경량 IoT 를 위한 오토 인코더 기반의 데이터 압축 기법

김연진 [1], 박나은 [2], 이일구 [3]
[1] 성신여자대학교 융합보안공학과 석사과정
[2] 성신여자대학교 미래융합기술공학과 박사과정
[3] 성신여자대학교 융합보안공학과, 미래융합기술공학과 교수

duswlsqhfk@gmail.com, nepark.cse@gmail.com, iglee@sungshin.ac.kr

Autoencoder-based Data Compression Technique for Lightweight IoT

Yeon-Jin Kim[1], Na-Eun Park [2], Il-Gu Lee[3]
[1]Dept. of Convergence Security Engineering, Sungshin Women's University
[2]Dept. of Future Convergence Technology Engineering, Sungshin Women's University
[3]Dept. of Convergence Security Engineering, Future Convergence Technology Engineering, Sungshin Women's University

요 약

IoT 가 전 산업에 널리 활용되면서 생성되는 데이터 양이 급증하고 있다. 그러나 경량, 저가, 저전력 IoT 는 대용량 데이터를 처리, 저장, 전송하기 어렵다. 그러나 이러한 문제를 해결하기 위한 종래의 방법들은 복잡도와 성능의 트레이드오프 문제가 있다. 본 논문은 IoT 기기의 효율적 리소스 사용을 위한 오토 인코더 데이터 압축 기법을 제안한다. 실험 결과에 따르면 제안한 기법은 종래 기술에 비해 평균 60.61% 축소된 데이터 크기를 보였다. 또한, 제안된 기법으로 압축된 데이터를 사용하여 모델 학습을 진행한 결과에 따르면 RNN 과 LSTM 모델에 제안한 방법을 적용했을 때 모두 97% 이상의 정확도를 보였다.

1. 서론

사물 인터넷(Internet of Things, IoT)이 전 산업과 일상생활에 널리 활용되면서 가전제품, 차량, 건강 모니터링 장치, 스마트 시티 시설 등 다양한 사물들이 서로 네트워크를 통해 연결되고 데이터를 교환할 수 있게 되었다. 그러나 IoT 기기가 여러 방면에 사용되면서 IoT 기기를 대상으로 한 사이버 공격이 증가하고 있다. 최근 악의적으로 IoT 기기의 제한된 리소스를 활용한 분산 서비스 공격을 수행하여 기기의 가용성을 침해하는 공격을 하거나, 취약한 초기 비밀번호 설정을 이용해 봇넷(botnet)을 구축하는 등의 공격으로 인한 피해가 발생하고 있다.

IoT 기기가 악의적 사용자들의 주요 공격 대상이 되는 이유는 IoT 기기의 제한된 리소스 환경으로 인해 큰 컴퓨팅 파워를 필요로 하는 침입 탐지 모델이나 복잡한 보안 메커니즘을 구축하기 어렵기 때문이다[1]. 최근에는 IoT 기기의 보안을 강화하고 부족한 리소스를 효율적으로 사용하기 위한 경량 탐지 모델이나 데이터 압축 방식에 대한 연구가 활발하게 진행되고 있다[2, 3].

특히, 웨어러블 기기나 디지털 헬스케어 기기는 일상 생활에 널리 활용되고 있으며 수많은 데이터를 생성하고 있다. 기기가 생성한 대량의 데이터를 효율적으로 저장하고 전송하기 위해서는 데이터 압축 기술이 필수적이다[4]. 대용량 데이터 압축은 IoT 기기의 리소스를 절약할 수 있지만, 압축 데이터로 기계 학습, 데이터 분석을 진행할 경우 모델 성능 저하나 추가적인 연산 비용이 발생할 수 있다[5].

이러한 문제를 해결하기 위해 본 논문은 IoT 기기의 데이터를 압축함으로써 발생하는 성능 저하와 추가적인 연산 비용 간 트레이드 오프를 완화할 수 있는 데이터 압축 기술을 제안한다. 비지도 학습을 수행하는 인공 신경망인 오토 인코더(Autoencoder, AE)를 활용하여 입력 데이터를 저차원의 잠재 표현으로 압축한다. 이후 압축된 데이터를 재구성하여 입력 데이터 손실 값을 최소화하는 디코딩 단계를 거친다. 이 과정을 통해 오토 인코더는 데이터의 주요 특징을 학습하여 효율적인 데이터 압축을 가능하도록 한다.

