

기계학습 기반 다중 레이블 분류를 이용한 실시간 전략 게임에서의 상대 행동 예측

신승수¹, 조동희², 김용혁^{3*}

¹광운대학교 소프트웨어학부 학사과정, ²광운대학교 컴퓨터과학과 석사과정, ³광운대학교 소프트웨어학부 교수

Opponent Move Prediction of a Real-time Strategy Game Using a Multi-label Classification Based on Machine Learning

Seung-Soo Shin¹, Dong-Hee Cho², Yong-Hyuk Kim^{3*}

¹Bachelor Student, Dept. Computer Software, Kwangwoon University

²Master Student, Dept. Computer Science, Kwangwoon University

³Professor, Dept. Computer Software, Kwangwoon University

요약 최근 많은 게임이 사용자의 게임 플레이와 관련된 데이터를 제공하고 있고, 이에 기계학습 기법을 결합하여 상대의 행동을 예측하는 연구들이 있다. 본 연구는 실시간 전략 게임(클래시로얄)의 경기 데이터와 기계학습 기반의 다중 레이블 분류를 사용하여 상대 플레이어의 행동을 예측한다. 초기 실험은 이진 형태의 카드 특성과 카드 배치 좌표 그리고 정규화된 시간 정보를 입력받아 카드 타입, 카드 배치 좌표를 랜덤포레스트와 다층 퍼셉트론을 이용하여 예측한다. 이후, 순차적으로 3 가지 전처리 방식을 사용하여 실험을 진행했다. 먼저 입력 데이터의 특성 정보 일부를 변환시켜 예측했다. 다음으로 입력 데이터를 연속된 카드 입력 방식까지 고려한 중첩 형태로 변환시켜 예측했다. 마지막으로 모든 이전 단계의 데이터들을 정규화된 시간 기준에 따라 초반, 후반으로 분할하여 예측했다. 그 결과 가장 개선을 보인 전처리 방식은 중첩 형태의 데이터를 초반으로 분할하였을 경우로 카드 타입이 약 2.6%, 카드 배치 좌표가 약 1.8% 개선을 보였다.

주제어 : 기계학습, 실시간 전략 게임, 다중 레이블 분류, 전처리, 예측

Abstract Recently, many games provide data related to the users' game play, and there have been a few studies that predict opponent move by combining machine learning methods. This study predicts opponent move using match data of a real-time strategy game named ClashRoyale and a multi-label classification based on machine learning. In the initial experiment, binary card properties, binary card coordinates, and normalized time information are input, and card type and card coordinates are predicted using random forest and multi-layer perceptron. Subsequently, experiments were conducted sequentially using the next three data preprocessing methods. First, some property information of the input data were transformed. Next, input data were converted to nested form considering the consecutive card input system. Finally, input data were predicted by dividing into the early and the latter according to the normalized time information. As a result, the best preprocessing step was shown about 2.6% improvement in card type and about 1.8% improvement in card coordinates when nested data divided into the early.

Key Words : Machine learning, Real-time strategy game, Multi-label classification, Data preprocessing, Prediction

*This research was supported by the MIST(Ministry of Science and ICT), under the National Program for Excellence in SW(2017-0-00096), supervised by the IITP(Institute for Information & communications Technology Promotion)

*Corresponding Author : Yong-Hyuk Kim(yhdfly@kw.ac.kr)

Received September 11, 2020

Accepted October 20, 2020

Revised October 6, 2020

Published October 28, 2020

1. 서론

과거와는 다르게 특정 게임에선 사용자의 게임 플레이와 관련된 다양한 데이터를 제공하고 있다. 이에 맞춰, 최근 게임 산업에서는 관련 데이터와 기계학습 기법을 결합한 다양한 연구들이 진행되고 있다. 그중에서도 강화학습 기법과 게임 내적인 요소들을 예측하는 연구를 기반으로 유저와 직접 게임을 진행할 수 있는 지능적 에이전트(intelligent agent) 연구[1]가 주목받고 있다. 게임의 승패, 행동 예측은 지능적 에이전트 연구의 있어 중요한 문제이다.

