JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지 Vol. 23, No. 5: 533~539, May 2019

임의 차원 데이터 대응 Dynamic RNN-CNN 멀웨어 분류기

임근영[1] · 조영복[2*]

Dynamic RNN-CNN malware classifier correspond with Random Dimension Input Data

Geun-Young Lim[1] · Young-Bok Cho[2*]

[1]Undergraduate student, Department of Information Security, Daejeon University, Daejeon, 34520 Korea
[2*]Associate Professor, Department of Information Security, Daejeon University, Daejeon, 34520 Korea

요 약

본 연구는 본 연구는 Microsoft Malware Classification Challenge 데이터 셋을 사용해 임의의 길이 입력 데이터에 대응할 수 있는 멀웨어 분류 모델을 제안한다. 우리는 기존 연구의 멜웨어 데이터를 이미지화 시키는 것을 기반으로 한다. 제안 모델은 멀웨어 데이터가 큰 경우는 많은 이미지를 생성하고, 작은 데이터는 적은 이미지를 생성한다. 생성된 이미지를 시계열 데이터로 Dynamic RNN으로 학습시킨다. RNN의 출력 값은 Attention 기법을 응용해 가장 가중치가 높은 출력만 사용하고, RNN 출력값을 다시 Residual CNN으로 학습시켜 최종적으로 멀웨어를 분류한다. 제안 모델을 실험한 결과 검증 데이터 셋에서 Micro-average F1 score 92%를 기록하였다. 실험 결과 특별한 특징 추출 및 차원 축소 없이 임의 길이의 데이터를 학습 및 분류할 수 있는 모델의 성능을 검증할 수 있었다.

ABSTRACT

This study proposes a malware classification model that can handle arbitrary length input data using the Microsoft Malware Classification Challenge dataset. We are based on imaging existing data from malware. The proposed model generates a lot of images when malware data is large, and generates a small image of small data. The generated image is learned as time series data by Dynamic RNN. The output value of the RNN is classified into malware by using only the highest weighted output by applying the Attention technique, and learning the RNN output value by Residual CNN again. Experiments on the proposed model showed a Micro-average F1 score of 92% in the validation data set. Experimental results show that the performance of a model capable of learning and classifying arbitrary length data can be verified without special feature extraction and dimension reduction.

키워드 : RNN, CNN, 멜웨어, 딥러닝, 마이크로 평균 F1 점수

Keywords : RNN, CNN, malware, Deep-learning, Micro-average F1 score

Received 17 December 2018, **Revised** 12 April 2019, **Accepted** 25 April 2019

*** Corresponding Author** Young-Bok Cho(E-mail:ybcho@dju.ac.kr, Tel:+82-42-280-2406)
Associate Professor, Department of Infromation Security, Daejeon University, Daejeon, 34520 Korea

Open Access http://doi.org/10.6109/jkiice.2019.23.5.533 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

인터넷이 발달하고 데이터 전송과 처리가 빨라짐에 따라 멀웨어 또한 급속도로 진화하였다. 기존 멀웨어 탐지나 분류의 경우 파일의 시그니처를 통하여 이루졌다. 하지만 쏟아지는 변종 멀웨어를 대응하기엔 파일 시그니처 방법은 신속하지 못한 것에 아쉬움이 있다. 따라서 최근 머신러닝/딥러닝을 이용하여 멀웨어 탐지 및 분류하는 방법에 대한 많은 연구가 진행되고 있다. 2015년 머신러닝/딥러닝 데이터 분석 경쟁 플랫폼 Kaggle에서 Microsoft Malware Dataset이 공개되고 이를 사용한 멀웨어 분류 문제 대회가 진행되었다[1,2,3]. 해당 경쟁의 상위권 팀은 멀웨어를 고정된 이미지로 변환하여 CNN 모델로 분류함으로써 높은 정확도를 기록하였다. 하지만 멀웨어는 고정된 크기를 가지고 있지 않기 때문에 고정된 이미지로 멀웨어를 변환하는 과정은 필히 데이터의 축소 혹은 확장이 필요하며 이는 의도하지 않은 잠재적 데이터의 손실이 발생할 수 있다[4,5]. 또한 축소・확장하는 방법에 따라 추가적인 시간이 걸리며 작업 또한 상대적으로 복잡해진다. 본 논문은 위와 같은 단점을 극복하고자 다양한 크기를 가지는 멀웨어 데이터를 특별한 데이터 축소 없이 학습하고 분류할 수 있는 모델을 제안한다.

