Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.5, Oct. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.5.1027

사이버 공격에 의한 시스템 이상상태 탐지 기법

윤 여 정,†‡ 정 유 진
국방과학연구소

Detection of System Abnormal State by Cyber Attack

Yeo-jeong Yoon,†‡ You-jin Jung
Agency for Defense Development

요 약

기존의 사이버 공격 탐지 솔루션은 일반적으로 시그니처 기반 내지 악성행위 분석을 통한 방식의 탐지를 수행하므로, 알려지지 않은 방식에 의한 공격은 탐지하기 어렵다는 한계가 있다. 시스템에서는 상시로 발생하는 다양한 정보들이 시스템의 상태를 반영하고 있으므로, 이들 정보를 수집하여 정상상태를 학습하고 이상상태를 탐지하는 방식으로 알려지지 않은 공격을 탐지할 수 있다. 본 논문은 정상상태 학습 및 탐지에 활용하기 위하여 문자열을 그 순서와 의미를 보존하며 정량적 수치로 변환하는 머신러닝 임베딩(Embedding) 기법과 이상상태의 탐지를 위하여 다수의 정상데이터에서 소수의 비정상 데이터를 탐지하는 머신러닝 이상치 탐지(Novelty Detection) 기법을 이용하여 사이버 공격에 의한 시스템 이상상태를 탐지하는 방안을 제안한다.

ABSTRACT

Conventional cyber-attack detection solutions are generally based on signature-based or malicious behavior analysis so that have had difficulty in detecting unknown method-based attacks. Since the various information occurring all the time reflects the state of the system, by modeling it in a steady state and detecting an abnormal state, an unknown attack can be detected. Since a variety of system information occurs in a string form, word embedding, ie, techniques for converting strings into vectors preserving their order and semantics, can be used for modeling and detection. Novelty Detection, which is a technique for detecting a small number of abnormal data in a plurality of normal data, can be performed in order to detect an abnormal condition. This paper proposes a method to detect system anomaly by cyber attack using embedding and novelty detection.

Keywords: Cyber Attack, Unknown Attack, Word Embedding, Novelty Detection, Anomaly Detection

I. 서 론

기존 공격에 대한 시그니처나 행위 분석 DB를 이용한 탐지 기법은 알려지지 않은 공격을 탐지하기 어렵다. 반면에 정상상태 모델을 기준으로 공격으로 인해 발생한 시스템의 비정상상태를 탐지 할 경우, 기존 공격 뿐 만 아니라 알려지지 않은 공격도 탐지 할 수 있다. 이는 곧 시스템의 정상상태에 대한 데이터를 기준으로 이상치 탐지(Novelty Detection)를 수행하는 것으로, 이상치 탐지란 다수의 정상 데이터와는 그 값이나 양상이 상이한 이상치 데이터를 판별해내는 기법이다. 즉 시스템에서 발생하는 다양한 데이터를 수집하여 정상적으로 운용될 때의 상태를 학습하고, 실시간으로 수집되는 데이터를 정상상태 모델과 비교하여 이상치 탐지를 수행하는 방식으로, 이를 통하여 알려지지 않은 공격에 의해 발생한 시스템의 이상상태를 탐지할 수 있다.

시스템에서 발생하는 정보들 중 메모리 사용량,

Received(07. 11. 2019), Accepted(08. 18. 2019)
† 주저자, befilledwithawe@gmail.com
‡ 교신저자, befilledwithawe@gmail.com (Corresponding author)

네트워크 통신량 등 정량적인 정보들이 있지만, 이러한 정보만으로 시스템의 상태를 파악하는 것은 한계가 있다. 시스템에서 발생하는 정보 중에는 프로세스의 생성과 종료, 파일의 생성과 수정 내역, 네트워크 통신 기록 등 비정량적인 데이터가 많고, 이들을 활용하여 이상치 탐지를 할 수 있도록 정량적인 정보로 환원할 필요가 있다. 이를 위해 자연어 처리 분야에서 널리 쓰이고 있는 임베딩(Embedding)기법을 활용할 수 있다. 자연어 처리에서 임베딩은 단어와 문장의 의미를 반영하여 벡터로 변환하는 기법을 말한다.

본 논문에서는 시스템 상태정보를 이용한 이상치 탐지 방법을 제안하기 위하여, 2장에서는 관련 연구로서 기존의 임베딩 기법 및 이상치 탐지 기법을 검토하며, 3장에서는 본 논문에서 제안하는 사이버 공격에 의한 이상상태 탐지 기법을 서술한다. 4장에서는 실험을 위해 생성한 데이터셋 및 실험 결과에 대하여 서술하며, 5장에서는 결론 및 향후 개선 방향을 서술한다.

II. 관련 연구

2.1 임베딩(Embedding) 기법

임베딩이란 자연어 처리 기법 중 단어나 문장을 그에 대응되는 수치, 벡터로 변환하는 기법을 의미한다. 오디오나 이미지 등의 고차원 데이터는 그 자체로 각각의 값이 연속된 실수 값을 가지기 때문에 머신러닝 등의 처리가 용이한 반면, 단어들로 이루어진 텍스트 데이터는 그렇지 않다. 임베딩 기법이 활용되기 이전에는 각 단어들을 이산된 표지로 치환하여 분석하는 방법이 사용되었으나, 단어에 대응되는 값이 다른 단어들과의 관계를 반영하지 않고, 다양한 단어들이 사용됨에 따라 데이터의 차원수가 너무 높아진다는 문제가 있었다. 임베딩 기법은 이를 해결하기 위해 인공신경망, 차원축소, 동시발생 행렬 등을 활용하여, 범주형 자료인 단어들의 집합을 연속적인 하나의 벡터 공간에 표현한다[1].

