SC2EGSet: StarCraft II Esport Replay and Game-state Dataset

Anonymous authors
Paper under double-blind review

Abstract

As a relatively new form of sport, esports offers unparalleled data availability. Despite the vast amounts of data that are generated by game engines, it can be challenging to extract them and verify their integrity for the purposes of practical and scientific use.

Our work aims to open esports to a broader scientific community by supplying raw and pre-processed files from StarCraft II esports tournaments. These files can be used in statistical and machine learning modeling tasks and related to various laboratory-based measurements (e.g., behavioral tests, brain imaging). We have gathered publicly available game-engine generated "replays" of tournament matches and performed data extraction and cleanup using a low-level application programming interface (API) parser library.

Additionally, we open-sourced and published all the custom tools that were developed in the process of creating our dataset. These tools include PyTorch and PyTorch Lightning API abstractions to load and model the data.

Our dataset contains replays from major and premiere StarCraft II tournaments since 2016. To prepare the dataset, we processed 55 tournament "replaypacks" that contained 17930 files with game-state information. Based on initial investigation of available StarCraft II datasets, we observed that our dataset is the largest publicly available source of StarCraft II esports data upon its publication.

Analysis of the extracted data holds promise for further Artificial Intelligence (AI), Machine Learning (ML), psychological, Human-Computer Interaction (HCI), and sports-related studies in a variety of supervised and self-supervised tasks.

1 Introduction

Electronic sports (esports) are a relatively new and exciting multidisciplinary field of study (Reitman et al., 2020; Chiu et al., 2021). There are multiple groups of stakeholders involved in the business of esports (Scholz, 2019). The application of analytics to sports aims to optimize training and competition performance. New training methods are derived from an ever increasing pool of data and research aimed at generating actionable insights (Pustišek et al., 2019; Giblin et al., 2016; Baerg, 2017; Chen et al., 2021; Rajšp & Fister jr, 2020; Kos & Umek, 2018). Rule changes in sports come at varying time intervals and frequently with unpredictable effects on their dynamics. It is especially relevant to share esports data to assess changes in game design and their impact on professional players, as such changes can occur more rapidly due to the (yet) relatively unstrctured nature of esports competition (Seif El-Nasr et al., 2013; Su et al., 2021).

Advancements in Artificial Intelligence (AI) and Machine Learning (ML) have shown that Reinforcement Learning (RL) agents are capable of outmatching human players in many different types of games (Vinyals et al., 2019; Jaderberg et al., 2019; Silver et al., 2018; Berner et al., 2019). Psychological research on neuroplasticity has also shown the great potential of video games to induce structural brain adaptation as a result of experience (Kowalczyk-Grębska et al., 2018). Further, previous studies have shown that playing video games can enhance cognitive functioining in a wide range of domains,

Redacted citations for double-blind review are marked as (REDACTED)
Dataset API Homepage: (REDACTED)

including perceptual, attentional and spatial ability (Green & Bavelier, 2003; 2012). Data obtained from esports titles – including those gathered from high-level tournament performance – may provide a path to improving the quality and reproducibility of research in this field, especially in contrast to the more variable data that is collected in laboratories and in less competitive settings. A lower technical overhead and more data available for modeling could assist further research in these areas (Alfonso et al., 2017; Ghasemaghaei, 2019; Zuiderwijk & Spiers, 2019).

The sparsity and methodological diversity of research on this topic remain as roadblocks in the study of how video games can affect mental functioning. Some scholars recommended further research on esports as a potential path forward (Reitman et al., 2020). Despite the digital nature of esports – which are their greatest asset with respect to data gathering – there seems to be a lack of high-quality pre-processed data published for scientific and practical use. The goal of our work is to mitigate this issue by publishing datasets containing StarCraft II replays and pre-processed data from esports events, classified as "Premiere" and "Major" by Liquipedia in the timeframe from 2016 until 2022 (Liquipedia, 2010).

A brief summary of the contributions stemming from this work is as follows: (1) The development of a set of four tools to work with StarCraft II data; (2) The collected esports data from various public sources; (3) The publication of a collection of raw replays after brief pre-processing (REDACTED) (4) The processing of raw data with our toolset and publication as a dataset (REDACTED) (5) and the preparation of an official API to interact with our data using PyTorch and PyTorch Lightning for ease of experimentation in further research (REDACTED)

2 Related Work

While reviewing StarCraft II related sources, we were able to find some publicly available datasets made in 2013 “SkillCraft1” (Blair et al., 2013) and 2017 “MSC” (Wu et al., 2017). These datasets are related to video games and in that regard could be classified as “gaming” datasets. However, it is not clear what percentage of games included within these datasets contain actively competing esports players and at what levels of skill. Using the SkillCraft1 dataset, the authors distinguished the level of players based on the data. They proposed a new feature in the form of the Perception-Action Cycle (PAC), which was calculated from the game data. This research can be viewed as the first step toward developing new training methods and analytical depth in electronic sports. It provided vital information describing different levels of gameplay and optimization in competitive settings (Thompson et al., 2013). In Table 1 we present a comparison of these two StarCraft II datasets to our own.

There are existing datasets in other games. Due to the major differences in game implementations, these could not be directly compared to our work. Despite that, such publications build upon a similar idea of sharing gaming or esports data for wider scientific audience and should be mentioned. Out of all related work, STARDATA dataset is notable in that it comes from prior generation of StarCraft game. This dataset seems to be the largest StarCraft: Brood War dataset available (Lin et al., 2021). Moreover, in League of Legends, a multimodal dataset including physiological data is available (Smerdov et al., 2020).

Table 1: StarCraft II dataset comparison

Dataset	esports	public	replays available	pre-processed	API available	replays	timespan
SC2EGSet	✓	✓	✓	✓	✓	17895	2016-2022
SkillCraft1 (Blair et al., 2013)	✗	✓	✗	✓	✗	3395	ND+
MSC (Wu et al., 2017)	✗	✓	✓*	✓	✓	36619	ND+

* provided by the game publisher
+ ND - not disclosed

Related publications focused on in-game player performance analyses and psychological, technical, mechanical or physiological indices. These studies were conducted with use of various video games such as: Overwatch (Braun et al., 2017; Glass & McGregor, 2020), League of Legends (Blom et al., 2019; Ani et al., 2019; Aung et al., 2018; Maymin, 2021; Lee et al., 2022), Dota 2 (Gourdeau & Archambault, 2020; Hodge et al., 2017; 2019; Cavadenti et al., 2016; Pedrassoli Chitayat et al., 2020), StarCraft (Sánchez-Ruiz & Miranda, 2017; Stanescu et al., 2021; Norouzzadeh Ravari et al., 2021), StarCraft II (Helmke et al., 2014; Lee et al., 2021; Lee & Ahn, 2021; Cavadenti et al., 2015; Volz et al., 2019), Heroes of the Storm (Gourdeau & Archambault, 2020), Rocket League (Mathonat et al., 2020), and Counter-Strike: Global Offensive (Khromov et al., 2019; Koposov et al., 2020; Smerdov et al., 2019; Xenopoulos et al., 2022; Jonnalagadda et al., 2021), among others (Galli et al., 2011). In some cases a comparison between professional and recreational players was conducted.

Most studies did not provide data as a part of their publication. In other publications, the authors used replays that were provided by the game publishers or were publicly available online, which are unsuitable for immediate data modeling tasks without prior pre-processing. The researchers used raw files in MPQ (SC2Replay) format with their custom code when dealing with StarCraft II (Blizzard, 2017; Wang et al., 2020). Other studies solved technical problems that are apparent when working with esports data and different sensing technologies, including visualization, but with no publication of data (Bednárek et al., 2017; Feitosa et al., 2015; Afonso et al., 2019; Stepanov et al., 2019; Korotin et al., 2019). Some researchers have attempted to measure tiredness in an undisclosed game via electroencephalography (EEG) (Melentev et al., 2020), and player burnout using a multimodal dataset that consisted of EEG, Electromyography (EMG), galvanic skin response (GSR), heart rate (HR), eyetracking, and other physiological measures in esports (Smerdov et al., 2021).

2.1 Game Description

Many of the related works introduce and communicate the properties of the games that they analyze. In case of StarCraft II, we recommend the following description: “StarCraft II: Legacy of The Void (SC2) contains various game modes: 1v1, 2v2, 3v3, 4v4, Archon, and Arcade. The most competitive and esports related mode (1v1) can be classified as a two-person combat, real-time strategy (RTS) game. The goal of the game for each of the competitors is either to destroy all of the structures, or to make the opponent resign.” Moreover, StarCraft II contains multiple matchmaking options: “Ranked game - Players use a built-in system that selects their opponent based on Matchmaking Rating (MMR) points. Unranked game - Players use a built-in system that selects their opponent based on a hidden MMR - such games do not affect the position in the official ranking. Custom game - Players join the lobby (game room), where all game settings are set and the readiness to play is verified by both players - this mode is used in tournament games. Immediately after the start of the game, players have access to one main structure, which allows for further development and production of units.” (Białecki et al., 2022).

3 Material and Methods

3.1 Dataset Sources and Properties

The files used in the presented information extraction process were publicly available due to a StarCraft II community effort. Tournament organizers for events classified as "Premiere" and "Major" made the replays available immediately after the tournament to share the information with the broader StarCraft II community for research, manual analysis, and in-game improvement. Sources include Liquipedia, Spawning Tool, Reddit, Twitter, and tournament organizer websites. All replay packs required to construct the dataset were searched and downloaded manually from the public domain. The critical properties of the presented dataset are as follows:

- To secure the availability of the raw replays for further research and extraction, the SC2ReSet: StarCraft II Esport Replaypack Set was created (REDACTED)
- The replays were processed under the licenses provided by the game publisher: End User License Agreement (EULA), and "Blizzard StarCraft II AI and Machine Learning License" which is available in subsection A.1 of the supplementary material.
- Our dataset was created by using open-source tools that were published with separate digital object identifiers (doi) minted for each of the repositories. These tools are indexed on Zenodo (REDACTED; REDACTED; REDACTED)
- We have made available a PyTorch (Paszke et al., 2019) and PyTorch Lightning (Falcon & The PyTorch Lightning team, 2019) API for accessing our dataset and performing various analyses. Our API is accessible in the form of a GitHub repository, which is available on

Zenodo with a separate doi. All of the instructions for accessing the data and specific field documentation are published there (REDACTED)

- The presented dataset is currently the largest that is publicly available, and contains information from prestigeous StarCraft II tournaments.
- The dataset can be processed under CC BY NC 4.0 to comply with Blizzard EULA and the aforementioned Blizzard StarCraft II AI and Machine Learning License.

