

# BattleNet: Capturing Advantageous Battlefield in RTS Games (Student Abstract)

**Donghyeon Lee, Man-Je Kim, Chang Wook Ahn**

Gwangju Institute of Science and Technology (GIST)  
 123, Cheomdangwagi-ro, Buk-gu, Gwangju, Republic of Korea  
 {cheetos, jaykim0104, cwan}@gist.ac.kr

### Abstract

In a real-time strategy (RTS) game, StarCraft II, players need to know the consequences before making a decision in combat. We propose a combat outcome predictor which utilizes terrain information as well as squad information. For training the model, we generated a StarCraft II combat dataset by simulating diverse and large-scale combat situations. The overall accuracy of our model was 89.7%. Our predictor can be integrated into the artificial intelligence agent for RTS games as a short-term decision-making module.

### Introduction

Real-time strategy (RTS) games attract the attention of artificial intelligence (AI) researchers. StarCraft II is an RTS game where players destroy opponents’ bases to win, with elaborate tactics and orders. Combat decision-making is difficult because an agent must consider factors such as the unit-compositions, the fog of war, and where to fight. It is important to anticipate the result of a battle between a player’s units and opponent’s units on various battlefields.

Preceding StarCraft II combat prediction models include the clustering army analyzer (Synnaeve and Bessière 2012), the global state evaluator (Erickson and Buro 2014), and the neural-network machine (Sánchez-Ruiz 2015). There is a StarCraft combat simulator, SparCraft (Churchill and Buro 2013), which efficiently simulates combats by simplifying unit collisions. Nonetheless, it is hard to simulate the effects of terrains accurately because the fog of war and ramp visions are not implemented. Uriarte and Ontañón (2018) evaluated the performance of combat models such as lifetime damage (LTD) model (Churchill, Saffidine, and Buro 2012), lanchester model (Stanescu, Barriga, and Buro 2015), decreasing/suspended damage per frame (DPF) model (Uriarte and Ontañón 2015). However, those combat models do not include factors of terrains.

Figure 1 shows the structure of our model, BattleNet, which has two submodels: SynergyNet (left) and TerrainNet (right). An input of SynergyNet is a squad combination, and an output is a squad synergy. Both input and output are a vector form. The input is represented as a squad vector

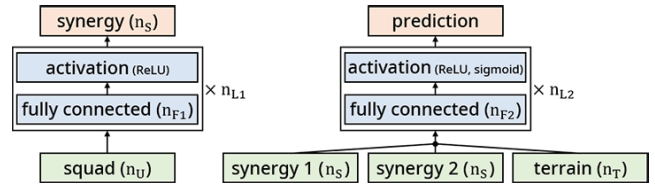


Figure 1: BattleNet, a combination of SynergyNet (left) and TerrainNet (right).

that each element represents the type of units in a squad, and the value of each element represents the number of the given unit type. The output vector represents a synergy of the given squad. TerrainNet takes the terrain information and the synergies of each confronting squads and gives a prediction of the combat outcome. The terrain information is a one-hot encoded vector. Two synergy vectors are made from SynergyNet. Then TerrainNet gives a prediction of the combat outcome, based on the two synergy vectors and a terrain vector.

### Experiment

For the experiment, we created a StarCraft II combat simulator which generates combat outcome dataset. The simulator generates two random squads, makes combat in a selected battlefield for five times, and records a winning rate of the combat. Equation (1) measures the winning rate. The number of the dataset is 15000 for each terrain. These datasets are separated into 60%, 20%, and 20% for the training set, validation set, and test set respectively.

$$p_{win} = \frac{n_{win} + 0.5 \times n_{draw}}{n_{win} + n_{draw} + n_{loss}} \quad (1)$$

Among the professional league maps of StarCraft II, we selected the categories of terrains that can represent most of the battlefields: plain, alley, narrow ramp, long alley, bush, two bushes, wide ramp, and foggy. A map can have several regions that fall into one of the categories. Plains have no special features such as obstacles, ramps, and alleys. Ramps have the high ground, the low ground, and a ramp that connects between them, while alleys have two grounds, and a

	Baseline-	Baseline+	BattleNet
Squad ( $n_U$ )	29	29	29
Synergy ( $n_S$ )	-	-	32
Terrain ( $n_T$ )	0	8	8
FC ( $n_{F(1,2)}$ )	512	512	256 / 512
#Layers ( $n_{L(1,2)}$ )	10	10	5 / 5

Table 1: Hyperparameters.  $n_U, n_S, n_T, n_{F(1,2)}$  are the number of features in one squad vector, one synergy vector, one terrain vector and one fully-connected layer, respectively.  $n_{L(1,2)}$  is the number of fully-connected layers.

narrow path between them. Bush-terrain is where units outside cannot see opponents’ units inside, but units inside can see both inside and outside. Foggy-terrain is similar to bush, except that units inside the fog have very small sight range, so they can only see units right next to themselves.