게임 내 행동 예측의 경우 실시간 전략 게임에서는 자원, 유닛, 맵, 제한되지 않는 시간 등 다양한 변수가 존재하기 때문에 상대방의 행동을 예상하기 위해 고려해야 할 게임 경우의 수가 무한하다. 본 연구에서 다루는 클래시로얄(ClashRoyale)은 카드 기반 실시간 전략 게임으로 두 명의 사용자가 지도에 카드를 배치하여 서로의 요새의 공성을 목표로 한다. 기존 실시간 전략게임과 달리 상대적으로 좁은 크기의 지도(map), 제한된 유닛(unit)의 움직임, 제한된 경기 시간 등의 특징이 있어 행동 예측에 비교적 적합하다.

본 연구는 클래시로얄의 경기 기록 데이터와 기계학습 기반의 다중 레이블 분류(multi-label classification)를 사용하여 상대 플레이어의 행동을 예측하고 이를 비교한다. 예측을 위한 기계학습 기법으로는 랜덤 포레스트(random forest)와 다층 퍼셉트론(multi-layer perceptron)이 사용되었다. 이 과정에서 초기 데이터를 개선시킨 3 가지 전처리 방식을 순차적으로 설명하고 실험한다.

2 절에서는 게임의 경기 데이터를 사용하여 기계학습을 진행한 선행 연구들에 대해 정리하였다. 3 절에서는 경기 데이터 수집 과정, 그리고 초기 데이터와 이를 순차적으로 개선시킨 3 가지의 전처리 방식에 대해 설명한다. 4 절에서는 2 가지 기계학습 기법과 각각의 파라미터를 설명한다. 5 절에서는 3 절의 순차적인 전처리 방식에 따른 모델들의 실험 결과를 비교한다. 6 절에서는 결론 및 향후 연구에 대해 소개한다.

2. 관련연구

본 연구에서 다루는 게임과 유사하게 실시간으로 진행되는 게임 속 플레이어의 행동을 예측한 연구가 존재한

다. Dockhorn 등[2]의 연구에서는 하스스톤 게임 상황에 대한 점수를 부여하는 방식과 몬테 카를로 트리 탐색(monte carlo tree search)기법을 이용하여 역전파(back propagation)를 통해 가장 높은 점수가 나오는 상대방의 카드를 평가하고 예측하였다.

Baek 등[3]의 연구에서는 스타크래프트2 리플레이 데이터 1,200 개에 대해 시간적 정보를 고려하여 시점별 리플레이 이미지가 가지는 공간적 정보를 이용해서 승패를 예측할 수 있는 3 차원 합성곱 신경망 모델을 제안하였다.

Silver 등[4]의 연구에서는 심층 신경망과 몬테 카를로 트리 탐색을 이용하여 바둑 경기에 최적화된 지능적 에이전트를 높은 수준으로 향상시켰다.

3. 데이터

3.1 데이터 수집

본 연구를 진행하기 위해 두 가지 형식의 데이터를 크롤링(crawling)[5] 하였다. 첫째는, 클래시로얄 응용 프로그램 프로그래밍 인터페이스(application programming interface; API)¹⁾를 사용하여 2020.03.20-2020.04.13 기간에 해당하는 상위 100 명의 플레이어 전적 기록 100,000 개를 추출했다. 그리고 RoyaleAPI²⁾의 전적 기록 검색 기능을 통해 카드 배치 정보를 담고 있는 경기 데이터를 추출했다. 양 플레이어가 게임 중 사용한 카드의 합이 40 회 이상인 게임을 정상 게임으로 판단하여, 100,000 개의 전적 기록 중 49,633 개의 데이터를 사용하였다.

