algorithm Approach and Results

Initially, we split the full dataset into training and validation (70%-30%), extracting a vector of 10 features representative of game state at certain points in the game. The main models trained were Linear Regression (LR), Artificial Neural Network (ANN), and Random Forest Classifier (RFC).

At the beginning of the game, accuracy is close to 50%, but as the game progresses the game state becomes more deterministic and the accuracy improves. ANN and LR outperform the RFC (Figure 2).

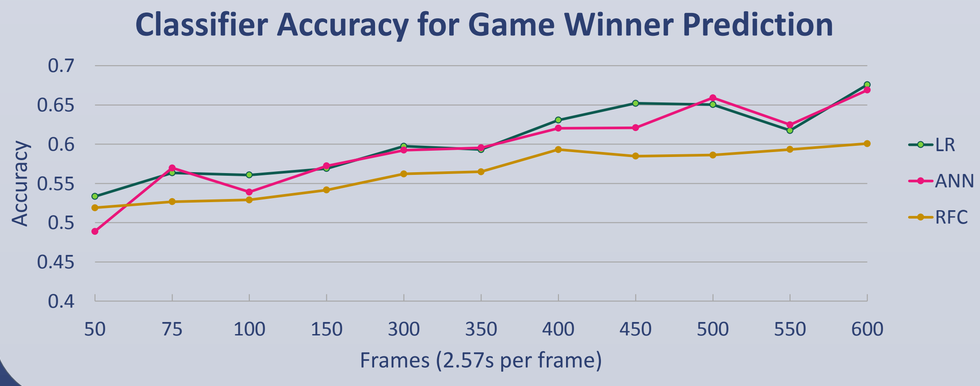

Figure 3. Comparison of model accuracy across time

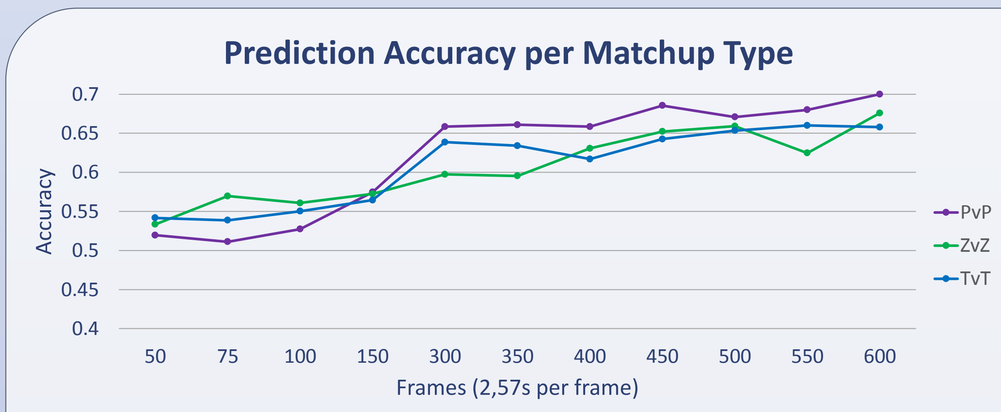

Figure 4. Automatic selection of highest performing model at each timepoint for each matchup

Visualization and Starlytics Web App and Model

Starlytics allows the user to upload a file, use a slider to move across time points, obtain a prediction based on that frame, and look at the condition of the minimap at that point. This enables players to view time points where their actions changed the likelihood of winning.

The web app was developed using a Django framework implementing a python backend consisting of a RFC with 50 trees with bootstrapping and features and depth not restricted. The model was trained on a subset of the data spanning all time frames of 1000 replays (1.6 GB) for approximately 200,000 datapoints. The model was chosen for local hosting computational speed.

The model has an average test dataset accuracy of ~58.5% on result prediction and ~78% on action prediction. The size of the resulting model for the two tasks were ~200MB and ~5.6GB respectively for result and action prediction.