본 장에서는 최근 가장 널리 활용되는 임베딩 기법 중 단어를 벡터로 변환하는 Word2Vec과, 문장을 벡터로 변환하는 Doc2Vec과 Autoencoder 임베딩에 대하여 서술한다.

2.1.1 Word2Vec

Word2Vec은 여러 문장들을 입력받아 문장을 구성하는 단어들 사이 관계를 학습하여 각 단어의 의미가 반영되도록 벡터 공간에 표현하는 방법이다[2]. Word2Vec은 같은 문장 등 가까운 위치에 있는 단어들끼리는 유사한 의미를 가진다고 보고, 함께 사용되는 빈도가 높은 단어들끼리는 유사한 벡터값을 갖도록 학습한다. 구체적으로는, 입력받은 문장 내에서 일부를 추출하는 window를 움직여 가며 학습하되, 주변 단어들의 정보로 중심 단어를 예측하는 방식(CBOW, Continuous Bag of Words)과, 중심 단어의 정보로 주변 단어들을 예측하는 방식(Skip-Gram)이 있다.

2.1.2 Doc2Vec

Doc2Vec은 문장을 벡터로 변환하는 기법으로, 단어들로 이루어진 문장의 의미와 그 단어들의 의미가 유사하리라는 전제를 기초로 한다[3]. Doc2Vec은 Word2Vec과 동일한 방식으로 학습을 진행하되 문장을 표현하는 벡터가 그 문장 속에 포함된 단어들의 벡터 값들과 유사한 값을 갖도록 학습한다. 즉, Word2Vec과 같은 학습을 진행하면서 문장의 벡터를 함께 학습시킨다. Doc2Vec도 마찬가지로 구체적 학습 방식에 있어 단어들과 문장 정보로 중심 단어를 예측하는 방식(PV-DM, Distributed Memory Model of Paragraph Vectors)과 문장을 입력하여 각 단어들의 도출 확률을 예측하는 방식(PV-DBOW, Distributed Bag of Words of Paragraph Vectors)이 있다.

2.1.3 Autoencoder 임베딩

Autoencoder는 입력과 비슷한 값을 출력하도록 비지도 학습으로 훈련되는 신경망을 말한다. 일반적으로 입력을 중간층으로 부호화하는 부분인 인코더(Encoder)와 중간층에서 다시 출력으로 복호화하는 부분인 디코더(Decoder)으로 구성되며, 입력을 통해 출력한 값과 입력을 비교해 오차를 줄이는 방향으로 학습이 이루어진다. 학습이 진행되면서 데이터의 중요한 자질(feature)이 반영될 수 있도록 인코더와 디코더의 가중치가 조정된다.

Autoencoder를 이용하여 문장을 벡터로 변환하

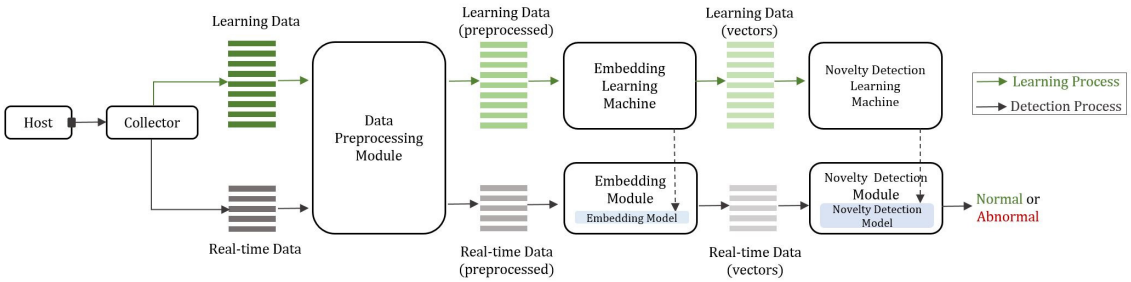


Fig. 1. Operation Architecture of system Abnormal Detection

는 방법으로서, Seq2Seq model 개념을 활용할 수 있다. Seq2Seq model이란 가변 길이의 여러 sequence를 입력받아 학습 과정을 거쳐 그에 대응되는 sequence를 출력하는 딥러닝 모델로, 번역기, 챗봇 등에서 오늘날 널리 활용되고 있다[4]. 가변 길이의 문장을 입력받아 학습하고, 단어 간 순서가 반영될 수 있도록 RNN(Recurrent Neural Network)을 활용하는데, 입력 sequence를 학습하는 RNN을 거쳐 Context Vector를 형성하고(encoder), 형성된 벡터에서 다시 출력 sequence를 형성하는(decoder) 구조로 이루어져 있다.

이처럼 입력에 맞는 출력을 도출할 수 있도록 학습하는 Seq2Seq 모델에서, 입력과 동일한 sequence를 출력하도록 학습을 진행시킨다면, 모델 내에서 학습되는 sequence의 내용이나 순서 정보, 특성이 반영되는 Context Vector를 생성할 수 있다. 이러한 방식으로 문장을 벡터로 변환하는 임베딩 모델을 만들 수 있다.

2.2 이상치 탐지(Novelty Detection) 기법

실 데이터에 대하여 이상치 탐지를 적용 할 경우, 데이터의 대부분은 정상 데이터이며, 다수의 정상 데이터에 비하여 소수의 이상 데이터를 탐지해야 한다. 이와 같은 정상 및 비정상 데이터의 비율에 대해서는 기존의 classification 알고리즘이나 clustering 알고리즘을 적용하기 어렵다.

본 장에서는 다수의 정상 데이터와 소수의 이상치 데이터에 대하여, 정상 데이터와 비교하여 많이 벗어나 있는 Novelty Data(Outlier)를 탐지하는 이상치 탐지 기법에 대하여 서술한다[5, 6].

