가우시안 혼합 변분 오토인코더를 활용한 네트워크 침입 탐지 시스템[1)]

성창민[O1], 송영록[2], 박종혁[2], 정윤경[2]

[1]성균관대학교 소프트웨어학과

[2]성균관대학교 인공지능학과

{tjdckdals,id4thomas,realkaya,aimecca}@ skku.edu

Network Intrusion Detection System using Gaussian Mixture Variational Autoencoder

ChangMin Seong[O1], YoungRok Song[2], JongHyeok Park[2], YunGyung Cheong[2]

[1]Department of Computer Software, Sungkyunkwan University

[2]Department of Artificial Intelligence, Sungkyunkwan University

요 약

네트워크 침입 탐지 시스템(NIDS)은 네트워크 상에서 발생하는 트래픽 중 악의적인 트래픽을 탐지하는 시스템이다. 본 논문에서는 비지도 클러스터링을 수행하는 가우시안 혼합 변분 오토인코더(Gaussian Mixture Variational Autoencoder) 모델로 학습된 잠재 군집 클래스 라벨 y로 정상 또는 공격을 판별하는 방식의 탐지 시스템을 제안한다. NIDS 평가 데이터셋인 NSL-KDD로 다른 지도 학습 모델들과의 탐지 성능을 비교하고, 테스트 데이터의 잠재 벡터 플롯을 그려 군집화 성능을 시각화하였다.

1. 서론

네트워크 침입 탐지 시스템(NIDS)은 네트워크 패킷들을 읽고, 그 중 비정상적인 패턴이 발생하는 악의적인 트래픽을 감지하는 시스템으로써 다양한 네트워크상의 보안 위협에 대한 효과적인 방어 방식으로 널리 사용되어 왔다. 최근에는 NIDS의 성능을 향상시키기 위하여 머신러닝(Machine Learning) 및 딥러닝(Deep Learning) 모델을 활용하는 방식이 활발하게 연구되고 있는데 크게 지도(Supervised)와 비지도(Unsupervised) 학습 방식으로 분류할 수 있다[1].

지도 학습 기반 방식은 정상 또는 공격으로 라벨링된 데이터로 학습되며 좋은 탐지 성능을 보이지만 라벨링된 데이터를 확보하는 것에 대해 비용적, 시간적 측면의 어려움이 있다. 반면 비지도 학습 기반 방식은 라벨 정보가 없는 데이터로도 데이터의 특징 정보를 추출하며 군집화(Clustering) 알고리즘 등으로 데이터를 정상과 공격으로 분리한다. 하지만 비지도 학습 기반 방식의 탐지 성능은 대체로 지도 학습 기반 방식보다 좋지 않다.

본 논문에서는 비지도 군집화를 수행하는 가우시안 혼합 변분 오토인코더(Gaussian Mixture Variational Autoencoder) 모델로 학습된 잠재 군집 분류 라벨 y로 정상 또는 공격을 판별하는 방식의 탐지 시스템을 제안한다. 해당 시스템의 효용성을 평가하기 위해 NIDS 평가 데이터셋인 NSL-KDD 데이터셋으로 다른 지도 학습 모델들과의 탐지 성능을 비교하고 테스트 데이터의 잠재 벡터(Latent Vector) 플롯을 그려 군집화 성능을 시각화하였다.

1) 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2020-0-00952, 5G+ 서비스 안정성 보장을 위한 엣지 시큐리티 기술 개발)

2. 관련 연구

2.1 비지도 기반 네트워크 침입 탐지 시스템

K-means 군집화 알고리즘은 비지도 학습 모델로써 데이터를 K개의 군집(Cluster)으로 묶는 알고리즘이다. 일반적으로 K-means는 L2-norm을 사용하여 각 데이터의 거리를 측정하여 같은 군집으로 묶는다. 안종하와 그의 동료는 p_logpack으로 압축한 로그 데이터 포인트의 차이를 정의하여 거리를 설정하고 K-means 알고리즘을 이용하여 학습을 하였다[2].

다른 환경에서 네트워크 침입 탐지 시스템을 적용한 이주화와 그의 동료는 비지도 학습 모델인 Stacked AutoEncoder 모델을 이용하여 IOT 환경에서의 네트워크 침입 탐지 시스템을 구축하였다. 이처럼 여러 분야에서 비지도 학습 기반의 탐지 방법에 대한 연구가 활발히 진행되고 있다[3].

