Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.4, Aug. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.4.691

산업제어시스템의 이상 탐지 성능 개선을 위한 데이터 보정 방안 연구

전 상 수,[1†] 이 경 호[2‡]
[1,2]고려대학교 (대학원생, 교수)

Research on Data Tuning Methods to Improve the Anomaly Detection Performance of Industrial Control Systems

SANGSO JUN,[1†] Kyung-ho Lee[2‡]
[1,2]Korea University (Graduate student, Professor)

요 약

머신러닝과 딥러닝의 기술이 보편화되면서 산업제어시스템의 이상(비정상) 탐지 연구에도 적용이 되기 시작하였다. 국내에서는 산업제어시스템의 이상 탐지를 위한 인공지능 연구를 활성화시키기 위하여 HAI 데이터셋을 개발하여 공개하였고, 산업제어시스템 보안위협 탐지 AI 경진대회를 시행하고 있다. 이상 탐지 연구들은 대개 기존의 딥러닝 학습 알고리즘을 변형하거나 다른 알고리즘과 함께 적용하는 앙상블 학습 모델의 방법을 통해 향상된 성능의 학습 모델을 만드는 연구가 대부분 이었다. 본 연구에서는 학습 모델과 데이터 전처리(pre-processing)의 개선을 통한 방법이 아니라, 비정상 데이터를 탐지하여 라벨링 한 결과를 보정하는 후처리(post-processing) 방법으로 이상 탐지의 성능을 개선시키는 연구를 진행하였고, 그 결과 기존 모델의 이상 탐지 성능 대비 약 10% 이상의 향상된 결과를 확인하였다.

ABSTRACT

As the technology of machine learning and deep learning became common, it began to be applied to research on anomaly(abnormal) detection of industrial control systems. In Korea, the HAI dataset was developed and published to activate artificial intelligence research for abnormal detection of industrial control systems, and an AI contest for detecting industrial control system security threats is being conducted. Most of the anomaly detection studies have been to create a learning model with improved performance through the ensemble model method, which is applied either by modifying the existing deep learning algorithm or by applying it together with other algorithms. In this study, a study was conducted to improve the performance of anomaly detection with a post-processing method that detects abnormal data and corrects the labeling results, rather than the learning algorithm and data pre-processing process. Results It was confirmed that the results were improved by about 10% or more compared to the anomaly detection performance of the existing model.

Keywords: Anomaly Detection, ICS Securiy, Time Sereies Data, HAI Dataset, SWaT

I. 서 론

산업제어시스템이란 전력, 가스 상하수도, 원력, 운송, 제조 등 산업현장을 모니터링하고 제어하는데 사용되는 시스템이다. 산업제어시스템은 발전소, 가스 생산 및 공급기지, 댐 등 국가와 사회유지에 필수

Received(07. 12. 2022), Modified(08. 02. 2022), Accepted(08. 02. 2022)

† 주저자, mysangs@daum.net
‡ 교신저자, kevinlee@korea.ac.kr(Corresponding author)

적인 기능을 제공하는 기반시설에서 주로 사용된다. 따라서 산업제어시스템이 해킹 등 사이버공격을 받아 피해를 입을 경우, 국가와 사회 기능이 마비되는 등 피해 규모가 매우 큰 경우가 대부분이다[10].

산업제어시스템은 많은 구성요소(예: 제어기, 센서, 액추에이터 등)를 포함하고 있고, 각 구성요소에서 생산하는 데이터를 통합하여 수집하면 데이터는 다변수 시계열로 표시될 수 있고 이러한 데이터에 존재하는 모든 비정상 혹은 이상은 장비 결함, 인적 요소, 사이버 공격과 같은 다양한 원인이 있을 수 있다[1]. 이러한 위협에 대응하기 위해 많은 연구가 진행되고 있고, 머신러닝과 딥러닝의 기술이 산업제어시스템의 이상(비정상) 탐지 성능의 개선을 위해 많은 연구가 진행되고 있다.

국가보안연구소(NSR)에서는 산업제어시스템의 이상 탐지를 위한 인공지능 연구를 활성화하고 장려하기 위하여 연구에 필요한 데이터셋를 개발하여 공개하였고, 산업제어시스템 보안위협 탐지 AI 경진대회(HAICon)를 시행하고 있다. HAI 데이터셋은 화력발전과 수력발전을 위한 테스트베드를 구축하여 시뮬레이터를 통해 운전하면서 제어기기로 부터 실시간으로 확보한 시계열 데이터셋이다. 현재는 세 번째 버전 HAI 22.04가 공개되어 있다[6][12].

산업제어시스템과 같은 대규모 시스템의 시계열 데이터에서 이상 현상은 일반적으로 거의 발생되지 않거나 정상 데이터의 흐름으로 인해 숨겨지거나 아주 드물게 나타나기 때문에 데이터 라벨의 지정이 어렵고 많은 시간이 필요하다. 따라서 많은 정상 시점에서 드물게 나타날 수 있는 이상 현상을 감지할 수 있는 기준을 도출해야 하며, 비지도학습 형태의 딥러닝 모델이 많은 성과를 보인다[2].

비지도 학습은 라벨(정답 혹은 목표치)이 없는 데이터로 학습하는 방법으로 정상 데이터로만 학습하여 현재시점에서 예측한 결과와 미래의 실제 데이터와의 차이를 비교하는 방법으로 비정상 데이터와 비교하여 이상 상황을 탐지할 수 있다. 이러한 이상 탐지의 방법은 [3][4][5][18][19]에서 관련 내용들을 확인할 수 있다.

HAICon 2020, 2021 경진대회에서 제공하는 HAI 데이터셋과 baseline 코드도 비지도 학습에 적합하도록 되어 있다. 경진대회에서 제공하는 HAI 데이터셋은 정상 데이터셋, 검증 데이터셋과 테스트 데이터셋으로 구성되어 있다. 정상 데이터셋은 훈련(학습)에 사용하고, 검증 데이터셋에는 공격 포인트 7개가 라벨링 되어 있고, 모델의 하이퍼파라미터를 보정하는데 사용한다. 테스트 데이터셋에는 라벨링 되어 있지 않은 공격 포인트가 포함되어 있고, 테스트 데이터셋의 공격 포인트를 탐지하는 것이 목표이며, eTaPR을 통해 성능평가를 수행한다. 제공된 Baseline 모델은 데이터 전처리 과정을 거쳐 Stacked GRU 알고리즘으로 정상 데이터를 학습한 후, 검증 데이터로 성능을 튜닝 한다. 학습 모델에 의해 예측한 데이터와 테스트 데이터셋의 차이가 일정 수준을 넘으면 비정상으로 라벨링 하여 주최 측에 제출하면 실제 라벨링 된 테스트 데이터와 비교하여 탐지성능을 계산한다[13].

시계열 데이터의 이상 탐지에 대한 기존의 연구는 탐지성능을 높이고 오탐을 줄일 수 있는 학습모델과 학습모델에 적합한 데이터 전처리에 대한 연구가 주를 이루어 왔다. LSTM 또는 GRU를 기반으로 하는 순환신경망(RNN)과 다변수 시계열에서 여러 변수 간의 동적 그래프를 학습하기 위한 그래프 신경망(GNN), self-attention map에서 각 시점의 시간적 연관성을 얻을 수 있는 Transformers 등이 사용되어 왔다[1][2][3][4].

본 연구는 학습 알고리즘과 데이터 전처리(pre-processing)의 과정이 아닌, 정상 데이터를 학습하여 추론한 결과의 비정상 데이터를 좀 더 세밀하게 보정하는 후처리(post-processing) 방법이며, 탐지한 공격의 시작과 끝의 정확도를 높이고 미탐지를 최소화하는 방법이다. 각 비정상 데이터가 보이는 그래프의 형태에 따라 실제 공격 구간과의 차이가 발생되는 것을 확인하고 이를 HAI 2.0(21.04) 데이터셋을 통해 보정값을 확인 한 후, HAICon 2021에서 제공된 데이터셋과 해외의 SWaT 데이터셋을 사용하여 탐지성능에 효과가 있는 것을 검증하였다.

II. 관련 연구

2.1 산업제어시스템의 구조

TTA(한국정보통신기술협회)의 산업제어시스템 보안요구사항에 따르면, 산업제어시스템의 네트워크 구성과 서비스 시나리오는 [그림1]과 같다. 센서, 엑츄에이터 등의 현장장치는 유무선 랜, 시리얼 케이블 또는 구리선에 의한 하드 와이어드 방법 등을 통해 PLC, DCS, RTU 등 제어 H/W와 연결되며, 제어 H/W는 이더넷 또는 시리얼 케이블 등을 통해

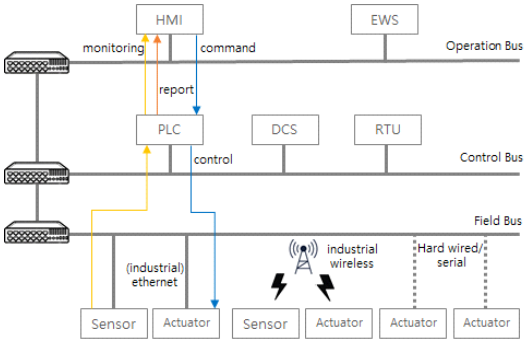

Fig. 1. Industrial Control System network structure and service scenario

HMI, EWS 등의 제어 S/W와 연결된다. 제어 H/W는 센서를 통해 수집된 압력, 온도 등의 현장장치 상태 데이터를 취합하여 제어 S/W로 전송하며, 사용자는 HMI 등 제어 S/W를 이용해서 취합된 현장장치 상태 데이터를 모니터링하고, 이를 통해 현장장치의 상태를 확인할 수 있다. 현장에 설치된 밸브의 개폐 등 액츄에이터를 제어하기 위해서 사용자는 제어 S/W를 통해서 제어 명령을 입력한다. 사용자가 입력한 제어 명령은 제어 H/W에게 전달되며, 제어 H/W는 제어 명령에 따라 현장장치를 제어한다. 또한, 제어 H/W는 이상 상황이나 설정된 이벤트가 발생하면, 제어 S/W에 보고(알람)를 통해 해당 상황을 알릴 수 있다. [그림 1]의 '명령', '제어', '보고', '모니터링'은 산업제어시스템의 필수 서비스이다[11].