둘째는, 첫째에서 수집한 경기 데이터를 전처리 하기 위해 Deckshop³⁾에서 클래시로얄의 모든 카드에 대한 특성 정보와 수치 정보를 크롤링하였고, Table 1과 같이 표현하였다.

3.2 데이터 전처리


데이터 전처리는 3.1절에서 수집한 경기데이터와 Table 1 형태의 게임 내 모든 카드 정보를 이용하여 초기 데이터를 생성한다. 그리고 이를 순차적으로 3가지 전처리 방식을 사용하여 개선시켰다.

1) <https://developer.clashroyale.com>

2) <https://royaleapi.com/blog/replay>

3) <https://www.deckshop.pro>

Table 1. An example card information of ClashRoyale

	Card information			
	Name	Knight	Hitpoints	1,452
	ID	26000000	Damage	167
	Type	Troop	Hit speed	1.2
	Elixir	3	Range	0
	Card Property	Ground troop	Speed	1
		Mini tank	Count	1
		Melee medium	Tower damage	0
		High damage anti ground	Spell radius	1
		Barbarian barrel bait	Radius	0
		Lightning bait	Shield	0
		Royal delivery bait	Building damage	0
		Card played at the bridge	Duration	0

3.2.1 초기 방식

입력 데이터는 카드 특성, 카드 배치 좌표, 그리고 시간 정보 총 3 가지로 구성된다. 카드 특성은 “Card Property”라 지칭하며 총 45 가지로 해당 카드의 특성 보유 여부 따라 이진 형태의 다중 레이블로 표현하였다. 카드 배치 좌표는 “Location Info”라 지칭하며 Fig. 1과 같은 경기 맵에 대한 분할 기준에 따라 상하, 좌우, 진영에서의 상하, 다리 금방의 유무 4 가지를 평가하여 이진 형태의 다중 레이블로 표현하였다.

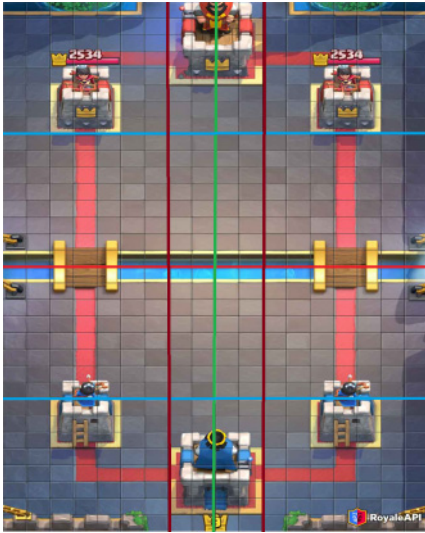


Fig. 1. Division rule map for converting card coordinates on ClashRoyale

시간 정보는 “Time Info”라 지칭하며 카드를 내는 순서를 정규화시킨 값이다. 출력 데이터는 Table 1에서의 카드 타입과 입력 데이터의 카드 배치 좌표를 이진 다중 레이블의 형식으로 구성하였다. 초기 방식의 실험 데이터는 Fig. 2로 표현되며 한 칸은 레이블의 명칭, 레이블의

형식, 레이블의 개수로 구성된다.

Card Property (Binary) 45	Location Info (Binary) 4	Time Info (Float) 1
------------------------------	-----------------------------	------------------------

Fig. 2. Format of initial input data

3.2.2 특성 정보 변환

3.2.1 절의 입력 데이터의 카드 특성 정보 중 19 개를 실수형 수치 정보가 담긴 12 개의 레이블로 변환시키고 이것을 “Card Ability”라 지칭한다. 이 과정에서 사용되는 수치 정보는 Table 1의 수치 정보가 사용된다. 해당 전처리 방식의 입력 데이터는 Fig. 3과 같이 구성된다.

