KIPS Trans. Softw. and Data Eng.
Vol.8, No.12 pp.483~490 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2019.8.12.483

Network Intrusion Detection System Using Feature Extraction Based on AutoEncoder in IOT environment

Joohwa Lee† · Keehyun Park††

ABSTRACT

In the Network Intrusion Detection System (NIDS), the function of classification is very important, and detection performance depends on various features. Recently, a lot of research has been carried out on deep learning, but network intrusion detection system experience slowing down problems due to the large volume of traffic and a high dimensional features. Therefore, we do not use deep learning as a classification, but as a preprocessing process for feature extraction and propose a research method from which classifications can be made based on extracted features. A stacked AutoEncoder, which is a representative unsupervised learning of deep learning, is used to extract features and classifications using the Random Forest classification algorithm. Using the data collected in the IOT environment, the performance was more than 99% when normal and attack traffic are classified into multiclass, and the performance and detection rate were superior even when compared with other models such as AE-RF and Single-RF.

Keywords : NIDS, IOT, Unsupervised Learning, Machine Learning, AutoEncoder

IOT 환경에서의 오토인코더 기반 특징 추출을 이용한 네트워크 침입탐지 시스템

이 주 화† · 박 기 현††

요 약

네트워크 침입 탐지 시스템(NIDS)에서 분류의 기능은 상당히 중요하며 탐지 성능은 다양한 특징에 따라 달라진다. 최근 딥러닝에 대한 연구가 많이 이루어지고 있으나 네트워크 침입탐지 시스템에서는 많은 수의 트래픽과 고차원의 특징으로 인하여 속도가 느려지는 문제점이 있다. 따라서 딥러닝을 분류에 사용하는 것이 아니라 특징 추출을 위한 전처리 과정으로 사용하며 추출한 특징을 기반으로 분류하는 연구 방법을 제안한다. 딥러닝의 대표적인 비지도 학습인 Stacked AutoEncoder를 사용하여 특징을 추출하고 Random Forest 분류 알고리즘을 사용하여 분류한 결과 분류 성능과 탐지 속도의 향상을 확인하였다. IOT 환경에서 수집한 데이터를 이용하여 정상 및 공격트래픽을 멀티클래스로 분류하였을 때 99% 이상의 성능을 보였으며, AE-RF, Single-RF와 같은 다른 모델과 비교하였을 때도 성능 및 탐지속도가 우수한 것으로 나타났다.

키워드 : 네트워크 침입탐지시스템, 사물인터넷, 비지도학습, 기계학습, 오토인코더

1. 서 론

최근 몇 년 동안 IOT(Internet of Things)의 확산은 전세계 사회에 널리 보급 되었다. IOT 장치의 수는 2017년에 270억 개에 달했고 이 IOT 장치는 시장 수요에 따라 기하급수적으로 증가할 것이다[1].

IOT 기술의 발달로 다양한 영역에서 민감 정보가 생성되고 있으며, 그 데이터의 양도 비약적으로 증가하고 있다. 이러한 정보들에 대한 보안이 어느 때 보다도 필요한 시점이므로 IOT 환경에서의 침입탐지 시스템은 반드시 필요하다 할 수 있다.

네트워크 침입 탐지 시스템(Network Intrusion Detection System:NIDS)은 네트워크 트래픽을 모니터링 하여 악의적인 활동을 탐지하는 시스템이다. 보안 방어의 중요한 기술인 침입탐지는 네트워크 보안의 핵심 기술이 되었다. 그러나 침입 방법이 다양해지고 새로운 형태의 침입이 증가함에 따라

※ 이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2018R1D1A1B07043982).
† 준 회 원 : 계명대학교 컴퓨터공학과 박사수료
†† 종신회원 : 계명대학교 컴퓨터공학과 교수
Manuscript Received : April 22, 2019
First Revision: June 3, 2019
Accepted : July 30, 2019
* Corresponding Author : Kee Hyun Park(khp@kmu.ac.kr)

전통적인 침입 탐지 방식은 현재의 네트워크 환경을 만족시킬 수 없었다. 이에 기계 학습을 응용한 침입탐지 방식이 증가하고 있는 추세이다[2].

침입탐지 시스템은 탐지 방식에 따라 오용탐지 방식과 비정상행위 탐지 방식으로 나눌 수 있다. 오용탐지 방식은 이미 알려져 있는 공격 행위로부터 특정 시그니처를 추출하고, 분석 대상에 그런 시그니처가 존재하는 경우 침입이라고 판단하는 방식이며, 비정상 탐지 방식은 정상적이고 평균적인 상태를 기준으로 하여, 이에 상대적으로 급격한 변화를 일으키거나 확률이 낮은 일이 발생하면 침입으로 규정하는 방식이다[3].

비정상탐지 기반 NIDS를 구현하기 위해서는 주로 기존에 많이 사용되었던 기계학습 알고리즘을 사용하거나 이미지 인식 및 음성 인식에서 높은 성능을 보이는 심층 인공 신경망을 사용하고 있다[4-8].