2.2.1 One-class SVM

One-class SVM은 정상 데이터를 원점으로부터 최대한 멀리 위치하게 만드는 초평면(Hyperplane)을 찾고, 새로운 데이터가 원점과 초평면 사이에 위치하면 이상치 데이터로 판별하는 알고리즘이다[7]. 기본 SVM 알고리즘과 마찬가지로 데이터 본래의 차원을 그대로 이용할 경우 linear decision boundary를 구하기 어렵기 때문에 kernel trick을 이용하여 데이터를 고차원으로 맵핑시킨다.

2.2.2 Isolation Forest

Isolation Forest는 소수 범주(이상치)는 개체수가 적고, 소수 범주 데이터는 정상 범주 데이터와는 특정 속성 값이 많이 다를 가능성이 있다는 특성을 이용한다. 이상치 탐지를 위해 먼저 하나의 객체를 고립(isolation) 시키는 tree를 생성한다[8].

각 입력 데이터를 해당 tree를 이용하여 고립시킬 때, 정상 데이터를 고립시키기 위해서는 많은 분기(split)가 필요한 반면, 이상치 데이터는 상대적으로 적은 분기만으로 고립이 가능하다. 이를 이용하여 특정 개체가 고립되는 말단 노드까지의 거리를 이상치 점수로 정의하고, 고립되는 말단 노트까지의 거리(depth)가 짧을수록 이상치 점수가 높아지도록 설정한다. 입력 데이터에 대한 이상치 점수가 특정 threshold 보다 높을 경우 이상치 데이터로 탐지한다.

III. 시스템 이상상태 탐지 기법

본 논문에서 제안하는 시스템 이상상태 탐지 기법은 Fig. 1과 같은 동작 구조를 가진다. 제안 기법은 탐지 대상 시스템과 프로세스의 정상 상태만을 기록

한 학습 데이터를 이용하여 임베딩 모델 및 이상치 탐지 모델을 학습하는 학습 과정과, 시스템에서 발생하는 실시간 데이터에 대하여 학습된 모델을 기반으로 정상 또는 비정상을 판단하는 탐지 과정으로 나뉜다.

학습 과정에서는 수집기를 통해 시스템의 상태 정보를 주기적으로 수집하되, 이 단계에서 시스템은 정상 행위만을 한다고 전제한다. 따라서, 이 때 수집된 모든 정보를 정상 데이터로 분류한다. 수집된 정보는 임베딩을 위한 전처리 과정을 거친 후, 임베딩 학습 모듈을 통해 학습되어 임베딩 모델을 생성한다. 이후 생성된 임베딩 모델을 이용하여 수집된 정보를 임베딩함으로써 벡터로 변환하고, 이 값은 이상치 탐지 학습 모듈에 입력되어 탐지 모델을 생성한다.

탐지 과정에서도 동일한 수집기를 통해 시스템 상태 정보를 실시간으로 수집한다. 실시간 수집 데이터에는 비정상 데이터가 포함될 수 있으며, 위 학습 과정에서 생성된 모델을 이용해 비정상 데이터를 탐지하는 것을 목표로 한다. 탐지 과정에서 수집된 실시간 상태 정보 역시 학습 과정과 동일하게 임베딩을 위한 전처리 과정을 거치고, 이후 학습 과정에서 생성된 임베딩 모델을 이용하여 벡터로 변환된다. 이후 탐지 모듈을 통하여 입력 데이터에 대한 정상 또는 비정상 여부를 판단한다.

본 장에서는 본 논문에서 제안하는 시스템 이상상태 탐지 기법의 각 단계에서 수행되는 정보 수집, 전처리와 임베딩 및 탐지 과정을 상세히 서술한다.

3.1 시스템 상태정보 수집

사이버 공격에 의한 시스템의 이상상태를 탐지하기 위해서, 시스템과 프로세스의 상태를 서술 할 수 있는 관련 이벤트를 수집한다. 수집 이벤트와 각 수집 이벤트에 대한 상세 수집 필드 항목의 예제는 Table. 1과 같다.

각 호스트에서 동작하는 수집 에이전트를 통하여 수집된 이벤트는 DB에 저장하여 이후 데이터 전처리 및 임베딩, 이상치 탐지 알고리즘의 입력 값으로 사용될 수 있도록 한다.

3.2 시스템 상태정보 전처리

3.1 장에서 서술한 시스템 및 프로세스의 상태정보 수집을 통해 수집된 이벤트를 이용하여 임베딩 및 이상치 탐지 의 입력 값으로 사용할 수 있게끔 전처

Table 1. Collection Event List

Class	Events	Example fields
Process	create	group ID, host ID, process ID, event timestamp, parent process ID, full path, etc.
	terminate	group ID, host ID, process ID, event timestamp, exit code, etc.
	load module	group ID, host ID, process ID, event timestamp, DLL path, etc.
	open	group ID, host ID, process ID, event timestamp, target process ID etc.
File	create	group ID, host ID, process ID, event timestamp, file size, hash, etc.
	change name	group ID, host ID, process ID, event timestamp, changed name, path, etc.
Registry	create key	group ID, host ID, process ID, event timestamp, path, etc.
	write value	group ID, host ID, process ID, event timestamp, size, value name, etc.
	delete value	group ID, host ID, process ID, event timestamp, value name, etc.
	change key name	group ID, host ID, process ID, event timestamp, new name, etc.
Network	tcp connect	group ID, host ID, process ID, event timestamp, client ip/port, server ip/port, transmission number, etc.
	tcp accept	group ID, host ID, process ID, event timestamp, client ip/port, server ip/port, transmission number, etc.

리를 수행한다. 또한 이상탐지 결과 정확도를 산출하기 위하여 각 데이터셋에 대하여 정상 행위 데이터와 비정상 행위 데이터에 대한 라벨링을 적용한다.

