James Watt

School of Engineering

Preliminary Report on MSc Project

ENG5059P (MSc) ✓

Student Name	Ziyuan Cheng
Student GUID Number	2786151c
Degree programme	Computer Systems Engineering MSc
Working Title of Project	Can we predict the future?
Name of Supervisor(s)	Prof. Martin Lavery
Academic year	2022-2023

James Watt

School of Engineering

Preliminary Report on MSc Project

Introduction

This project plans to employ advanced machine learning techniques to develop a predictive model for forecasting the outcomes of professional Dota 2 matches. Dota 2 is a complex multiplayer online battle game that entails a rich array of strategic decisions, such as hero selection, item choice, and time management. These complexities make it a rich source of data for predictive modeling. The aim of this project is to investigate a variety of factors that might affect game outcomes, such as team composition diversity and item selection at various stages of early gameplay. We plan to utilize supervised learning techniques, especially deep neural networks, to predict the ultimate winner based on given combinations of heroes and item status in three different periods of early gameplay (5 minutes, 10 minutes, 15 minutes).

Aims/Objectives of project

1. **Data Collection & Processing Stage**: Approximately 2000 professional Dota 2 matches are played each month. To reduce the variability introduced by version updates, we choose to use about 6000 matches from the last three months for comprehensive analysis, studying the relationship between team composition diversity, item selection, and other significant factors and the outcomes of professional Dota 2 matches. This involves collecting and processing relevant data and using exploratory data analysis techniques to understand variables influencing game outcomes.

2. **Model Design & Training Stage**: Design and develop a machine learning model using deep neural networks, renowned for their ability to handle complex and high-dimensional data. The model should be able to predict wins or losses based on the provided data about hero combinations and item choices in three different periods of early gameplay.

3. **Model Evaluation**: Evaluate the performance of the predictive model, splitting the data into training and testing sets, and making predictions about victories based on the hero combinations and item selection situation at 5 minutes, 10 minutes, and 15 minutes using the test set, comparing prediction accuracies to select the most suitable prediction period. If time permits, the project will add factors for prediction to make the model more accurate, or increase the use of other models to explore differences between different models, as well as accelerate model training through Nvidia's CUDA parallel computing.

James Watt

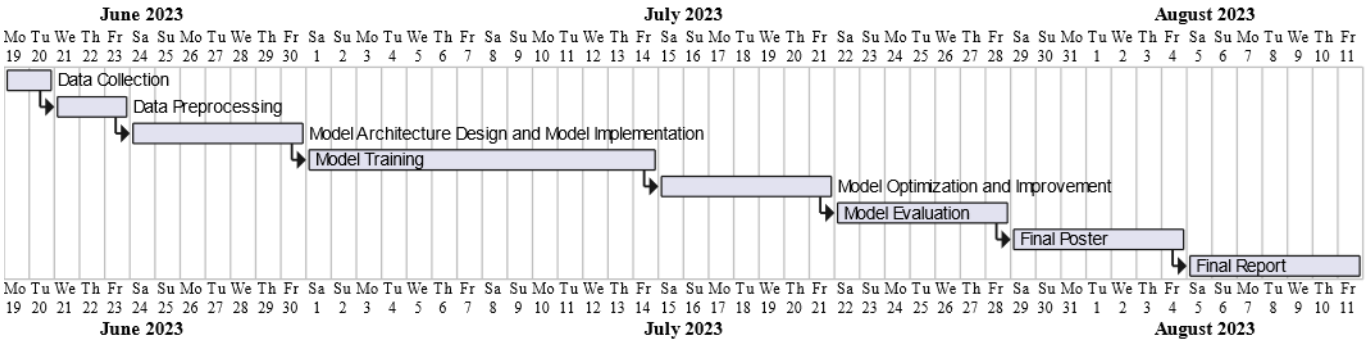
School of Engineering

Preliminary Report on MSc Project

Resources required

1. **Dota 2 match dataset**: including hero combinations, item choices, and match outcomes, obtained from the **OpenDota website API**.
2. **Machine learning libraries** for model development and training, such as TensorFlow or PyTorch: installed via **Anaconda**.
3. **Python** programming language for data preprocessing, model implementation, and evaluation: installed via **Anaconda**.
4. **Computational resources with GPU capabilities** for efficient deep learning model training: this project will use the **NVIDIA GeForce RTX 3060**.
5. **Data visualization libraries** for visualizing data patterns and model performance: **Matplotlib**.

GANTT chart



Risk assessment

1. **Data Security Risk**: The professional Dota 2 match data used in this project is completely public and does not contain any sensitive information, so there are no privacy or security risks associated with the data used in this project.
2. **Model Development Risk**: During the model development process, we might encounter technical issues such as overfitting, underfitting, or failure to converge. To mitigate these risks, we will employ strategies such as cross-validation and early stopping. Furthermore, we will utilize hyperparameter tuning tools and other optimization algorithms to enhance the model's performance.
3. **Software and Hardware Failure Risk**: During the model development and training process, there could be software or hardware failures that may cause delays in the project. To minimize this risk, we

will use version control tools like GitHub to back up all our code and data. We will also rely on high-quality hardware and tested software libraries.