3.2 Dataset Pre-Processing

Dataset pre-processing required the use of a custom toolset. Initially, the Python programming language was used to process the directory structure which held additional tournament stage information. We include this information in the dataset in a separate file for each tournament, effectively mapping the initial directory structure onto the resulting unique hashed filenames. Moreover, a custom tool for downloading the maps was used; only the maps that were used within the replays were downloaded (REDACTED) Finally, to ensure proper translation to English map names in the final data structures, a custom C++ tool implementation was used. Information extraction was performed on map files that contained all necessary localization data (REDACTED) The entirety of our processing pipeline is visualized in Figure 1, and additional visualizations are available in the Appendix, subsection A.4.

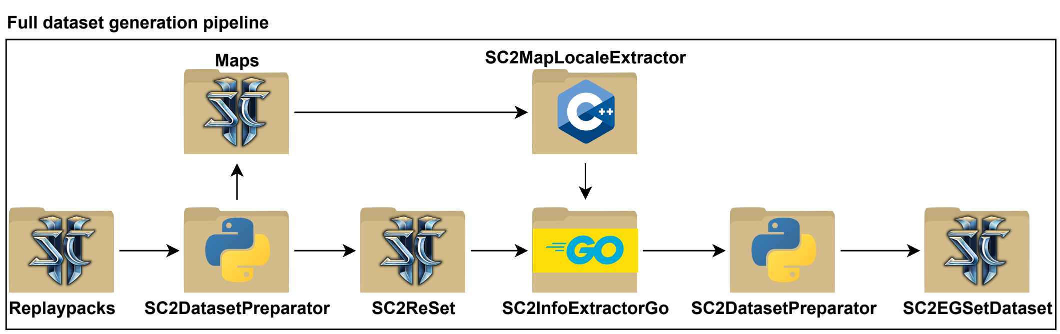

Figure 1: Pre-processing and processing steps of our pipeline that result in SC2ReSet (REDACTED) and SC2EGSetDataset (REDACTED) We used a custom data processing toolset including the SC2DatasetPreparator (REDACTED) SC2MapLocaleExtractor (REDACTED) and SC2InfoExtractorGo (REDACTED) .

3.3 Data Processing

Custom software was implemented in the Go programming language (Golang) and built upon authorized and public repositories endorsed by the game publisher (Belicza, 2016). The tool was used to perform information extraction from files in MPQ format with the SC2Replay extension. Information extraction was performed for each pre-processed directory that corresponded to a single tournament. Depending on the use case, different processing approaches are possible by providing command line arguments (REDACTED) .

3.4 Data Parsing and Integrity

The parsing capabilities of the tooling were defined with a Golang high-level parser API available on GitHub (Belicza, 2016). After initial data-structures were obtained, the next step checked the integrity of the data. This was accomplished by comparing information available across different duplicate data structures that corresponded to: the number of players, map name, length of the player list, game version, and Blizzard map boolean (signifying whether a map was published by Blizzard). If a replay parser or custom integrity check failed, the replay was omitted.

3.5 Data Filtering and Restructuring

Filtering for different game modes was omitted as collected replay files were a part of esports tournament matches. Most often, StarCraft II tournament matches are played in the form of one versus one player combat. Therefore, it was assumed that filtering for the number of players was not required at this step. Custom data structures were created and populated at this stage. This allowed for more control over the processing, summary generation, and final output. Merging data structures containing duplicate information was performed where applicable.

3.6 Summarization and JSON Output to zip archive

Replay summarization was required in order to provide information that can be accessed without unpacking the dataset. Finally, the data was converted from Golang data structures into JavaScript Object Notation (JSON) format, and compressed into a zip archive.

3.7 Dataset Loading

Interacting with the dataset is possible via PyTorch (Paszke et al., 2019) and PyTorch Lightning (Falcon & The PyTorch Lightning team, 2019) abstractions. Our implementations exposes a few key features: (1) Automatic dataset downloading and extraction from Zenodo archive; (2) Custom validators that filter or verify the integrity of the dataset; (3) The ability of our abstractions to load and use any other dataset that was pre-processed using our toolset. The required disk space to succesfully download and extract our dataset is approximately 170 gigabytes. We showcase the example use of our API in Listing 1. Please note that the API is subject to change and any users should refer to the official documentation for the latest release features and usage information. Additional listing showcasing the use of SC2EGSetDataset is available in the Appendix, subsection A.5. Additionally, we include human readable examples of JSON data structures in the Appendix, subsection A.6.

```python
from sc2_datasets.torch.sc2_egset_dataset import SC2EGSetDataset
from sc2_datasets.available_replaypacks import (
    EXAMPLE_SYNTHETIC_REPLAYPACKS
)

if __name__ == "__main__":
    # Initialize the dataset:
    sc2_egset_dataset = SC2EGSetDataset(
        unpack_dir="./unpack_dir_path",
        download_dir="./download_dir_path",
        download=True,
        names_urls=EXAMPLE_SYNTHETIC_REPLAYPACKS,
    )

    # Iterate over instances:
    for i in range(len(sc2_egset_dataset)):
        sc2_replay_data = sc2_egset_dataset[i]
```

Listing 1: Example use of the SC2EGSetDataset with PyTorch with a synthetic replaypack prepared for testing.

4 Dataset Description

The collected dataset consisted of 55 tournaments. Within the available tournaments, 18309 matches were processed. The final processing yielded 17895 files. While inspecting the processed data, we observed three major game versions. Each tournament in the dataset was saved with an accompanying JSON file that contains descriptive statistics such as: (1) Game version histogram, (2) Dates at which the observed matches were played, (3) Server information, (4) Picked race information, (5) Match length, (6) Detected spawned units, (7) Race picked versus game time histogram. Figure 2 depicts the frequency with which each of the races played against the other and the distribution of races observed within the tournaments. Figure 4 depicts the distribution of match times that were observed.

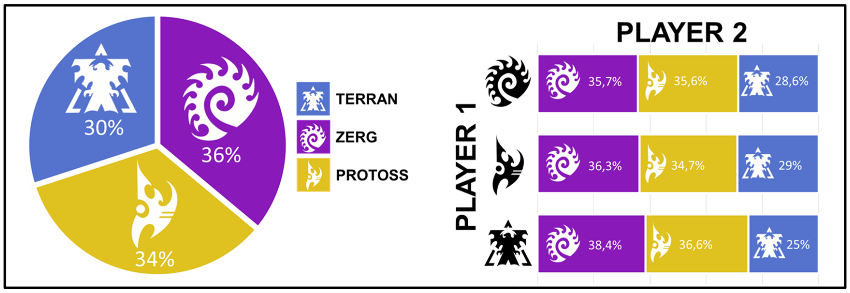

Figure 2: Distribution of player races and race matchup information.

The oldest observed tournament was IEM 10 Taipei, which was played in 2016. The most recent observed tournament was IEM Katowice, which finished on 2022.02.27. The game contains different "races" that differ in the mechanics required for the gameplay. Figure 3 shows visible differences in the distribution of match time for players that picked different races.

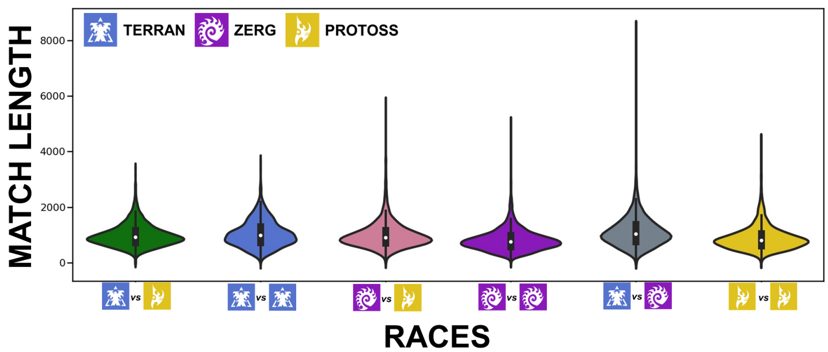

Figure 3: Match time distribution split by races: Terran (blue), Protoss (yellow), and Zerg (purple).

The published data resulting from our work is distributed under the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) license and is available in a widely recognized scientific repository - Zenodo.

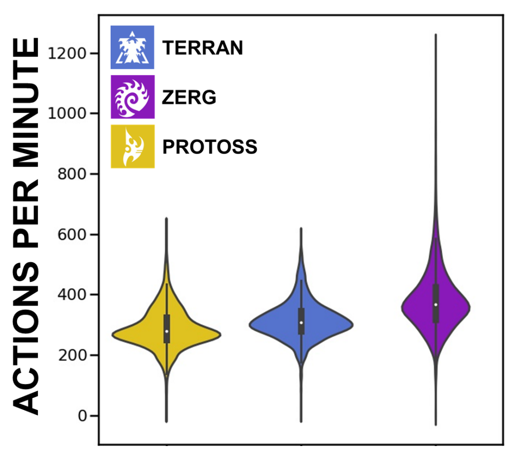

Figure 4: Actions per minute (APM) by player race.

5 EXPERIMENTS AND FUTURE RESEARCH

5.1 WINNER PREDICTION AND PLAYER PERFORMANCE EVALUATION

Within section 2 we have referenced multiple articles that dealt with player performance evaluation. These works performed data mining tasks on game engine generated replays and other sources of player related information.

Experiments regarding winner prediction can uncover interesting information about the optimal strategy of play. Prior analyses in this task with a small sample of esports players have shown the importance of some key indicators. The proposed dataset can help with the reproduction and facilitation of various claims, some of which are based on anecdotal evidence (Białecki et al., 2022). The sample analysis below describes a basic attempt at predicting match outcome using only data related to player economy to demonstrate the potential for gleaning insights from replay data.