논문의 기여점은 다음과 같다.

- 경량 IoT 를 위한 오토 인코더 기반의 고효율 데이터 압축 기법을 제안한다.
- IoT 의 데이터 전송 방식의 효율을 비교 분석할 수 있는 평가 프레임워크를 제안한다.

본 논문의 구성은 다음과 같다. 2 장에서는 데이터 압축 기술을 활용한 네트워크 탐지 기술에 대한 종래 연구 분석을 진행한다. 3 장에서는 오토 인코더를 활용한 데이터 압축 기법을 제안하고, 4 장에서는 실험 환경을 설명하고 결과를 분석한다. 5 장에서는 결론을 맺는다.

2. 선행 연구 분석

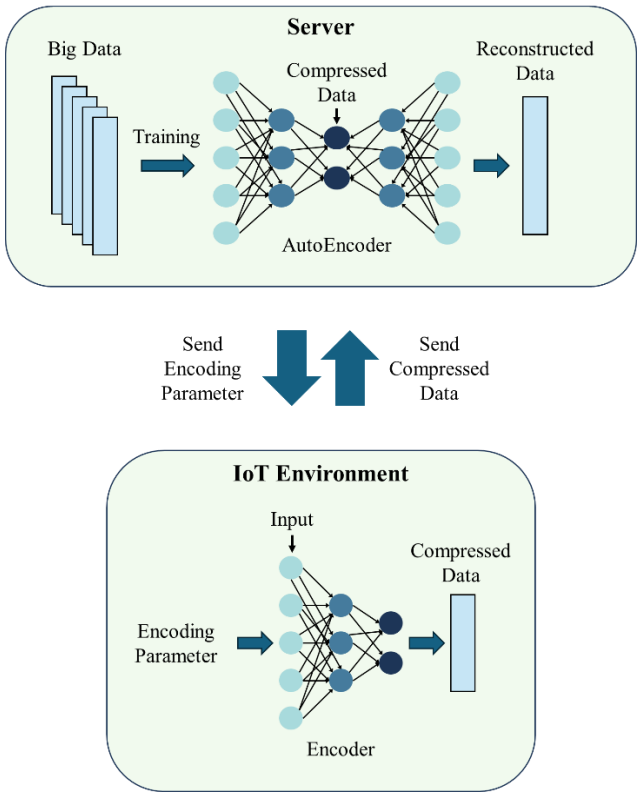
Sarhan, M [2]은 IoT 네트워크 기반 침입 탐지 시스템 기술 개선을 위해 특성 축소 기술과 기계 학습 기술의 성능을 평가했다. 본 연구에서는 실험 결과 데이터 세트와 특성 축소 기술, 그리고 기계 학습 알고리즘의 다양한 조합을 탐색하고 각 데이터 세트에 가장 적합한 기술 조합을 찾아냈다. 그러나 탐지 모델 성능 평가를 위해 정확도와 수신자 운영 특성만을 지표로 고려하여, 사물 인터넷 환경에 적합한 모델을 식별하기 위한 평가 지표가 부족하다는 한계를 가지고 있다.

Alaghbari, K.A [3]은 사물 인터넷 대상의 사이버 공격을 다중 분류하기 위한 이상 탐지 및 특징 추출 심층 오토 인코더를 제안했다. 제안 기술의 검증을 위해 이상 탐지를 수행하는 오토 인코더 성능 분석 실험을 진행했다. 제안한 오토 인코더의 성능은 OC-SVM(One-Class Support Vector Machine)과 동일했으나 학습 및 탐지 시간이 약 37% 개선되었다. 추가로 주성분 분석, 선형 판별 분석과 제안된 오토 인코더를 사용해 특징의 차원을 축소하고 DT(Decision Tree), XTree(eXtreme Tree), RF(Random Forest), DNN(Deep Neural Network) 모델을 학습한 결과를 비교했다. 제안한 기술은 차원 축소를 진행하지 않은 모델 대비 약 2%의 정확도가 감소했으나 학습 및 탐지 시간이 약 13% 개선되었다. 그러나 이 연구는 오토 인코더의 은닉층에 대한 뉴런(neural) 최적화 과정 거치지 않고 실험을 진행하여, 논문에서 제안하는 오토 인코더가 최적의 성능을 보이는 데에 대한 근거가 부족하다는 한계점이 있다.