Card Property (Binary) 24	Card Ability (Float) 12	Location Info (Binary) 4	Time Info (Float) 1
------------------------------	----------------------------	-----------------------------	------------------------

Fig. 3. Format of property converted input data

3.2.3 중첩 입력 방식

3.2.2 절의 데이터를 본 연구에서 다루는 게임의 특징으로 카드가 연속적일 수 있다는 것을 고려하여 다음과 같이 전처리 하였다. 기존의 1:1 카드 대응 방식과 더불어 연속된 카드 입력 방식인 N:1 방식을 고려하여 입력 데이터를 중첩하여 구성하였다. 해당 전처리 방식의 입력 데이터는 Fig. 4와 같이 구성된다.

Card Property	Card Ability	Location Info	Time Info
Card Property	Card Ability	Location Info	Time Info
Card Property	Card Ability	Location Info	Time Info

Fig. 4. Format of nested input data

3.2.4 시간 기준에 따른 분할

이전 단계의 모든 데이터들은 시간 정보 값 0.5를 기준으로 초반, 후반으로 나뉜다. 게임의 특성상 진영의 건물이 무너졌을 때 양 플레이어가 카드를 낼 수 있는 좌표 범위가 증가하며, 게임의 시간이 1 분 미만이 될 때 카드 배치 비용의 충전 시간이 단축된다. 이 점을 고려하여 경기의 초반과 후반의 플레이어 카드 배치 양상의 변화가 생길 것으로 판단하였다[3].

Fig. 5는 3.2.1 절부터 3.2.4 절에 해당하는 순차적인 전처리 방식의 순서도이다. 이 과정에서 각 단계마다 생성된 데이터를 알파벳에 해당하는 시나리오로 치환하였다.

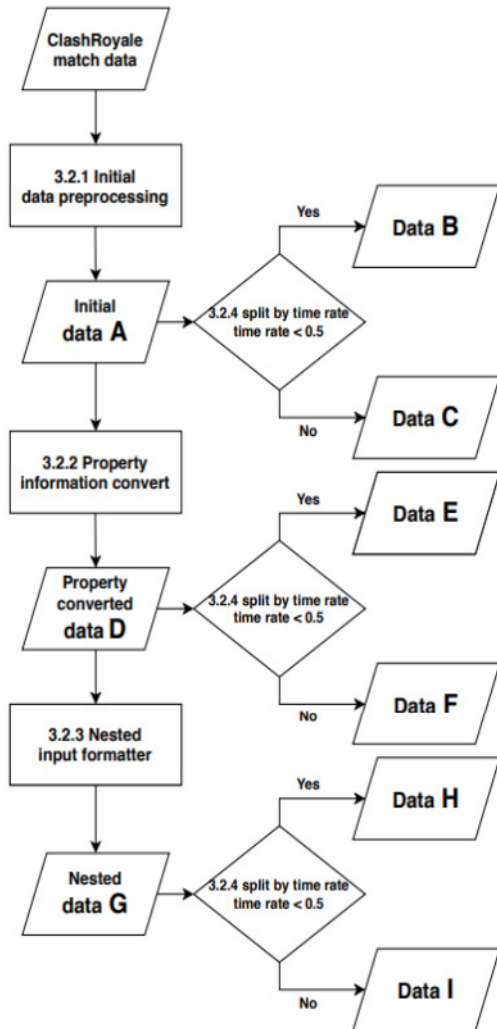


Fig. 5. Flow chart of data preprocessing & data scenarios

Table 2. Data scenarios for multi-label classification

Data scenario	The number of data	
	Train	Test
A	2,777,842	308,650
B	1,378,488	153,166
C	1,399,354	155,484
D	2,777,842	308,650
E	1,378,488	153,166
F	1,399,354	155,484
G	1,926,000	214,000
H	951,382	105,710
I	974,617	108,291

4. 기계학습 기법

실험에는 Fig. 5의 9 가지의 데이터 시나리오를 사용했다. 각 데이터 시나리오마다 1,000,000개 이상의 데이터가 존재하므로 최대한 많은 데이터를 학습에 사용하기 위해 각 데이터의 90%를 학습세트, 10%를 평가세트로 분할하였으며 Table 2와 같이 표현된다.