특히 머신러닝 알고리즘을 대규모 침입탐지 시스템에 실제로 사용하는 경우 대부분 시간 복잡성과 공간 복잡성의 한계에 직면한다. 본질적인 원인은 고차원의 특징과 비선형 특성을 갖는 입력 데이터에 기인한다[9].

따라서 고차원의 특성을 저차원으로 축소하는 것이 침입탐지의 필수적인 단계라고 할 수 있다. 또한, IOT 환경에서의 네트워크 공격이 다양해짐에 따라 새로운 유형의 공격이 발생하였을 때 탐지를 못하거나 희소의 공격 데이터인 경우 학습의 양이 적어 탐지가 어렵다는 것이 현재의 문제점이라 할 수 있다. 또한, IOT 환경에서의 데이터 수집은 한계가 있고 공인 데이터셋의 종류도 다양하지 않다는 문제점도 있다.

위와 같은 문제를 해결하기 위해 본 논문에서는 IOT 환경에서 수집한 데이터셋을 사용하여 딥러닝과 트리기반 통계적 학습의 장점을 결합하여 정확성과 효율성을 향상시키는 하이브리드 방식을 제안하고자 한다. 대표적인 비지도 학습인 AutoEncoder를 기반으로 침입 행위 정보의 고수준 특징을 압축한 후 최소의 시간으로 최적의 분류를 할 수 있는 모델을 제안 한다.

기존의 NIDS는 데이터의 모든 특징을 사용하거나 중요한 특징을 선택하여 분류에 사용하였다. 그러나 본 논문에서 제안한 AutoEncoder는 데이터가 가지고 있는 모든 특징을 고차원에서 저차원으로 압축하므로 분류 성능을 향상시키고 학습 및 분류시간을 줄일 수 있다.

실험 방법은 IOT 환경에서의 침입탐지 외에 기존 네트워크 환경에서도 제안한 모델이 적용되는지 확인하기 위하여 최신의 공격이 포함된 CICIDS 2017 데이터셋을 사용하여 성능을 평가하고 비교한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련연구와 연구에 포함된 관련기술에 대해 살펴본다, 제 3장에서는 본 논문에서 제안하는 모델의 설계와 구현에 대해 기술한다. 제 4장에서는 제안한 모델의 실험과 성능을 평가하고 마지막으로 5장에서는 결론과 향후 연구 계획을 기술한다.

2. 관련 연구

2.1 관련 연구

침입탐지 시스템과 관련하여 기존의 네트워크 환경에서의 연구는 많이 이루어지고 있다. 그러나 IOT 환경에서의 침입탐지시스템은 복잡한 네트워크 구조와 다양한 침입 경로 및 공인 데이터셋의 부족으로 인하여 연구의 어려움이 있다.

최근 IOT의 침입탐지 연구로 머신러닝을 적용한 모델들이 연구 되고 있다. IOT와 스마트 시티 보안을 위하여 반지도 심층 강화 학습 모델을 적용하여 레이블이 없는 데이터에서도 높은 성능을 보이는 연구를 제안하였다[10].

대부분의 연구에서 IOT 공인 데이터셋의 부족으로 인하여 기존 네트워크 환경의 데이터셋을 이용한 실험을 하고 있다. 최신의 공격이 포함된 CICIDS2017 데이터셋을 이용하여 딥러닝 모델과 머신러닝의 알고리즘을 비교 평가한 연구가 있으며[11] CICIDS2017 데이터셋과 IOT 데이터셋인 UNSW-15 데이터셋을 이용하여 하이브리드 비정상 행위 탐지 모델을 제안하였다[12].

IOT 공인데이터셋을 사용한 연구로 사물의 인터넷 보안 모니터링을 위해 개발된 “detection_of_IoT_botnet_attacks_N_ BaIoT”[13] 데이터셋을 이용하여 5가지 머신러닝 알고리즘으로 분류 성능을 측정하였으며 실험결과 공격 탐지의 정확도가 높게 나타났다[14].

특징 추출에 관한 연구로 기존 방식에서 많이 사용된 방법은 네트워크 트래픽의 데이터 차원을 줄이기 위한 전처리 단계로 일부의 특징을 선택하여 사용하는 방법이 있다[15].

좀 더 나아가 정보 손실 없이 특징의 크기를 줄이는 연구가 이루어졌다. 주성분분석(Principal Component Analysis, PCA)을 이용한 특징감소와 선형판별분석을 이용한 특징 감소 방법이 연구되었으며 PCA의 성능이 선형판별보다 더 높게 나타났다[16].

그러나 PCA는 특징 간의 비선형 상관관계를 포착하지 못하는 선형 변환이다. 네트워크 이상 탐지의 대부분의 특징은 비선형적이므로 많은 거짓 정보를 유발할 수 있다[17].

따라서 최근의 특징 추출 연구로 SAE(Sparse Autoencoder)와 SVM(Support Vector Machine)을 결합하여 STL(Self Taught Learning) 프레임 워크에 기반한 학습법을 제안하여 특징 학습과 차원 감소에 사용하였다. 학습 및 테스트 시간을 대폭 줄이고 SVM의 예측 정확도를 효과적으로 향상시켰다[18].