Ⅱ. 관련연구

Window NT 계열 운영체제 기준 실행파일에서 호출 API, API 호출 시퀀스 추출과 실행파일 이미지화를 통해 머신러닝/딥러닝 모델을 학습시키는 연구들이 진행되었다[3,6,7,8]. 본 논문에서 제시하는 모델은 이들 중 이미지화 방법을 기반으로 하고 CNN 뿐만 아니라 ConvLSTM과 Attention Mechanism을 사용한다.

2.1. 악성코드 이미지화

멀웨어 바이너리는 바이트 단위로 그레이 스케일 이미지로 시각화 시킬 수 있다[7,8]. 많은 멀웨어 변종의 경우 동일한 패밀리에 속한 멀웨어들의 경우 이미지화 했을 경우 레이아웃이나 텍스쳐 면에서 매우 유사하게 나타난다. 해당연구에서는 멀웨어를 시각화 했을 경우의 나타나는 텍스쳐를 연구하여 Feature Vector를 생성한다. 본 논문에서 제안하는 모델은 해당 연구의 멀웨어 이미지화의 장점은 가져오면서, 복잡한 Feature 분석 및 처리가 필요하지 않다.

2.2. ConvLSTM

입력과 예측 대상이 모두 시공간 순서로 된 문제를 풀기 위해 고안된 모델이다. 해당 연구의 경우 강수량을 예측 문제를 근사하는 모델에 ConvLSTM을 제안했다. 입력 – 상태 및 상태 – 상태 전환 모두에서 Convolution 구조를 갖도록 Fully Connected LSTM(FC-LSTM)을 확장히여 Convolution LSTM을 제안하였다. 본 논문에서는 멀웨어를 이미지화하여 분류 문제를 접근한다. ConvLSTM에서 이미지의 시공간처리를 본 논문에서는 가변 길이의 데이터 차원을 대응하는 방면으로 활용할 것이다.

2.3. Attention Mechanism

Attention Mechanism은 RNN 네트워크의 장기 의존성 문제를 극복하기 위한 하나의 수단으로 사용되며 시퀀스 입력의 출력 값들의 가중치를 따져 특정 출력 값의 집중(Attention) 가능 하게 한다[9,10]. Attention Mechanism은 NLP 문제를 푸는데 효과적인 것이 증명이 되었다. 본 연구는 가변 차원 데이터를 Encoder를 통하여 고정된 차원의 데이터로 출력하는데 Attention Mechanism을 응용한다. 멀웨어는 프로그램이 하는 악성 행동에 따라 그 종류가 정해진다. 멀웨어가 하는 행동은 일부의 코드에서 발생할 것이라는 추측을 통해, ConvLSTM 모델의 출력 시퀀스에 Attention Mechanism을 응용한다.

2.4. 멀웨어 좌표 기반 이미지화

멀웨어는 해당 연구는 바이너리들의 유사성을 측정하기 한 방법을 제안한다. 각 바이리에 해당하는 고정된 크기의 이미지를 생성하고, 이미지를 특징정보로 이용하여 악성코드들을 분류한다[10,11,12]. 해당 연구에서 멀웨어의 바이너리 데이터를 두 개의 1바이트 좌표로 활용하여 256 × 256 고정된 이미지를 생성한다. 1바이트 씩 이동하며 좌표를 읽어 해당하는 픽셀 좌표의 값을 1증가시킨다. 해당 연구는 본 논문에서 제시하는 모델과 가변 입력 데이터에 대응 할 수 있다는 점에서 비슷하다. 하지만 1바이트씩 이동하면서 좌표를 읽는 전처리 방식은 상대적으로 많은 전처리 연산이 필요하다. 본 논문에서 제시하는 모델은 상대적으로 적은 연산의 전처리로 멀웨어를 분류할 수 있는 모델을 제안한다.