2.2 공개 데이터셋(국내/해외)

산업제어시스템은 전용 네트워크로 운영하는 기존 방식과는 달리, 효율성을 높이고 비용을 절감하기 위해 외부와의 연결성이 늘어나면서 보안위협이 증가하고 있다. 따라서 보안위협에 대응하기 위한 연구가 활발히 진행되고 있지만, 실제 운영 현장에서 보안 기술을 개발하거나 검증하는 것은 무중단 운전 환경과 관제 시스템의 고가용성을 위해 갖추어야 하는 제약으로 인해 상당히 어렵다. 또한 이러한 제약으로 인해 보안 연구를 위한 데이터셋을 확보하는 데에도 한계가 존재한다. 이러한 제약과 한계를 극복하여 원활한 연구를 수행할 수 있도록 산업제어시스템 테스트베드를 구축하였고, 이 테스트베드에서 정상 운전 및 비정상 상태의 데이터셋을 수집하여 공개하였다[10].

이렇게 공개된 국내외의 데이터셋을 정리하였다.

2.2.1 국내의 HAI 데이터셋

HAI(HIL-BASED AUGMENTED ICS) 보안 데이터셋은 국가보안연구소에서 화력 발전 및 수력 발전을 에뮬레이트 하는 HIL(Hardware in the loop) 시뮬레이터를 활용하여 자체 제작한 실제 테스트베드에서 수집한 데이터셋이다[6].

보안뉴스(https://www.boannews.com/media/view.asp?idx=106391)의 HAI 데이터셋에 대한 기사에 따르면, "AI 기반 산업제어시스템 보안 연구를 위해서는 양질의 데이터셋 확보가 필수적이나 활용 가능한 데이터셋이 현저히 부족하고, 일부 공개된 데이터셋은 연구 활용에 한계가 있다. 이에 국가보안연구소는 자동화된 공격 재현과 라벨링으로 이러한 한계점을 극복하여 데이터 신뢰성을 보장하고, 단순한 공격 수준을 뛰어넘어 실제 사이버 공격과 유사한 공격이 재현된 데이터셋을 개발함으로써 더 정밀한 성능평가를 가능하도록 했다." 라고 하였다.

테스트베드는 보일러, 터빈, 수처리 부품 및 HIL 시뮬레이터로 구성하였다. 보일러, 터빈 및 수처리 프로세스는 각 공정별 컨트롤러에 의해 제어되는데, Emerson Ovation 분산 제어 시스템(DCS)은 보일러 공정에서 수위, 유량, 압력, 온도, 급수 펌프 및 히터를 제어하기 위해 사용하였고, 터빈 공정에서의 GE의 Mark VIe DCS는 속도 제어 및 진동 모니터링 용도로 사용하였다. Siemens S7-300 PLC는 수위와 펌프를 제어하기 위해 수처리 공정에 사용하였다. dSPACE® SCALEXIO 시스템은 HIL 시뮬레이션을 위해 사용하였으며, Siemens S7-1500 PLC 및 ET200 원격 IO 장치를 사용하여 보일러, 터빈, 수처리 공정을 운영한다[7].

정상 데이터셋을 수집하기 위한 동작으로 HMM(Hidden Markov Model)을 사용하였다. 3개의 HMM는 각 공정별 컨트롤러가 정상 운영을 할 수 있도록 구성하였고, 운영에 필요한 설정 값들은 정상범위 내에서 무작위(확률적)로 결정하였다. 이러한 정상 운전 상황에서 각 포인트의 데이터를 수집하여 정상 데이터셋을 수집하였다. 비정상 데이터셋은 32개의 단일 공격과 26개의 공격 조합을 포함하여 총 58개의 공격을 수행하여 수집하였다[7].

가장 최근에 공개된 HAI 3.0(22.04) 데이터셋은

Table 1. HAI dataset summary

Type		HAI 20.07	HAI 21.03	HAI 22.04
Points (per sec)		59	78	86
Normal Dataset	Interval (days)	7	11	11
	Size(MB)	225	471	534
	Normal Scenario	5	10	-
Abnormal Dataset	Interval (days)	5	5	4
	Size(MB)	181	205	196
	Attack Scenario	38	50	58

정상 데이터는 약 11일간, 비정상 데이터 약 4일간 수집하였고, 정상 데이터는 6개의 CSV 파일(train1.csv~train6.csv), 비정상 데이터는 4개의 CSV 파일(test1.csv~test4.csv)로 구성되어 있다. 각 데이터셋은 Timestamp, 수집 포인트(필드 디바이스 86개)와 공격 라벨로 구성되어 있다[6][7]. HAI 데이터셋의 히스토리는 표1에 요약하였다.

2.2.2 해외 데이터셋

공개되어 있는 해외의 산업제어시스템 데이터셋 중에 특징이 있는 4개를 선정하여 정리하였고, Choi et al.[9]의 조사 내용을 기반으로 표2에 정리 하고,각 공개된 웹사이트를 확인하여 변경된 사항을 업데이트 하였다.

SWaT(Secure Water Treatment) 데이터셋은 필드 디바이스(총 51개의 센서, 액추에이터 등) 데이터와 네트워크 트래픽으로 구성되어 있으며, 정상 7일 및 공격 시나리오 4일, 총 11일 동안 수집되었다. 특히 SWaT 데이터셋은 대규모 테스트베드에서 가장 많은 양의 데이터를 제공한다. SWaT는 공격 대상 디바이스와 물리적 포인트를 정의하고 각각의 공격을 필드 신호 및 네트워크 트래픽과 관련된 총 41개의 공격 시나리오를 수행하였다[10][14].

Morris 데이터셋은 발전, 가스 및 수처리와 관련된 다섯 가지 데이터 세트로 구성되어 있다. 필드 디바이스 데이터, 네트워크 트래픽과 디바이스 로그를 csv와 arff 파일 포맷으로 제공한다. Morris-1 데이터셋은 발전기, IED(Intelligent Electronic Device), 차단기, 스위치, 라우터로 구성된 전력계통 테스트베드의 정상/비정상 이벤트뿐만 아니라, 37개의 전력 시스템 이벤트 시나리오로 구성되어 있다. Morris-2, Morris-3 및 Morris-4 데이터셋은 가스 파이프라인 테스트베드에서 RS-232 와 이더넷을 연결하여 필드 디바이스와 HMI 간의 통신 데이터를 포함하고 있다. Morris-5 데이터셋은 실제 에너지 관리 시스템에서 30일 이상 수집하여 다른 데이터셋에 비해 가장 긴 시간 동안 수집하였다[10][15].

Table 2. Public ICS Datasets

Dataset	System Domain	Collect Target	Recent rel.	Source
SWaT	Water	Field, Network	2015	[13]
Morris 1-5	Power	Field, Network, Device log	2017	[14]
	Gas Pipeline			
	Water Storage Tank			
	EMS			
Rodofile	Mining Refinery	Network, Device log	2017	[16]
Lemay	SCADA	Network	2016	[15]

Rodofile 데이터셋은 Siemens S7-300 및 S7-1200 PLC를 사용하는 광물 정제 시스템에서 수집한 데이터셋이며, 테스트베드는 컨베이어, 세척 탱크, 파이프라인 반응기, 마스터 PLC 및 슬레이브 PLC로 구성되어 있다. 공격은 네트워크를 통해 PLC에 액세스하고 제어 프로세스에서 오작동을 일으키는 프로세스 공격을 수행하였으며, 약 9시간의 네트워크 트래픽을 수집하여 공개하였다[10][17].

Lemay 데이터셋은 SCADA(Supervisory Control and Data Acquisition) 분야에서 명령 및 제어와 관련된 네트워크 트래픽 데이터셋을 수집하였다. SCADA 네트워크는 공용 툴인 SCADA Sandbox를 이용하여 구축하였고, SCADA BR을 이용하여 2개의 마스터 단말장치를 구현하였다. 데이터셋은 컨트롤러의 수와 폴링 주기를 변경하고 운영자의 수동 조작 등의 공격이 수행된 데이터가 포함되어 있다[10][16].

본 논문에서는 HAI 2.0(21.03)과 1.1(20.07)을 사용하여 제안 내용을 정리하고, HAICon 2021 경진대회에서 제공된 데이터셋과 해외의 SWaT 데이터셋을 사용하여 검증한다.

2.3 eTaPR 평가도구

시계열 데이터는 동일한 시간 간격으로 반복 측정을 통해 얻은 값의 시퀀스로 구성된다. 기존의 평가지표는 일반적으로 (1)예측된 결과가 얼마나 정확한지(Precision, 정밀도)와 (2)얼마나 많은 이상징후를 탐지하였는지(Recall, 재현율)의 두 가지 측면을 기반으로 고려한다. 그런데, 비정상적인 이벤트는 유사한 패턴을 가진 비정상적인 값을 연속으로 유발하기 때문에 재현율(recall-like score)을 정확하게 계산하지 못한다. 어떤 탐지 방법도 정확한 범위의 이상을 탐지할 수 없는데, 기존의 평가지표는 시계열 이상 탐지에서 부정확하거나 불충분한 경우를 모두 간과하게 되고, 이상 탐지 성능을 과대평가하는 경향이 있다. eTaPR(Enhanced Time-series Aware precision and recall)은 이러한 기존의 이상 탐지 평가지표의 한계를 극복하기 위하여 제안한 방식이다. eTaP(precision) 와 eTaR(recall)로 구성된 평가지표이며, eTaP는 예측결과가 얼마나 이상 탐지를 오탐 없이 잘 찾아내는지 이고, eTaR은 비정상 범위를 얼마나 잘 탐지하는지를 평가한다. eTaPR은 두 번의 HAICon 2020과 2021 경진대회를 통해서 실제로 검증하였다[7][8].

본 논문에서도 eTaPR을 이용하여 이상 탐지 성능을 평가한다.

2.4 HAICon 2020, 2021

국가보안기술연구소와 국가정보원에서 주최한 HAICon(산업제어시스템 보안위협 탐지 AI 경진대회) 2020 과 2021에서는 비지도 학습에 적합한 데이트를 제공하고, 우수한 이상 탐지 성능을 보여준 탐지 모델을 수상하고, 보안위협 탐지연구를 활성화하고 기술 보급을 위한 목적으로 해당 모델 코드를 공개하고 있다[13]. 지난 2년간의 경진대회의 수상작 10편의 모델 코드가 공개되었고 관련 논문[21][22][23][24][25]이 발표되었다.