2.2 AutoEncoder

AE(AutoEncoder)는 Input Layer, Hidden Layer 및 Output Layer로 구성되어 있으며 비지도 3계층 신경망 모델이다[19]. AE는 Input Layer의 데이터를 짧은 코드로 압축한 다음 해당 코드를 원래 데이터와 거의 일치하도록 하는 인공 신경망 모델이다. 비지도 학습 모델이므로 레이블이 없는 데이터의 특징도 추출할 수 있는 장점이 있다. AE의 구조

는 Fig. 1과 같다.

input vector $X \in [0,1]^D$이고, hidden representation $Y \in [0,1]^d$, reconstructed vector $Z \in [0,1]^D$이다. Input Layer에서 Hidden Layer로의 코딩 프로세스는 Equation (1)과 같다.

$$y = f_\theta(X) = s(WX + b) \tag{1}$$

Hidden Layer에서 Output Layer로의 디코딩 프로세스는 Equation (2)와 같다.

$$Z = g_{\theta'}(Y) = s(W'Y + b') \tag{2}$$

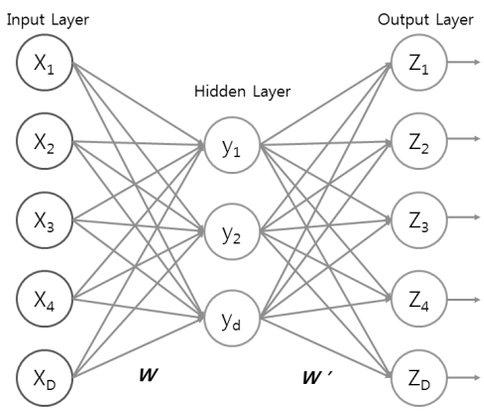

Fig. 1. The Structure of Basic AutoEncoder

이때 mapping 하기 전의 Input Layer의 weight matrix인 전치행렬 W^T가 remapping의 weight matrix인 W'와 같은 경우 해당 AE는 tied weight를 가졌다고 한다.

b와 b'는 Input Layer와 Hidden Layer의 각각의 바이어스 벡터이다. f_θ와 $g_{\theta'}$는 Hidden Layer 뉴런과 Output Layer뉴런의 활성 함수이다. 이 연구에서는 Relu 함수를 사용하였다. Relu 함수는 $f(x) = \max(0,x)$로 표현할 수 있으며 $x > 0$이면 기울기가 1인 직선이고, $x < 0$이면 출력값은 항상 0이 된다.

인코더와 디코더의 매개 변수를 조정하면 출력된 데이터와 원본 데이터 간의 오차를 최소화 할 수 있다. Hidden Layer로 출력된 데이터가 원본 데이터의 최적의 저차원 특징이다.

2.3 Stacked AutoEncoder

SAE(Stacked AutoEncoder)는 여러개의 Hidden Layer를 가지는 AE이며, Layer를 추가할수록 AE가 더 복잡한 코딩을 학습할 수 있다. SAE는 Fig. 2와 같이 가운데 Hidden Layer(코딩층)을 기준으로 대칭인 구조를 가진다. 본 연구에서는 Encoding 부분만 사용하여 특징을 압축한다.

SAE가 기존의 AE와의 가장 큰 차이점은 DBN(Deep Beilef Network)의 구조라는 점이다. SAE는 AE의 Hidden Layer를 여러 개 쌓아서 구현한 것이다. 또한 차원을 계속해서 줄여가는 구조를 가지고 있다. 따라서 이전 Layer에서 얻은 특징을 좀 더 간결하게 표현한다.

DBN은 Layer와 뉴런의 수가 많아질수록 지속적으로 작은 값들이 Update 되는 과정에서 Weight의 오차가 점점 줄어드는 Vanishing Gradient[20] 문제와 Overfitting[21]의 문제가 발생한다. 이 방법을 해결하기 위하여 SAE는 Greedy Layer-Wise Training[19] 방법으로 학습을 시킨다.

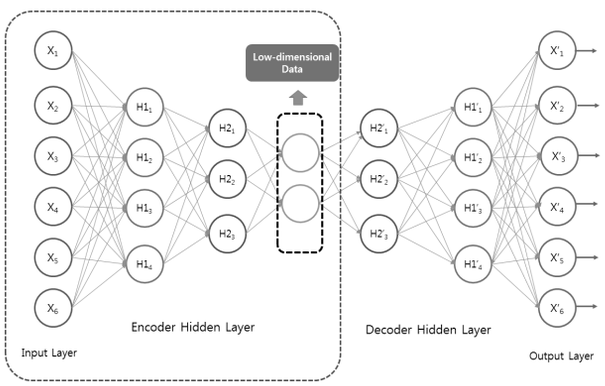

Fig 2. The Structure of Stacked AutoEncoder

Greedy Layer-Wise Training은 Fig. 3과 같이 크게 2단계의 과정으로 구분된다. 1단계는 Pretraining 단계로 각 Layer를 계층별로 비지도 방식으로 학습한다. 2단계는 Fine-Tuning 단계로 Pretraining 학습이 완료되면 학습된 모델에 추가 파라미터를 업데이트하는 방법이다.