2.2 가우시안 혼합 변분 오토인코더(GMVAE)

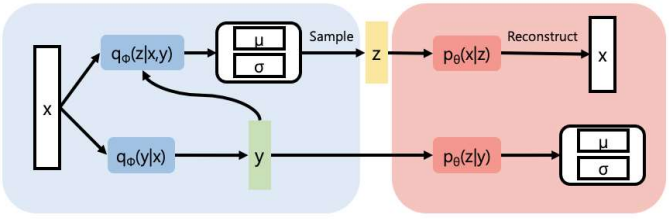
가우시안 혼합 변분 오토인코더, 즉 GMVAE 모델은 Rui Shu가 Kingma의 준지도 학습(Semi-supervised) 기반 생성 모델인 M2모델을 수정한 모델이다[4][5]. 일반적인 오

토인코더 모델은 크게 인코더(Encoder)와 디코더(Decoder)로 구성되어 있는데, 인코더로 입력 데이터를 압축한 뒤 디코더로 복원하는 과정을 거치면서 복원 손실을 최소화하도록 학습된다. 이러한 학습 과정을 거치면 인코더는 입력 데이터를 저차원의 잠재 벡터(Latent Vector)로 압축할 수 있다. 압축된 저차원의 잠재 공간(Latent Space)에서 변분 추론(Variational Inference)을 통해 잠재 벡터의 확률 분포를 추정하고, 추정된 잠재 공간의 확률 분포를 통해 복원하는 모델을 변분 오토인코더(Variational Autoencoder)라고 하며, 사전 확률(Prior Distribution)로 가우시안 혼합(Gaussian Mixture)을 사용하는 변분 오토인코더를 가우시안 혼합 변분 오토인코더라고 한다.

3. 실험 모델 및 실험 설계

3.1 실험 모델

본 논문에서 제시한 NIDS에 사용되는 GMVAE 모델의 구조는 [그림 1]과 같다. GMVAE는 2개의 추론 네트워크



[그림 1] GMVAE 모델 구조

인 $q_\Phi(y|x)$, $q_\Phi(z|x,y)$와 2개의 생성 네트워크인 $p_\theta(x|z)$, $p_\theta(z|y)$로 구성된다. 이때 x는 모델의 입력값, y는 군집 레이블 잠재 변수, z는 데이터 잠재 변수를 의미한다. 입력 x는 추론 단계와 생성 단계를 거친 후 복원된다.

추론 단계에서 x가 $q_\Phi(y|x)$ 네트워크의 입력으로 Softmax 활성화 함수를 거쳐 군집 레이블 y가 출력된다. 이후 x와 y가 $q_\Phi(z|x,y)$ 네트워크의 입력으로 주어져 평균(μ)과 분산(σ)이 출력된다. 이때 출력된 $q_\Phi(z|x,y)$ 네트워크의 분산은 Softplus 활성화 함수를 거친다. 이후 출력된 평균과 분산을 가지는 정규 분포에서 z가 샘플링된다.

생성 단계에서는 샘플링된 z가 $p_\theta(x|z)$ 네트워크를 통과하여 복원된 x가 출력된다. 이때 $p_\theta(x|z)$의 마지막 레이어를 통과 후 Sigmoid함수로 활성화 된다. $p_\theta(z|y)$ 네트워크는 y를 입력받아서 손실(loss) 함수 계산에 이용되는 사전 평균(Prior Mean)과 사전 분산(Prior Variance)을 출력하게 된다. 사전 분산 또한 Softplus 활성화 함수를 거친다.

GMVAE 모델은 모델의 ELBO를 최대화하도록 학습이 되며 ELBO의 수식은 다음과 같다.

$$E_{q_\phi(y,z|x)}[\log p_\theta(x|z) + \log p_\theta(z|y) - \log q_\phi(z|x,y) - \log q_\phi(y|x)]$$

본 논문에서는 ELBO의 복원 손실(Reconstruction Loss)로 이진 교차 엔트로피 손실(Binary Cross Entropy Loss) 함수를 사용하였다.

$q_\Phi(y|x)$로 예측된 군집 레이블 y를 공격 탐지 레이블로 변환하기 위하여 Jiang과 그의 동료들이 군집 레이블과 데이터셋 레이블간 최적의 매핑을 찾기 위해 활용한 KuhnMunkres 알고리즘을 사용하였다[6].