각 대회마다 주최 측에서는 데이터셋과 함께 베이스라인 모델 코드를 제공하고 있는데, 학습 알고리즘으로 RNN 기반의 GRU를 채택하고 있다. 이외에

Table 3. HAICon Baseline model vs ORIDORI Model(1st winner)

Type		Baseline Model	ORIDORI Model
Dataset	Train	11 days of normal data (6 files)	
	Validatation	1 day of normal/attack data (7 attack points, 1 file)	
	Test	1 day of normal/attack data (No attack label, 3 files)	
Preprocessing	Normalization	Min-Max Normalization	
	Weight	Exponential Weighted Mean(ewm=0.9)	
Training Model	Learning Type	Unsupervised Learning	
	Loss Func.	MSE	
	Optimizer	AdamW	
	Algorithm	StackedGRU	Stacked LSTM + Attention
		(rnn): GRU(86, 100, num_layers=3, bidirectional=True) (fc): Linear(in_features=200, out_features=86, bias=True)	(rnn): LSTM(24, 200, num_layers=3, dropout=0.1, bidirectional=True) (fc): Linear(in_features=400, out_features=24, bias=True) (relu): LeakyReLU(negative_slope=0.1) (sigmoid): Sigmoid() (dense1): Linear(in_features=24, out_features=12, bias=True) (dense2): Linear(in_features=12, out_features=24, bias=True)
	Epoch	32	130
Optimiza-tion	Filtering	-	Lowpass filter (signal.butter(1, 0.02, btype='lowpass'))
Metric & Eval.		Exceeding the threshold of the average (|predicted value-actual value|) of the entire field for each second	
		Difference prediction of anomaly detection vs actual attack duration	
		eTaPR (F1 value)	

도 대부분의 공개된 수상작 모델 코드들도 이 베이스라인 모델을 기반으로 이상 탐지 성능을 향상시키기 위해 학습 알고리즘을 변경한 모델을 사용하고 있다.

이 중에 2021년 대회에서 최종 1등을 수상한 오리도리팀의 모델은 주최 측에서 제공한 베이스라인 모델을 기반으로 학습 모델을 변경하였고, 테스트 데이터와 학습모델을 통하여 추론한 결과에 대해 Low Pass Filter를 적용한 특징을 가지고 있다. 베이스라인 모델의 특징과 오리도리팀 모델의 차이를 표3에서 정리하였다.

2.5 비지도 시계열 이상탐지에 대한 기존 연구

비지도 이상 탐지에 대한 연구는 지속적으로 진행되어왔다. 비정상이라고 판단하는 기준에 따라 분류하자면, 밀도 기반(density estimation), 클러스터링 기반(clustering-based), 재구성 기반(reconstruction-based) 등의 방법으로 나눌 수 있다. 밀도 추정 방법의 경우, 고전적인 방법으로 지역 이상치 인자(local outlier factor(LOF), Breunig et al.(2000)) 및 연결성 이상치 인자(connectivity outlier factor(COF), Tang et al.(2002))가 있다. DAGMM(Zong et al, 2018)및 MPPCACD (Yairi et al., 2017)는 가우시안 분포가 혼합된 모델(Gaussian Mixture Model)을 통합하여 표현 밀도를 추정한다. 군집 기반 방법에서는 가장 가까운 것과의 거리가 긴 것을 이상치(outlier)라고 본다. SVDD(Tax & Duin, 2004) 및 Deep SVDD(Ruff et al., 2018)는 일반 데이터의 표현을 압축 클러스터로 수집하고, THOC(Shen et al., 2020)는 각 레이어 마다 multiscale temporal feature를 추출한 후, 추출된 feature를 통합하고 이를 통해 이상 점수를 구하는 방법이고, ITAD(Shin et al., 2020)는 분해된 텐서에 대해 클러스터링을 수행하는 방법이다. 재구성 기반 모델은 재구성 오류로 이상을 감지하는 방법이며, Parket al. (2018)은 시간 모델링을 위해 LSTM을 사용하고 재구성을 위해 Variational Autoencoder (VAE)를 사용하는 LSTM-VAE 모델을 제시하였고, Su et al.이 제안한 Omni-Anomaly (2019)는 정규화 흐름으로 LSTM-VAE 모델을 추가로 확장하고 탐지를 위해 재구성 확률을 사용한다. Li et al.의 InterFusion (2021)은 백본을 계층적 VAE로 변형하여 여러 시리즈 간의 상호 및 내부 종속성을 동시에 모델링한다. GANs(Goodfellow et al., 2014)은 재구성 기반 이상 감지(Schlegl et al., 2019; Li et al., 2019a; Zhou et al., 2019)에도 사용되며 적대적인 정규화(adversarial regularization)를 수행한다[2].

III. 데이터 후처리 보정을 통한 성능 개선

3.1 제안의 배경

비지도 학습에서 정상 데이터를 이용하여 학습한 후, 예측한 결과와 실제 결과의 차이를 통해 이상 탐지를 하는 방법의 흐름은 그림2와 같다[3].

타겟 라벨이 없는 훈련 데이터를 알고리즘의 특성에 맞게 데이터 전처리를 한다. 입력할 특성을 선택하고 비어있는 데이터를 보충하고, 튀어 나와 있는 이상치(outlier)를 제거하는 표준화 작업을 한다[18][19]. 전처리된 데이터는 훈련(학습)에 입력되고 학습 알고리즘이 구성된 모델을 통해 학습과정을 거치게 된 후, 검증 데이터를 준비하였다면, 검증 데

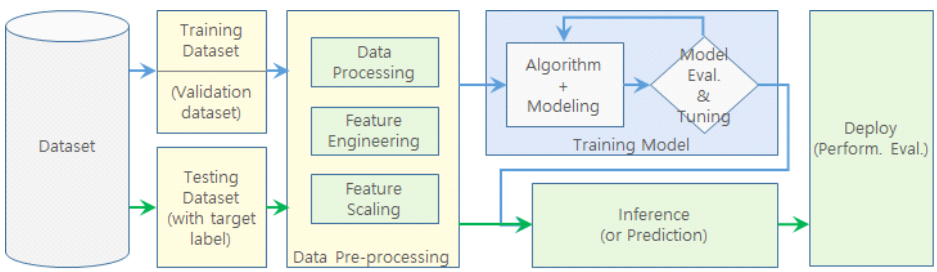

Fig. 2. Unsupervised Learning Workflow

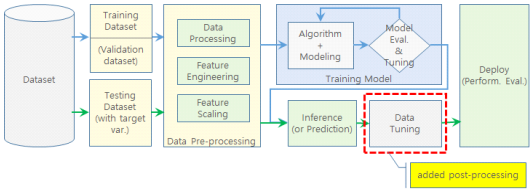

Fig. 3. Unsupervised Learning Workflow with post-processing

이터를 통해 해당 모델을 평가하고 보정하는 과정을 반복한다. 타겟 라벨이 있는 테스트 데이터와 모델을 통해 예측하는 과정을 거쳐 제품에 적용이 되거나 또는 입력된 데이터와 예측한 미래의 데이터와의 관계를 추론하는 단계를 거쳐 제품에 적용이 되거나 성능 평가를 한다[20].

본 연구에서 제안하고자 방법은 산업제어시스템의 데이터셋을 통해 이상 탐지 성능을 향상시키기 위한 방법으로, 적절한 학습 알고리즘을 변경하는 방법이 아닌, 추론 단계 이후의 결과 데이터를 보정하여 이상 탐지 성능을 향상시키고자 하였다. 학습 알고리즘의 개선을 통한 접근 방법이 아닌 데이터 후처리 보정을 통해 접근하고자 하였다. 그림3에 추가된 과정을 도식화 하였다.

3.2 데이터 보정의 필요성

데이터 보정의 과정을 위해 HAICon 2021의 1위 수상작으로 공개된 오리도리팀의 모델을 일부 변경하여 사용하였고, 데이터셋은 HAICon 2021에서 제공한 데이터셋이 아닌, HAI 2.0(21.03) 데이터셋을 이용하였다.

경진대회에서 제공한 데이터셋에는 검증 데이터셋에서 공격 포인트를 7개 밖에 제공하고 있지 않아서 연구에 필요한 비정상 형태의 특징을 찾아내기 어렵기 때문이다. HAI 2.0(21.03)의 데이터셋에는 테스트 데이터셋이 50개의 공격 포인트를 포함하고 있다. 각 공격 포인트에서 보인 비정상의 형태를 분석하여 보정의 방향을 잡고 최종적으로는 HAICon 2021의 데이터셋을 적용하여 eTaPR의 F1 Score로 수상작들과 비교하도록 한다.

HAI 2.0(21.03) 데이터셋은 정상 데이터를 가진 11일간의 훈련 데이터 파일 3개와 공격 포인트 50개를 가진 5일간의 테스트 파일 5개로 구성되어 있다.

3개의 훈련 데이터를 입력받아 전처리 과정을 거쳐 학습을 한 후, 테스트 데이터로 추론(예측결과와 실제 데이터와의 차이)하여 필터링한 결과를 그래프로 확인하면 그림4와 같다.

그림4에서 윗부분은 학습모델을 통한 예측치와 공격 구간을 포함하고 있는 테스트 데이터셋의 차이를 추론한 결과이고, 밑부분(주황색)은 50개의 실제 공격 포인트, 중간부분은 실선 Threshold(0.020) 설정을 나타낸다.

Threshold를 넘어서 표시된 것을 공격으로 간주하여 이상 탐지로 정한다. 다시 말하면, 공격이 있었을 때, 산업제어시스템의 각 필드 데이터에서 정상일 때 보다 큰 값을 출력하였다는 것을 의미한다.

eTaPR(v21.8.2-py3-none-any)을 통해 평가한 이상 탐지 성능은 다음과 같다.

- F1: 0.695 (TaP: 0.735, TaR: 0.659), # of detected anomalies : 40 / (공격 50)

F1 Score는 0.695이고, 실제 공격 50개 중에 40개를 탐지하였다. 그런데, Threshold를 넘어선 것으로 표시된 비정상 그래프의 수를 확인해 보면 45개가 된다. 본 연구의 시발점이 된 것은 이 차이에 대한 궁금증에서 시작되었다.

그림4의 그래프를 일부 확대하여 비정상 그래프가 있는 부분을 보면 그림 5와 같다.

밑부분의 네모 막대가 나타내는 것은 연속적으로 공격이 있었던 시간적 구간을 나타내는데, 첫 번째 그래프의 경우에는 Threshold에 걸쳐진(비정상 그래프와 Threshold의 교차하는 부분) 구간과 공격 구간과의 시간적 차이가 많이 어긋난 것을 확인할 수 있다. eTaPR의 평가산식에 의해 실제 공격 구간과 일정 비율 이상의 차이가 발생하여 탐지 제외를 한 것이다.

해당 부분을 좀 더 확대한 후, Threshold에 의해 탐지된 구간과 실제 공격 구간과의 차이를 표시해

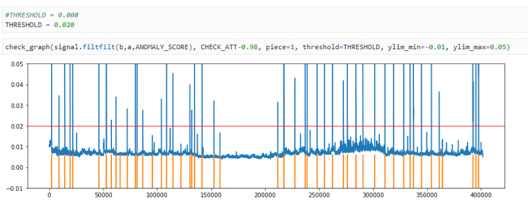

Fig. 4. Anomaly detection(AD) graph of HAI 2.0 dataset

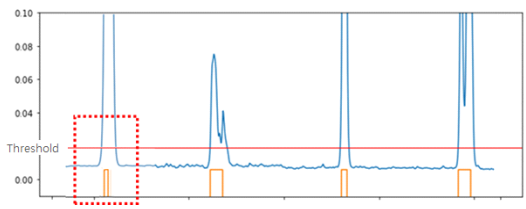

Fig. 5. An enlarged part of AD graph

보면 그림 6과 같다.