All moving ground units which can attack ground units were included in the simulation. Air units were excluded because their action is uninfluenced by the map’s terrain. Total 29 types (Zerg 8, Terran 11, Protoss 10) of units are used.

In order to simulate diverse large-scale combat situations, we made a restriction of maximum resources and number of unit types. The total resource of forming a squad should not exceed 10000 minerals, 5000 gases, and 150 supplies. The supply limit is 150, which is 50 less than a player’s maximum supply limit (200), assuming the player reserved 50 for workers. Assuming a one versus one game, all units in each squad have the same race. Unit compositions are determined by first assigning the total resource to unit types in the ratio of  $\mathbf{p}$ , where  $\mathbf{p} \sim \text{Dir}(\alpha)$  with  $\alpha = (1, 1, \dots, 1)$ , and then constructing a squad with the maximum number of units from the assigned resources.

## Results and Conclusion

The aforementioned experiment attributes were applied to three outcome prediction models (Baseline-, Baseline+, and BattleNet), which were trained and tested with the generated combat dataset for ten times. Baseline+ is a deep neural network (DNN) with squad and terrain information given, while Baseline- has squad information only. Hyperparameters are in Table 1. In 200 epoch of training, we used Adam optimizer and Binary Cross-Entropy loss with a learning rate =  $1e-4$  and a batch size = 1000. After 10 trainings and testings, the average accuracy of predicting combat outcomes on eight terrains is shown in Table 2. BattleNet showed the best accuracy regardless of battlefields, followed by Baseline+ and Baseline-. Additional terrain information increased the accuracy, considering Baseline+’s 9.2%p better overall accuracy than Baseline-’s. The overall accuracy of BattleNet is 89.7%, which is 2.0%p better than the Baseline+.

We presented a new combat outcome predictor module for StarCraft II. We generated StarCraft II combat dataset and proposed a model, BattleNet, which utilizes the battlefield information. On the dataset of large-scale combat simulation, the accuracy of BattleNet was 89.7%. With the terrain analysis tool, our combat outcome predictor can be im-

Battlefields	Baseline-	Baseline+	BattleNet
Plain	0.831	0.880	0.901
Alley	0.748	0.890	0.914
Narrow ramp	0.785	0.891	0.917
Long alley	0.758	0.879	0.905
Bush	0.782	0.870	0.890
Two bushes	0.789	0.820	0.830
Wide ramp	0.841	0.892	0.902
Foggy	0.778	0.894	0.918
Overall	0.789	0.877	0.897

Table 2: The average accuracy of models.

plemented as a module of the StarCraft II artificial intelligence agent, with additional decision-making modules for long-term problems such as operation and build order.

## Acknowledgments

This work was partially supported by the LG Electronics (LGE) grant, and IITP grant funded by the Korea government (MSIT) (No.2019-0-01842, AI Graduate School Support). The authors gratefully thank Dr. Kim, Sungjin and Mr. Kim, Junho for their helpful comments and supports.

## References

- Churchill, D., and Buro, M. 2013. Portfolio greedy search and simulation for large-scale combat in StarCraft. In *2013 IEEE Conference on Computational Intelligence in Games (CIG)*, 1–8.
- Churchill, D.; Saffidine, A.; and Buro, M. 2012. Fast heuristic search for RTS game combat scenarios. In *Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 112–117.
- Erickson, G., and Buro, M. 2014. Global state evaluation in StarCraft. In *Proceedings of the Tenth Annual AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 2014)*, 112–118.
- Sánchez-Ruiz, A. A. 2015. Predicting the outcome of small battles in StarCraft. In *Proceedings of the ICCBR 2015 Workshops.*, 33–42.
- Stanescu, M.; Barriga, N.; and Buro, M. 2015. Using lanchester attrition laws for combat prediction in StarCraft. In *Proceedings of the Eleventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-15)*, 86–92.
- Synnaeve, G., and Bessière, P. 2012. A dataset for StarCraft AI and an example of armies clustering. In *Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 25–30.
- Uriarte, A., and Ontañón, S. 2015. Automatic learning of combat models for RTS games. In *Proceedings of the Eleventh AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-15)*, 212–218.
- Uriarte, A., and Ontañón, S. 2018. Combat models for RTS games. *IEEE Transactions on Games* 10(1):29–41.