Data Preparation Matches were initially filtered to only include those exceeding or equaling a length of 9 minutes, which is approximately the 25th percentile of match length values. Next, a set of features was generated from the available economy-related indicators. Additional features were generated by combining mineral and vespene indicators into summed resource indicators. Match data were then aggregated by averaging across match time for each player, resulting in 22,230 samples of averaged match data (from 11,115 unique matches). Standard deviations were computed in addition to averaged values where applicable. Further features were then generated by computing ratios of resources killed/resources lost for army, economy and technology contexts, along with a ratio of food made to food used. As a final step, prior to feature standardization, each feature was filtered for outliers (replacing with median) that exceeded an upper limit of 3 standard deviations from the feature mean.

Feature Selection The feature count was reduced by first computing point biserial correlations between features and match outcome, selecting for features with a statistically significant (α = .001) coefficient value exceeding that of $\pm$ .050. Next, a matrix of correlations was computed for the remaining features and redundant features were removed. 17 features remained after this process, of which 8 were basic features (mineralsLostArmy, mineralsKilledArmy, mineralsLostEconomy, mineralsKilledEconomy, and the SD for each).

Modelling Modelling was conducted on features (economic indicators) that represented the global average gamestate, in which all time points were aggregated into a single state, and also as a time series in which the gamestate was represented at a sampling rate of approx. 7 seconds. Three algorithms were chosen for comparative purposes: Logistic Regression (sklearn.linear_model.LogisticRegression), Support Vector Machine (sklearn.svm.SVC) (Pedregosa et al., 2011; Buitinck et al., 2013), and Extreme Gradient Boosting (xgboost.XGBClassifier) (Chen & Guestrin, 2016). Each algorithm was initiated with settings aimed at binary classification and with typical starting hyperparameters. A 5-fold cross validation procedure was implemented across the models.

Two sets of models were trained for the average gamestate and one for the gamestate as a time series. In the first averaged set of models the input features represented the economic gamestate of a single player without reference to their opponent, with the model output representing outcome prediction accuracy for that player - a binary classification problem on scalar win/loss classes. The second averaged set of models differed in that it used the averaged economic gamestate of both players as input features, and attempted to predict the outcome of "Player 1" for each match. Finally, the time series models consisted of a feature input vector containing the economic gamestate at 7 second intervals – the task here was also to predict the outcome of a match based on only a single player's economic features, as in the single-player averaged set of models.

Label counts were equalized to the minimal label count prior to generating the data folds, resulting in 10,744 samples of "Win" and "Loss" labels each for the single-player averaged models and the time series models. For the two-player set of averaged models (containing the features of both players in a given match), the total number of matches used was 10,440. Accuracy was chosen as the model performance evaluation metric in all three cases. Computation was performed on a standard desktop-class PC without additional resources.

Results As the results indicate (see Table 2), good outcome prediction can be achieved from economic indicators only, even without exhaustive optimization of each model's hyperparameters. For the one-player averaged set of models, SVM and XGBoost displayed similar performance, with the logistic classifier lagging slightly behind. For the two-player averaged set of models, all three algorithms performed essentially equally well. Feature importances were taken from a single-player XGBoost model (with identical hyperparameters) that was applied to the entire dataset for illustrative purposes. Figure 6 available in Appendix, subsection A.3, depicts the top five features by importance. It is interesting to note that importance was more heavily centered around mineral-related features than those tied to vespene, which is likely tied to how mineral and vespene needs are distributed across unit/building/technology costs. Further feature investigation is required to verify this tendency.

Table 2: Classification models and their performance metrics for two separate win prediction models. The “One Player Prediction” models attempt to correctly output if one of the players won or lost. The “Two Player Prediction” models have access to the data for both of the players and attempts to output if "Player 1" won or lost.

Classifier	Accuracy	SD	Hyperparameters
One Player Prediction			
Support Vector Machine - RBF	0.8488	0.0075	kernel='rbf', C=10, gamma='auto'
XGBoost	0.8397	0.0064	Booster='gbtree', eta=0.2, max_depth=5
Logistic Regression	0.8118	0.0057	C=10, penalty='l2'
Two Player Prediction			
Support Vector Machine - RBF	0.9071	0.0055	kernel='rbf', C=10, gamma='auto'
XGBoost	0.8924	0.0063	Booster='gbtree', eta=0.2, max_depth=5
Logistic Regression	0.8916	0.0063	C=10, penalty='l2'

Figure 5 depicts the time series application of these models as an illustration of outcome prediction accuracy over time. It should be noted that these time series results are not based on any form of data aggregation, and as such only basic economic features could be used for classification (18 features in total).

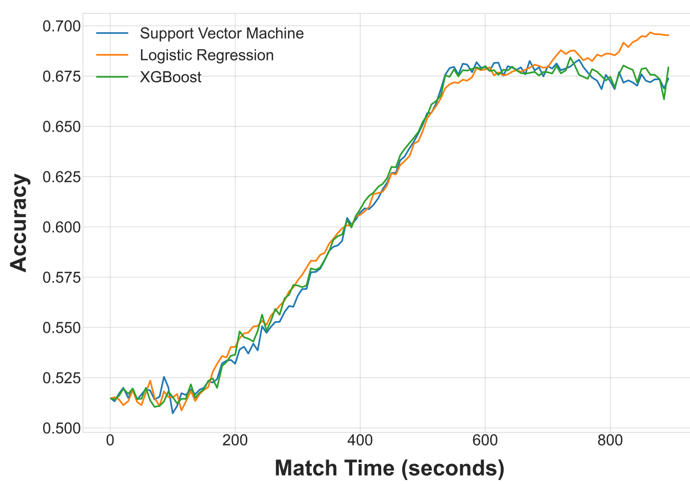

Figure 5: Accuracy comparison of applied classification models.

Each timepoint contains the average accuracy for 5-fold cross validation, with a minimum match length of 9 minutes and a maximum match length of approx. 15 minutes. All three algorithms provided similar performance over time, although this may be an effect of the minimal hyperparameter optimization that was performed. Further, it is also interesting to note and that all three algorithms meet a classification performance asymptote at approx. the same match time (~550 seconds), which may indicate that this is where economic indicators begin to lose their predictive power and (presumably) other factors such as army size, composition, and their application become the primary determinants. The code for our experiments is available at a dedicated GitHub repository (REDACTED).

5.2 Future Research

5.2.1 Game Style Analysis

Game style analysis can be treated as a task to be solved via supervised or self-supervised methods. Using algorithms such as Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) or t-Distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten & Hinton, 2008) for the data that we provided could uncover interesting insights depending on the direction of the analysis. Such game style analysis could be investigated using sequence analysis methods or use per game statistics.

5.2.2 Combat Encounter Analysis

Combat analysis as a task can be researched using AI, ML, and classic algorithms in various esports (Uriarte & Ontañón, 2018). There were some related works that analyzed unit encounters in StarCraft II (Lee et al., 2021). Although our pre-processed dataset cannot be used to directly reproduce combat encounter analyses, we provide raw replays published as SC2ReSet (REDACTED).

6 Limitations

We acknowledge that our work is not without limitations. The design and implementation of our dataset do not consider the ability to obtain StarCraft II data through game-engine simulation at a much higher resolution. Because of this, the extracted dataset cannot reflect exact unit positioning. Replays in their original MPQ (SC2Replay) format contain all necessary information to recreate a game using game-engine API. Therefore, we plan to continue our research and provide more datasets that will expand the scientific possibilities within gaming and esports. Further, it should be noted that the experiments described here are more illustrative than investigative in nature, and could be greatly expanded upon in future work. We recommend further research to use SC2ReSet (REDACTED) to compute game-engine simulated information. We also do not provide simulation observation data that allows more detailed spatiotemporal information to be extracted at a higher computational cost. Moreover, it is worth noting that the dataset completeness was dependent on which tournament organizers and tournament administrators decided to publish replay packs.

7 Discussion

Future authors may want to filter out replays that ended prematurely due to unknown reasons. Our dataset may contain replays that are undesirable for esports research. We have decided against the deletion of replays to preserve the initial distributions of data. Additionally, as filtering was omitted (besides that performed for the purposes of the described experiments), there is a risk that the dataset contains matches that were a part of the tournament itself but did not count towards the tournament standings. Due to the timeframe of the tournaments and game version changes, despite our best efforts, some information might be missing or corrupted and is subject to further processing and research.

Our dataset is the largest publicly available pre-processed esports dataset. Moreover, in preparing the data, we defined and published the software used for the data extraction process and other tasks. Future research on StarCraft II may be built upon these tools and our dataset (REDACTED; REDACTED; REDACTED) .

The dataset may also serve to increase knowledge regarding the in-game behavior of players, i.e. the relationship between the variables and overall strategies used by the players at high levels of advancement. Such information can be used in comparisons to non-gamers or intermediate players in the process of studying the relationship between game proficiency, cognitive functioning, and brain structure (Jakubowska et al., 2021).

Moreover, a report done in the area of clinical medicine highlighted the lack of compliance of many authors with their data availability statement (DAS). It is clear that publishing the data and tools required for modeling is a key component of ensuring reproducible scientific work (Gabelica et al., 2022).

Other noteworthy applications of the dataset include comparing gameplay styles, action sequence classification, and their relation to victory. To that end, we encourage using different statistical methods and Machine Learning (ML) algorithms, including supervised and self-supervised approaches.

ACKNOWLEDGEMENTS

Available in the final version. Redacted for double-blind review process.

DECLARATIONS

AUTHORS' CONTRIBUTIONS

Available in the final version. Redacted for double-blind review process.

AUTHOR STATEMENT

We acknowledge that we as authors bear all responsibility in case of violation of rights.

FUNDING

This publication was self-funded.

CONFLICTS OF INTEREST/COMPETING INTERESTS

Authors declare no conflict of interest.

AVAILABILITY OF DATA AND MATERIAL

Extracted data is published as a dataset in a scientific repository (REDACTED; REDACTED; REDACTED).