HAICon에서는 참가자들이 생성한 모델의 성능 평가를 위하여 비정상으로 탐지한 시간적 구간을 1로 라벨링 하여 제출하면 실제 공격 구간과 비교하여 eTaPR로 성능을 평가하게 된다. 하지만, Threshold와 교차하는 포인트를 탐지구간의 시작과 끝으로 정하여 이상 탐지 구간으로 라벨링을 하게 되면, 실제 공격 구간과 오차가 발생할 수밖에 없고, 이 오차가 클수록 탐지 성능은 낮아지게 된다.

결국, 이 오차를 최소화하면, 실제 공격 구간과의 차이가 줄어들게 되고, 이는 실제 공격 구간에 탐지하는 정확도가 높아지게 되고, 비정상의 시작과 끝을 좀 더 정확히 판단할 수 있다는 것은 실제 산업제어 시스템을 운전하는 시간적 흐름 속에서 이상징후의 분석에 드는 비용을 줄일 수 있는 효과로 나타나게 된다.

하지만, 일반적인 상황에서는 공격 구간을 알지 못하게 되고, 비정상의 시작점과 끝점을 보정하는 것은 쉬운 일이 아니다. 공격 구간을 모르는 상황에서 확인할 수 있는 것은 비정상 그래프의 형태인데, 이 그래프의 형태에서 공통적인 특징을 찾아내어 그 그래프의 보편적인 보정값을 찾아낼 수 있다면, 공격

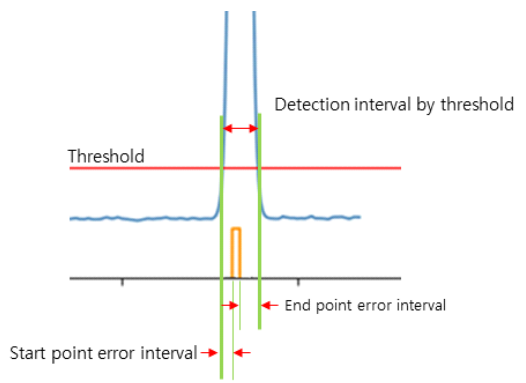

Fig. 6. difference of detection by threshold and actual attack interval

구간을 탐지하는 정확도를 향상시킬 수 있게 된다.

3.3 실험 과정

비정상 그래프의 형태를 분석하고 공통적인 특징을 찾아내기 위해 다음과 같이 진행하였다.

1) Threshold로 탐지한 비정상의 시작점과 끝점을 그래프 상에서 비정상이 되는 시작점과 끝점으로 확장한다.
2) 실제 공격 구간과 확장한 시작점과 끝점의 구간 사이의 차이를 구하고, 기본 보정값(보정1)을 구한다.
3) 비정상 그래프의 형태를 높이와 넓이(폭)의 각 3단계로 분류한다.
4) 각 분류별로 시작점과 끝점의 차이가 + 와 - 의 공통적인 형태가 있는지 확인한다.
5) 확인된 공통적인 형태에서 F1 Score가 크게 나오는 추가 보정값(보정2 ~ 보정3)을 구한다.
6) 기본 보정값을 적용한 상태에서 4)~5)를 반복하여 추가 보정값을 찾는다.

3.3.1 Threshold에 의해 탐지된 비정상의 시작점과 끝점의 확장

Threshold에 의해서 탐지된 구간은 비정상 그래프의 시작점과 끝점과는 차이가 있다. 그림 7에서 볼 수 있듯이 거의 대부분의 경우에 Threshold에 의해서 탐지된 구간을 비정상이 시작된 포인트로 확장해야 실제 비정상의 시작과 끝이 된다. Threshold에 의해 탐지된 구간은 실제 비정상의 시작과 끝의 차이가 있다.

비정상의 시작과 끝을 확장하는 방법은 다음과 같다.

- Threshold와 비정상 그래프가 만나는 시작과 끝의 포인트에서 공격 구간이 가질 수 있는 충분한 시간 간격(300~500) 만큼 떼어낸다.(그림8 (a))
- 이상치(outlier) 탐색 방법을 이용하여 최적의 시작점과 끝점을 찾는다.(그림8 (b))
- 기존 Threshold에 의해서 찾은 비정상의 시작과 끝을 새로 찾아낸 시작점과 끝점으로 대체한다.(그림8 (c))

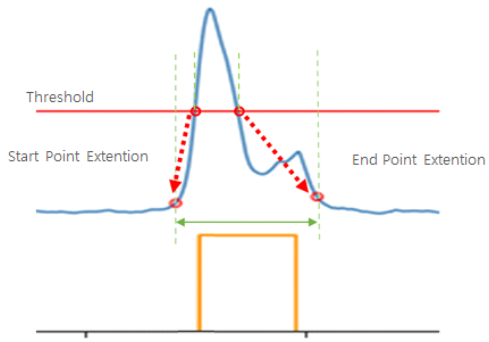

Fig. 7. extension of start/end point of abnormal graph

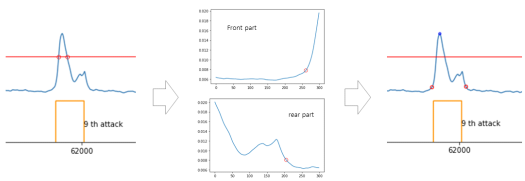

Fig. 8. Extension procedure of start/end point of abnormal graph

이 과정에서 중요한 점은 이상치(outlier) 포인트를 어떻게 찾을 것인지 이다. 이것을 위하여 '이상치 탐지(https://zephyrus1111.tistory.com/96)'와 stackoverflow를 검색하여 유사 질문의 답변 코드(https://stackoverflow.com/questions/32486697/find-start-and-end-of-abrupt-changes-in-an-arrary-in-python)를 참조하여 수행하여 보았으나. 그림 9와 같이 원하는 결과가 나오지 않거나, 특정 비정상 그래프에는 맞지 않는 부분들이 있었다.

그래서 pandas의 rolling.mean() 함수를 사용하여 구간 평균을 구하고, 이전 값과의 차이를 이용하는 방법으로 그림10의 코드를 통해 그림7의 결과

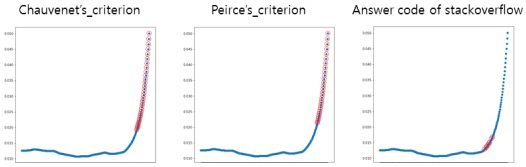

Fig. 9. (failure) finding outliers of abnormal graph

```
# Function that finds the point where the abnormal starts from the received abnormal data.
def get_abn_start_point(abn_dataset) :

    # Find a moving average with a window of 20.
    avg_data = np.array(pd.DataFrame(abn_dataset).rolling(20).mean() )
    temp1 = np.round(avg_data[1:] - avg_data[:-1], 5)   # a[x+1] - a[x]
    temp2 = np.nonzero([np.sum(temp1[i:i+30])>=0.0030 for i in range(len(temp1))])
    # A condition that is more relaxed than the previous condition
    if len(temp2[0]) == 0 :
        temp2 = np.nonzero([np.sum(temp1[i:i+20])>=0.0020 for i in range(len(temp1))])

        return temp2[0][0]  # abnormal starting point

get_abn_end_point = get_abn_start_point
```

Fig. 10. (success) finding outliers code of abnormal graph

와 같은 원하는 시작점 끝점을 찾을 수 있었다.

코드에서 사용된 숫자는 테스트를 통해 찾아낸 값이며, 데이터셋이 바뀌면 상황에 맞게 변경되는 값이다.

3.3.2 실제 공격 구간과 확장한 시작점과 끝점의 차이에 의한 기본 보정값 계산

각 비정상 그래프 형태의 시작점과 끝점을 확장한 후, HAI 2.0(21.03) 데이터셋에 포함된 실제 공격 포인트의 시작점과 끝점의 차이를 각각 계산하면 그림11의 표와 같다.

그림11의 표에서도 볼 수 있듯이, 대부분의 경우에 시작점은 + 크기의 차이를 보이고, 끝점은 - 크기의 차이를 보이고 있다. 이 차이를 통해서 찾아낸 기본 보정(보정1)의 값은 다음과 같다.

- 보정 1 : 시작점 +45, 끝점 -30% (비정상 구간의 크기에 대한 비율)

보정값은 시작점과 끝점의 최소값 부터 평균값에 이르기까지 전수 조사하여 F1 Score가 가장 높게 나타나는 보정값을 선정하였다.

Attack Count	Attack			Difference		Anomaly detection		
	Start point	End point	Interval	Start point	End point	Start point	End point	Interval
1	2072	2264	192	-5	-33	2077	2297	220
2	8852	8950	98	30	-49	8822	8999	177
3	14312	14502	190	44	-78	14268	14580	312
4	19227	19287	60	59	-64	19168	19351	183
5	21762	21851	89	59	-70	21703	21921	218
6	45943	46026	83	51	-152	45892	46178	286
7	52603	53025	422	20	-149	52583	53174	591
		⋮			⋮		⋮	
43	337498	337701	203	51	-85	337447	337786	339
44	344932	345085	153	52	-7	344880	345092	212
45	353994	354073	79	74	-135	353920	354208	288
46	361315	361366	51	48	-83	361267	361449	182
47	364080	364321	241				미탐지	
48	392450	392712	262	46	-151	392404	392863	459
49	395209	395329	120	64	-81	395145	395410	265
50	397668	397930	262	42	-113	397626	398043	417

Fig. 11. Interval difference of actual attack and AD

끝점의 경우에는 특정한 고정값으로 보정값을 정하는 것보다 비정상 구간의 넓이(폭)의 비율로 보정하는 것이 F1 Score가 더 높게 나왔다.

시작점의 보정값(+45)은 탐지된 45개의 비정상 그래프들의 시작점에 45초를 더하면 실제 공격 구간의 시작점과 평균적으로 좀 더 비슷해지는 것이고, 끝점의 보정값(-30%)은 비정상 그래프들의 끝점에 비정상 구간의 넓이에 30%를 빼주면 실제 공격 구간의 끝점에 좀 더 가까워지는 것이다.

시작점의 보정값은 공격이 시작한 이후 필드 디바이스들이 영향을 받기 시작하는 시점이 대체로 고정적이라는 것을 의미하며, 한편으로는 그래프가 Low pass filter를 통과한 영향도 존재한다. 끝점의 경우에는 공격 기간, 공격 종류, 타겟에 따라 필드 디바이스들이 영향을 받는 정도의 차이가 있다는 것을 의미한다.