Ⅲ. Dynamic RNN-CNN

본 논문에서는 실험을 위한 절차는 다음과 같다. 멀웨어 데이터 셋의 문자열 바이너리 덤프 데이터를 읽어 바이너리로 복구 한다. 그 다음 바이너리를 용량만큼의 256 × 256 사이즈의 이미지 여러 장으로 변환한 뒤, 모델을 학습시킨다.

3.1. 데이터 셋

머신러닝/딥러닝 데이터를 공유하고 공유 된 데이터를 두고 경쟁하는 Kaggle에서는 멀웨어 분류를 위한 Microsoft Malware Classification의 데이터가 공유되었다. 본 연구에서는 이 멀웨어 데이터를 사용하여 멀웨어 분류 딥러닝 모델을 학습시켰다. 멀웨어 학습 데이터는 10,868개의 멀웨어 분석 파일들로 이루어져 있다. 데이터 셋은 멀웨어 당 2개의 포맷으로 나뉘어져 있다. 16진법 표현으로 멀웨어 바이너리 바이트들을 스트링 포맷으로 기록한 bytes 파일과, 바이너리 파일에서 IDA disassembler로 추출 된 각종 메타 정보가 기록된 파일인 asm 파일이 있다. 제안 모델을 위하여 8:2비율로 학습데이터와 검증데이터를 나누었다. 본 논문에서 제안하는 모델은 멀웨어의 메타 정보는 사용하지 않고 순수하게 멀웨어 바이너리를 이미지로 입력받는 모델이다. 따라서 ida 포맷의 데이터 셋은 사용하지 않았다. asm 파일을 읽어 멀웨어 바이너리로 복구한 뒤, 복구한 바이너리 파일로 실험을 진행하였다. 용량이 6MB가 넘는 이상치 데이터 2개는 데이터 셋에서 제외하였다. 학습과 검증 데이터 셋의 클래스별 데이터 샘플 개수는 표1과 같다.

Table. 1 Number of training and validation data for each classes

Class	Train ing data	Validation data
Ramnit	1231	308
Lollipop	1982	496
Kelihos_ver3	2354	588
Vundo	380	95
Simda	34	8
Tracur	601	150
Kelihos_ver1	318	80
Obfuscator.ACY	982	246
Gatak	810	203
Total	8692	2174

3.2. 실험 환경

실험을 위한 프로그래밍 언어 환경은 Python 3.6.2를 사용해 구현하였고, Dynamic RNN-CNN 모델을 구현을 위해 Tensorflow-gpu 1.9 버전을 사용하였다. 또한 악성코드 데이터 처리를 위하여 numpy 1.15.0과 pandas 0.23.4 버전을 사용하였다. 실험에 사용된 컴퓨터의 사양은 표 2와 같다.

Table. 2 Computer Hardware Specifications for Experiments

H/W	Description
CPU	Intel(R) Core(TM) i3-8100 CPU @ 3.60GHz (4 CPUs), ~3.6GHz
GPU	NVIIDIA GeForce GTX 1060 6GB
Memory	8192MB

3.3. 제안 모델의 Encoder와 Classifier

Dynamic RNN-CNN 모델은 크게 두 부분으로 구성되어 있다. 가변 차원의 파일을 고정된 차원의 결과물을 출력하는 Encoder와 Encoder의 결과물을 가지고 악성코드를 분류하는 Classifier로 구성 된다. Encoder는 그림1과 같이 Dynamic ConvLSTM으로 구성되어 있다.