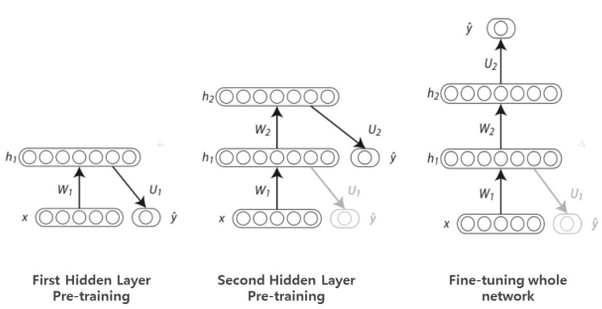

Fig. 3. The Structure of Greedy Layer-Wise Training

3. 설계 및 구현

3.1 설계

본 논문에서 제안하는 SAE를 이용한 침입탐지시스템은 Fig. 4와 같으며 크게 특징 추출과 분류 부분으로 나누어진다. 데이터셋은 IOT 환경에서 수집된 데이터셋을 사용하였으며 특징 추출은 AE 중 SAE 모델을 기반으로 특징을 추출하였다. 분류 알고리즘은 다수의 결정 트리들을 학습하고 다중 클래스 알고리즘 특성을 가지고 있는 Random Forest를 사용하였다.

3.2 특징 추출

AE는 특정한 특징 벡터를 추상적인 특징 벡터로 점진적으로 변환할 수 있어 고차원 데이터 공간에서 저차원 데이터 공간으로의 비선형 변환을 할 수 있다.

데이터셋의 60%를 학습용으로 사용하였으며 학습용 데이터를 SAE 모델을 이용하여 비지도 학습을 시킨다. SAE 학습의 목적은 Input data와 동일하게 Output data를 만드는 것이다. 이때 error function을 최소화 하기 위하여 여러 하이퍼파라미터를 설정한다. 실험에 사용한 하이퍼파라미터로 Batch size는 500, learning rate는 0.001, Epoch은 10회로 설정한다. 학습된 SAE를 이용하여 학습용 데이터와 테스트용 데이터의 특징을 저차원으로 압축 후 재구성 시 Batch size는 10000으로 한다. 학습과 압축은 Hidden Layer 수만큼 반복하며 학습 시간과 재구성의 시간을 확인한다. 또한 실험 시 Hidden Layer의 수, 뉴런 수와 같은 파라미터를 조정하여 가장 최적의 모델을 찾는다.

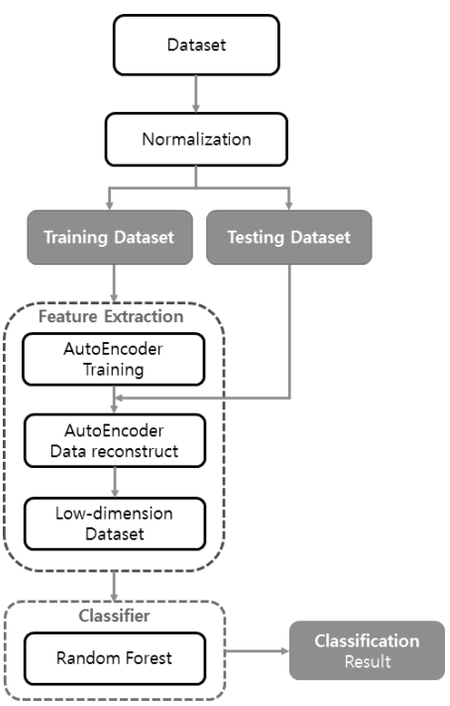

Fig. 4. Flowchart of the Proposed Method

3.3 분류

본 논문에서는 추출한 저차원의 특징을 기반으로 분류를 하기 위한 알고리즘으로 머신러닝의 대표적인 알고리즘인 Random Forest(RF)[22]를 사용하였다. 머신러닝에서 여러 개의 모델을 학습시켜 그 모델들의 예측결과들을 이용해 하나의 모델보다 더 나은 값을 예측하는 방법을 앙상블 학습이라 하며 대표적인 예가 RF이다. RF는 여러 개의 Decision Tree들을 생성한 다음, 각각의 트리에서의 예측한 값 중에서 가장 많은 선택을 받은 클래스를 예측하는 알고리즘이다.

Decision tree는 학습데이터에 따라 생성되는 트리가 매우 달라지기 때문에 일반화가 쉽지 않고 Overfitting이 되기 쉽다. 또한 계층적 접근 방식이므로 중간에 에러가 발생하면 다음 단계로 계속 에러를 전파하는 특성 때문에 IOT 환경에서는 적합한 분류 방법이 아니다.

이를 해결하기 위하여 RF는 데이터를 bagging 해서 Forest를 구성한다. bagging(bootstrap aggregation)은 훈련용 데이터 집합으로부터 크기가 같은 표본을 여러 번 동일한 크기로 랜덤하게 반복적으로 추출해서 각각에 대한 알고리즘을 적용하여 분류기를 생성하는 방법이다. 또 각 노드에서 특성의 일부만 사용하기 때문에 트리의 각 분기는 각기 다른 특성 부분집합을 사용한다. 이 두 메커니즘이 합쳐져서 RF의 모든 트리가 서로 달라지도록 만든다. 따라서 RF는 OverFitting을 막고 일반화에 적합한 알고리즘이라 할 수 있다.