학습 데이터를 이용하여 다중 레이블 분류(multi-label classification)[6]를 진행하기 위해 해당 레이블의 부분 집합을 무작위로 표현한 새로운 단일 레이블로 분류하는 레이블 멍집합(label power-set)[7]방식을 사용한다. 단일 레이블을 분류하기 위한 기계학습 기법은 랜덤포레스트(random forest)[8]와 다층 퍼셉트론(multi-layer perceptron)[9]이 사용되었다.

랜덤포레스트는 다수의 의사결정트리[10]를 생성하고, 다수결로 결과를 결정하는 방법이다. 의사결정트리의 개수는 100 개로 구성되어있으며, 각 의사결정트리의 노드(node) 분할기준은 트리의 유사성을 피하기 위해 지니(Gini) 계수[11]를 사용하였다.

다층 퍼셉트론은 100개의 노드로 구성된 1개의 은닉층(hidden layer)[12]을 가지고 있으며, 활성화 함수로는 정류된 선형 유닛(rectified linear unit; ReLu)[13]을 사용했다. 출력층(output layer)은 이진 형태의 값을 반환하며 카드 타입, 카드 배치 정보에 대해서 각각 3, 4개의 노드로 구성된다. 손실 값 최적화를 위해 사용하는 함수는 아담(Adam)[14]함수를 채택하였으며, 에포크(epoch) 수는 20, 배치 크기(batch size)는 200이다.

Table 3. Results of multi-label classification using machine learning methods

Training Model											
Experiment using random forest						Experiment using multi-layer Perceptron					
Output label	Data scenario	Train		Test		Output label	Data scenario	Train		Test	
		Accuracy	F1-score	Accuracy	F1-score			Accuracy	F1-score	Accuracy	F1-score
Card type	A	0.84	0.84	0.58	0.58	Card type	A	0.68	0.68	0.68	0.68
	B	0.84	0.84	0.60	0.60		B	0.70	0.70	0.70	0.70
	C	0.83	0.83	0.56	0.56		C	0.65	0.65	0.65	0.65
	D	0.84	0.84	0.58	0.58		D	0.68	0.68	0.68	0.68
	E	0.84	0.84	0.60	0.60		E	0.70	0.70	0.70	0.70
	F	0.83	0.83	0.56	0.56		F	0.65	0.65	0.65	0.65
	G	0.93	0.93	0.58	0.58		G	0.68	0.68	0.68	0.68
	H	0.93	0.93	0.60	0.60		H	0.70	0.70	0.71	0.71
	I	0.93	0.93	0.56	0.56		I	0.68	0.66	0.65	0.65
Coord	A	0.69	0.87	0.21	0.67	Coord	A	0.32	0.75	0.32	0.75
	B	0.69	0.87	0.23	0.67		B	0.34	0.75	0.34	0.75
	C	0.68	0.87	0.20	0.67		C	0.30	0.74	0.30	0.75
	D	0.69	0.88	0.21	0.67		D	0.31	0.75	0.31	0.76
	E	0.69	0.87	0.23	0.67		E	0.33	0.77	0.33	0.75
	F	0.68	0.87	0.20	0.67		F	0.30	0.76	0.30	0.76
	G	0.86	0.94	0.22	0.67		G	0.32	0.75	0.32	0.75
	H	0.93	0.93	0.24	0.68		H	0.34	0.74	0.34	0.74
	I	0.86	0.94	0.20	0.67		I	0.31	0.76	0.31	0.76

5. 실험 결과

Fig. 5의 데이터 시나리오를 4 절의 기계학습 기법으로 상대방의 카드 타입과, 카드 배치 좌표를 각각 예측한 학습 및 평가 결과는 Table 3과 같다. 학습 및 평가 지표는 정확도와 F1점수(F1-score)[15]를 사용하였으며 다중 레이블 이진 분류 시 k개의 레이블이 존재 할 때 F1점수는 식 (1)과 같이 표현된다.