본 논문에서는 프로세스 행위 이벤트 데이터셋과 단위시간 당 시스템 이벤트 데이터셋을 이용하였다.

3.2.1 프로세스 행위 이벤트 데이터셋

정상 프로세스에 대한 행위 이벤트를 학습하고, 이에 대하여 다른 경향성을 가지는 비정상 프로세스를 탐지하기 위하여 각 프로세스의 생성 이벤트부터 종료 이벤트까지 발생한 모든 이벤트를 임베딩 및 탐지의 단위로 설정하였다. 또한 각 프로세스에 대하여 발생 이벤트 수가 200개를 초과할 경우, 각 입력 값을 200개의 이벤트 단위로 나누어 전처리를 수행하였다. 이는 특정 프로세스가 생성부터 종료까지 대량의 실행 이벤트가 발생할 경우 입력 값의 용량이 비정상적으로 커지므로 임베딩 및 탐지 시 실험이 불가한 경우가 발생함에 따라, 실험 환경에 적용 가능하도록 입력 데이터를 비슷한 크기로 조정하기 위함이다.

각 수집 이벤트에 대하여 패킷 수, 이미지 생성 시각 등 값을 갖는 형태의 필드 값은 임베딩 대상에서 제외하도록 하였다. 동일한 문서에서 발견되는 단어의 빈도가 높을수록 단어 간 의미의 유사도를 높게 책정할 뿐 단어를 구성하는 문자 자체의 유사도를 비교하지 않는 임베딩 메커니즘 특성상, 수치로 표현된 값은 그 의미를 충분히 반영할 수 없기 때문이다. 또한, 플래그의 형태로 수집되는 필드 값 등 서로 다른 필드의 데이터라도 같은 단어로 표현되나 그 의미가 상이한 단어들의 경우에는 필드명을 함께 표시하는 등 의미 차이를 반영할 수 있도록 하였다.

3.2.2 단위시간 당 시스템 이벤트 데이터셋

시스템 전체의 관점에서 단위시간 동안 발생하는 시스템 이벤트 정보에서 이상상태가 드러날 수 있으므로, 수집한 정보를 데이터의 종류 및 발생시간으로 구분하여 임베딩 및 탐지의 단위로 할 수 있다. 본 논문에서는 수집된 시스템 이벤트 데이터를 토대로 임베딩 및 이상상태 탐지에 앞서, 이벤트의 종류에 따라 네트워크 이벤트, 레지스트리 이벤트로 데이터를 나누고, 이를 발생 시간 30초마다 다시 나누어 임베딩 및 탐지의 1단위가 되도록 하는 방식의 전처리를 수행하였다.

기타 수집되는 데이터 중 임베딩의 특성 및 데이터의 성질에 따른 전처리 과정은 3.2.1에서 서술한 프로세스 행위 이벤트 데이터 셋과 동일하게 수행하였다.

3.3 데이터셋 임베딩

전처리가 완료된 데이터셋은 문자열로 구성되어 있으므로 이를 탐지 알고리즘의 입력 값 형식인 수치형 벡터로 변환하기 위하여 2.1장에서 서술한 임베딩 기법을 활용한다.

임베딩 기법 중 Doc2Vec은 전체 문서들 사이의 비교를 통하여 상대적인 거리를 구하는 임베딩 기법이다. 따라서 정상 데이터와 탐지 데이터 모두를 모델 학습시에 활용하여야 하지만, 본 논문에서는 평상시에 축적된 학습 정상 데이터를 기반으로 임베딩 모델을 학습 후 새로이 수집된 테스트 데이터를 임베딩 및 탐지하고자 하므로 별도의 조치가 필요하였다.

본 논문에서는 학습 정상 데이터만을 대상으로 임베딩을 수행하여 정상 데이터의 벡터값을 추출한 후, 탐지 데이터는 Gensim Doc2Vec 라이브러리의 infer_vector 함수를 활용하여, 학습 데이터와의 비교를 통해 정상 데이터들과의 관계에서 해당 데이터가 가질 벡터값을 유추하였다. 학습 데이터에서 학습되지 않았던 새로운 단어의 경우 unknown 토큰으로 처리하였다.

3.4 이상치 탐지 기법 학습

임베딩된 정상상태 데이터셋을 입력 값으로 하여 2.2장에서 서술한 이상치 탐지 기법에 대한 학습을 통해 정상상태 모델을 생성한다. 본 논문에서는 One-class SVM과 Isolation Forest 알고리즘을 이용하여 모델을 생성하였다.

3.5 이상상태 탐지

정상상태 데이터를 이용하여 이상치 탐지 기법을 학습한 후, 실시간으로 입력되는 상태정보 데이터를 임베딩하고, 학습모델을 이용하여 이상치 점수(Novelty score)를 산출, 특정 threshold 보다 높을 경우 이상상태로 탐지한다.

기존의 One-class SVM은 정상데이터에 대한

초평면(Hyperplane)을 기준으로 입력 데이터가 초평면보다 위에 존재하면 정상, 아래에 존재하면 이상치 데이터로 탐지하나 본 논문에서는 오탐율을 줄이기 위하여 이상치 점수(입력데이터와 초평면까지의 거리)가 특정 threshold 이하일 경우 이상치 데이터로 탐지하였다.

Isolation Forest는 입력 데이터를 고립시키는 말단 노드까지의 depth를 이용하여 Novelty score를 출력하며, 이 값이 전체 Novelty score 범위의 특정 비율 이상이면 이상치 데이터로 탐지한다.