종래 연구는 IoT 환경에서 탐지 모델의 연산 복잡도를 개선하기 위해 데이터를 압축하고 모델의 성능을 유지하는 방안을 연구했다. 그러나 종래 연구에서 제안하는 기술은 리소스 제한된 IoT 환경을 고려하지 않았고, 탐지 모델의 성능에 집중된 연구가 많았다.

3. 오토 인코더를 활용한 데이터 압축 기법

본 장에서는 오토 인코더를 활용한 데이터 압축 기법에 대해 설명한다. 제안하는 압축 기법을 활용한 전체적인 구조는 그림 1 과 같다.



(그림 1) Flowchart of the proposed technique

서버는 대량의 데이터를 사용해 오토 인코더를 학습한다. 오토 인코더는 입력 데이터를 저차원으로 압축하는 인코더(encoder)와 압축된 데이터를 복원하는 디코더(decoder)로 구성된다. 오토 인코더는 입력 데이터와 재구성된 출력 데이터 간의 손실을 최소화하는 것을 목표로 학습한다. 50 에포크(epoch) 동안 학습하며, 검증 데이터에 대한 손실 값과 정확도가 임계점에서 수렴하는 결과를 보이면 학습을 조기 종료하여 학습 시간을 단축한다.

학습이 완료된 오토 인코더는 인코딩 파라미터를 IoT 기기로 전송한다. IoT 기기는 수신한 인코딩 파라미터로 인코더를 재구성해 데이터를 압축한다. IoT 기기에 저장된 데이터는 압축된 상태로 서버에 전송되며, 서버는 디코더를 사용하여 압축된 데이터를 복원함으로써 IoT 기기의 제한된 리소스를 효율적으로 활용할 수 있다.

4. 성능 평가 및 분석

본 논문에서 제안하는 데이터 압축 기법의 검증을 위해 파이썬(python)을 이용하여 실험을 진행했다. 제

안 기법은 전통적인 데이터 압축 기술인 주성분 분석과 비교했다. 실험에 사용된 데이터 세트는 UNSW-NB15 이며, 총 2,540,047 개의 데이터를 포함하고 있다. 이 데이터 세트는 총 44 개의 특성을 가지고 있으며, 전 처리 과정을 통해 모델 성능에 영향을 줄 수 있는 특성과 범주형 데이터 특성을 삭제하여 총 38 개의 특성을 사용했다.

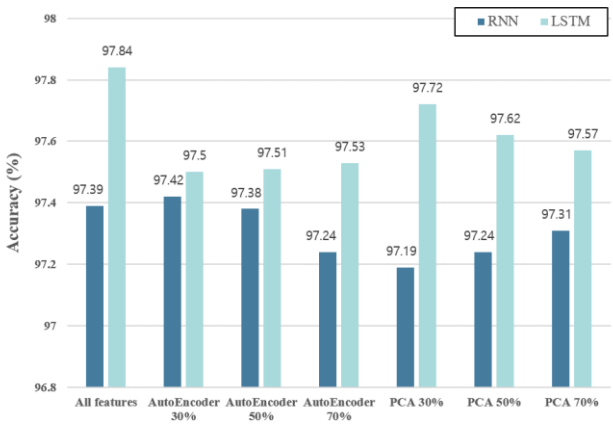
오토 인코더를 활용하여 데이터를 각각 30%, 50%, 70%로 압축하고, 종래 기술인 주성분 분석 또한 동일한 압축률로 데이터를 압축하여 실험했다. 각각의 압축 단계에서 압축된 데이터의 크기와 압축 데이터 복원율을 비교한 결과는 표 1 과 같다. 표 1 의 데이터 복원율은 평균 제곱 오차 계산을 통해 산출했다.