3.2 NSL-KDD 데이터셋

NIDS를 평가하기 위해 KDD Cup 99 데이터셋과 중복된 데이터를 제거한 NSL-KDD 데이터셋이 많이 사용되어 왔다[7][8]. 이에 따라, 본 논문에서는 NSL-KDD 데이터셋을 통해 실험을 하였다[9]. NSL-KDD 데이터셋은 훈련 데이터와 테스트 데이터로 나누어져 있고, 훈련 시 검증을 하기 위해 훈련 데이터의 10%를 검증 데이터로 사용하였다. [표 1]은 NSL-KDD 데이터셋의 통계를 나타낸 표이다.

[표 1] NSL-KDD 데이터 통계

	분류	훈련	검증	테스트
정상	정상	60,642	6,701	9,711
공격	DoS	41,325	4,602	7,458
	U2R	52	2	533
	R2L	906	87	2,421
	Probe	10,451	1205	2,421

공격 데이터로는 DoS, U2R, R2L, 그리고 Probe가 있으나 U2R 공격의 경우 데이터의 수가 적으므로 본 논문에서는 해당 데이터를 모두 제거하고 실험을 진행하였다.

3.3 데이터셋 전처리

먼저, 데이터 내에서 숫자 형태가 아닌 경우 숫자 형태로 바꾸어 주었다. 다음으로, 프로토콜(Protocol) 관련 부분은 3차원 벡터, 서비스(Service) 관련 부분은 86차원 벡터, 플래그(Flag) 관련 부분은 13차원 벡터로 나타내었다. 송신 포트 번호(Source Port Number)와 수신 포트 번호(Destination Port Number)는 각각 3차원 벡터로 나타내었다. 나머지 부분들도 벡터로 표현한 뒤 수치의 정규화를 위해 최소-최대 스케일러(Min-Max Scaler)를 사용하였고, 최종적으로 114차원의 벡터로 변환되었다.

3.3 성능 평가 지표

본 연구에서 NIDS의 성능 평가 지표로 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1점수를 사용하였으며 각 지표의 수식은 다음과 같다. 수식에서 TP는 True Positive, FP는 False Positive, TN은 True Negative, FN은 False Negative를 의미하며 공격을 양성 클래스라고 설정한다.

$$\text{accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \quad (1)$$

$$\text{precision} = \frac{TP}{TP+FP} \quad (2)$$

$$\text{recall} = \frac{TP}{TP+FN} \quad (3)$$

$$\text{f1} = \frac{2\times\text{recall}\times\text{precision}}{\text{recall}+\text{precision}} \quad (4)$$

4. 실험 결과

4.1 학습 파라미터 설정

에포크(Epoch)는 2000, 배치 크기(Batch size)는 8192, 학습률(Learning rate)은 1e-3으로 설정하여 Adam 최적화

알고리즘으로 학습을 진행하였다[10]. GMVAE의 은닉층(Hidden Layer) 활성화 함수는 ReLU 함수를 사용하였다.

4.2 잠재 벡터 크기에 따른 성능 비교

잠재 벡터의 크기에 따른 GMVAE 모델의 성능 변화를 비교하기 위해 잠재 벡터의 크기를 변화시키며 실험을 진행하였으며 각 성능 지표별 결과는 [표 2]와 같다.

[표 2] 잠재 벡터 z의 크기에 따른 탐지 성능

z 크기	accuracy	precision	recall	F1
2	**0.872**	0.920	**0.844**	**0.880**
4	0.8167	0.972	0.692	0.808
8	0.849	**0.977**	0.748	0.847
16	0.852	0.969	0.760	0.852
32	0.852	0.965	0.764	0.852

잠재 벡터의 크기가 2인 경우에 Accuracy, Recall 그리고 F1 점수가 가장 높았으며, 잠재 벡터의 크기가 8인 경우에 Precision 점수가 가장 높았다. [표 2]에서도 알 수 있듯이 잠재 벡터의 크기가 커질수록 오히려 전반적인 점수가 낮아지는 경향을 보였다.

4.3 GMVAE 모델과 지도 학습 모델의 성능 비교

GMVAE 모델과 지도 학습 모델의 성능 차이를 알아보기 위해 지도 학습 모델을 선정하여 같은 조건 하에 학습을 진행하였다. 지도 학습 모델로는 Random Forest, SVM, XGBoost를 선정하였으며 모델별 성능 점수는 [표 3]과 같다.