IV. 실험 결과

이상의 시스템 이상상태 탐지 기법의 각 단계에서 적용할 수 있는 임베딩 기법(Doc2Vec, Autoencoder) 및 탐지 기법(One-class SVM, Isolation Forest) 각각의 성능을 평가하기 위하여 실험을 진행하였다. 운용 단계에서는 정상 데이터만으로 임베딩 및 학습이 진행되나, 본 실험에서는 각 기법의 정확도를 정량적으로 평가하기 위하여 정상과 악성 데이터를 생성하여 탐지 데이터로 활용하였다. 즉, 생성된 정상 데이터로 임베딩 및 탐지 모델을 학습하고, 정상과 악성 데이터로 구성된 탐지 데이터를 활용하여 각 알고리즘을 이용한 예측의 결과와 비교함으로써 정확도를 산출하였다.

본 장에서는 본 논문에서 제안하는 시스템 이상상태 탐지 기법의 효과 및 각 단계에서 적용할 임베딩 및 탐지 기법의 성능을 확인하기 위한 실험의 데이터셋 생성 방안과, 각 기법별 탐지 결과를 서술한다.

4.1 사용 데이터 셋

본 논문에서는 실험 및 제안 기법의 정확도 산출을 위하여 데이터셋을 생성하였다. 정상상태와 비정상상태를 모의하고 각각 3.1 장에서 서술했던 시스템 상태정보를 수집하였으며, 각 정상행위와 비정상행위 이벤트에 대한 라벨링을 통해 정확도를 산출할 수 있도록 하였다.

4.1.1 정상상태 모의

이상치 탐지 기법 학습을 위해서는 정상 데이터셋이 필요하므로, 본 논문에서는 테스트베드 상에서 사용자의 정상 행위를 모의하였다. PC와 서버에 대하여 정상 행위 모의를 수행하였으며, PC에서는 사용자의 역할 분류에 따른 행위 모의 모델을 구축하고 자동화된 모의 소프트웨어를 통하여 행위 모의를 실행하였다. 서버에서는 웹서버, 메일서버, FTP 서버와 같은 사용자의 행위를 지원해주는 서버 기능을 모의하였다. 사용자의 역할은 문서 담당자, 개발자, 서버 관리자, 프로젝트 매니저 등 업무에 대한 역할 기준으로 분류하였으며 각 역할에 따라 행위 분류(문서/메일/웹/엔지니어링/파일 관리/유휴 시간)에 대한 비율을 지정하였다. 각 행위 분류에 대한 세부 행위 항목의 예제는 Table. 2와 같으며, 모의 프로그램 항목 및 주요 행위는 Table. 3과 같다.

Table 2. Examples of detailed behavioral items for behavior classification

behavior classification	detailed behavioral items
document	create DOC document
	create XLS document
	create PPT document
	create HWP document
	read PDF document
	create TXT document

Table 3. Simulation program items and key activities

Program items	Key activities
MS Word, MS Power Point, MS Excel, Hangeul, Notepad	create and save document
PDF reader	open and save PDF document
MS Outlook	send and receive E-mail
Web browsing	access web site, login, create board post
Ping	ping communication
SSH	access server, command execution and termination
SNMP	request and terminate server information
File managing	local file / shared file copy, alternate and delete
FTP	access server, send file, terminate
device control	virtual USB, CD connecting and disconnecting

업무 시간, 역할 정보 및 사용자 행위 시간(출근/점심/퇴근)을 입력받아 랜덤하게 사용자 행위 시나리오를 구성하고, 해당 시간에 맞춰 각 프로그램의 행위를 수행한다.

4.1.2 비정상상태 모의

이상 탐지 기법에 대한 정확도 산출을 위해서는 비정상 데이터셋이 필요하므로, 본 논문에서는 테스트베드 상에서 사이버 공격이 발생했을 때의 상태와 행위를 모의하였다. 비정상 행위 모의를 위해 MITRE에서 주요 공격 행위를 정의한 ATT&CK 모델을 이용하였으며, ATT&CK에 정의된 공격 행위 중 약 150여개의 행위를 모의하였다[9]. 또한 공격 시뮬레이션에 필요한 C&C 서버를 제작하여 각 단일 행위를 다수개 연결하여 APT 공격을 시뮬레이션 할 수 있도록 했다. ATT&CK를 기반으로 모의한 비정상 행위 목록의 일부는 Table. 4와 같다.

Table 4. Examples of abnormal behavior

Behavior classiciation	Detailed behavioral items
document	create DOC document
	create XLS document
	create PPT document
	create HWP document
	read PDF document
	create TXT document

4.1.3 데이터셋 라벨링

본 논문에서는 이벤트 정보를 묶어 문서로 제작한 뒤 임베딩 및 탐지 과정에서 정상 데이터로 분류된 문서만을 이용해 학습하고, 검증 데이터셋에 대한 문서별 악성 및 정상을 탐지하므로, 각 입력값에 대한 라벨링이 별도로 요구된다.

프로세스 행위 이벤트 데이터셋은 각 실행 프로세스에 대해 프로세스 생성부터 종료까지 실행했던 모든 이벤트를 입력 값의 단위로 처리하므로, 실행 프로세스 단위로 비정상 행위 모의에서 사용되었던 프로세스를 추적하여 악성으로 라벨링을 수행하였다.

단위시간 당 시스템 이벤트 데이터 셋은 각 문서마다 실행 주체(프로세스)를 달리하는 다양한 이벤트 정보가 포함되므로, 문서에 포함된 이벤트들 중 악성으로 라벨링 된 이벤트가 1개 이상 포함된 경우 해당 문서를 악성으로 분류하였다.

4.2 이상상태 탐지 실험 결과

4.2.1 프로세스 행위 이벤트 이상상태 탐지

본 실험을 위하여 약 5일간 정상상태 및 비정상상태 행위모의를 진행하며 수집한 시스템 이벤트를 기반으로 프로세스 행위 이벤트 데이터셋을 생성하기 위한 전처리 및 라벨링을 수행하였으며, one-class SVM과 Isolation Forest를 이용한 이상치 탐지 결과를 확인하였다. 모델 생성에는 20485개의 정상 학습 데이터가 사용되었으며, 정확도 산출을 위해 5763개의 정상 검증 데이터, 473개의 비정상 검증 데이터를 사용하였다.