<표 1> Comparison results of reconstruction rate and data size

Data Compression Technique	Reconstruction Rate(%)	Data Size(MB)
AutoEncoder 30%	98.38	454
AutoEncoder 50%	97.96	373
AutoEncoder 70%	98.06	229
PCA 30%	99.05	1230
PCA 50%	94.31	922
PCA 70%	76.82	529

오토 인코더로 압축된 데이터는 모두 97%가 넘는 데이터 복원율을 보였다. 반면에 종래 기술로 압축된 데이터는 50%의 압축률까지 94%가 넘는 데이터 복원율을 유지했으나, 데이터 압축률을 70%까지 높이면, 복원율이 17.49% 감소했다. 압축된 데이터의 크기를 비교하면 오토 인코더로 압축된 데이터가 종래 기술을 통해 압축된 데이터보다 평균적으로 541.67MB 절감할 수 있었다. 실험 결과에 따르면 오토 인코더를 활용하여 데이터를 압축함으로써 종래의 기술에 비해 높은 복원율을 유지하면서도 데이터 크기를 작게 압축할 수 있음을 확인할 수 있었다.

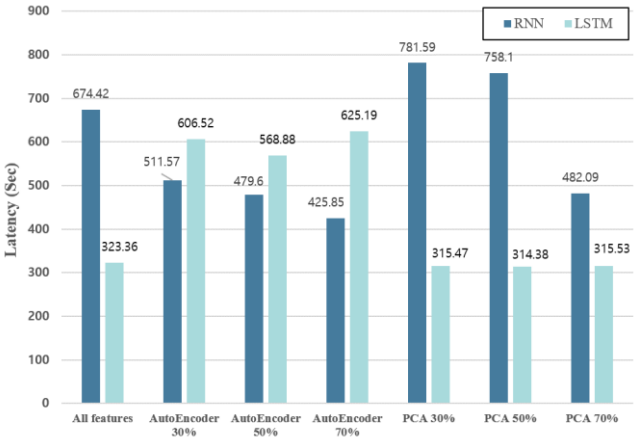
또한, 원본 데이터로 학습한 모델과의 성능, 학습 및 탐지 시간의 차이를 비교하기 위해 실험을 수행했다. 원본 데이터, 오토 인코더로 압축된 데이터, 종래 기술로 압축된 데이터를 사용하여 RNN(Recurrent Neural Network), LSTM(Long Short-Term Memory) 모델을 학습한 후 얻은 정확도 그래프는 그림 2 와 같다.



(그림 2) Accuracy of RNN and LTSM for various conditions

RNN 과 LSTM 의 기계 학습을 진행하고 정확도를 측정한 결과에 따르면 모든 모델이 97%를 넘는 정확도를 보였다. RNN 을 사용하면, 제안된 기법으로 30% 압축된 데이터를 이용해 학습된 모델이 97.42%의 가장 높은 정확도를 기록했다. LSTM 을 사용하면 제안한 기법이 원본 데이터로 학습된 모델보다 정확도가 평균 0.33% 감소했다. 종래의 기술과 비교하면 제안된 기법을 사용한 모델의 정확도가 평균 0.13% 감소되었다.

정확도를 비교한 실험 결과를 통해 제안 기법으로 데이터를 압축하여 모델 학습을 진행해도 모델의 성능이 원본 데이터를 사용해 학습된 모델과 수렴하는 정확도를 보이고 있음을 확인할 수 있었다. 이는 제안 기법으로 압축된 데이터가 원본 데이터의 주요 특성을 충분히 유지한다는 것을 시사한다. 모델의 학습 및 탐지 시간을 측정한 그래프는 그림 3 과 같다.



(그림 3) Latency of RNN and LTSM for various conditions

RNN 모델에 제안한 압축 기법을 적용했을 때 가장 적은 학습 및 탐지 시간이 소요되었다. 원본 데이터

를 사용하여 학습한 모델보다 평균적으로 29.02%의 학습 및 탐지 시간을 단축했고, 종래 기술보다 학습 및 탐지 시간을 평균 27.41% 단축했다. LSTM 모델에 제안한 압축 기법을 적용했을 때 학습 및 탐지 시간이 약 300 초 증가했다. 총 학습 및 탐지 시간이 증가한 이유는 오토 인코더의 학습 및 탐지를 위한 데이터 처리 시간 때문인 것으로 분석되었다. 오토 인코더를 학습하고 데이터를 압축한 후, LSTM 모델의 학습을 수행한 모델의 학습 및 탐지에 소요된 시간은 표 2 와 같다.