Random Forest 분류는 Scikit learn의 라이브러리를 사용하였으며 하이퍼파라미터로 Random_state = 1, n_estimators는 100으로 설정하였다. Random_state는 변수선택의 임의화를 고정하여 실험 시 동일한 결과를 도출하도록 하였으며 n_estimators는 의사결정 트리의 개수를 의미한다.

4. 실험 및 분석

4.1 실험 환경

하드웨어 실험 환경은 3.30 GHz의 Intel(R) core I9-7900X CPU와 64GB RAM, linux Ubuntu 16.04 운영체제가 설치된 데스크톱에서 실험하였다.

머신러닝 프레임워크 중 가장 많이 사용하는 Tensorflow와 scikit-learn을 사용하여 실험 시뮬레이션을 수행하였으며 Python을 프로그래밍 언어를 사용하였다.

4.2 Dataset

본 논문에서 제안한 방법을 실험하기 위한 데이터셋으로 IOT 환경에서 수집한 데이터셋인 Danmini Doorbell[15]을 사용하였다.

Danmini Doorbell 데이터셋은 UCI Machine Learning Reposity에서 제공하는 IOT 환경의 공인 데이터셋으로 9개의 모바일 IOT 장치 중 Danmini Doorbell 장치의 네트워크 트래픽을 수집한 데이터이다. 이 데이터셋은 115개의 특징을 가지고 있으며 정상데이터와 10개의 공격 클래스로 구성되어 있다. 공격 클래스는 가장 잘 알려진 IOT 봇넷인 Gafgyt와 Mirai 악성코드로 구성되어 있다.

훈련 데이터셋과 테스트 데이터셋은 Table 1과 같이 각각 전체 데이터셋의 60%와 40%로 분할하여 사용하였다.

4.3 성능 평가

본 논문에서는 결과를 측정하기 위하여 Confusion matrix에 기반한 메트릭을 사용한다. Confusion matrix의 정의는 Table 2와 같다.

Table 1 The Number of Training and Test Data in the Danmini Doorbell Dataset

Traffic		Training Data	Testing Data
Normal		29,728	19,280
Gafgyt Attack	Combo	29,126	23,888
	Junk	18,169	11,628
	Scan	16,180	11,940
	Tcp	64,679	36,857
	Udp	61,583	42,350
Mirai Attack	Ack	61,317	40,878
	Scan	64,611	43,074
	Syn	58,655	49,030
	Udp	142,599	95,066
	Udpplain	49,189	32,793

Table 2. Confusion Matrix

		Actual	
		Positive	Negative
Predicted	Positive	TP	FP
	Negative	FN	TN

실험 성능 평가는 Accuracy(정확도), Precision(정밀도), Recall(재현율), F1-score를 측정하였다. 성능 측정의 방법은Equation (3), (4), (5), (6)과 같다.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

$$\mathrm{Precision} = \frac{TP}{TP+FP} \tag{4}$$

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

$$F1-Score = 2 \times \frac{\mathrm{Precision} \times Recall}{\mathrm{Precision} + Recall} \tag{6}$$

4.4 실험 결과

1) 네트워크 구조에 따른 성능 평가

Deep Neural Network에서는 Hidden Layer의 수와 각 Layer의 뉴런 수에 따라 성능의 차이를 확인할 수 있다. Hidden Layer의 수가 적고 각 Layer의 뉴런 수가 부족하면 고차원의 특징을 효과적으로 압축할 수 없다.

Table 3은 서로 다른 수의 Hidden Layer를 적용하였을 때 성능 및 SAE의 학습시간과 압축시간을 비교한 것이다.

실험결과 네트워크가 깊을수록 학습시간은 길어지고 압축시간은 줄어든다는 것을 볼 수 있다. 그리고 네트워크가 깊을수록 성능이 높아지는 것은 아니며 적당한 뉴런의 수에 따라 최적의 특징을 압축한다는 것을 보여준다. 이 실험에서는 [100, 75, 50] 네트워크에서 가장 성능이 우수하였으며 학습시간과 압축시간도 다른 네트워크와 비교하여 느리지 않았다.

2) 특징 추출에 따른 성능 평가

실험은 네트워크 구조에 따른 성능 중 가장 우수한 [100, 75, 50] 네트워크를 사용하여 특징을 저차원으로 압축하였다. 특징 추출 없이 분류하는 Single-RF 모델과 기본 AE로 특징 추출 후 분류하는 AE-RF 모델, 그리고 본 논문에서 제안하는 여러 네트워크를 거쳐 고차원에서 저차원으로 추출한 후 분류하는 SAE-RF 모델의 성능을 비교하였다. Table 4는 Single RF, AE-RF, 제안방식인 SAE-RF 모델을 Multi class로 분류한 학습시간과 분류 시간을 비교한 것이다. Table 3에서와 같이 SAE로 특징을 학습하고 압축하는 시간이 35.9초로 측정되었기 때문에 학습 시간과 분류시간에 SAE 학습과 압축 시간을 합치더라도 Single-RF, AE-RF 모델 보다 본 논문에서 제안한 SAE-RF 모델이 빠르다는 것을 확인할 수 있다.