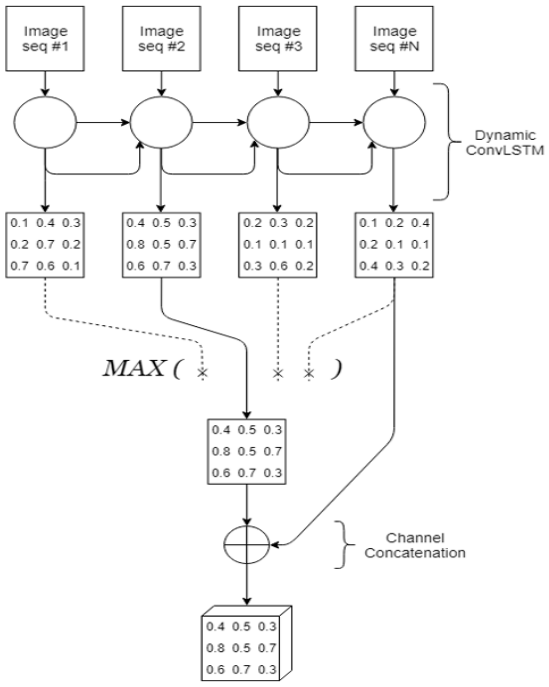

Fig. 1 EVariable length encoder

멀웨어 바이너리를 용량만큼 256 × 256 이미지 여러 장으로 변환한다. 각 이미지를 time step 시퀀스로 입력하게 되는데, 용량이 큰 파일은 time step이 길어지게 된다. Dynamic한 ConvLSTM 연산이기 때문에 time step의 길이와 관계없이 인코딩이 가능하다. ConvLSTM의 입력 time step 길이만큼의 출력 결과물이 나오게 된다. time step에 상관없이 고정된 차원 데이터 출력을 위하여 Attention 메커니즘의 아이디어를 차용하였다. 그 출력 결과물 중 가장 가중치가 큰 값을 가지는 출력과 가장 마지막 출력을 Channel Concatenation을 한 결과물이 Encoder의 최종 출력이다.

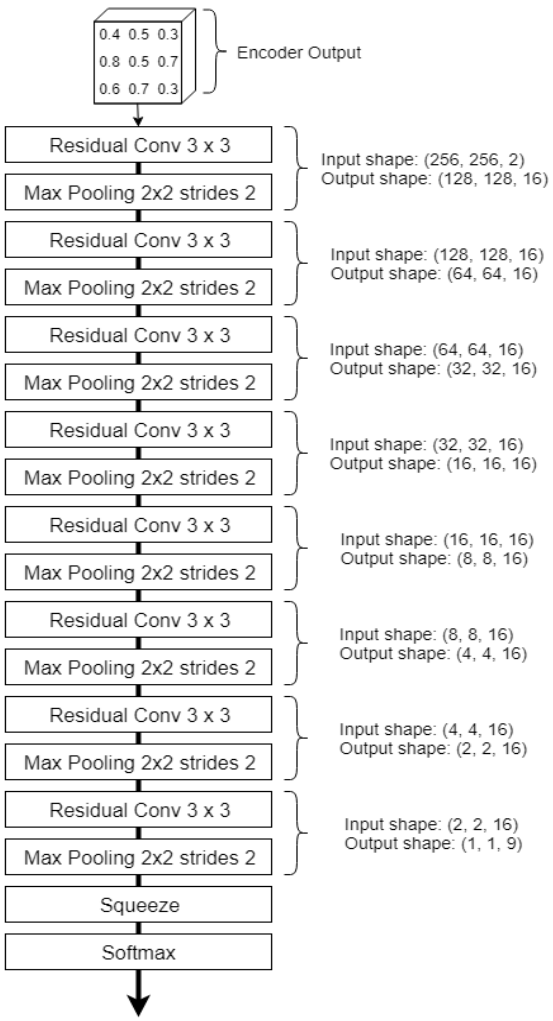

Fig. 2 Classifier with Residual Convolutional Layers

Classifier는 그림 2와 같이 구성되어 있다. Fully Connected 레이어 없이 Convolution 레이어로만 Classifier를 구성하였다. Convolution 레이어는 Residual Block을 추가하였다. Convolution의 Kernal Size는 3 x 3으로 Filter 개수는 16으로 마지막 레이어는 9로 설정하였다.