3.3.3 비정상 그래프 형태의 분류에 의한 추가 보정 보정값 계산

공격에 의해 비정상을 보이는 그래프의 일부 샘플은 그림5와 그림14에서 확인할 수 있다. 이와 같이 비정상 그래프들은 특정한 모양을 가지고 있다. 그리고 각 비정상 그래프의 형태를 넓이(폭), 높이와 이중피크로 다음과 같이 분류하였다.

분류의 기준은 넓이와 폭의 분포를 기준으로 공통적인 실험적 특성을 고려하여 그룹화 하였다. 예를 들면, 비정상 그래프의 높이가 0.45 이상인 형태에서 끝점의 차이가 상대적으로 크게 나타났고, 이중피크가 보여 지는 그래프의 형태에서도 끝점의 차이가 상대적으로 큰 값을 가지는 것이 확인되었다.

- 넓이(폭)에 의한 분류 : 좁다(225이하), 보통(350이하), 넓다(350이상) (그림12)
- 높이에 의한 분류 : 낮다(0.1이하), 보통(0.45이하), 넓다(0.45이상) (그림13)
- 이중피크 여부에 의한 분류 : O/X

이중피크의 형태는 그림14의 형태와 같은 것들을 말하며, 파이썬 라이브러리 scipy에 포함되어 있는 find_peaks() 함수를 사용하여 정의하였다.

분류의 기준인 옵션 값은 끝점의 차이에서 특징적인 형태가 나타나는 값을 수동으로 찾아 설정하였다.

이렇게 분류하여 기본보정값(보정1)을 적용한 이후의 공격 구간의 시작점과 끝점의 차이를 정리하면 그림15의 표와 같다.

그림15의 표에 있는 비정상 모양(Anomaly Shape)을 기준으로 각 비정상 형태별로 시작점과 끝점의 차이를 확인하여 + 와 - 가 공통적으로 나타나는 형태를 찾은 결과, 이중피크(multi-peak)일 때 끝점의 차이가 + 형태(그림16)를 보인다.

이중피크일 때, F1 Score가 가장 높게 나타나는 보정값을 수동으로 찾아보니 끝점 +20 이다.

- 보정 2 : 끝점 +20

보정2를 적용한 후, 다시 반복하여 각 비정상 형태별로 시작점과 끝점의 차이를 확인하여 + 와 - 가 공통적으로 나타나는 형태를 찾은 결과, 높이가 높고

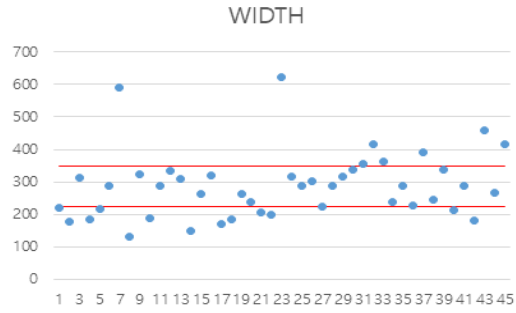

Fig. 12. Classification by width of AD

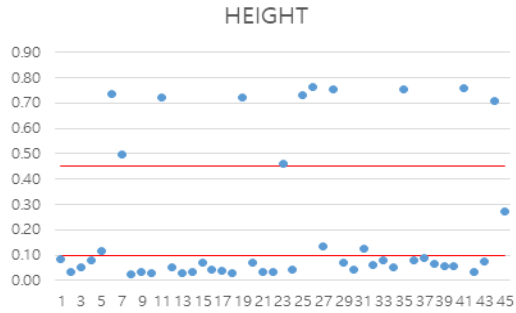

Fig. 13. Classification by height of AD

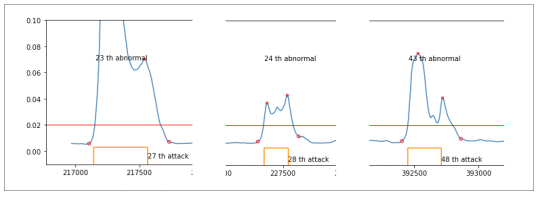

Fig. 14. Anomaly graph with multi-peak

Attack Count	Attack			Difference		Anomaly detection (After Tuning-1)					
	Start point	End point	Interval	Start point	End point	Start point	End point	Anomaly shape			Interval
1	2072	2264	192	-50	33	2122	2231	narrow	low		109
2	8852	8950	98	-15	4	8867	8946	narrow	low		79
3	14312	14502	190	-1	15	14313	14487	mid-width	low	multi-peak	174
4	19227	19287	60	14	-10	19213	19297	narrow	low		84
5	21762	21851	89	14	-5	21748	21856	narrow	mid-height		108
6	45943	46026	83	6	-67	45937	46093	mid-width	high		156
7	52603	53025	422	-25	28	52628	52997	wide	high	multi-peak	369
⋮											
43	337498	337701	203	6	16	337492	337685	mid-width	low	multi-peak	193
44	344932	345085	153	7	56	344925	345029	narrow	low		104
45	353994	354073	79	29	-49	353965	354122	mid-width	high		157
46	361315	361366	51	3	-29	361312	361395	narrow	low		83
47	364080	364321	241			미탐지					
48	392450	392712	262	1	-14	392449	392726	wide	low	multi-peak	277
49	395209	395329	120	19	-2	395190	395331	mid-width	high		141
50	397668	397930	262	-3	12	397671	397918	wide	mid-height	multi-peak	247

Fig. 15. Interval difference of actual attack and AD (with classification)

Attack Count	Attack			Difference		Anomaly detection (After Tuning-1)					
	Start point	End point	Interval	Start point	End point	Start point	End point	Anomaly shape			Interval
3	14312	14502	190	-1	15	14313	14487	mid-width	low	multi-peak	174
7	52603	53025	422	-25	28	52628	52997	wide	high	multi-peak	369
18	114466	114720	254	-17	58	114483	114662	mid-width	low	multi-peak	179
27	217145	217566	421	-9	21	217154	217545	wide	high	multi-peak	391
28	227353	227542	189	4	15	227349	227527	mid-width	low	multi-peak	178
33	262266	262521	255	16	94	262250	262427	mid-width	low	multi-peak	177
35	276147	276409	262	-15	43	276162	276366	wide	mid-height	multi-peak	204
36	284068	284331	263	24	40	284044	284291	wide	low	multi-peak	247
37	290549	290807	258	5	53	290544	290754	wide	low	multi-peak	210
43	337498	337701	203	6	16	337492	337685	mid-width	low	multi-peak	193
48	392450	392712	262	1	-14	392449	392726	wide	low	multi-peak	277
50	397668	397930	262	-3	12	397671	397918	wide	mid-height	multi-peak	247

Fig. 16. Interval difference of multi-peak

Attack Count	Attack			Difference		Anomaly detection (After Tuning-1+Tuning-2)					
	Start point	End point	Interval	Start point	End point	Start point	End point	Anomaly shape			Interval
6	45943	46026	83	6	-67	45937	46093	mid-width	high		156
12	79603	79671	68	22	-66	79581	79737	mid-width	high		156
22	134447	134631	184	-95	-51	134542	134682	mid-width	high		140
29	237067	237173	106	38	-14	237029	237187	mid-width	high		158
30	238865	238949	84	31	-51	238834	239000	mid-width	high		166
32	254947	255057	110	31	-17	254916	255074	mid-width	high		158
39	310969	311141	172	7	21	310962	311120	mid-width	high		158
45	353994	354073	79	29	-49	353965	354122	mid-width	high		157
49	395209	395329	120	19	-2	395190	395331	mid-width	high		141

Fig. 17. Interval difference of high-height

(high), 이중피크가 아닌 경우에 시작점과 끝점에서 + 와 - 의 공통된 형태(그림17)가 확인되었다.

마찬가지로 보정값을 찾은 결과는 다음과 같다.

- 보정 3 : 시작점 +19, 끝점 -17

3.3.4 각 보정별 F1 Score 비교

HAI 2.0 데이터셋의 train, test 파일들을 사용하여 학습하고 추론한 결과를 eTaPR로 성능 평가한 F1 Score와 보정1,2,3을 적용한 F1 Score를 비교하면 그림18과 같다.

보정 전의 F1 Score 0.695와 비교하여 보정 후의 F1 Score는 0.782로 약 12.52%의 이상 탐지 성능 개선이 있었고, 탐지 개수도 4개 증가하였다.

여기서 주의할 점은 보정2와 보정3은 각각 이전의

Type	Tuning Value	eTaPR			Remarks
		F1 score	(%)	Detection Count	
Before Tuning	-	0.695	-	40/50	TaP: 0.735, TaR: 0.659
Tuning-1	Start point +45 End point -30%	0.756	8.78%	43/50	TaP: 0.807, TaR: 0.710
Tuning-2	End point +20	0.766	10.22%	43/50	TaP: 0.820, TaR: 0.718
Tuning-3	Start point +19 End point -17	0.782	12.52%	44/50	TaP: 0.838, TaR: 0.733

Fig. 18. eTaPR comparison, tuning before and after of HAI 2.0

보정값을 적용한 결과에서 찾아낸 보정값이라는 것이다. 보정2는 보정1을 적용한 이후의 데이터에서 나타난 특징을 찾아 보정한 값이며, 보정3은 보정1과 보정2를 적용한 이후의 데이터를 통해 찾아`낸 값이다.

3.3.5 WINDOW size 축소 후 결과 비교

Window size를 축소하였을 때의 탐지 결과 및 비정상 형태 분류의 비교를 위해 기존 Window size 40을 20으로 1/2로 줄여서 동일한 과정의 실험을 진행하였다. 결과는 그림19와 같다.

HAI2.0 데이터셋을 사용하여 Window size 20으로 축소하여 진행한 실험 결과에서 보정 전, 후의 F1 Score는 기존 대비하여 거의 차이를 보이지 않았다. Window size 20에서는 최적의 탐지를 위하여 Threshold를 0.017로 변경하였고, 나머지 조건은 모두 동일하게 진행하였고, 보정에 필요한 시작점 끝점의 파라미터와 형태 분류에 필요한 조건값도 모두 동일한 값으로 진행하였다. 추론(inference)과정 후의 그래프 형태도 크게 다르지 않고 유사하여 동일한 값을 적용하였다. 다만, 일부 비정상 그래프의 형태에서 끝점이 맞지 않는 형태가 3건(그림20)이 있었지만, 수정하지 않고 진행하였고, Window size의 크기에 따른 보정의 조건은 바뀔 필요가 없었다. 보정 전의 F1 Score는 기존 Window size 40일 때보다 약 0.014(약 2%) 가량 작은 수치가 나왔다.