CODE AVAILABILITY

The code used for data extraction is available as open source implementations published by the authors (REDACTED; REDACTED; REDACTED) . The code used for experiments is available for preview in a GitHub repository (REDACTED).

In the process of preparing this article, PyTorch Lightning has changed its name into Lightning. We have decided to use the old form of the name, following the citation template provided by the Lightning project on GitHub (Falcon & The PyTorch Lightning team, 2019).

REFERENCES

Ana Paula Afonso, Maria Beatriz Carmo, and Tiago Moucho. Comparison of Visualization Tools for Matches Analysis of a MOBA Game. In *2019 23rd International Conference Information Visualisation (IV)*, pp. 118–126, 2019. doi:10.1109/IV.2019.00029. URL https://doi.org/10.1109/IV.2019.00029.

Fernando Alfonso, Karlen Adamyan, Jean-Yves Artigou, Michael Aschermann, Michael Boehm, Alfonso Buendia, Pao-Hsien Chu, Ariel Cohen, Livio Dei Cas, Mirza Dilic, Anton Doubell, Dario Echeverri, Nuray Enç, Ignacio Ferreira-González, Krzysztof J. Filipiak, Andreas Flammer, Eckart Fleck, Plamen Gatzov, Carmen Ginghina, Lino Goncalves, Habib Haouala, Mahmoud Hassanein, Gerd Heusch, Kurt Huber, Ivan Hulín, Mario Ivanusa, Rungroj Krittayaphong, Chu-Pak Lau, Germanas Marinskis, François Mach, Luiz Felipe Moreira, Tuomo Nieminen, Latifa Oukerraj, Stefan Perings, Luc Pierard, Tatjana Potpara, Walter Reyes-Caorsi, Se-Joong Rim, Olaf Rødevand, Georges Saade, Mikael Sander, Evgeny Shlyakhto, Bilgin Timuralp, Dimitris Tousoulis, Dilek Ural, J.J. Piek, Albert Varga, and Thomas F. Lüscher. Data Sharing: A New

Editorial Initiative of the International Committee of Medical Journal Editors. Implications for the Editors' Network. *Revista Portuguesa de Cardiologia*, 36(5):397–403, 2017. ISSN 0870-2551. doi:10.1016/j.repc.2017.02.001. URL https://doi.org/10.1016/j.repc.2017.02.001.

R. Ani, Vishnu Harikumar, Arjun K. Devan, and O.S. Deepa. Victory prediction in League of Legends using Feature Selection and Ensemble methods. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pp. 74–77, 2019. doi:10.1109/ICCS45141.2019.9065758. URL https://doi.org/10.1109/ICCS45141.2019.9065758.

Myat Thura Aung, Valerio Bonometti, Anders Drachen, Peter Ivan Cowling, Athanasios Vasileios Kokkinakis, Christian Yoder, and Alexander Robert Patrick Wade. Predicting skill learning outcomes in a large, longitudinal MOBA dataset. In *Proceedings of the IEEE Computational Intelligence in Games*. IEEE, 2018.

Andrew Baerg. Big Data, Sport, and the Digital Divide: Theorizing How Athletes Might Respond to Big Data Monitoring. *Journal of Sport and Social Issues*, 41(1): 3–20, 2017. doi:10.1177/0193723516673409. URL https://doi.org/10.1177/0193723516673409.

David Bednárek, Martin Krulis, Jakub Yaghob, and Filip Zavoral. Data Preprocessing of eSport Game Records - Counter-Strike: Global Offensive. pp. 269–276, 01 2017. doi:10.5220/0006475002690276. URL https://doi.org/10.5220/0006475002690276.

András Belicza. s2prot. https://github.com/icza/s2prot, 2016. Acessed: 2021-10-13.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Andrzej Białecki, Jan Gajewski, Piotr Białecki, Ashwin Phatak, and Daniel Memmert. Determinants of victory in Esports - StarCraft II, sep 2022. ISSN 1573-7721. URL https://doi.org/10.1007/s11042-022-13373-2.

M. Blair, J. Thompson, A. Henrey, and B. Chen. SkillCraft1 Master Table Dataset. UCI Machine Learning Repository, 2013. URL https://archive-beta.ics.uci.edu/ml/datasets/skillcraft1+master+table+dataset. Acessed: 2022-06-03.

Blizzard. s2client-proto, 2017. URL https://github.com/Blizzard/s2client-proto. Acessed: 2021-08-26.

Paris Mavromoustakos Blom, Sander Bakkes, and Pieter Spronck. Towards Multi-modal Stress Response Modelling in Competitive League of Legends. In *2019 IEEE Conference on Games (CoG)*, pp. 1–4, 2019. doi:10.1109/CIG.2019.8848004. URL https://doi.org/10.1109/CIG.2019.8848004.

Peter Braun, Alfredo Cuzzocrea, Timothy D. Keding, Carson K. Leung, Adam G.M. Padzor, and Dell Sayson. Game Data Mining: Clustering and Visualization of Online Game Data in Cyber-Physical Worlds. *Procedia Computer Science*, 112:2259–2268, 2017. ISSN 18770509. doi:10.1016/j.procs.2017.08.141. URL http://dx.doi.org/10.1016/j.procs.2017.08.141.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pp. 108–122, 2013.

Olivier Cavadenti, Victor Codocedo, Jean-François Boulicaut, and Mehdi Kaytoue. When cyberathletes conceal their game: Clustering confusion matrices to identify avatar aliases. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, 2015. doi:10.1109/DSAA.2015.7344824. URL https://doi.org/10.1109/DSAA.2015.7344824.

Olivier Cavadenti, Victor Codocedo, Jean-François Boulicaut, and Mehdi Kaytoue. What Did I Do Wrong in My MOBA Game? Mining Patterns Discriminating Deviant Behaviours. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 662–671, 2016. doi:10.1109/DSAA.2016.75. URL https://doi.org/10.1109/DSAA.2016.75.

Mark A. Chen, K. Spanton, P. van Schaik, I. Spears, and D. Eaves. The Effects of Biofeedback on Performance and Technique of the Boxing Jab. *Perceptual and Motor Skills*, 128(4): 1607–1622, 2021. doi:10.1177/00315125211013251. URL https://doi.org/10.1177/00315125211013251. PMID: 33940988.

Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pp. 785–794, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi:10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

Weisheng Chiu, Thomas Chun Man Fan, Sang-Back Nam, and Ping-Hung Sun. Knowledge Mapping and Sustainable Development of eSports Research: A Bibliometric and Visualized Analysis. *Sustainability*, 13(18), 2021. ISSN 2071-1050. doi:10.3390/su131810354. URL https://doi.org/10.3390/su131810354.

William Falcon and The PyTorch Lightning team. PyTorch Lightning, 3 2019. URL https://github.com/PyTorchLightning/pytorch-lightning.

Victor R. M. Feitosa, José G. R. Maia, Leonardo O. Moreira, and George A. M. Gomes. GameVis: Game Data Visualization for the Web. In *2015 14th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, pp. 70–79, 2015. doi:10.1109/SBGames.2015.21. URL https://doi.org/10.1109/SBGames.2015.21.

Mirko Gabelica, Ružica Bojčić, and Livia Puljak. Many researchers were not compliant with their published data sharing statement: mixed-methods study. *Journal of Clinical Epidemiology*, May 2022. ISSN 0895-4356. URL https://www.sciencedirect.com/science/article/pii/S089543562200141X.

Luca Galli, Daniele Loiacono, Luigi Cardamone, and Pier Luca Lanzi. A cheating detection framework for Unreal Tournament III: A machine learning approach. In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, pp. 266–272, 2011. doi:10.1109/CIG.2011.6032016.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé, and Kate Crawford. Datasheets for Datasets, 2018. URL https://arxiv.org/abs/1803.09010.

Maryam Ghasemaghaei. Does data analytics use improve firm decision making quality? The role of knowledge sharing and data analytics competency. *Decision Support Systems*, 120:14–24, 2019. ISSN 0167-9236. doi:10.1016/j.dss.2019.03.004. URL https://doi.org/10.1016/j.dss.2019.03.004.

Georgia Giblin, Elaine Tor, and Lucy Parrington. The impact of technology on elite sports performance. *Sensoria: A Journal of Mind, Brain & Culture*, 12, 11 2016. doi:10.7790/sa.v12i2.436.

Jonah Glass and Carolyn McGregor. Towards Player Health Analytics in Overwatch. In *2020 IEEE 8th International Conference on Serious Games and Applications for Health (SeGAH)*, pp. 1–5, 2020. doi:10.1109/SeGAH49190.2020.9201733. URL https://doi.org/10.1109/SeGAH49190.2020.9201733.

Daniel Gourdeau and Louis Archambault. Discriminative neural network for hero selection in professional Heroes of the Storm and DOTA 2. *IEEE Transactions on Games*, pp. 1–1, 2020. doi:10.1109/TG.2020.2972463. URL https://doi.org/10.1109/TG.2020.2972463.

C. S. Green and D. Bavelier. Learning, attentional control, and action video games. *Current biology : CB*, 22(6):R197–R206, mar 2012. ISSN 1879-0445. doi:10.1016/j.cub.2012.02.012. URL https://doi.org/10.1016/j.cub.2012.02.012.

C. Shawn Green and Daphne Bavelier. Action video game modifies visual selective attention. *Nature*, 423(6939):534–537, may 2003. ISSN 1476-4687. doi:10.1038/nature01647. URL https://doi.org/10.1038/nature01647.

Ian Helmke, Daniel Kreymer, and Karl Wiegand. Approximation Models of Combat in StarCraft 2, 2014. URL https://arxiv.org/abs/1403.1521.

Victoria Hodge, Sam Devlin, Nick Sephton, Florian Block, Anders Drachen, and Peter Cowling. Win Prediction in Esports: Mixed-Rank Match Prediction in Multi-player Online Battle Arena Games, 2017. URL https://arxiv.org/abs/1711.06498.

Victoria Hodge, Sam Devlin, Nick Sephton, Florian Block, Peter Cowling, and Anders Drachen. Win Prediction in Multi-Player Esports: Live Professional Match Prediction. *IEEE Transactions on Games*, pp. 1–1, 2019. doi:10.1109/TG.2019.2948469. URL https://doi.org/10.1109/TG.2019.2948469.