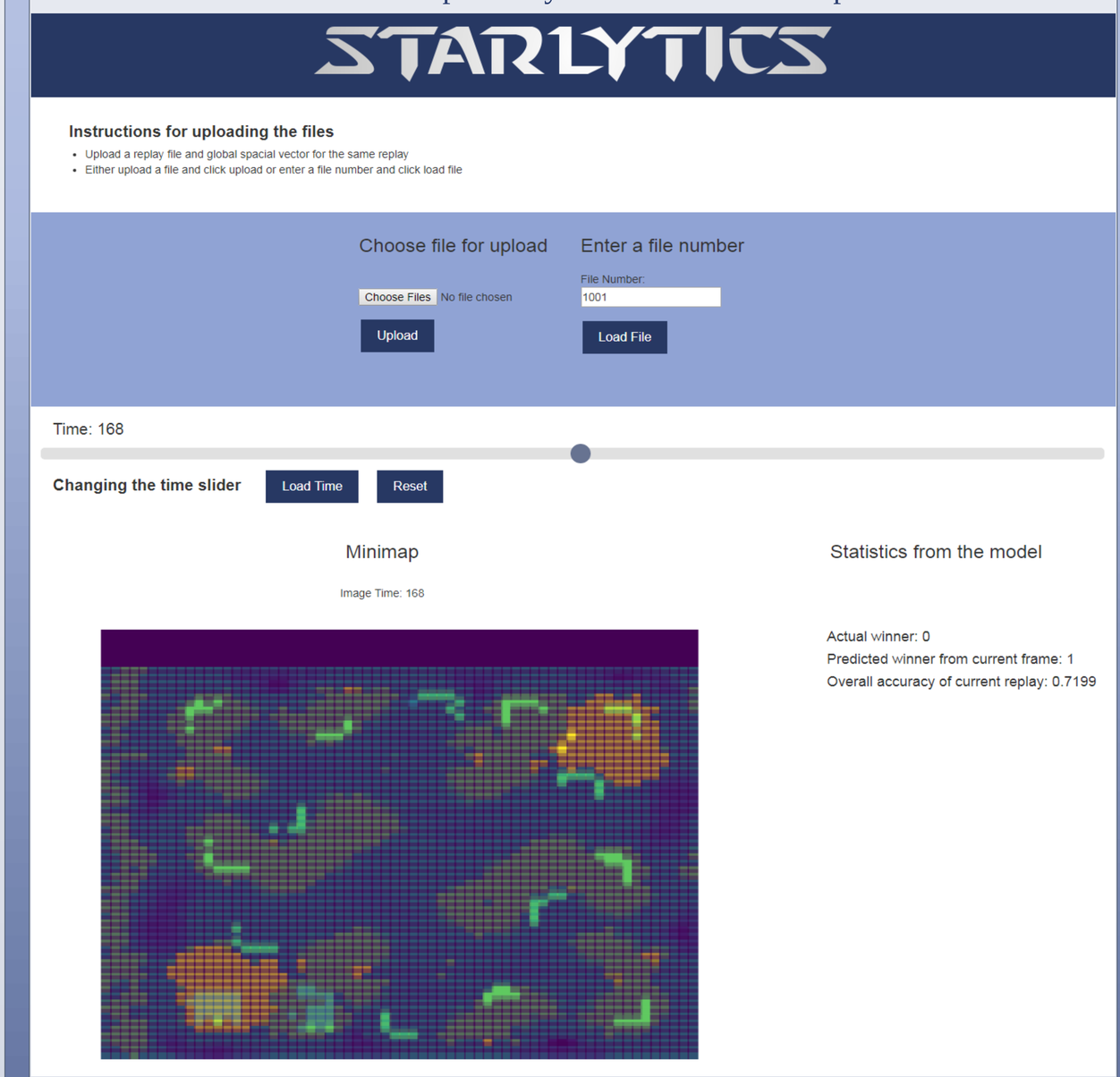

Figure 5. Starlytics Strategic Analytics for Starcraft II Web App

Conclusions and Future Directions

In order to surpass local hosting limitations in the proof of concept, future models will be hosted and trained on AWS EC2 instances in order to incorporate full 36,619 replays used in preliminary model comparisons and allow highest-performing model.

Starlytics brings a combination of AI and post-hoc game analysis to provide gamers with strategic guidance on gameplay, using the Starcraft II API in a new and publicly consumable manner. In addition, the high volume of analyses can be used to help guide future AI development decision making, shifting the window for AI processing real-time decisions in complex feature spaces.

References

1. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, K., et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484-489
2. Wu, Huikai, Zhang, J., Huang, K. MSC: Dataset for Macro-Management in Starcraft II. https://github.com/wuhuikai/MSC. Institution of Automation, Chinese Academy of Sciences (2017).