4.2.1.1 one-class SVM 을 이용한 이상상태 탐지

one-class SVM을 이용할 경우 커널함수의 종류, gamma 및 nu 값을 hyper parameter로 설정하여야 한다. 본 논문에서는 rbf 커널을 사용하였으며, grid search를 통해 가장 높은 auroc를 기록하는 gamma와 nu값을 탐색하였다.
먼저 Doc2Vec과 Autoencoder 임베딩 기법의 유효성 및 정확도를 산출하기 위하여, 각 임베딩 기법을 통해 생성된 벡터에 대하여 one-class SVM 이상치 탐지를 수행하였으며, 각 임베딩 기법에 대한 one-class SVM 정확도 결과 값은 Fig. 2, 3 및 Table. 5와 같다.

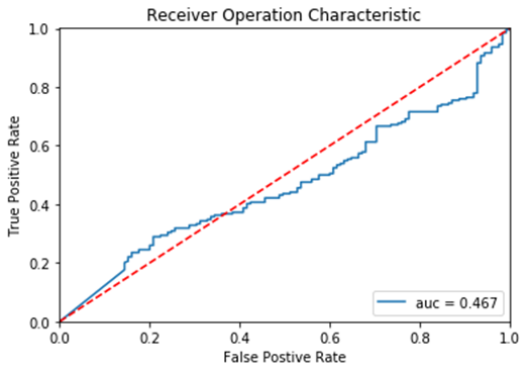


Fig. 2. ROC curve on OneSVM detection of process action event data embedded by Doc2Vec

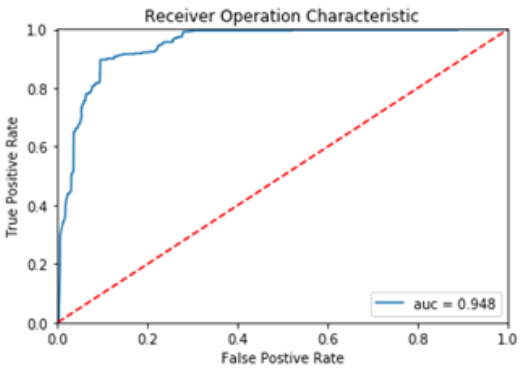


Fig. 3. ROC curve on OneSVM detection of process action event data embedded by Autoencoder

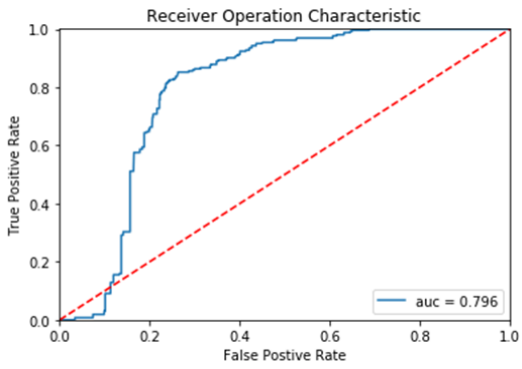


Fig. 4. ROC curve on Isolation Forest detection of process action event data embedded by Autoencoder

Table 5. Detection Results by Embedding Algorithms

Embedding Algorithm	gamma	nu	AUROC
Doc2Vec	0.1	0.8	0.467
Autoencoder	0.8	0.8	0.948

Autoendoer 임베딩 벡터에 대한 one-class SVM 이상탐지 결과 AUROC 최대값이 0.948을 기록하였으나, Doc2Vec 임베딩 벡터에 대한 one-class SVM 이상탐지 결과는 AUROC 최대값이 0.467으로, Autoendoer 임베딩 벡터에 대한 이상탐지 결과에 비하여 매우 낮은 정확도 수치를 보였다. 이에 따라 본 논문의 실험에 사용한 데이터셋의 경우 Doc2Vec보다 Autoencoder 임베딩 기법을 적용하는 것이 정확도가 더 높음을 확인할 수 있었다.

4.2.1.2 Isolation Forest를 이용한 이상상태 탐지

Isolation Forest를 이용할 경우 최대 tree 분할 개수를 의미하는 estimator 값을 hyper parameter로 설정하여야 한다. 본 논문에서는 grid search를 통해 가장 높은 auroc를 기록하는 estimator값을 탐색하였다.

4.2.1.1에서 서술한 실험결과를 통하여 본 논문의 실험용 데이터셋의 경우 Doc2Vec보다 Autoencoder 임베딩 기법을 적용하는 것이 더 적합함을 알 수 있었다. 따라서 본 장에서는 Autoencoder 임베딩 기법을 활용한 벡터를 이용하여 Isolation Forest 이상 탐지를 수행한 결과를 서술한다.

실험 결과 estimator 200일 때 AUROC 최대값인 0.773을 기록하였다. 실험 결과 auc 그래프는 Fig. 4.와 같다.

4.2.1.3 탐지 알고리즘별 결과 비교

Table 6.은 Autoencoder 임베딩 기법을 적용하여 얻은 벡터를 입력 값으로 하여, One-class SVM과 Isolation Forest를 통한 탐지 결과를 비교하였다. One-class SVM을 이용한 이상탐지 정확도가 더 높은 것을 확인할 수 있다.