Max Jaderberg, Wojciech M. Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castañeda, Charles Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019. doi:10.1126/science.aau6249. URL https://doi.org/10.1126/science.aau6249.

Natalia Jakubowska, Paweł Dobrowolski, Alicja Anna Binkowska, Ibrahim V. Arslan, Monika Myśliwiec, and Aneta Brzezicka. Psychophysiological, but Not Behavioral, Indicator of Working Memory Capacity Predicts Video Game Proficiency. *Frontiers in Human Neuroscience*, 15, 2021. ISSN 1662-5161. doi:10.3389/fnhum.2021.763821. URL https://doi.org/10.3389/fnhum.2021.763821.

Aditya Jonnalagadda, Iuri Frosio, Seth Schneider, Morgan McGuire, and Joohwan Kim. Robust Vision-Based Cheat Detection in Competitive Gaming. *The Proceedings of the ACM in Computer Graphics and Interactive Techniques*, 4(1), apr 2021. doi:10.1145/3451259. URL https://doi.org/10.1145/3451259.

Nikita Khromov, Alexander Korotin, Andrey Lange, Anton Stepanov, Evgeny Burnaev, and Andrey Somov. Esports Athletes and Players: A Comparative Study. *IEEE Pervasive Computing*, 18(3): 31–39, 2019. doi:10.1109/MPRV.2019.2926247. URL https://doi.org/10.1109/MPRV.2019.2926247.

Denis Koposov, Maria Semenova, Andrey Somov, Andrey Lange, Anton Stepanov, and Evgeny Burnaev. Analysis of the Reaction Time of eSports Players through the Gaze Tracking and Personality Trait. In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, pp. 1560–1565, 2020. doi:10.1109/ISIE45063.2020.9152422. URL https://doi.org/10.1109/ISIE45063.2020.9152422.

Alexander Korotin, Nikita Khromov, Anton Stepanov, Andrey Lange, Evgeny Burnaev, and Andrey Somov. Towards Understanding of eSports Athletes' Potentialities: The Sensing System for Data Collection and Analysis. In *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 1804–1810, 2019. doi:10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00319. URL https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00319.

Anton Kos and Anton Umek. Smart sport equipment: SmartSki prototype for biofeedback applications in skiing. *Personal and Ubiquitous Computing*, 22, jun 2018. doi:10.1007/s00779-018-1146-1. URL https://doi.org/10.1007/s00779-018-1146-1.

Natalia Kowalczyk-Grębska, Feng Shi, Mikołaj Magnuski, Maciek Skorko, Pawel Dobrowolski, Bartosz Kossowski, Artur Marchewka, Maksymilian Bielecki, Malgorzata Kossut, and Aneta Brzezicka. Real-time strategy video game experience and structural connectivity - A diffusion tensor imaging study. *Human Brain Mapping*, 39, 06 2018. doi:10.1002/hbm.24208. URL https://doi.org/10.1002/hbm.24208.

Chan Min Lee and Chang Wook Ahn. Feature Extraction for StarCraft II League Prediction. *Electronics*, 10(8), 2021. ISSN 2079-9292. doi:10.3390/electronics10080909. URL https://doi.org/10.3390/electronics10080909.

Donghyeon Lee, Man-Je Kim, and Chang Wook Ahn. Predicting combat outcomes and optimizing armies in StarCraft II by deep learning. *Expert Systems with Applications*, 185:115592, 2021. ISSN 0957-4174. doi:10.1016/j.eswa.2021.115592. URL https://doi.org/10.1016/j.eswa.2021.115592.

Hojoon Lee, Dongyoon Hwang, Hyunseung Kim, Byungkun Lee, and Jaegul Choo. DraftRec: Personalized Draft Recommendation for Winning in Multi-Player Online Battle Arena Games. In *Proceedings of the ACM Web Conference 2022*, WWW '22, pp. 3428–3439, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi:10.1145/3485447.3512278. URL https://doi.org/10.1145/3485447.3512278.

Zeming Lin, Jonas Gehring, Vasil Khalidov, and Gabriel Synnaeve. STARDATA: A StarCraft AI Research Dataset. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 13(1):50–56, jun 2021. URL https://ojs.aaai.org/index.php/AIIDE/article/view/12929.

Liquipedia. Portal:Leagues. https://liquipedia.net/starcraft2/Portal:Leagues, 2010. Acessed: 2021-08-26.

Romain Mathonat, Jean-Francois Boulicaut, and Mehdi Kaytoue. A Behavioral Pattern Mining Approach to Model Player Skills in Rocket League. In *2020 IEEE Conference on Games (CoG)*, pp. 267–274, 2020. doi:10.1109/CoG47356.2020.9231739. URL https://doi.org/10.1109/CoG47356.2020.9231739.

Philip Z. Maymin. Smart kills and worthless deaths: eSports analytics for League of Legends. *Journal of Quantitative Analysis in Sports*, 17(1):11–27, 2021. doi:doi:10.1515/jqas-2019-0096. URL https://doi.org/10.1515/jqas-2019-0096.

L. McInnes, J. Healy, and J. Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, feb 2018.

Nikita Melentev, Andrey Somov, Evgeny Burnaev, Irina Strelnikova, Galina Strelnikova, Elizaveta Melenteva, and Alexander Menshchikov. eSports Players Professional Level and Tiredness Prediction using EEG and Machine Learning. In *2020 IEEE SENSORS*, pp. 1–4, 2020. doi:10.1109/SENSORS47125.2020.9278704. URL https://doi.org/10.1109/SENSORS47125.2020.9278704.

Yaser Norouzzadeh Ravari, Snader Bakkes, and Pieter Spronck. StarCraft Winner Prediction. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 12(2): 2–8, jun 2021. URL https://ojs.aaai.org/index.php/AIIDE/article/view/12887.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Alan Pedrassoli Chitayat, Athanasios Kokkinakis, Sagarika Patra, Simon Demediuk, Justus Robertson, Oluseji Olarewaju, Marian Ursu, Ben Kirmann, Jonathan Hook, Florian Block, and Anders Drachen. WARDS: Modelling the Worth of Vision in MOBA's. In Kohei Arai, Supriya Kapoor, and Rahul Bhatia (eds.), *Intelligent Computing*, pp. 63–81, Cham, 2020. Springer International Publishing. ISBN 978-3-030-52246-9. doi:10.1007/978-3-030-52246-9_5. URL https://doi.org/10.1007/978-3-030-52246-9_5.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Matevž Pustišek, Yu Wei, Yunchuan Sun, Anton Umek, and Anton Kos. The role of technology for accelerated motor learning in sport. *Personal and Ubiquitous Computing*, 08 2019. doi:10.1007/s00779-019-01274-5. URL `https://doi.org/10.1007/s00779-019-01274-5`.

Alen Rajšp and Iztok Fister jr. A Systematic Literature Review of Intelligent Data Analysis Methods for Smart Sport Training. *Applied Sciences*, 10, 04 2020. doi:10.3390/app10093013. URL `https://doi.org/10.3390/app10093013`.

Jason G. Reitman, Maria J. Anderson-Coto, Minerva Wu, Je Seok Lee, and Constance Steinkuehler. Esports Research: A Literature Review. *Games and Culture*, 15(1): 32–50, 2020. doi:10.1177/1555412019840892. URL `https://doi.org/10.1177/1555412019840892`.

Antonio A. Sánchez-Ruiz and Maximiliano Miranda. A machine learning approach to predict the winner in StarCraft based on influence maps. *Entertainment Computing*, 19:29–41, 2017. ISSN 18759521. doi:10.1016/j.entcom.2016.11.005. URL `https://doi.org/10.1016/j.entcom.2016.11.005`.

Tobias M. Scholz. *A Short History of eSports and Management*, pp. 17–41. Springer International Publishing, Cham, 2019. ISBN 978-3-030-11199-1. doi:10.1007/978-3-030-11199-1_2. URL `https://doi.org/10.1007/978-3-030-11199-1_2`.

Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa (eds.). *Game Analytics: Maximizing the Value of Player Data*. Springer London, London, 2013. ISBN 9781447147688 9781447147695. doi:10.1007/978-1-4471-4769-5. URL `https://doi.org/10.1007/978-1-4471-4769-5`.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018. doi:10.1126/science.aar6404. URL `https://doi.org/10.1126/science.aar6404`.

Anton Smerdov, Evgeny Burnaev, and Andrey Somov. eSports Pro-Players Behavior During the Game Events: Statistical Analysis of Data Obtained Using the Smart Chair. In *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pp. 1768–1775, 2019. doi:10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00314. URL `https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00314`.

Anton Smerdov, Bo Zhou, Paul Lukowicz, and Andrey Somov. Collection and Validation of Psychophysiological Data from Professional and Amateur Players: a Multimodal eSports Dataset, 2020. URL `https://arxiv.org/abs/2011.00958`.

Anton Smerdov, Andrey Somov, Evgeny Burnaev, Bo Zhou, and Paul Lukowicz. Detecting Video Game Player Burnout With the Use of Sensor Data and Machine Learning. *IEEE Internet of Things Journal*, 8(22):16680–16691, 2021. doi:10.1109/JIOT.2021.3074740. URL `https://doi.org/10.1109/JIOT.2021.3074740`.

Marius Stanescu, Nicolas Barriga, and Michael Buro. Using Lanchester Attrition Laws for Combat Prediction in StarCraft. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 11(1):86–92, Jun. 2021. URL `https://ojs.aaai.org/index.php/AIIDE/article/view/12780`.

Anton Stepanov, Andrey Lange, Nikita Khromov, Alexander Korotin, Evgeny Burnaev, and Andrey Somov. Sensors and Game Synchronization for Data Analysis in eSports. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pp. 933–938, 2019. doi:10.1109/INDIN41052.2019.8972249. URL https://doi.org/10.1109/INDIN41052.2019.8972249.

Yanhui Su, Per Backlund, and Henrik Engström. Comprehensive review and classification of game analytics. *Service Oriented Computing and Applications*, 15(2):141–156, Jun 2021. ISSN 1863-2394. doi:10.1007/s11761-020-00303-z. URL https://doi.org/10.1007/s11761-020-00303-z.