Type	WINDOW 40 (Threshold : 0.020)			WINDOW 20 (Threshold : 0.017)		
	F1 score	(%)	AD Count	F1 score	(%)	AD Count
Before Tuning	0.695	-	40	0.681	-	40
Tuning-1 (SP+45, EP-30%)	0.756	8.78%	43	0.740	8.66%	43
Tuning-2 (EP+20)	0.766	10.22%	43	0.746	9.54%	43
Tuning-3 (SP+19, EP-17%)	0.782	12.52%	44	0.759	11.45%	44

Fig. 19. eTaPR comparison of WINDOW size 40 and 20 (same condition of HAI 2.0)

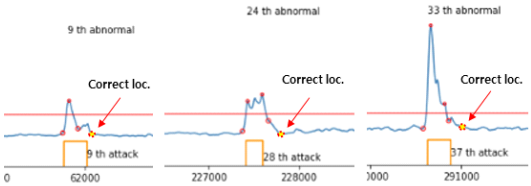

Fig. 20. Anomaly graph with incorrect end point

3.4 HAI 1.1 데이터셋을 이용한 동일 실험

이번에는 HAI 1.1 데이터셋을 통해 동일한 실험 과정을 진행하였다. HAI 2.0 데이터셋에서의 비정상 그래프를 유발한 공격 구간의 숫자가 충분하지 않을 수 있기 때문이다. HAI 2.0은 HAI1.1을 수집한 테스트베드를 업그레이드한 버전의 테스트베드에서 수집한 것이기 때문에 데이터셋의 특성이 유사할 것으로 판단하였다.

HAI 1.1 데이터셋은 훈련 데이터셋(train1.csv, train2.csv)과 테스트 데이터셋(test1.csv, test2.csv)로 구성되어 있는데, HAI 2.0과 달리 지도학습용 훈련 데이터셋(train2.csv)도 포함하고 있다. 공격 구간을 라벨링하지 않은 비지도 학습으로 진행하였기에 훈련 데이터셋은 train1.csv 파일 1개만 사용하였다. 그리고 HAI 1.1 데이터셋을 사용할 때는 테스트 데이터셋의 일부를 제거하였다.

동일 학습모델과 파라미터를 적용하여 공격 구간이 라벨링 되어 있는 테스트 데이터셋의 추론 결과를 확인해 보니, 데이터셋의 특성 때문인지, 학습과정이 부족한 이유인지는 알 수 없으나, 데이터셋의 앞부분과 뒷부분의 일정 영역에서 이상 패턴을 보이는 현상이 발생하였다. 그래서 앞부분은 약 13000초(1.5일)를 제거하였고, 뒷부분은 약 60000초(0.7일)를 제거한 후 테스트를 진행하였다. 본 연구의 목적이 학습 알고리즘 및 파라미터 설정에 관련한 부분이 아니기 때문에 실험 비용을 최소화하기 위함이다.

테스트 데이터셋에는 총 24개의 공격 구간이 있고, HAI 2.0에서 진행하였던 동일한 학습모델로 학습하여 테스트 데이터의 비정상을 탐지한 후, 비정상의 형태를 분류한 결과는 그림 21의 표와 같다.

Threshold의 최적값은 0.485이며, HAI 2.0의 0.02와는 많은 차이를 보인다. Threshold를 넘어선 비정상 그래프의 넓이와 높이 분포를 기준으로 실험적 특성을 고려하여 다음과 같이 분류하였다.

Attack Count	Attack		Difference		Anomaly detection(before tuning)				
	Start point	End point	Start point	End point	Start point	End point	Anomaly shape		
1	31327	31673	163	-222	31164	31895	mid-width	mid-height	multi-peak
2	37762	38275	-33	98	37795	38177	mid-width	mid-height	
3	41562	41906	175	-140	41387	42046	mid-width	mid-height	multi-peak
4	48263	48628	109	-123	48154	48751	mid-width	low	
5	52163	52793	98	248	52065	52545	mid-width	low	
6	56442	56788	140	-152	56302	56940	mid-width	mid-height	multi-peak
⋮									
15	145045	145597	86	-141	144959	145738	wide	mid-height	multi-peak
16	148762	149275							
17	160207	160593	-80	-242	160287	160835	mid-width	low	
18	166764	167161	-60	-7	166824	167168	narrow	mid-height	
19	224923	225301	32	75	224891	225226	narrow	low	
20	227923	228213	-7	-199	227930	228412	mid-width	low	
21	231723	232068	178	-51	231545	232119	mid-width	mid-height	multi-peak
22	235743	236089	131	-56	235612	236145	mid-width	mid-height	multi-peak
23	243833	246721	115	-166	243718	246887	wide	mid-height	multi-peak
24	249533	249868	50	-195	249483	250063	mid-width	high	

Fig. 21. Interval difference of actual attack and AD (with classification, HAI 1.1)

- 넓이(폭)에 의한 분류 : 좁다(350이하), 보통(750이하), 넓다(750이상)
- 높이에 의한 분류 : 낮다(0.6이하), 보통(1.0이하), 넓다(01.0이상)
- 이중피크 여부에 의한 분류 : O/X

(find_peaks()의 옵션 : distance=150, height=THRESHOLD)

비정상 그래프의 형태는 HAI 2.0에서 보여졌던 그림5, 그림14, 그림20의 모양과 유사하였다.

우선, 이전 실험에서 사용하였던 보정 1, 2, 3의 보정값을 그대로 사용하여 그림22과 같은 결과를 얻었다.

HAI 2.0의 보정에서와 마찬가지로 보정3까지 적용하였을 때, F1 Score 0.708로 보정 전과 비교하여 약 11.15%의 탐지 성능 개선이 있었다. 다음으로는 HAI 1.1 테스트 데이터셋에 있는 공격 라벨링의 값을 참조하여 실험을 통해 가장 높은 F1 Score를 가지는 보정값을 찾아보니, 보정값 적용이 가능한 비정상 형태의 분류가 HAI 2.0에서와 동일하게 나타났다.

Type	Tuning Value	eTaPR(v21.8.2)			Remarks
		F1 score	(%)	Detection Count	
Before Tuning	-	0.637	-	19/24	TaP: 0.730, TaR: 0.565
Tuning-1 (defalt)	Start point +45 End point -30%	0.698	9.58%	20/24	TaP: 0.775, TaR: 0.635
Tuning-2 (multi-peak)	End point +20	0.708	11.15%	20/24	TaP: 0.787, TaR: 0.644
Tuning-3 (high)	Start point +19 End point -17	0.708	11.15%	20/24	TaP: 0.787, TaR: 0.644

Fig. 22. eTaPR comparison, tuning before and after (tuning value of HAI 2.0)

Attack Count	Attack		Difference		Anomaly detection				
	Start point	End point	Start point	End point	Start point	End point	Anomaly shape		
1	31327	31673	63	-33	31264	31706	mid-width	mid-height	multi-peak
3	41562	41906	75	27	41487	41879	mid-width	mid-height	multi-peak
6	56442	56788	40	9	56402	56779	mid-width	mid-height	multi-peak
11	131432	131996	74	92	131358	131904	wide	mid-height	multi-peak
12	135632	136196	-4	27	135636	136169	mid-width	mid-height	multi-peak
15	145045	145597	-14	62	145059	145535	wide	mid-height	multi-peak
21	231723	232068	78	91	231645	231977	mid-width	mid-height	multi-peak
22	235743	236089	31	73	235712	236016	mid-width	mid-height	multi-peak
23	243833	246721	15	754	243818	245967	wide	mid-height	multi-peak

Fig. 23. Interval difference of multi-peak of HAI 1.1

Type	Tuning Value	eTaPR(v21.8.2)			Remarks
		F1 score	(%)	Detection Count	
Before Tuning	-	0.637	-	19/24	TaP: 0.730, TaR: 0.565
Tuning-1 (defalt)	Start point +45 End point -30%	0.710	11.46%	20/24	TaP: 0.793, TaR: 0.643
Tuning-2 (multi-peak)	End point +20	0.725	13.81%	20/24	TaP: 0.811, TaR: 0.655
Tuning-3 (high)	Start point +19 End point -17	Not applied			

Fig. 24. eTaPR comparison, tuning before and after (new tuning value)

보정1은 형태 분류가 아닌 수치에 의한 기본 보정값이고, 보정2(이중피크)에 대한 분류는 그림23의 표와 같다.

보정3(높이높다+이중피크제외)의 경우는 1개 밖에 분류되지 않아 적용에서 제외하였다.

HAI 1.1에 최적화된 보정값을 적용하여 얻은 탐지성능 결과는 그림 24와 같다.

보정 전과 대비하여, F1 Score 0.725 이면서 약 13.81% 개선을 한 것으로 결과가 나왔고, 탐지개수는 1개 증가하였다.

IV. 성능 개선 검증

지금까지는 HAI 2.0과 HAI 1.1 데이터셋을 사용하여 실제 공격 구간을 알고 있는 상태에서 보정할 비정상 형태와 보정값을 찾는 방식에 대해 실험하였다. 이 보정의 방법을 검증하기 위해서 HAICon 2021 경진대회에서 제공된 데이터셋을 사용하여 실제 경진대회의 진행방식과 동일한 방법을 통해 검증하고자 한다. 또한, 특정 데이터셋에만 특화된 보정의 방법인지 확인을 위해 HAI 데이터셋이 아닌 해외의 SWaT 데이터셋을 사용하여 검증을 진행하였다.

4.1 HAICon 2021 데이터셋을 사용한 검증

HAICon 2021에서 제공된 데이터셋은 훈련 데이터셋, 검증 데이터셋, 테스트 데이터셋으로 구성되어 있다. 검증 데이터셋에 포함된 7개의 공격 포인트를 통해 하이퍼파라미터를 튜닝하고, 이상 탐지를 위한 Threshold를 정하게 된다. 테스트 데이터를 통해 탐지한 비정상의 구간을 라벨링 하여 제출하면 주최 측에서 실제 공격 구간이 라벨링 된 데이터를 사용하여 eTaPR로 탐지성능을 평가하게 된다.

HAICon 2021 경진대회 웹사이트의 리더보드 게시판에는 Public Score 순위와 Private Score 순위, 그리고 최종 순위가 있는데, 본 검증에서는 확인이 가능한 Public Score의 비교를 통해 추론 데이터 보정에 의한 탐지 성능 개선 효과를 확인하도록 한다.

HAICon 2021 데이터셋을 사용한 진행과정은 이전의 과정과 동일하게 진행하였다.