$$\begin{aligned}
 \text{Micro-}PRE &= \frac{TP_1 + \dots + TP_k}{TP_1 + \dots + TP_k + FP_1 + \dots + FP_k} \\
 \text{Micro-}REC &= \frac{TP_1 + \dots + TP_k}{TP_1 + \dots + TP_k + FN_1 + \dots + FN_k} \\
 \text{Micro-}F_1 &= 2 \cdot \frac{\text{Micro-}PRE \cdot \text{Micro-}REC}{\text{Micro-}PRE + \text{Micro-}REC}
 \end{aligned} \quad (1)$$

기계학습 기법에 따른 평가세트의 정확도와 F1점수는 다층 퍼셉트론을 사용했을 때 모든 데이터 시나리오에서 우위를 나타냈지만, 학습세트의 정확도와 F1점수는 모든 데이터 시나리오에서 랜덤 포레스트가 우위를 보였다. 순차적 전처리 방식별 성능 차이를 비교해보면, 초기 실험과 특성 정보 변환에 대해서는 성능 차이를 크게 보이지 못했다. 다음으로 연속된 카드 입력에 대한 중첩 처리를 해주었을 때는 평가세트의 평균 정확도는 랜덤포레

스트에서 약 0.4% 다층 퍼셉트론에서 약 0.8% 향상되었음을 보였다. 마지막으로 이전 단계의 모든 데이터 시나리오들을 정규화된 시간 기준에 따라 초반과 후반으로 분할한 방식에서는 이전 단계의 모든 데이터 시나리오에 대해 초반에서만 성능 개선이 일어났다.

가장 높은 성능 개선을 보인 조합은 중첩 입력 형태의 데이터를 시간에 따라 초반으로 분할한 Fig. 5의 H데이터와 다층 퍼셉트론에 적용한 경우이며, 이 때 개선 전 단계의 Fig. 5의 G 데이터 보다 카드 타입이 약 2.6% 카드 배치 좌표가 약 1.8% 개선되었다.

6. 결론 및 향후 연구

본 논문에서는 실시간 전략게임(클래시로얄)의 경기 데이터와 기계학습 기반의 다중 레이블 분류를 사용하여 상대 플레이어의 행동을 예측했다. 이 과정에서 순차적인 3 가지 전처리 방식을 제안하고 이들의 성능을 비교 하였다. 그 결과 3 가지 전처리 방식 중 마지막 단계에서 중첩 입력 형태의 데이터를 시간에 따라 초반으로 분할한 경우 가장 높은 성능 개선을 보였다. 이로서 상대방의 행동 예측에 대해서 게임의 시점이 영향을 끼치는 요소라는 것을 알 수 있었다.

하지만 본 연구에서 사용된 경기 데이터는 경기의 배치 정보만을 알 수 있었으며, 해당 배치 시간의 게임 상황을 반영하지 못한다는 한계가 있다. 경기의 배치 기록이 아닌 경기의 리플레이 데이터를 사용한다면, 추후의 연구에서는 기존의 행동을 예측하는 연구[16]처럼 높은 성능의 행동 예측 연구가 가능할 것이다.