오토 인코더로 압축한 데이터를 사용한 LSTM 모델의 학습 및 탐지 시간은 평균 317.61 초로 측정되었다. 원본 데이터로 LSTM 모델을 학습하는 것보다 학습 및 탐지 시간을 1.78% 단축할 수 있었다.

<표 2> LSTM model latency after data compression using autoencoder

Compression Rate	30%	50%	70%
AE Training Time (Sec)	231.07	192.34	248.9
Data Compression Time (Sec)	58.6	58.71	58.13
LSTM Training Time (Sec)	307.56	308.63	309.07
LSTM Testing Time (Sec)	9.29	9.2	9.09

5. 결론

IoT 기술이 발전함에 따라 IoT 기기를 통해 생성되는 데이터의 양이 폭발적으로 증가하고 있다. 그러나 종래의 IoT 기기는 제한된 리소스 환경으로 생성된 데이터를 처리하고 전송하는 데에 어려운 문제가 있다. 이러한 문제를 해결하기 위해서 데이터 압축 기술에 관한 다양한 연구가 진행 중이지만 과도한 데이터 압축은 데이터 분석 비용을 증가시키거나, 압축 데이터로 학습된 기계 학습 모델의 성능이 저하될 수 있다는 문제가 있다.

본 논문에서는 종래의 문제를 해결하기 위해 오토 인코더를 활용한 데이터 압축 기법을 제안했다. 제안하는 기법의 성능 검증을 위해 데이터 복원율과 압축된 데이터의 크기를 종래의 기술과 비교 분석했다. 실험 결과에 따르면, 제안 기법이 종래 기술에 비해 평균 8.08% 개선된 압축 데이터 복원율을 보였으며, 평균 60.61%의 데이터 크기가 줄었다.

또한, 제안한 기법을 활용하여 압축된 데이터로 RNN 및 LSTM 모델을 학습한 결과에 따르면 모든 모델이 97% 이상의 정확도를 보였다. 그리고 제안한 기법을 사용한 RNN 모델의 학습 및 탐지 시간은 원본 데이터로 학습한 모델에 비해 평균 29.02% 감소했다. 이와 같이 본 연구에서 제안한 IoT 기기 데이터 압축 기법이 데이터의 크기를 줄이고 원본 데이터의 중요 특성은 유지할 수 있음을 입증했다.

그러나 압축된 데이터의 복원율이나 압축된 데이터로 학습된 모델의 성능, 학습 및 탐지 시간은 사용되는 데이터 세트에 의해서 달라진다는 것을 확인했다. 따라서 후속 연구로 데이터 세트와 상관없이 최적의 데이터 압축률과 압축 데이터의 성능을 보이는 오토 인코더 활용 압축 기법을 연구, 적용하여 성능을 검증할 계획이다.

ACKNOWLEDGEMENT

본 논문은 2024 년도 산업통상자원부 및 한국산업기술진흥원의 산업혁신인재성장지원사업 (RS-2024-00415520)과 과학기술정보통신부 및 정보통신기획평가원의 ICT 혁신인재 4.0 사업의 연구결과로 수행되었음 (No. IITP-2022-RS-2022-00156310)

참고문헌

[1] Yaacoub, J-P.A et al., "Ethical hacking for IoT: Security issues, challenges, solutions and recommendations," in Internet of Things and Cyber-Physical Systems, vol. 3, pp. 280-308, 2023.

[2] Sarhan, M et al., "Feature extraction for machine learning-based intrusion detection in IoT networks," in Digital Communications and Networks, vol. 10, 1, pp. 205-216, 2024.

[3] Alaghbari, K.A et al., " Deep Autoencoder-Based Integrated Model for Anomaly Detection and Efficient Feature Extraction," in IoT Networks, vol. 4, 3, pp. 345-365, 2023.

[4] Correa, J.D.A et al., "Lossy Data Compression for IoT Sensors: A Review," in Internet of Things, vol. 19, pp. 100516, 2022.

[5] Hafeez et al., "Using Dynamic Perceptually Important Points for Data Reduction in IoT," in Proceedings of the 11th International Conference on the Internet of Things, New York, pp. 33-39, 2022.