Table 4. Training and Testing Time of Single-RF and AE-RF and SAE-RF

Method	Training Time(sec)	Testing Time(sec)
Single-RF	171.75	3.52
AE-RF	178.68	4.48
SAE-RF (Proposed Method)	91.00	5.39

Table 3. Performance Evaluation According to Structure of Hidden Layer

Hidden Layer Structure	Accuracy	Precision	Recall	F1-Score	AutoEncoder Training time(s)	Data reconstruction time(s)
[100, 75, 50, 25, 10]	95.94	96.88	96.68	96.76	41.00	0.8
[100, 75, 50, 25]	94.50	96.65	94.58	94.36	33.75	1.38
[100, 75, 50]	**99.45**	**99.45**	**99.49**	**99.46**	**32.25**	**3.65**
[75, 50]	98.56	97.46	98.21	94.56	31.27	5.37
[50]	90.94	94.63	90.88	88.13	27.22	4.64

Fig. 5는 Single RF와 AE-RF, 제안방식인 SAE-RF를 multi class로 분류한 성능을 보여준다. 실험 결과 Single-RF와 AE-RF의 성능은 거의 차이가 없었다. 따라서 하나의 Hidden Layer 로 구성된 AE는 특징 추출을 통한 성능 향상이 어렵다는 결과를 보여 주었다. 실험결과 Accuracy, Precision, Recall, F1-Score 모두 세 가지의 모델 중 제안방식의 성능이 가장 높다는 것을 확인할 수 있다. 따라서 여러 개의 네트워크를 거쳐 고차원의 특징을 저차원으로 압축한 SAE-RF 모델의 속도와 성능이 모두 우수하다는 것을 확인할 수 있다..

3) 다른 알고리즘과의 성능 비교

IGOR KOTENKO[13]등은 Danmini Doorbell 데이터셋을 기반으로 여러 머신 러닝 알고리즘으로 성능을 측정하였다.

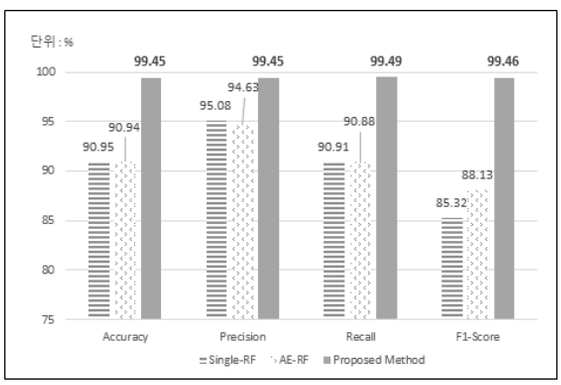

Fig. 5. Performance Evaluation of Single-RF and AE-RF and SAE-RF Based Danmini Doorbell Dataset

비교 알고리즘으로 Decision Tree, Gaussian Naive Bayes[23], Linear SVM[24] 알고리즘으로 분류한 성능과 본 논문에서 제안한 방법을 비교 한다.

실험결과는 Fig. 6과 같이 다른 알고리즘과 비교하였을 때 본 논문에서 제안한 방법의 성능이 가장 높게 나타났다.

머신러닝의 특성 상 특징이 많을수록 성능이 우수하지만 본 논문에서 제안한 방식은 특징의 수를 줄이지 않고 고차원을 저차원으로 압축하였기 때문에 기존의 머신러닝 보다 탐지의 속도는 빨라지고 성능은 높아진 것을 확인할 수 있다.

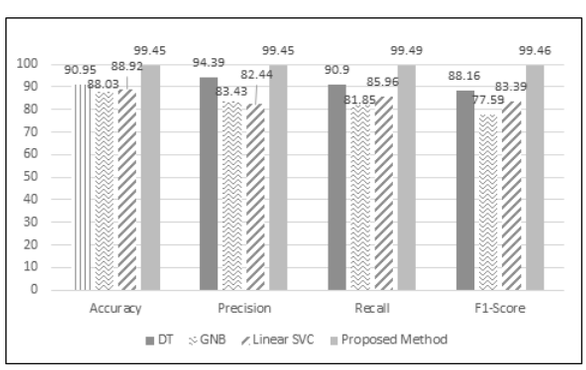

Fig. 6. Performance Comparison with Other Algorithm

4) 기존 네트워크 환경에서의 침입탐지 성능 결과

IOT 환경의 데이터셋을 사용하여 본 논문에서 제안한 모델로 실험하였을 때 성능이 향성된 것을 확인하였다. 따라서 기존 네트워크의 침입탐지에도 효과적인지 실험하였다.

CICIDS2017 데이터셋[25]으로 실험하였으며 이 데이터셋은 최신의 공격을 반영하였고 희소 클래스도 존재하므로 성능을 측정하기에 적합하다.