Table 6. Detection Results by Detecting Algorithms

Detecting Algorithm	parameters		AUROC
OneSVM	gamma	0.8	0.948
	nu	0.8	
Isolation Forest	n_estimator	200	0.796

4.2.2 단위시간 당 시스템 이벤트 이상상태 탐지

프로세스 행위 이벤트와 마찬가지로, 5일간 행위 모의 결과 수집된 데이터를 기반으로 임베딩 및 이상상태 탐지를 수행하였다. 네트워크 데이터와 레지스트리 데이터 각각에 대하여 실험을 수행하였으며, 네

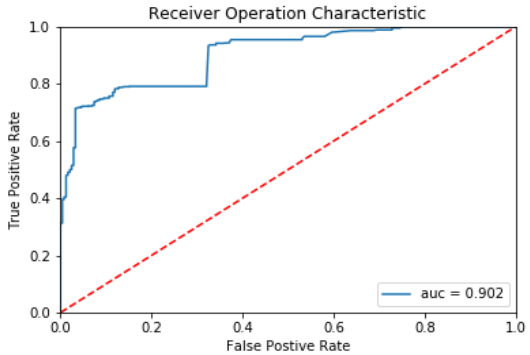
트워크 데이터의 경우 모델 생성에 2372개의 정상 학습 데이터, 정확도 산출을 위해 243개의 정상 검증 데이터 및 176개의 악성 비정상 검증 데이터를 사용하였다. 레지스트리 데이터는 모델 생성에 5002개의 정상 학습 데이터를 사용하였고, 정확도 산출을 위해 500개의 정상 검증 데이터 및 100개의 비정상 검증 데이터를 사용하였다.

4.2.2.1 one-class SVM을 이용한 이상상태 탐지

임베딩 기법별 생성된 벡터를 활용하여 one-class SVM 알고리즘을 통한 이상치 탐지 결과를 Table 7과 같이 비교하였다. 네트워크 이벤트 데이터에 대하여, Autoencoder를 이용한 임베딩 결과 벡터의 경우 Fig. 5와 같이 AUROC 최대값 0.902를 기록하였으나, Doc2Vec을 활용해 생성한 벡터는 Fig. 6과 같이 auroc 최대값이 0.499를 기록하는 데 그쳤다. 레지스트리 이벤트 역시 Autoencoder를 이용한 임베딩 결과가 Fig. 7과 같이 0.857을 기록한 반면, Doc2Vec 임베딩 결과는 Fig. 8과 같이 0.734로 상대적으로 낮았다.

Table 7. Detection Results by Embedding Algorithms

Data	Embedding Algorithm	gamma	nu	AUROC
Network	Doc2Vec	0.03	0.2	0.499
	Autoencoder	0.3	0.1	0.902
Registry	Doc2Vec	1	0.5	0.734
	Autoencoder	1	0.9	0.857



Fig, 5. ROC curve on OneSVM detection of network event data embedded by Autoencoder

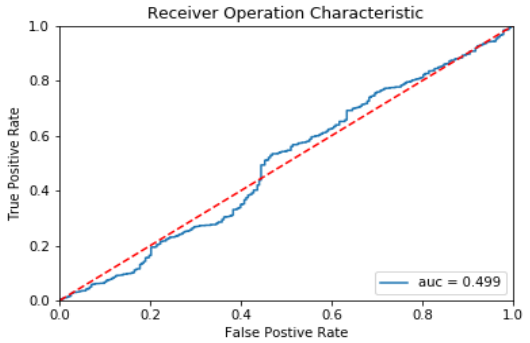


Fig. 6. ROC curve on OneSVM detection of network event data embedded by Doc2Vec

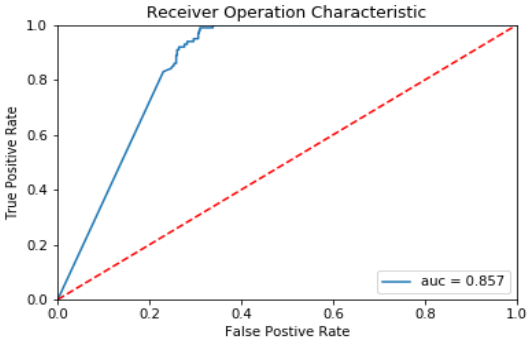


Fig. 7. ROC curve on OneSVM detection of registry event data embedded by Autoencoder

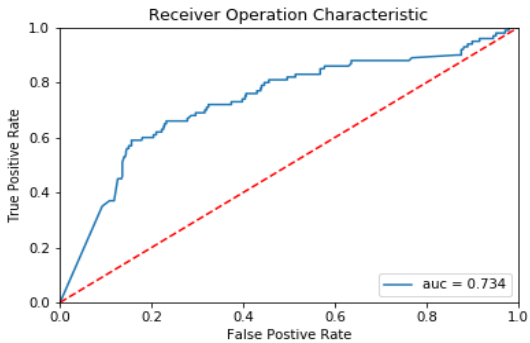


Fig. 8. ROC curve on OneSVM detection of registry event data embedded by Doc2Vec

이상의 결과와 같이 서로 다른 종류의 데이터라도 Autoencoder를 이용한 임베딩 결과 벡터가 Doc2Vec을 이용한 결과 벡터보다 탐지율이 더 높게 나타나, Autoencoder를 이용한 임베딩이 더 적

합한 임베딩 기법임을 확인할 수 있었다.

4.2.2.2 Isolation Forest를 이용한 이상상태 탐지

상기 검토한 바와 같이 one-class SVM 탐지 결과 더 적합하다고 판단된 임베딩 기법인 autoencoder를 통한 방식으로 임베딩 및 라벨링된 벡터를 이용하여 Isolation Forest 탐지를 수행하였다.

네트워크 이벤트 데이터를 이용한 Isolation Forest 탐지 수행 결과, Fig. 9와 같이 estimator 수가 20일 때 가장 높은 auroc 값인 0.773을 기록하였다. 레지스트리 이벤트 데이터를 이용한 Isolation Forest 탐지 수행 결과, Fig. 10과 같이 estimator 수가 100일 때 가장 높은 auroc 값인 0.693을 기록하였다.