정상 데이터를 가지고 있는 훈련 데이터셋 파일을 불러온 후, 데이터 전처리 과정에서 데이터를 Normalization 한 후, 딥러닝 학습 모델을 통해 학습을 진행하고, 검증데이터를 통해 추론한 후, 공격에 의한 비정상 그래프의 탐지 결과에 대해 가장 높은 eTaPR의 F1 Score가 나오는 Threshold를 찾아낸다. 테스트 데이터셋에 동일한 Threshold를 설정하여 탐지한 비정상의 구간을 라벨링 하여 Public Score를 구하는 과정으로 진행된다.

경진대회를 주최한 국가보안연구소 담당자에 문의하여 HAICon 2021에서 사용한 Public Score 데이터셋을 확보하였다. HAI 3.0(22.04) 데이터셋의 테스트 데이터셋 파일의 test2.csv ~ test4.csv의 공격 라벨링과 동일하다는 답변을 받았다. HAI 3.0(22.04) 데이터셋은 HAICon 2021에서 제공된 데이터셋으로 만들었다는 것을 알 수 있다.

확인 결과, HAI3.0의 테스트 데이터셋은 경진대회의 테스트 데이터셋과 동일하였고, Public Score의 F1 Score를 측정할 수 있는 라벨링을 가지고 있었다.

테스트 데이터셋에 사용하여 이상 탐지한 Public Score(보정 전)는 다음과 같다. Threshold 0.006을 설정하여 평가한 결과이다.

- F1: 0.695 (TaP: 0.756, TaR: 0.642), # of detected anomalies: 38

HAI 2.0 데이터셋에서 사용하였던 보정 형태와 보정값을 변경 없이 그대로 사용하여 검증을 진행하였다.

보정은 다음과 같이 진행하였다.

HAI 2.0 과 HAICon 2021 데이터셋은 데이터의 종류(필드 디바이스)가 추가 되었고, 공격의 종류가 추가되었고, 훈련데이터에 대한 학습 결과 등의 여러 다른 부분이 있지만, 우선적으로는 HAI 2.0 데이터셋에서 적용하였던 보정1, 2, 3의 보정 형태와 보정값을 동일하게 적용하여 탐지성능을 확인한 후, 추가적으로 최고의 F1 Score를 나타내는 보정값을 찾기 위한 실험을 진행하였다.

다만, 비정상 그래프의 형태를 높이와 넓이(폭)의 각 3단계로 분류하는 기준값은 변경되어야 했다. HAI 2.0 데이터셋에서는 Threshold가 0.02이었지만, HAICon 2021 데이터셋에서는 Validation 데이터셋에서 찾은 Threshold가 0.006에서 최고의 탐지 성능을 보여준 것과 같이 차이가 크기 때문이다.

비정상 그래프의 형태를 구분하기 위해 탐지된 비정상의 분포와 공통적인 특징을 보이는 기준값을 정한 결과는 다음과 같다.

- 넓이(폭)에 의한 분류 : 좁다(80이하), 보통(300이하), 넓다(300이상)
- 높이에 의한 분류 : 낮다(0.03이하), 보통(0.10이하), 넓다(0.10이상)
- 이중피크 여부에 의한 분류 : O/X
  (find_peaks()의 옵션 : distance=70, height=0.002)

HAI 2.0 데이터셋 보정시 사용하였던 보정 형태와 보정값을 그대로 적용한 Public Score의 F1 Score를 보정 이전과 비교한 결과는 그림25의 표와 같다.

보정을 수행한 결과는 보정 전과 대비하여 약 7.77%의 성능 향상이 나타났다. HAI 2.0 데이터셋의 12.52%에 비해 작은 비율 이지만 성능 개선의 효과가 있음을 확인하였다.

HAI 2.0 데이터셋에서 찾았던 보정값이 아닌, HAICon 2021 데이터셋을 통해 Public Score가 가장 높게 나타난 보정값을 찾아낸 결과는 그림26의 표와 같다. HAI 2.0에서 찾은 보정값에서 +/- 5의 값을 변경하면서 찾은 값이다.

보정1 ~ 보정3 까지 적용하였을 때, F1 Score가 0.766으로 보정 전의 값에 비해 약 10.22%의

Type	eTaPR			Remarks
	F1 score	(%)	AD Count	
Before Tuning	0.695	-	38/51	TaP: 0.756, TaR: 0.642
Tuning-1	0.716	3.02%	38/51	TaP: 0.775, TaR: 0.665
Tuning-2	0.725	4.32%	38/51	TaP: 0.786, TaR: 0.672
Tuning-3	0.749	7.77%	40/51	TaP: 0.811, TaR: 0.695

Fig. 25. eTaPR comparison, tuning before and after (HAI 2.0 tuning value applied to HAICon 2021)

Type	eTaPR			Remarks
	F1 score	(%)	AD Count	
Before Tuning	0.695	-	38/51	TaP: 0.756, TaR: 0.642
Tuning-1	0.740	6.47%	41	TaP: 0.800, TaR: 0.687
Tuning-2	0.750	7.91%	41	TaP: 0.814, TaR: 0.696
Tuning-3	0.766	10.22%	41	TaP: 0.829, TaR: 0.711

Fig. 26. eTaPR comparison, tuning before and after (new tuning value applied to HAICon 2021)

탐지성능 개선이 있었다. HAI 2.0 데이터셋을 사용한 실험결과(12.52%)와 유사한 비율의 이상 탐지 성능의 개선효과를 확인하였다.

Public Score의 F1 Score 0.766은 HAICon 2021의 리더보드의 Public Score 1위에 해당하는 점수이다. 현재 리더보드 상에서 Public Score 1위의 점수는 0.716 이다. (HAICon 2021 리더보드: https://dacon.io/competitions/official/235757/leaderboard)

4.2 SWaT 데이터셋을 사용한 검증

SWaT(Secure Water Treatment) 데이터셋은 SUTD(Singapore University of Technology and Design)의 iTrust 연구소에서 테스트베드를 만들어 수집한 데이터셋이며, HAI 데이터셋과 유사하게 제어시스템의 필드 디바이스로부터 수집한 데이터셋을 가지고 있다.

41개의 공격 시나리오에 대해 공격이 수행되었지만, 실제로 물리적인 영향을 준 것은 36개 이며, 이중에서 2개의 공격은 시간차가 없이 연속적으로 수행되어 실제로는 1개의 공격 구간으로 표시된다. 따라서, 본 실험에서는 35개 공격 구간으로 간주하고 진행하였다.

진행된 과정은 HAI 데이터셋의 진행과정과 동일하게 진행하였으나, 기존의 학습 모델에서 사용한 LSTM 알고리즘이 아닌 GRU 알고리즘으로 바꾸어 진행하였다. LSTM을 사용했을 때의 F1 Score보

다 GRU를 사용했을 때의 결과가 더 좋게 나왔기 때문이다. 전처리 과정의 Normalization 코드는 사이킷런의 MinMaxScaler() 함수로 변경하여 진행하였다.

사용된 학습모델은 다음과 같다.

```
StackedGRU(
  (rnn): GRU(51, 200, num_layers=3, dropout=0.1, bidirectional=True)
  (fc): Linear(in_features=400, out_features=51, bias=True)
  (relu): LeakyReLU(negative_slope=0.1)
  (sigmoid): Sigmoid()
  (dense1): Linear(in_features=51, out_features=25, bias=True)
  (dense2): Linear(in_features=25, out_features=51, bias=True)

BATCH_SIZE = 2024
epoch = 130
WINDOW_SIZE = 40
```

정상 데이터셋으로 학습하고, 공격 구간 35개를 가지고 있는 테스트 데이터셋으로 탐지한 결과는 다음과 같다.

F1: 0.420 (TaP: 0.753, TaR: 0.291)

# of detected anomalies: 13

Threshold를 0.035로 설정하였을 때, F1 Score는 0.420 이며, 이상 탐지한 공격 구간은 13개이다. 탐지한 13개의 비정상 구간에 대해 이전의 다른 시험과 동일한 보정을 과정을 진행하였고, 탐지한 전체 비정상 그래프는 그림 27과 같다.

탐지한 비정상 (13개) 의 시작점과 끝점을 확장하여 표시한 형태는 그림 29와 같다. 개별 그래프의 양 옆에 있는 원형의 점이 확장한 시작점과 끝점을 나타내며, 밑부분의 네모 상자 부분이 공격 구간을 나타낸다.

비정상 그래프의 시작점과 끝점을 확장한 이후에 공격 구간과의 시간적 차이를 표로 나타내면 그림

```
TaPR = etapr.evaluate_haicon(anomalies=ATTACK_LABELS, predictions=FINAL_LABELS)
print(f"F1: {TaPR['f1']:.3f} (TaP: {TaPR['TaP']:.3f}, TaR: {TaPR['TaR']:.3f})")
print(f"# of detected anomalies: {len(TaPR['Detected_Anomalies'])} ")
# print(f"(total {len(TaPR['Detected_Anomalies'])+TaPR['N False Alarm'])})")
# print(f"Detected anomalies: {TaPR['Detected_Anomalies']}")
F1: 0.420 (TaP: 0.753, TaR: 0.291)
# of detected anomalies: 13
```

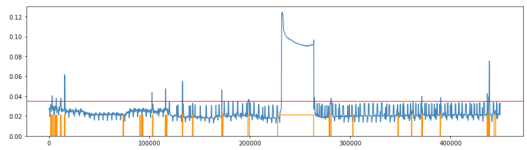

Fig. 27. Total anomaly graph of SWaT

Attack Count	Attack		Difference		Anomaly detection (before tuning)	
	Start point	End point	Start point	End point	Start point	End point
2	3029	3472	102	-53	2927	3525
5	7216	7412	101	-63	7115	7475
7	11371	12335	70	4	11301	12331
8	15341	16062	48	-139	15293	16201
13	103053	103770	518	808	102535	102962
15	116104	116499	148	-72	115956	116571
17	132879	133342	157	-115	132722	133457
19	172229	172550	123	106	172106	172444
21	198257	199702	96	-67	198161	199769
22	227789	263689	-3765	-1302	231554	264991
24	280021	281192	121	-480	279900	281672
28	371401	371502	43	-131	371358	371633
30	389602	390142	-372	-441	389974	390583
31	436463	436932	54	-159	436409	437091
33, 34	438069	438840	131	-505	437938	439345

Fig. 28. Interval difference of SWaT Dataset

28과 같다.

공격 구간과 이상 탐지의 시작점과 끝점의 차이를 보면, 시작점은 +형태, 끝점은 -형태의 공통적인 특징이 나타나고 있다. 보정값의 +/- 형태를 추정할 수 있으며, F1 Score가 가장 높게 나오는 보정값을 구해보면 다음과 같다.