CICIDS2017 데이터셋의 특징 수는 77개이며 데이터 클래스의 구성은 Table 5와 같다. CICIDS2017 데이터셋은 정상 트래픽이 80% 이상을 차지하며 12개의 최신 공격 트래픽으로 구성되어 있다. 특히 Infilteration, Heartbleed 공격과 같은 비중이 0.01%미만의 희소 공격이 포함되어 있다.

이 실험도 특징 추출에 따른 성능 평가와 같이 제안 방식과 Single RF의 성능을 비교하였다.

실험 결과는 Fig. 7과 같이 Accuracy는 거의 차이가 없으며 Precision와 Recall, F1-Score의 성능은 제안방식이 더 높게 나타났다. 따라서 기존 네트워크 환경에서도 본 논문에서 제안한 방법이 침입탐지 성능 향상에 적용할 수 있을 것이라 기대한다.

Table 5. CICIDS2017 Dataset

Flow Type		Number	Percentage
Benign		2,273,097	80.3004%
DDoS		128,027	4.5227%
Port Scan		158,930	5.6441%
Bot		1,966	0.0695%
Infilteration		36	0.0013%
Web Attack	Brute Force	2,180	0.0770%
	SQL Injection		
	XSS		
FTP-Patator		7,938	0.2804%
SSH-Patator		5,897	0.2083%
DoS GoldenEye		10,293	0.3636%
DoS Hulk		231,073	8.1630%
DoS Slowhttptest		5,499	0.1943%
DoS Slowloris		5,796	0.2048%
Heartbleed		11	0.0004%

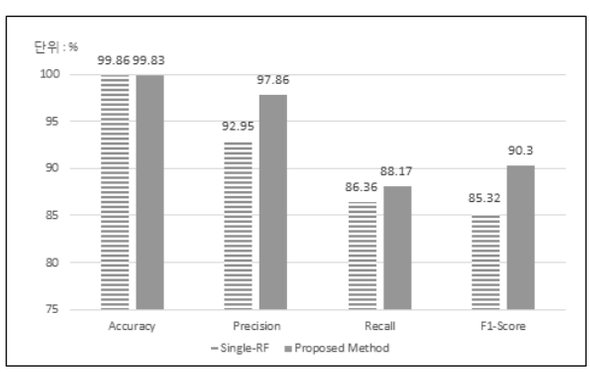

Fig. 7. Performance Evaluation of SAE-RF and Single RF based CICIDS2017 Dataset

5. 결론 및 향후 연구

본 논문에서는 제안한 방법은 IOT 환경에서 수집된 Danmini Doorbell 데이터셋을 기반으로 특징 학습 및 차원 감소를 위해 SAE를 사용하였으며 분류를 위하여 softmax 대신 Random Forest로 분류하였다. 제안된 방법의 실험 결과는 특징 추출을 하지 않고 분류한 기존의 머신러닝 알고리즘 보다 성능이 향상되었다는 것을 보여 주었다. 또한 특징 추출 시 모든 특징을 압축하여 저차원으로 감소시키기 때문에 99% 이상의 정확도와 학습 및 분류 시간 또한 향상되었음을 확인하였다.

또한 기존의 네트워크 환경에서도 제안한 방법을 적용한 결과 특징 추출을 하지 않고 분류한 방법보다 성능이 우수하다는 것을 확인하였다.

기존 네트워크 환경의 데이터에서 희소 클래스로 인하여 성능이 상대적으로 낮게 나온 것을 확인할 수 있었다. 따라서 향후 희소 클래스 문제를 해결하기 위한 연구를 통해 성능을 높여나갈 예정이다.

References

[1] I. Alrashdi, A. Alqazzaz, E. Aloufi, R. Alharthi, M. Zohdy and H. Ming, "AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning," *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp.305-310, 2019.

[2] Yu Su, Kaiyue Qi, Chong Di, Yinghua Ma, and Shenghong Li, "Learning Automata based Feature Selection for Network Traffic Intrusion Detection," *2018 IEEE Third International Conference on Data Science in Cyberspace*, pp.622-627, 2018.

[3] Marzieh Bitaab and Sattar Hashemi, "Hybrid Intrusion Detection: Combining Decision Tree and Gaussian Mixture Model," *2017 14th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*, pp.8-12, 2017.

[4] Saeid Soheily-Khah, Pierre-Franc, ois Marteau and Nicolas B´echet, "Intrusion Detection in Network Systems Through Hybrid Supervised and Unsupervised Machine Learning Process: A Case Study on the ISCX Dataset," *International Conference on Data Intelligence and Security*, pp.219-226, 2018.

[5] Xiaoming Ye, Xingshu Chen, Dunhu Liu, Wenxian Wang, Li Yang, Gang Liang and Guolin Shao, "Efficient Feature Extraction using Apache Spark for Network Behavior Anomaly Detection," *Tsinghua Science and Technology*, Vol.23, No.5, pp.561-573, 2018.

[6] Ahmad I., Basheri M., Iqbal MJ. and Rahim A., "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," *IEEE Access*, Vol.6, pp.33789-33795, 2018.