Joseph J. Thompson, M. Blair, Lihan Chen, and Andrew J. Henrey. Video Game Telemetry as a Critical Tool in the Study of Complex Skill Learning. *PLoS ONE*, 8, 2013. doi:10.1371/journal.pone.0075129. URL https://doi.org/10.1371/journal.pone.0075129.

Alberto Uriarte and Santiago Ontañón. Combat Models for RTS Games. *IEEE Transactions on Games*, 10(1):29–41, 2018. doi:10.1109/TCIAIG.2017.2669895. URL https://doi.org/10.1109/TCIAIG.2017.2669895.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P Agapiou, Max Jaderberg, Alexander S Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. ISSN 0028-0836. doi:10.1038/s41586-019-1724-z. URL https://doi.org/10.1038/s41586-019-1724-z.

Vanessa Volz, Mike Preuss, and Mathias Kirk Bonde. Towards Embodied StarCraft II Winner Prediction. In Tristan Cazenave, Abdallah Saffidine, and Nathan Sturtevant (eds.), *Computer Games*, pp. 3–22, Cham, 2019. Springer International Publishing. ISBN 978-3-030-24337-1.

Xiangjun Wang, Junxiao Song, Penghui Qi, Peng Peng, Zhenkun Tang, Wei Zhang, Weimin Li, Xiongjun Pi, Jujie He, Chao Gao, Haitao Long, and Quan Yuan. SCC: an efficient deep reinforcement learning agent mastering the game of StarCraft II. *CoRR*, abs/2012.13169, 2020. URL https://arxiv.org/abs/2012.13169.

Huikai Wu, Junge Zhang, and Kaiqi Huang. MSC: A Dataset for Macro-Management in StarCraft II, 2017. URL https://arxiv.org/abs/1710.03131.

Peter Xenopoulos, William Robert Freeman, and Claudio Silva. Analyzing the Differences between Professional and Amateur Esports through Win Probability. In *Proceedings of the ACM Web Conference 2022*, WWW '22, pp. 3418–3427, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450390965. doi:10.1145/3485447.3512277. URL https://doi.org/10.1145/3485447.3512277.

Anneke Zuiderwijk and Helen Spiers. Sharing and re-using open data: A case study of motivations in astrophysics. *International Journal of Information Management*, 49:228–241, 2019. ISSN 0268-4012. doi:10.1016/j.ijinfomgt.2019.05.024. URL https://doi.org/10.1016/j.ijinfomgt.2019.05.024.

A APPENDIX

A.1 BLIZZARD STARCRAFT II AI AND MACHINE LEARNING LICENSE

BLIZZARD® STARCRAFT® II AI AND MACHINE LEARNING LICENSE

IMPORTANT NOTICE:

YOU SHOULD CAREFULLY READ THIS AGREEMENT (THE "AGREEMENT") BEFORE INSTALLING OR USING BLIZZARD'S ("BLIZZARD") STARCRAFT II AI AND MACHINE LEARNING SOFTWARE AND ENVIRONMENT (THE "SOFTWARE"). IF YOU DO NOT AGREE WITH ALL OF THE TERMS OF THIS AGREEMENT, YOU MAY NOT INSTALL OR OTHERWISE ACCESS THE SOFTWARE.

Subject to the terms of this Agreement, your use of the Software is governed by Blizzard's End User License Agreement ("EULA"), which is incorporated by reference herein and is available for review here. (http://us.blizzard.com/en-us/company/legal/eula.html) Please carefully review the EULA and this Agreement prior to installing or using the Software. IF YOU DO NOT AGREE TO THE TERMS OF THE EULA AND THIS AGREEMENT, YOU ARE NOT PERMITTED TO INSTALL, COPY, OR USE THE SOFTWARE.

1. Use Of The Software.
   - A. AI Testing And Machine Learning Use Only: Subject to your compliance with this Agreement, Blizzard grants you a limited, revocable, non-sublicensable license to use the Software for purposes of AI testing, machine learning, and related research only.
   - B. Blizzard Account Not Required: Notwithstanding the requirements of Section 1.A of the EULA, creation of a Blizzard Account is not required in order to use the Software. Legal entities other than natural persons are authorized to use the Software. However, other than as specifically excepted in this Agreement, the remaining provisions and requirements of the EULA are controlling.
   - C. EULA Exceptions: The terms of Blizzard's EULA govern your use of the Software, subject to the following narrow exceptions:
     - i. Derivative Works: Section 1.C.i of the EULA shall not be read to prohibit the authorized use of the Software or data generated or collected from such use. However, no portion of this Agreement shall give you the right to create, distribute, or otherwise exploit unauthorized derivative works of the Software.
     - ii. Automation: The provisions of Section 1.C.ii of the EULA prohibiting the use of automation processes or software do not apply to use of the Software.
     - iii. Commercial Use: The provisions of Section 1.C.iii of the EULA govern your use of the Software, except that you are authorized to use and exploit data derived from using the Software in connection with AI and machine learning programs for personal or internal use, despite that such use of the data may ultimately be for a commercial purpose. You may not otherwise use or exploit the Software for any commercial purpose.
     - iv. Data Mining: The provisions of Section 1.C.iv of the EULA shall not prohibit the authorized use of the Software or data generated or collected from such use.
     - v. Matchmaking: The provisions of Section 1.C.vi of the EULA shall not prohibit the authorized use of the Software or data generated or collected from such use.
2. Ownership.
   - A. The provisions of Section 2 of the EULA apply in full force to the Software (including generated by or collected through the authorized use of the Software.

A.2 DATASHEETS FOR DATASETS

Datasheets for datasets (Gebru et al., 2018) are defined and available as a part of the original pre-processed dataset publication (REDACTED).

A.3 Additional Visualizations

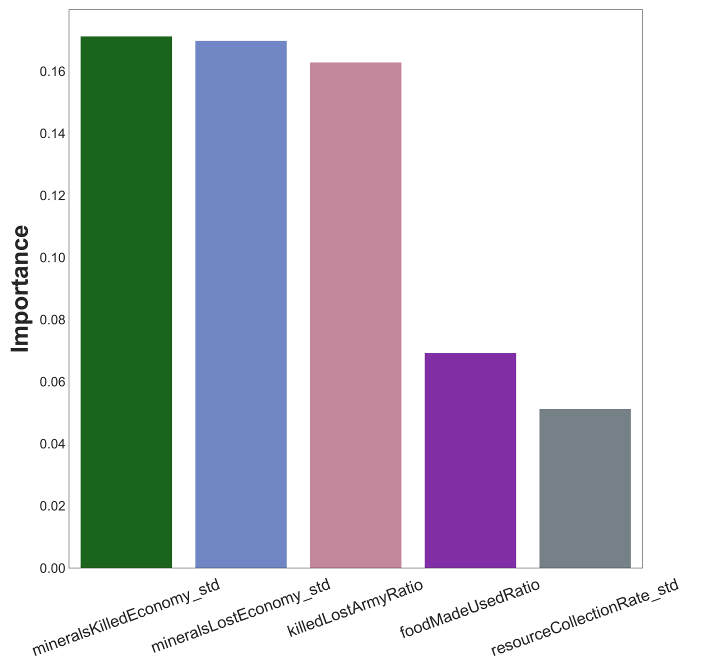

Figure 6: Percentages of feature importances based on XGBoost fit to all data.

A.4 Pipeline Visualizations

Due to the relative complexity of our infrastructure, we include additional visualizations of the processing pipeline for all potential users on Figures 7 - 9. The dataset pre-processing is shown on Figure 7, highlighting the use of a set of tools named SC2DatasetPreparator (REDACTED) . The dataset processing is shown on Figure 8, highlighting the use of SC2MapLocaleExtractor (REDACTED) to acquire the english map names, SC2InfoExtractorGo (REDACTED) to extract the data, and SC2DatasetPreparator (REDACTED) to collect the final .zip archives. The dataset post-processing and experiments are briefly visualized on Figure 9 and highlight the use of PyTorch (Paszke et al., 2019), and Lightning (Falcon & The PyTorch Lightning team, 2019).

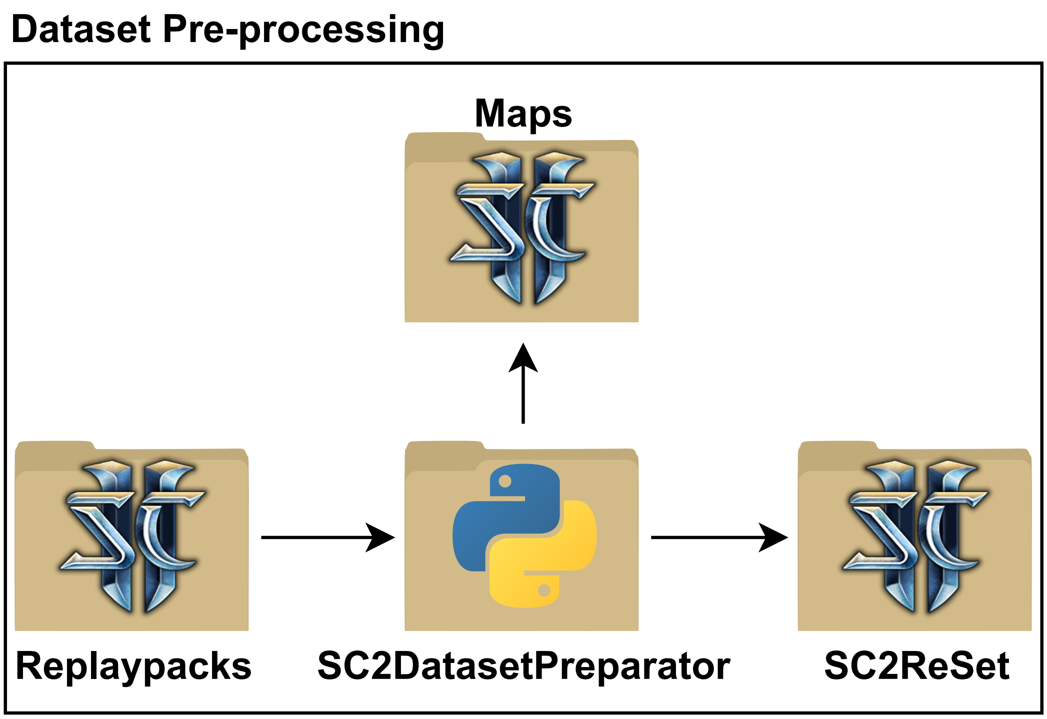

Figure 7: Pre-processing steps of our pipeline that result in SC2ReSet (REDACTED) . We are using a custom data processing toolset including SC2DatasetPreparator (REDACTED) .