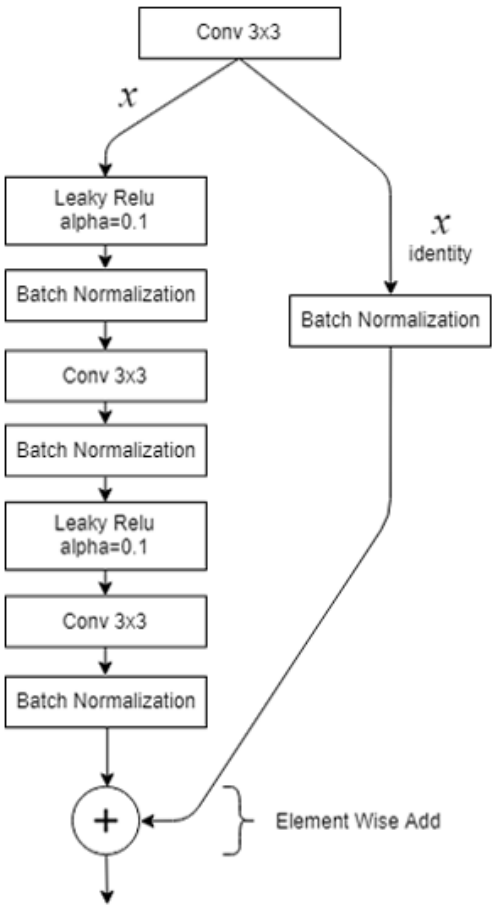

Fig. 3 Residual Conv 3×3 Layer in Fig. 2.

그림3은 그림2의 Classifier의 레이어 중 Residual Conv 레이어의 구조를 나타낸 것이다. Convolution 레이어 중간 중간에 Batch Normalization을 수행하였으며, 활성화 함수로는 Leaky Relu를 사용하였다.

3.4. 모델 학습

Mini-Batch 사이즈는 20으로 설정하고 40 epoch을 학습하였다. 검증 데이터 셋의 계산은 100 step마다 수행하고 검증 데이터 셋의 정확도 측정기준은 confusion matrix를 사용해 정확도(accuracy)를 사용하였다. 손실함수의 metric은 multiclass cross entropy를 사용하였다. 제안 모델의 학습 최적화 알고리즘은 Adam[11]을 사용하였으며 학습률은 0.0001값을 사용하였다. 메인 메모리의 용량 부족으로 멀웨어 바이너리 파일은 Mini-Batch 단위로 처리하였다.

Ⅳ. 결 과

학습 과정 및 결과는 그림4의 그래프에서 볼 수 있다. 본 논문에서 제안한 모델은 검증 데이터 셋 정확도 96%의 성능을 보였다.

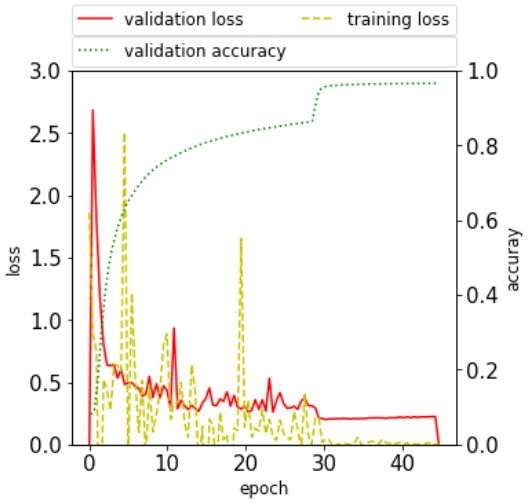

Fig. 4 Loss and accuracy graph.

각 클래스 별 모델 분류와 라벨 데이터를 비교하여 오차행렬(Confusion Matrix)을 나타낸 것이 그림 5이다.