REFERENCES

- [1] K. Arulkumaran, A. Cully & J. Togelius. (2019). Alphastar: an evolutionary computation perspective. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. (pp. 314-315). Prague : ACM.
DOI : 10.1145/3319619.3321894
- [2] A. Dockhorn, M. Frick, Ü. Akkaya & R. Kruse. (2018, May). Predicting opponent moves for improving hearthstone AI. *Information Processing and Management of Uncertainty in Knowledge-based Systems. Theory and Foundations* (pp. 621-632). Cadiz : Springer.
DOI : 10.1007/978-3-319-91476-3_51
- [3] I. S. Baek et al. (2020). Real-time strategy game win-lose prediction model using 3d convolutional neural networks. *Journal of The Korean Institute of Industrial Engineers*, 46(4), 349-355.
DOI : 10.7232/JKIE.2020.46.4.349
- [4] D. Silver et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529, 484-489.
DOI : 10.1038/nature16961
- [5] C. Olston & M. Najork. (2010). Web crawling. *Foundations and Trends in Information Retrieval*, 4(3), 175-246.
- [6] T. Grigorios & L. Katakis. (2007). Multi-label classification: an overview. *International Journal of Data Warehousing and Mining*, 3(3), 1-13.
- [7] S. L. Enrique, C. Bielza, E. F. Morales, P. Hernandez-Leal, J. H. Zaragoz & P. Larrañaga. (2014). Multi-label classification with bayesian network-based chain classifiers. *Pattern Recognition Letters*, 41, 14-22.
- [8] L. Andy & M. Wiener. (2002). Classification and regression by randomforest. *R News*, 2(3), 18-22.
- [9] M. Riedmiller. (1994). Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms. *Computer Standards & Interfaces*, 16(3), 265-278.
- [10] S. R. Safavian & D. Landgrebe. (1991). A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3), 660-674.
- [11] L. Ceriani & P. Verme. (2012). The origins of the gini index: extracts from *variabilità e mutabilità* (1912) by corrado gini. *The Journal of Economic Inequality*, 10(3), 421-443.
DOI : 10.1007/s10888-011-9188-x
- [12] T. Hastie, R. Tibshirani & J. Friedman. (2009). *The elements of statistical learning: data mining, inference, and prediction*. New york : Springer.
- [13] V. Nair & G. E. Hinton. (2010). Rectified linear units improve restricted Boltzmann machines. *International Conference on Machine Learning*. (pp. 807-814). Haifa : Omnipress.
- [14] D. P. Kingma & J. Ba. (2014). Adam: a method for stochastic optimization. *3rd International Conference for Learning Representation*. San Diego : ICLR.
- [15] F. Akinori, I. Hideki & J. Suzuki. (2008). Multi-label text categorization with model combination based on f1-score maximization. *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*. (pp 823-828).
- [16] Y. W. Nam & Y. H. Kim. (2018). Prediction of drifter trajectory using evolutionary computation. *Discrete Dynamics in Nature and Society* 2018.
DOI : 10.1155/2018/6848745

신 승 수(Seung-Soo Shin)

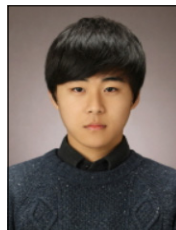
[학생회원]



- 2016년 3월 ~ 현재 : 광운대학교 소프트웨어학부 학사과정
- 관심분야 : 인공지능, 유전알고리즘, 기계학습
- E-Mail : tmd0707@naver.com

조 동 희(Dong-Hee Cho)

[학생회원]



- 2020년 8월 : 광운대학교 컴퓨터소프트웨어학부 (공학사)
- 2020년 9월 ~ 현재 : 광운대학교 컴퓨터과학과 석사과정
- 관심분야 : 인공지능, 유전알고리즘, 기계학습
- E-Mail : whehd16@naver.com

김 용 혁(Yong-Hyuk Kim)

[정회원]



- 1999년 2월 : 서울대학교 전산과학사 (이학사)
- 2001년 2월 : 서울대학교 전기컴퓨터 공학부(공학석사)
- 2005년 2월 : 서울대학교 전기컴퓨터 공학부(공학박사)
- 2005년 3월 ~ 2007년 2월 : 서울대학교 반도체공동연구소 연구원
- 2007년 3월 ~ 2017년 2월 : 광운대학교 컴퓨터소프트웨어 학과 조교수/부교수
- 2017년 3월 ~ 현재 : 광운대학교 소프트웨어학부 교수
- 관심분야 : 최적화, 진화연산, 지식공학
- E-Mail : yhdfly@kw.ac.kr