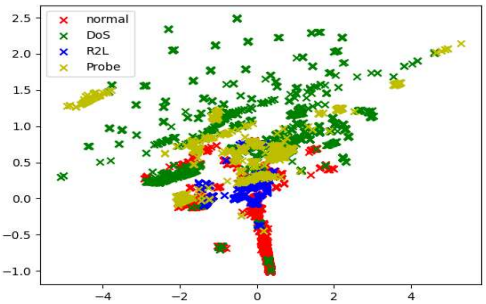
[표 3] 지도 학습 모델들과 탐지 성능 비교

모델	accuracy	precision	recall	F1
Random Forest	0.797	**0.968**	0.659	0.784
SVM	0.801	0.923	0.703	0.798
XGBoost	0.799	**0.968**	0.663	0.787
GMVAE	**0.872**	0.920	**0.844**	**0.880**

일반적으로 비지도 학습은 지도 학습에 비해 성능이 떨어진다고 알려져 있다. 하지만, GMVAE 모델의 성능은 전반적으로 지도 학습 모델보다 더 좋은 성능을 보여주는 것을 알 수 있다.

4.4 카테고리별 성능 비교

GMVAE 모델을 이용하여 카테고리별 성능을 비교하였다. 공격 카테고리별 탐지율은 Probe가 0.997로 가장 높았으며, 다음으로는 DoS가 0.9063으로 높았고, 마지막으로 R2L이 0.50으로 높았다.



[그림 2] 테스트 데이터의 잠재 벡터 z의 분포

[그림 2]는 크기가 2인 잠재 벡터 z를 각 카테고리별로 시각화 한 그림이다. 정상 데이터와 공격 카테고리별 데이터의 군집이 잘 분리되는 모습을 보여준다.

5. 결론 및 향후 연구 계획

본 논문에서는 비지도 학습 방식의 GMVAE 모델을 통한 NIDS를 제안하여 네트워크 침입 탐지 성능을 평가하였다. 실험 결과 GMVAE 모델이 지도 학습 방식의 모델보다 좋은 성능을 보이는 것을 확인할 수 있었다. 또한 테스트 데이터의 잠재 벡터 z를 시각화여 정상 데이터와 공격 데이터의 잠재 벡터가 분리되는 것을 확인할 수 있었다. 향후 R2L과 같이 탐지 성능이 낮은 공격 카테고리에 대해 정상과 같이 군집 되는 원인을 분석함으로써 탐지 성능을 높일 수 있는 방향으로 연구해 볼 수 있을 것이다.

참 고 문 헌

[1] Liu, Hong and Bo Lang. “Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey.” *Applied Sciences* 9, pp.4396, 2019.

[2] 안종하,김대원.“K-means 클러스터링을 이용한 압축 기반 이상탐지”,*정보과학회논문지: 소프트웨어 및 응용 39.8*, pp.605-612, 2012.

[3] 이주화,박기현.“IOT 환경에서의 오토인코더 기반 특징 추출을 이용한 네트워크 침입탐지 시스템”,*정보처리학회논문지/소프트웨어 및 데이터 공학 제 8(12)*, pp.483-490, 2019.

[4] Rui Shu, “Gaussian Mixture VAE: Lessons in Variational Inference, Generative Models, and Deep Nets”,http://ruishu.io/2016/12/25/gmvae/, 2016.

[5] Kingma, Diederik P. et al. “Semi-supervised Learning with Deep Generative Models.”,*NIPS*, pp.3581-3589, 2014.

[6] Jiang, Zhuxi et al. “Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering.”,*IJCAI*, 2017.

[7] Suchet Sapre, Pouyan Ahmadi and Khondkar Islam, “A Robust Comparison of the KDDCup99 and NSL-KDD IoT Network Intrusion Detection Datasets Through Various Machine Learning Algorithms” ,*arXiv:1912.13204*, 8, 2019.

[8] K. Siddique, Z. Akhtar, F. Aslam Khan and Y. Kim, "KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research," *in Computer, vol. 52, no. 2*, pp. 41-51, 2019.

[9] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set ", *IEEE symposium on computational intelligence for security and defense applications*, pp.1-6, 2009.

[10] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", *International Conference on Learning Representations (ICLR), San Diego, CA, USA*, pp.1-15, 2015.