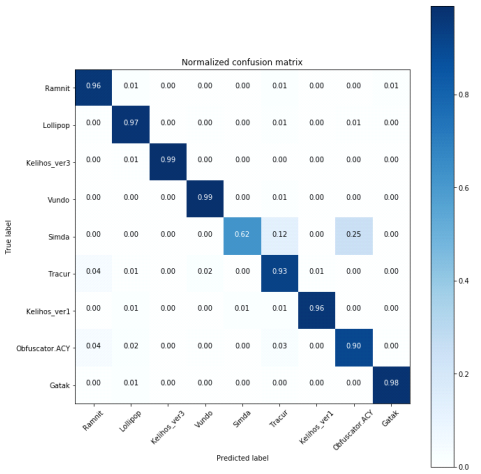

Fig. 5 Normalized confusion matrix.

오차 행렬에서 관찰 할 수 있듯이, Simda 클래스 분류가 상대적으로 낮은 정확도를 보였다. 이는 Simda 클래스의 데이터가 다른 클래스의 데이터보다 적은 수를 가지고 있어 나타난 현상으로 보인다. 불균형한 데이터 셋의 크기를 고려하여 Micro-average F1 스코어로 모델을 평가했을 때, 92%의 결과가 나왔다. 그림 6과 7은 각각 Micro-average와 Macro-average의 ROC 커브 그래프이다.