[7] K. Park, Y. Song and Y. Cheong, "Classification of Attack Types for Intrusion Detection Systems Using a Machine Learning Algorithm," *Proc. of 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*, pp.282-286, 2018.

[8] INGHAO YAN and GUODONG HAN, "Effective Feature Extraction via Stacked Sparse Autoencoder to Improve Intrusion Detection System," *IEEE Access*, Vol.6, pp.41238-41248, 2018.

[9] Mehdi Mohammadi, Ala Al-Fuqaha, Mohsen Guizani and Jun-Seok Oh, "Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services," *IEEE Internet of Things Journal*, Vol.5, No.2, pp.624-635, 2018.

[10] Monika Roopak, Gui Yun Tian and Jonathon Chambers, "Deep Learning Models for Cyber Security in IoT Networks," *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp.452-457, 2019.

[11] Imtiaz Ullah and Qusay H. Mahmoud, "A Two-Level Hybrid Model for Anomalous Activity Detection in IoT Networks," *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp.1-6, 2019.

[12] Machine Learning Repository [Internet], https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet _attacks_N_BaIoT

[13] Igor Kotenko, Igor Sanko and Alexander Branitskiy, "Framework for Mobile Internet of Things Security Monitoring Based on Big Data Processing and Machine Learning," *IEEE ACCESS*, Vol.6, pp.72714-72723, 2018.

[14] H. Chae and S. H. Choi, "Feature Selection for efficient Intrusion Detection using Attribute Ratio," *International Journal of Computers and Communications*, Vol.8, pp. 134-139, 2014.

[15] R. Datti and S. Lakhina, "Performance Comparison of Features Reduction Techniques for Intrusion Detection System," *International Journal of Computer Science And Technology*, Vol.3, No.1, pp.332-335, 2012.

[16] Al-Qatf MAjjed, Lasheng Yu, Al-Habib Mohammed, and Al-Sabahi Kamal, "Deep Learning Approach Combining Sparse Autoencoder With SVM for Network Intrusion Detection," *IEEE Access*, Vol.6, pp.52843-52856, 2018.

[17] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee and Chiew Tong Lau, "Autoencoder-based Network Anomaly Detection," *2018 Wireless Telecommunications Symposium (WTS)*, pp.1-5, 2018.

[18] S. Squartini, A. Hussain and F. Piazza, "Preprocessing Based Solution for the Vanishing Gradient Problem in Recurrent Neural Networks," *Proceedings of the 2003 International Symposium on Circuits and Systems*, 2003. ISCAS '03. pp.713-716, 2003.

[19] Tie Luo and Sai G. Nagarajan, "Distributed Anomaly Detection using Autoencoder Neural Networks in WSN for IoT," *2018 IEEE International Conference on Communications (ICC)*, pp.1-6, 2018

[20] Imanol Bilbao and Javier Bilbao, "Overfitting Problem and the Over-training in the Era of Data: Particularly for Artificial Neural Networks," *2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp.173-177, 2017.

[21] Telmo Amaral, Luís M. Silva, Luís A. Alexandre, Chetak Kandaswamy, Jorge M. Santos and Chetak Kandaswamy, "Using Different Cost Functions to Train Stacked Auto-Encoders," *2013 12th Mexican International Conference on Artificial Intelligence*, pp.114-120, 2013.

[22] J. Zhang and M. Zulkernine, "A Hybrid Network Intrusion Detection Technique using Random Forests," *First International Conference on Availability, Reliability and Security (ARES'06)*, pp.262-269, 2006.

[23] Marcin Mizianty,Lukasz Kurgan and Marek Ogiela, "Comparative Analysis of the Impact of Discretization on the Classification with Naïve Bayes and Semi-Naïve Bayes Classifiers," *2008 Seventh International Conference on Machine Learning and Applications*, pp.823-828, 2008.

[24] Jianxin Wu and Hao Yang, "Linear Regression-Based Efficient SVM Learning for Large-Scale Classification," *IEEE Transactions on Neural Networks and Learning Systems*, Vol.26, No.10, pp.2357-2369, 2015.

[25] Iman Sharafaldin, Arash Habibi Lashkari and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pp.108-116, 2018.

이 주 화

https://orcid.org/0000-0002-7855-261X
e-mail : yezi1004@gmaiil.com
2012년 평생교육진흥원 컴퓨터공학(학사)
2015년 계명대학교 전산교육전공(석사)
2015년 ~ 현 재 계명대학교 컴퓨터공학 박사수료
관심분야 : 네트워크 보안, 사물인터넷, 기계학습, 심층학습

박 기 현

https://orcid.org/0000-0002-3208-4216
e-mail : khp@kmu.ac.kr
1979년 경북대학교 전자공학과(학사)
1981년 한국과학기술원 전자계산학과(석사)
1990년 미국 밴드빌트 대학교 컴퓨터공학과(박사)
1981년 ~ 현 재 계명대학교 컴퓨터공학과 교수
관심분야 : 병렬/분산 운영체제, 모바일 통신 소프트웨어, 사물인터넷, 네트워크 보안 등