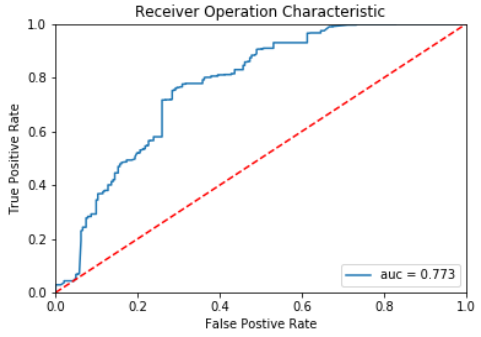


Fig. 9. ROC curve on Isolation Forest detection of network event data embedded by Autoencoder

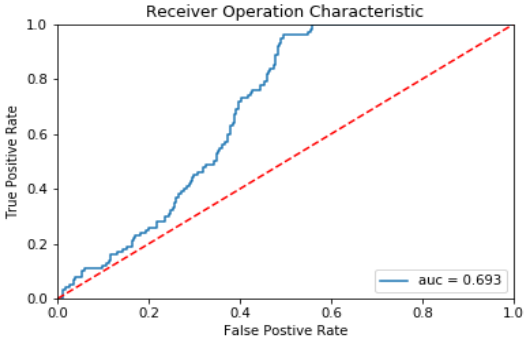


Fig. 10. ROC curve on Isolation Forest detection of registry event data embedded by Autoencoder

4.2.2.3 탐지 알고리즘별 결과 비교

Autoencoder를 이용한 임베딩 결과 벡터를 기반으로 OneSVM을 통한 탐지와 Isolation Forest를 통한 탐지를 수행해 본 바, Table 8과 같이 OneSVM을 통한 탐지가 Isolation Forest에 비하여 다소 높은 탐지율을 기록하는 것을 알 수 있었다.

Table 8. Detection Results by Detecting Algorithms

Data	Detecting Algorithm	parameters		AUROC
Network	OneSVM	gamma	0.3	0.893
		nu	0.1	
	Isolation Forest	n_estimator	20	0.773
Registry	OneSVM	gamma	1	0.857
		nu	0.9	
	Isolation Forest	n_estimator	100	0.693

V. 결론 및 향후 개선 방향

본 논문에서는 알려지지 않은 공격을 탐지하기 위해 정상 상태 데이터를 이용하여 시스템 정상 모델을 학습하고, 이를 이용하여 이상치 탐지를 수행하는 기법을 제안하였다. 2장에서 Word2Vec, Doc2Vec, Autoencoder를 이용한 임베딩 및 이상치 탐지 방법을 검토하였고, 3장에서 본 논문에서 제안하는 사이버 공격에 의한 시스템 이상상태 탐지 기법에 대하여 상세히 서술하였다. 4장에서는 실험에 사용된 데이터셋에 대하여 서술하고, 임베딩 및 탐지 알고리즘별 실험 결과를 비교하였다. 임베딩 알고리즘별 탐지 결과를 비교했을 때 Doc2Vec에 비하여 Autoencoder를 이용한 임베딩 결과가 월등했으며, 탐지 알고리즘별 결과를 비교했을 때 OneSVM이 Isolation Forest에 비해 우수한 결과를 기록하는 것을 확인할 수 있었다.

본 논문의 수집 데이터에서 사용된 여러 종류의 시스템 정보 중 정상상태와 비정상상태의 차이를 반영하지 못하여 탐지에 유용하지 아니한 정보가 있을 수 있으므로 이를 판별하여 수집 데이터 전처리 방식

을 개선할 수 있다. 따라서 향후 실 운용 환경에서 보다 유용하게 활용될 수 있도록 수집 데이터의 최적화를 위한 연구를 진행할 필요가 있다.

References

[1] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, & J. Dean, "Distributed representations of words and phrases and their compositionality," Advances in neural information processing systems, pp. 3111-3119, 2013.

[2] T. Mikolov, K. Chen, G. Corrado, & J. Dean, "Efficient estimation of word representations in vector space," International Conference on Learning Representations, Jan. 2013.

[3] Q. Le, & T. Mikolov, "Distributed representations of sentences and documents," International Conference on Machine Learning, May. 2014.

[4] I. Sutskever, O. Vinyals, & Q. V. Le, "Sequence to sequence learning with neural networks," Advances in Neural Information Processing Systems, pp. 3104-3112, 2014.

[5] MARKOU, M. AND SINGH, S., "Novelty detection: A review-part 1: Statistical approaches. Sig. Proc. 83, 12," pp. 2481-2497, 2003.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys (CSUR), 41(3):15, 2009.

[7] CHEN, Y., ZHOU, X., AND HUANG, T. S., "One-class SVM for learning in image retrieval," Proceedings of the IEEE International Conference on Image Processing (ICIP), 2002.

[8] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," Proceedings of the 8th IEEE International Conference on Data Mining, pp. 413-422, 2008.

[9] B. E. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, D. P. Miller, C. Wampler, S. M. Whitley, and R. D. Wolf, "Finding cyber threats with ATT&CK-based analytics," 2017.

〈저자소개〉

윤 여 정 (Yeo-jeong Yoon) 정회원
2009년: 서강대학교 수학과 졸업(이학사, 공학사)
2011년: 한국과학기술원 전산학과 졸업(공학석사)
2011년~현재: 국방과학연구소 선임연구원
〈관심분야〉 정보보호

정 유 진 (Yujin Jung) 정회원
2017년: 고려대학교 정보보호학부 졸업(공학사)
2017년~현재: 국방과학연구소 현역파견원
〈관심분야〉 정보보호