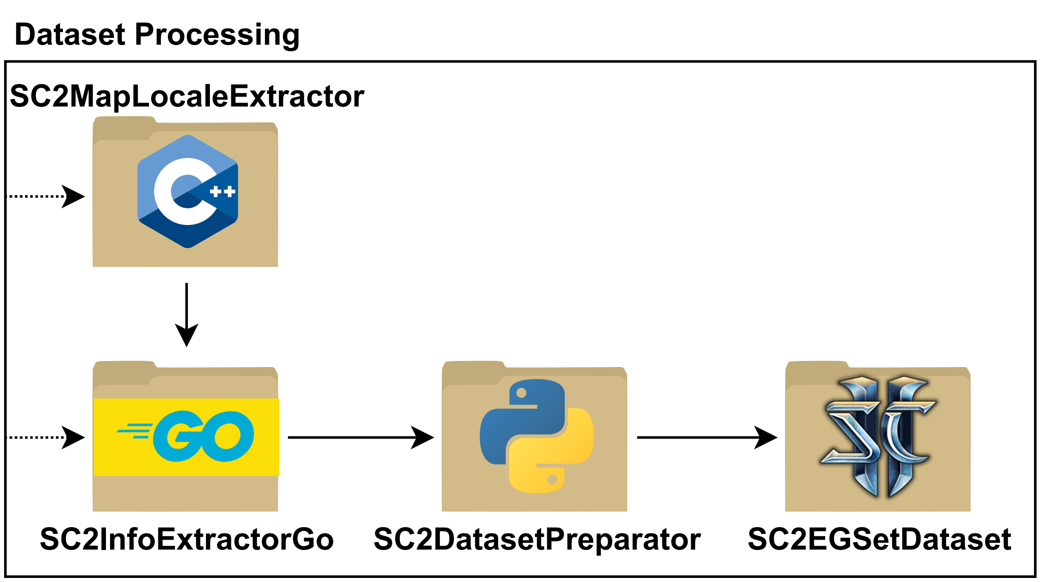

Figure 8: Processing steps of our pipeline that result in SC2EGSetDataset (REDACTED) . We are using a custom data processing toolset including SC2DatasetPreparator (REDACTED), SC2MapLocaleExtractor (REDACTED) , and SC2InfoExtractorGo (REDACTED) .

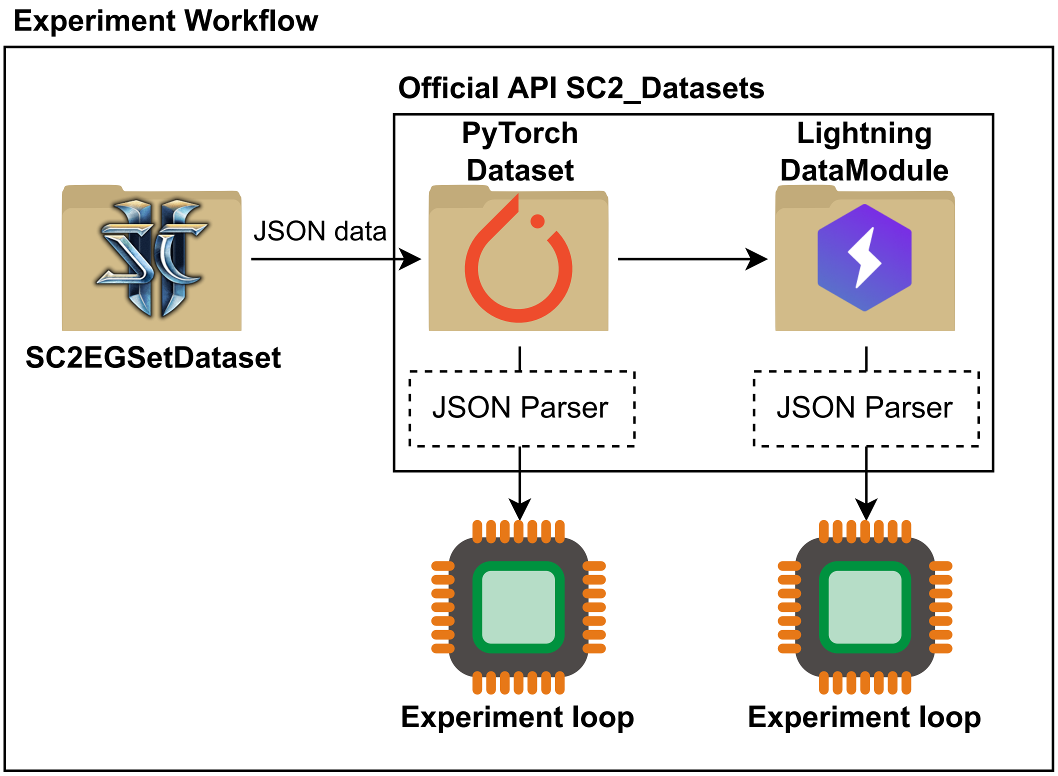

Figure 9: Using the SC2EGSetDataset (REDACTED) with the officially provided API (REDACTED) to conduct experiments.

A.5 Dataset Usage Examples

There are various ways to use our dataset; one way includes using the custom PyTorch dataset class which was briefly introduced in subsection 3.7, Listing 1. Due to the page limit we were unable to visualize all of the potential uses of our infrastructure in the main text; Listing 2 showcases the most basic use of the Lightning custom DataModule class that we implemented for our dataset. For further information please refer to the official documentation.

```python
from sc2_datasets.lightning.sc2_egset_datamodule import (
    SC2EGSetDataModule
)
from sc2_datasets.available_replaypacks import (
    EXAMPLE_SYNTHETIC_REPLAYPACKS
)

if __name__ == "__main__":
    # Initialize the datamodule:
    sc2_egset_datamodule = SC2EGSetDataModule(
        unpack_dir="./unpack_dir_path",
        download_dir="./download_dir_path",
        download=True,
        replaypacks=EXAMPLE_SYNTHETIC_REPLAYPACKS,
    )

    # Initializing the PyTorch dataset within the DataModule class:
    sc2_egset_datamodule.prepare_data()
    # Obtaining the splits for training, testing, and validation:
    sc2_egset_datamodule.setup()
```

Listing 2: Example use of the SC2EGSetDataModule with Lightning using a synthetic replaypack prepared for testing.

It is important to note that our classes by default return a custom SC2ReplayData class which is a serialization of the initial pre-processed JSON files. To construct a custom tensor required for further modeling, users should use the exposed keyword argument “transform”, which should be a function that transforms the default SC2ReplayData into some custom tensor required for further modeling.

A.6 DATASET STRUCTURE EXAMPLES

We include human-readable examples of various fields showcase on Listings 3-24; these are a part of the SC2EGSet dataset JSON files. Users should refer to the respective parts of the official documentation for more information and a list of all of the available fields. Access to these can be used to define parsers in other programming languages.

A.6.1 TOP LEVEL FIELDS

```json
{
    ...
    "header": {
        "elapsedGameLoops": 7855,
        "version": "3.4.0.44401"
    },
    ...
}
```

Listing 3: Example header field containing a JSON object.

```
{
    ...
    "initData": {
        "gameDescription": {
            "gameOptions": {
                "advancedSharedControl": false,
                "amm": false,
                "battleNet": true,
                "clientDebugFlags": 265,
                "competitive": false,
                "cooperative": false,
                "fog": 0,
                "heroDuplicatesAllowed": true,
                "lockTeams": true,
                "noVictoryOrDefeat": false,
                "observers": 0,
                "practice": false,
                "randomRaces": false,
                "teamsTogether": false,
                "userDifficulty": 0
            },
            "gameSpeed": "Faster",
            "isBlizzardMap": true,
            "mapAuthorName": "5-S2-1-1",
            "mapFileSyncChecksum": 360400735,
            "mapSizeX": 152,
            "mapSizeY": 152,
            "maxPlayers": 2
        }
    },
    ...
}
```


Listing 4: Example initData field containing a JSON object with nested information.


```
{
    ...
    "details": {
        "gameSpeed": "Faster",
        "isBlizzardMap": true,
        "timeUTC": "2016-07-29T04:50:12.5655603Z"
    },
    ...
}
```


Listing 5: Example details field containing a JSON object.


```
{
    ...
    "metadata": {
        "baseBuild": "",
        "dataBuild": "",
        "gameVersion": "",
        "mapName": "Galactic Process LE"
    },
    ...
}
```


Listing 6: Example metadata field containing a JSON object.

```
{
    ...
    "metadata": {
        "baseBuild": "",
        "dataBuild": "",
        "gameVersion": "",
        "mapName": "Galactic Process LE"
    },
    ...
}
```

Listing 7: Example metadata field containing a JSON object.

```
{
    ...
    "ToonPlayerDescMap": {
        "5-S2-1-7361539": {
            "nickname": "somePlayerNickname",
            "playerID": 2,
            "userID": 5,
            "SQ": 105,
            "supplyCappedPercent": 4,
            "startDir": 1,
            "startLocX": 127,
            "startLocY": 131,
            "race": "Zerg",
            "selectedRace": "",
            "APM": 0,
            "MMR": 0,
            "result": "Win",
            "region": "China",
            "realm": "China",
            "highestLeague": "Unknown",
            "isInClan": false,
            "clanTag": "",
            "handicap": 100,
            "color": {
            "a": 255,
            "b": 0,
            "g": 66,
            "r": 255
            }
        },
        "5-S2-1-7361634": {
            "nickname": "AnotherPlayerNickname",
            "playerID": 1,
            "userID": 1,
            "SQ": 115,
            "supplyCappedPercent": 7,
            "startDir": 7,
            "startLocX": 24,
            "startLocY": 20,
            "race": "Zerg",
            "selectedRace": "",
            "APM": 0,
            "MMR": 0,
            "result": "Loss",
            "region": "China",
            "realm": "China",
            "highestLeague": "Unknown",
            "isInClan": false,
            "clanTag": "",
            "handicap": 100,
            "color": {
            "a": 255,
            "b": 180,
            "g": 20,
            "r": 30
            }
        }
    }
    ...
}
```


Listing 8: Example ToonPlayerDescMap field containing a JSON object mapping player statistics to unique toon id.