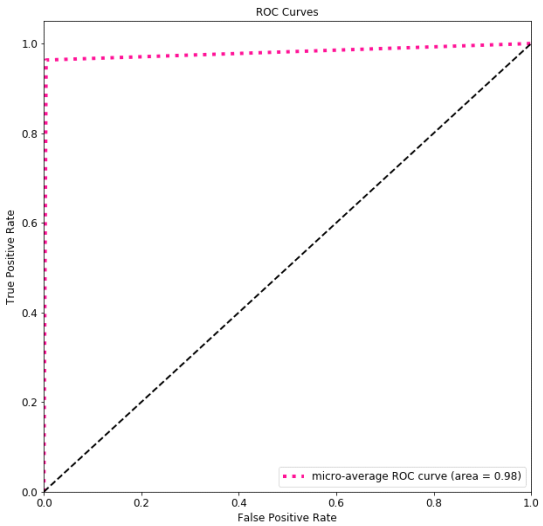

Fig. 6 Micro-average ROC curve

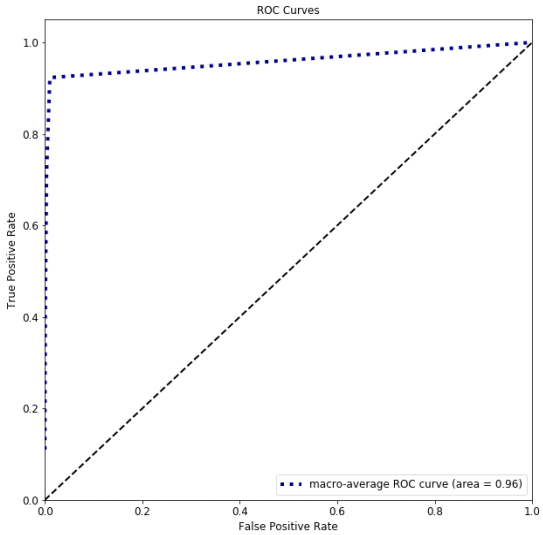

Fig. 7 Macro-average ROC curve

Ⅴ. 결 론

본 연구에서는 다양한 크기를 가진 멀웨어 파일을 복잡한 Feature Engineering과 많은 연산이 필요한 전처리 없이 Dynamic RNN과 CNN을 사용해 효과적으로 분류할 수 있는 딥러닝 모델을 제안하였다. 멀웨어 데이터에 대한 메타 데이터는 사용하지 않았고 바이너리를 256 × 256 크기의 여러 장의 이미지로 전처리하여 사용하였다. 실험 결과 검증 데이터 셋 정확도 96%를 기록하였다. 데이터 셋 불균형으로 인해 특정 클래스에서 상대적으로 분류 정확도가 떨어지는 현상이 관찰되었다. Micro-average로 평가하였을 때, 92%의 결과가 나왔다. 데이터 셋 불균형으로 인한 성능 저하는 데이터 셋의 개수가 많아지면 극복 될 것으로 기대된다. 또한 Bidirectonal RNN과 Inception 구조 등으로 모델을 변경한다면 조금 더 나은 성능을 기대할 것으로 생각된다. 본 연구에서 제안한 논문은 가변 입력 데이터를 대응할 수 있으며, 입력 데이터에 대한 복잡한 전처리가 필요없는 것이 장점이나, ConvLSTM Encoder와 Residual Block등을 추가하여 모델이 복잡해 분류에서 많은 연산을 필요로 한다는 점은 추가 연구가 필요하다.

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2018 R1C1B5083789).

REFERENCES

[1] S. J. Park, S.M. Choi, H.J. Lee, and J.B. Kim, “Spatial analysis using R based Deep Learning,” *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, vol. 6, no. 4, pp. 1-8, Apr. 2016.

[2] J.M. Kim, and J.H. Lee, “Text Document Classification Based on Recurrent Neural Network Using Word2vec,” *Journal of korean Institute of Intelligent System*, vol. 27, no.6, pp. 560-565, Jun. 2017.

[3] P. Baudis, S. Stanko, and J.Sedivy, “Joint Learning of Sentence Embeddings for Relevance and Entailment,” in *The Workshop on Representation Learning for NLP*, Berlin, Germany, pp. 18-26, 2016.

[4] J.Y. Kim, and E. H. Park, “e-Learning Course Reviews Analysis based on Big Data Analytics,” *Journal of the Korea Institute of Information and Communication Engineering*, vol. 21, no. 2, pp. 423-428, Feb. 2017.

[5] J.M. Kim, and J.H. Lee, “Text Document Classification Based on Recurrent Neural Network Using Word2vec,” *Journal of Korean Institute of Intelligent Systems*, vol. 27, no. 6, pp. 560-565, Dec. 2017.

[6] J. Mueller, and T. Aditya “Siamese Recurrent Architectures for Learning Sentence Similarity.” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, Arizona, pp. 2786-2792, 2016.

[7] Y. Kim, Y. Jernite, S. David, and M. R. Alexander, “Character-Aware Neural Language Models,” *CoRR, abs/1508.06615*, 2015.

[8] Naver ai hackerton 2018 Team sadang solution [Internet]. Available:https://github.com/moonbings/naver-ai-hackathon-2018.

[9] R. Dey, and F. M. Salem. “Gate-variants of gated recurrent unit (GRU) neural networks,” CoRR, abs/1701.05923, 2017.

[10] wiki fast .ai Logloss [Internet]. Available: http://wiki.fast.ai/index.php/Log_Loss

[11] Diederik P. Kingma, Jimmy Ba, “Adam: A Method for

Stochastic Optimization" in *3rd International Conference for Learning Representations*, San Diego, 2015.

[12] G. Y. Lim, and Y. B. Cho, "The Sentence Similarity Measure Using Deep-Learning and Char2Vec." *Journal of the Korea Institute of Information and Communication Engineering*, vol. 22, no. 10: 1300-1306, Oct. 2018.

임근영 (Geun-Young Lim)

입학년도-현재 :대전대학교 정보보안학과 학부4학년
※관심분야 : Deep-learning, NLP, Computer Vision

조영복(Young-Bok Cho)

2005: 충북대학교 전자계산학과 공학석사
2012: 충북대학교 전자계산학과 공학박사
2016: 충북대학교 의학과 박사과정수료
2012-2018: 충북대학교 소프트웨어학과 초빙교수
현 재 : 대전대학교 정보보안학과 조교수
※관심분야: 의료영상처리, 정보보안, 의료정보보호, 모바일보안