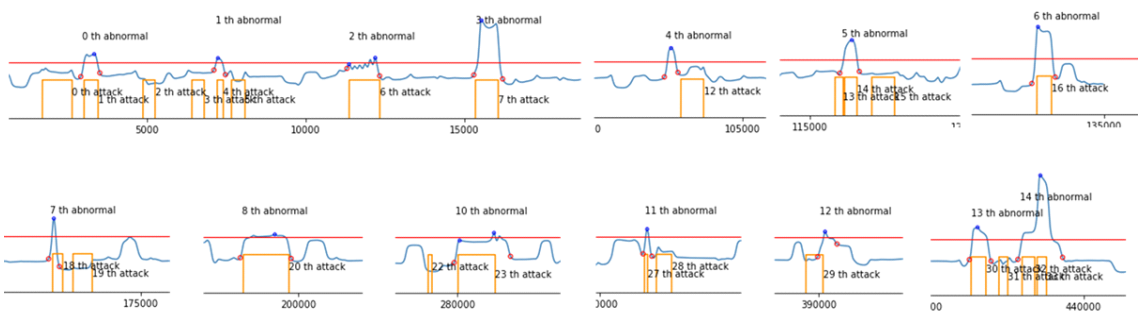

Fig. 29. Individual anomaly graph after start/end point extension(SWaT)

Type	Tuning Value	eTaPR(v21.8.2)			Remarks
		F1 score	(%)	Detection Count	
Before Tuning	-	0.420	-	13/35	TaP: 0.753 TaR: 0.291
Tuning-1	SP +30 EP -60	0.475	13.10%	15/35	TaP: 0.816 TaR: 0.335

Fig. 30. eTaPR comparison, tuning before and after (SWaT)

- 기본보정(보정1) :시작점 +30, 끝점 -60
- F1: 0.475 (TaP: 0.816, TaR: 0.335)
  # of detected anomalies: 15

기본보정(보정1) 이외의 비정상 형태 분류에 의한 보정값은 찾을 수 없었다. 추가적인 보정값을 찾기 위한 그래프 형태의 분류에서 공통점을 찾기에는 비정상 그래프의 수가 너무 적었기 때문에 더 이상의 보정값을 찾을 수 없었다.

기본 보정값(SP +30, EP -60)을 적용하기 이전과 이후를 비교하면 그림 30과 같다.

보정 전과 비교하여 13.1%의 이상탐지 성능 개선이 있으며, 탐지 개수는 2개 증가하였다.

탐지된 비정상의 개수가 적어서 비정상 그래프의 형태 분류에 의한 추가 보정은 할 수 없었지만, 기본 보정만으로도 후처리 보정의 효과를 확인할 수 있었다.

V. 결 론

본 논문에서는 산업제어시스템 데이터셋으로 국가보안연구소에서 공개한 HAI 2.0과 HAI 1.1 데이터셋을 이용하여 비지도학습에 의한 이상 탐지 성능을 향상시키기 위해 후처리(Post-processing) 데이터 보정의 과정을 추가하는 방법을 적용하였고, 보정 전과 후의 성능을 평가하고 비교하였다.

동일한 방법으로 HAICon 2021 경진대회에서 제공하였던 데이터셋을 사용하여 보정을 한 후의 탐지성능이 향상된 것을 확인할 수 있었고, 다른 종류의 SWaT 데이터셋에 동일한 보정 방법을 적용하여 성능 개선 효과를 확인하였다.

보정의 방법을 요약하면 다음과 같다.

딥러닝 비지도 학습 모델에 정상 데이터셋을 학습시킨 후, 공격구간이 포함된 테스트 데이터셋으로 추론을 한 후, 최적의 Threshold에 의해 탐지된 비정상 그래프의 시작점과 끝점을 확장하고, 공격 구간과의 시간적 차이 값에서 공통적인 +/- 의 기본 보정(보정1) 값을 구하고, 추가로 비정상 그래프의 형태를 구분하여 공통적인 특성을 찾아낸 후, 추가 보정값을 적용하는 과정이다.

본 연구를 통해 보정 이후에 이상 탐지 개수가 증가할 뿐만 아니라, 공격의 시작점과 종료 시점을 좀 더 정확하게 파악할 수 있게 되었다.

산업제어시스템은 운전 중에 공격을 받거나, 오동작하거나, 작업자의 실수와 같은 오류가 발생하는 경우에 각각의 필드 디바이스(컨트롤러, 센서, 액츄에이터 등)별로 나타나는 특성과 현상이 유사하기 때문에 비정상 형태 분류에 의한 보정의 효과가 나타난다. 이는 이상 탐지 성능 개선을 위해 최적의 딥러닝 학습 모델을 찾는 방법 이외의 또 다른 방법이 될 수 있다는 것을 보여준다.

비정상을 유발시키는 공격 구간을 충분히 많이 확보할 수 있다면, 보정의 형태와 보정값을 찾기 위한 방법으로 머신러닝을 이용할 수도 있고, 이는 실험을 통해 보정값을 찾는 방법보다 좀 더 정확한 형태의 분류와 보정값을 찾을 수 있다.

딥러닝 학습에 사용할 수 있는 데이터가 충분히 많아서 후처리 보정 없이 이상 탐지의 성능이 높게 나온다고 할지라도 후처리 보정을 진행한다면 개선율이 낮아지겠지만, 좀 더 정확한 이상 탐지 효과를 기대할 수 있다.

지금까지는 산업제어시스템의 이상 탐지 성능개선을 위해 주로 학습 모델 구성에 대한 연구가 진행되고 있는데, 데이터 전처리/후처리 과정을 통한 성능 개선의 지속적인 연구도 필요하다.

References

[1] Shalyga, Dmitry, Pavel Filonov, and Andrey Lavrentyev. "Anomaly detection for water treatment system based on neural network with automatic architecture optimization." arXiv preprint arXiv:1807.07282 (2018).

[2] Xu, Jiehui, et al. "Anomaly transformer: Time series anomaly detection with association discrepancy." arXiv preprint arXiv:2110.02642 (2021).

[3] Blázquez-García, Ane, et al. "A review on outlier/anomaly detection in time

series data." ACM Computing Surveys (CSUR) 54.3 (2021): 1-33.
[4] Braei, Mohammad, and Sebastian Wagner. "Anomaly detection in univariate time-series: A survey on the state-of-the-art." arXiv preprint arXiv:2004.00433 (2020).
[5] Filonov, Pavel, Andrey Lavrentyev, and Artem Vorontsov. "Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model." arXiv preprint arXiv:1612.06676 (2016).
[6] Shin, Hyeok-Ki, et al. "HAI 1.0: HIL-based Augmented ICS Security Dataset." 13th USENIX Workshop on Cyber Security Experimentation and Test (CSET 20). 2020.
[7] HAI v3.0(22.04), "hai dataset technical details v3.0", Retrieved from https://github.com/icsdataset/hai/blob/master/hai_dataset_technical_details_v3.0.pdf. Last accessed 15 Jun. 2022
[8] Hwang, Won-Seok, et al. "Do you know existing accuracy metrics overrate time-series anomaly detections?." Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing. 2022.
[9] Audibert, Julien, et al. "Usad: Unsupervised anomaly detection on multivariate time series." Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020.
[10] Choi, Seungoh, Jeong-Han Yun, and Sin-Kyu Kim. "A comparison of ICS datasets for security research based on attack paths." International Conference on Critical Information Infrastructures Security. Springer, Cham, 2018.
[11] Lee, Jong-Hu, Kim, U-Nyeon, "Industrial Control System Security Requirements Standard Introduction", TTA, 2017
[12] Hyeok-Ki Shin, Woomyo Lee, Jeong-Han Yun and Byung-Gil Min, "ICS security dataset", 2022. GitHub, Available at: https://github.com/icsdataset.
[13] DACON, Industrial Control Systems Security Threat Detection AI Competition https://dacon.io/competitions/official/235624. Last accessed 23 Jun. 2022
[14] iTrust: Swat datasets. https://itrust.sutd.edu.sg/itrust-labs_datasets/. Last accessed 23 Jun. 2022
[15] Morris, T.H.: Industrial control system (ics) cyber attack datasets. https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets. Last accessed 23 Jun. 2022
[16] Lemay, A.: Scada network datasets. https://github.com/antoine-lemay/Modbus_dataset. Last accessed 23 Jun. 2022
[17] Rodofile, N.R.: S7comm datasets. https://github.com/qut-infosec/2017QUT_S7comm. Last accessed 23 Jun. 2022
[18] Into The Data, data sciense wiki - anomaly detection, https://intothedata.com/02.scholar_category/anomaly_detection. Last accessed 25 Jun. 2022
[19] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.
[20] DATA SCIENCE BLOG, 2018, https://www.datascienceblog.net/post/commentary/inference-vs-prediction. Last accessed 25 Jun. 2022
[21] Bian, Xingchao. "Detecting Anomalies in Time-Series Data using Unsupervised Learning and Analysis

on Infrequent Signatures." Journal of IKEEE 24.4 (2020): 1011-1016.
[22] Bae, Sungho, Chanwoong Hwang, and Taejin Lee. "Research on Improvement of Anomaly Detection Performance in Industrial Control Systems." International Conference on Information Security Applications. Springer, Cham, 2021.
[23] Seong, ChangMin, et al. "Towards Building Intrusion Detection Systems for Multivariate Time-Series Data." Silicon Valley Cybersecurity Conference. Springer, Cham, 2021.
[24] HyoSeok Kim, Yong-Min Kim. "Abnormal Detection for Industrial Control Systems Using Ensemble Recurrent Neural Networks Model." Journal of The Korea Institute of Information Security & Cryptology 31.3 (2021).
[25] Kim, Doyeon, Chanwoong Hwang, and Taejin Lee. "Stacked-autoencoder based anomaly detection with industrial control system." International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing. Springer, Cham, 2021.
[26] ORIDORI, DACON, https://dacon.io/competitions/official/235757/codeshare/4600?page=1&dtype=recent. Last accessed 3 Jul. 2022

〈저자소개〉

전 상 수 (SANGSO JUN) 정회원
2000년 2월: 한양대학교 전자,전자통신,전파공학과 졸업
2017년 9월~현재: 고려대학교 정보보호대학원 석사과정
2000년 1월~현재: 삼성전자, SK인포섹 등 근무
〈관심분야〉 산업제어시스템(OT/ICS) 보안, 딥러닝 이상탐지, 모의해킹

이 경 호 (Kyung-ho Lee) 종신회원
1989년 8월: 서강대학교 수학과 학사
1997년 8월: 서강대학교 정보통신대학원 석사
2009년 8월: 고려대학교 정보보호대학원 박사
1994년 2월~현재: 삼성그룹, 네이버(주), 시큐베이스 등 근무
2011년 9월~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 위험관리, 정보보호컨설팅, 정보보호 및 개인정보보호정책