A.6.2 GAME EVENTS

All of the game events that were recorded by the game engine are available in one of the fields named "gameEvents", all of the events that are available are presented in listings below.


```
[
    ...
    {
        "baseBuildNum": 44401,
        "buildNum": 44401,
        "cameraFollow": false,
        "debugPauseEnabled": false,
        "developmentCheatsEnabled": false,
        "evtTypeName": "UserOptions",
        "gameFullyDownloaded": true,
        "hotkeyProfile": "\u003ccustom\u003e",
        "id": 7,
        "isMapToMapTransition": false,
        "loop": 0,
        "multiplayerCheatsEnabled": false,
        "platformMac": false,
        "syncChecksummingEnabled": false,
        "testCheatsEnabled": false,
        "useGalaxyAsserts": false,
        "userid": {
                "userId": 0
            },
        "versionFlags": 0
        },
    ...
]
```


Listing 9: Example UserOptions game event JSON object.


```
[
    ...
    {
        "distance": null,
        "evtTypeName": "CameraUpdate",
        "follow": false,
        "id": 49,
        "loop": 2,
        "pitch": null,
        "reason": null,
        "target": {
            "x": 0.7109375,
            "y": 0.5469970703125
        },
        "userid": {
            "userId": 6
        },
        "yaw": null
    },
    ...
]
```


Listing 10: Example CameraUpdate game event JSON object.

```
[
    ...
    {
        "controlGroupId": 10,
        "delta": {
            "addSubgroups": [
            {
                "count": 1,
                "intraSubgroupPriority": 1,
                "subgroupPriority": 32,
                "unitLink": 108
            }
            ],
            "addUnitTags": [
            56885249
            ],
            "removeMask": {
            "None": null
            },
            "subgroupIndex": 0
        },
        "evtTypeName": "SelectionDelta",
        "id": 28,
        "loop": 12,
        "userid": {
            "userId": 5
        }
        },
    ...
]
```


Listing 11: Example SelectionDelta game event JSON object.


```
[
    ...
    {
        "abil": {
            "abilCmdData": null,
            "abilCmdIndex": 0,
            "abilLink": 188
        },
        "cmdFlags": 256,
        "data": {
            "None": null
        },
        "evtTypeName": "Cmd",
        "id": 27,
        "loop": 15,
        "otherUnit": null,
        "sequence": 1,
        "unitGroup": null,
        "userid": {
            "userId": 5
        }
    },
    ...
]
```


Listing 12: Example Cmd game event JSON object.

```
[
    ...
    {
        "evtTypeName": "CmdUpdateTargetUnit",
        "id": 105,
        "loop": 37,
        "target": {
            "snapshotControlPlayerId": 0,
            "snapshotPoint": {
            "x": 64.5,
            "y": 68.75,
            "z": 5.994140625
            },
            "snapshotUnitLink": 369,
            "snapshotUpkeepPlayerId": 0,
            "tag": 2883585,
            "targetUnitFlags": 111,
            "timer": 0
        },
        "userid": {
            "userId": 5
        }
    },
    ...
]
```


Listing 13: Example CmdUpdateTargetUnit game event JSON object.


```
[
    ...
    {
        "evtTypeName": "CommandManagerState",
        "id": 103,
        "loop": 37,
        "sequence": 3,
        "state": 1,
        "userid": {
            "userId": 5
        }
    },
    ...
]
```


Listing 14: Example CommandManagerState game event JSON object.

```
[
    ...
    {
        "controlGroupIndex": 1,
        "controlGroupUpdate": 2,
        "evtTypeName": "ControlGroupUpdate",
        "id": 29,
        "loop": 1639,
        "mask": {
            "None": null
        },
        "userid": {
            "userId": 1
        }
    },
    ...
]
```


Listing 15: Example ControlGroupUpdate game event JSON object.


```
[
    ...
    {
        "evtTypeName": "CmdUpdateTargetPoint",
        "id": 104,
        "loop": 2965,
        "target": {
            "x": 19.133056640625,
            "y": 26.369140625,
            "z": 5.73388671875
        },
        "userid": {
            "userId": 5
        }
    },
    ...
]
```


Listing 16: Example CmdUpdateTargetPoint game event JSON object.


```
[
    ...
    {
        "evtTypeName": "GameUserLeave",
        "id": 101,
        "leaveReason": 0,
        "loop": 7845,
        "userid": {
            "userId": 5
        }
    },
    ...
]
```


Listing 17: Example CmdUpdateTargetPoint game event JSON object.

A.6.3 TRACKER EVENTS

All of the game events that were recorded by the game engine are available in one of the fields named `"trackerEvents"`, all of the events that are available are presented in listings below.

```json
[
    ...
    {
        "evtTypeName": "PlayerSetup",
        "id": 9,
        "loop": 0,
        "playerId": 1,
        "slotId": 0,
        "type": 1,
        "userId": 1
    },
    ...
]
```

Listing 18: Example PlayerSetup tracker event JSON object.

```
[
    ...
    {
        "evtTypeName": "PlayerStats",
        "id": 0,
        "loop": 1,
        "playerId": 1,
        "stats": {
            "scoreValueFoodMade": 57344,
            "scoreValueFoodUsed": 49152,
            "scoreValueMineralsCollectionRate": 0,
            "scoreValueMineralsCurrent": 50,
            "scoreValueMineralsFriendlyFireArmy": 0,
            "scoreValueMineralsFriendlyFireEconomy": 0,
            "scoreValueMineralsFriendlyFireTechnology": 0,
            "scoreValueMineralsKilledArmy": 0,
            "scoreValueMineralsKilledEconomy": 0,
            "scoreValueMineralsKilledTechnology": 0,
            "scoreValueMineralsLostArmy": 0,
            "scoreValueMineralsLostEconomy": 0,
            "scoreValueMineralsLostTechnology": 0,
            "scoreValueMineralsUsedActiveForces": 0,
            "scoreValueMineralsUsedCurrentArmy": 0,
            "scoreValueMineralsUsedCurrentEconomy": 1050,
            "scoreValueMineralsUsedCurrentTechnology": 0,
            "scoreValueMineralsUsedInProgressArmy": 0,
            "scoreValueMineralsUsedInProgressEconomy": 0,
            "scoreValueMineralsUsedInProgressTechnology": 0,
            "scoreValueVespeneCollectionRate": 0,
            "scoreValueVespeneCurrent": 0,
            "scoreValueVespeneFriendlyFireArmy": 0,
            "scoreValueVespeneFriendlyFireEconomy": 0,
            "scoreValueVespeneFriendlyFireTechnology": 0,
            "scoreValueVespeneKilledArmy": 0,
            "scoreValueVespeneKilledEconomy": 0,
            "scoreValueVespeneKilledTechnology": 0,
            "scoreValueVespeneLostArmy": 0,
            "scoreValueVespeneLostEconomy": 0,
            "scoreValueVespeneLostTechnology": 0,
            "scoreValueVespeneUsedActiveForces": 0,
            "scoreValueVespeneUsedCurrentArmy": 0,
            "scoreValueVespeneUsedCurrentEconomy": 0,
            "scoreValueVespeneUsedCurrentTechnology": 0,
            "scoreValueVespeneUsedInProgressArmy": 0,
            "scoreValueVespeneUsedInProgressEconomy": 0,
            "scoreValueVespeneUsedInProgressTechnology": 0,
            "scoreValueWorkersActiveCount": 12
        }
    },
    ...
]
```


Listing 19: Example PlayerStats tracker event JSON object.

```
[
    ...
    {
        "evtTypeName": "UnitTypeChange",
        "id": 4,
        "loop": 15,
        "unitTagIndex": 218,
        "unitTagRecycle": 1,
        "unitTypeName": "Egg"
    },
    ...
]
```


Listing 20: Example UnitTypeChange tracker event JSON object.


```
[
    ...
    {
        "controlPlayerId": 1,
        "evtTypeName": "UnitBorn",
        "id": 1,
        "loop": 652,
        "unitTagIndex": 238,
        "unitTagRecycle": 1,
        "unitTypeName": "Drone",
        "upkeepPlayerId": 1,
        "x": 23,
        "y": 17
    },
    ...
]
```


Listing 21: Example UnitBorn tracker event JSON object.


```
[
    ...
    {
        "evtTypeName": "UnitTypeChange",
        "id": 4,
        "loop": 652,
        "unitTagIndex": 203,
        "unitTagRecycle": 1,
        "unitTypeName": "Larva"
    },
    ...
]
```


Listing 22: Example UnitTypeChange tracker event JSON object.

```
[
    ...
    {
        "evtTypeName": "UnitDied",
        "id": 2,
        "killerPlayerId": null,
        "killerUnitTagIndex": null,
        "killerUnitTagRecycle": null,
        "loop": 652,
        "unitTagIndex": 203,
        "unitTagRecycle": 1,
        "x": 23,
        "y": 17
    },
    ...
]
```


Listing 23: Example UnitDied tracker event JSON object.


```
[
    ...
    {
        "evtTypeName": "UnitPositions",
        "firstUnitIndex": 265,
        "id": 8,
        "items": [
            0,
            50,
            27,
            14,
            49,
            26,
            3,
            49,
            28,
            9,
            48,
            28,
            18,
            50,
            26,
            11,
            50,
            27,
            21,
            55,
            23
        ],
        "loop": 6000
    }
    ...
]
```


Listing 24: Example UnitPositions tracker event JSON object.