JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 30 No. 1, pp. 23-32, January 2025
https://doi.org/10.9708/jksci.2025.30.01.023

Research on AI-based anomaly detection for ship combat system weapon system interlocking control

Hyun-Ho Na*

*Engineer, Naval R&D Center, Hanwha Systems, Gumi, Korea

[Abstract]

This paper proposes the use of anomaly detection using LSTM AutoEncoder to verify the possibility of anomaly detection function through AI in the control environment of the interlocking weapon system of a naval combat system.. Performance data such as combat system logs and metric data were collected and time-series preprocessed with the ELK Stack. The LSTM AutoEncoder model, which uses the LSTM network-based Eocder to compress and dimensionally reduce data, and the Decoder to restore the input data to a similar form, was trained using only normal environmental data. Afterwards, the performance was evaluated using test data generated by simulating normal and abnormal situations, and a high score of Accuracy 0.99, Precision 0.97, Recall 0.87, and F1-Score 0.92 was output. This study confirmed the applicability of the model generated through machine learning in the detection of anomalies in the control environment of the interlocking weapon system of a naval combat system.

▸ **Key words**: Artificial Intelligence, Machine Learning, LSTM-Auto Encoder, ELK, Combat Systems

[요 약]

본 논문은 함정전투체계의 무장 체계 연동 통제 환경에서의 AI를 통한 이상 탐지 기능의 가능성을 확인하고자 LSTM AutoEncoder를 이용한 이상 탐지 활용방안에 대해 제안한다. 전투체계 로그와 Metric 데이터 등 성능 데이터를 ELK Stack으로 수집하고 시계열화하는 전처리를 수행하였다. LSTM 네트워크 기반의 Eocder를 사용하여 데이터를 압축 및 차원을 축소하고, Decoder를 이용해 입력데이터와 유사한 형태로 복원하는 LSTM AutoEncoder 모델을 정상 환경 데이터만으로 훈련 시켜 완성하였다. 이후, 정상과 비정상 상황을 모의하여 생성한 시험 데이터를 이용하여 성능 평가를 하여 Accuracy 0.99, Precision 0.97, Recall 0.87, F1-Score 0.92로 높은 점수가 출력되었다. 이를 통해 함정전투체계 무장 체계 연동 통제 환경 이상 탐지 시에 기계학습을 통해 생성한 모델의 적용 가능성을 확인하였다.

▸ **주제어**: 인공지능, 기계학습, LSTM-Auto Encoder, ELK, 전투체계

• First Author: Hyun-Ho Na, Corresponding Author: Hyun-Ho Na
*Hyun-Ho Na (na.hh1@hanwha.com), Naval R&D Center, Hanwha Systems
• Received: 2024. 11. 26, Revised: 2024. 12. 09, Accepted: 2024. 12. 24.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

최근 국방부는 국방혁신 4.0 기본계획 발표를 통해 제2창군 수준의 국방 재설계, AI과학기술강군 육성을 계획하고 있다. AI 기반 유·무인 복합체계를 활용한 경계 작전 개념을 발전시키고, 이를 위한 중대급 또는 대대급 시범부대를 내년부터 운용할 계획이다. 또한, AI 기반의 고성능 무기체계와 전력지원체계 개발 및 운용을 위해 양질의 국방 데이터를 선제적으로 구축하고, 국방 데이터의 체계적 관리를 시작하고 있으며, 국방 AI 분야를 전담하는 국방 AI 센터의 창설 및 발전을 위한 법과 제도적 기반도 마련해 나가고 있다[1]. '국방혁신 4.0을 통한 첨단과학기술군 육성' 측면에서, 또한, 국방부는 '국방정책방향' 세부 추진과제 이행방안에서 AI 기술 수준과 발전단계를 고려하여 Fig. 1.과 같이 '국방AI 발전모델'을 정립하였으며, 이에 따라 우리 군에 대한 AI 기술 적용을 단계적으로 확대해 나갈 것이라고 밝혔다. 따라서, 국방 AI의 도입은 필수이며, 함정 전투체계(Naval Combat Management Systems)의 AI 적용 방향성에 대해서도 고민이 필요한 현실이다[2].

Fig. 1. Defense AI 3-stage development model

함정전투체계는 함정에 탑재된 각종 센서를 이용하여 표적을 탐지하여 추적하고, 무장을 통제하여 교전을 수행하는 함정 작전 수행하는 역할을 담당하고 있다[3]. 무장통제는 함정전투체계 내에서 발사관과 무장을 이용하여 무장이 발사되도록 하며, 무장 통제(Weapon Control) 소프트웨어는 전투체계와 무장 체계를 이어주는 역할로 전투체계에서 식별한 위협 상황에 대해 무장 운용을 위한 소프트웨어이다. 그러므로 함정 전투체계에서 복합적이고 동시다발적인 전장 상황에 대응하기 위한 무장 통제 능력은 필수적이다[4]. 무장 통제 소프트웨어의 정상적인 운용이 필수적으로 필요하며, 현재 함정전투체계는 체계 상태 감시 기능으로 구동되는 소프트웨어의 상태정보, 연동되는 무장 관련 장치들의 연결정보 등을 보여줌으로써 실시간 상태를 감시할 수 있다. 이러한 소프트웨어 상태 및 연결 상태 감시뿐 아니라, 소프트웨어와 하드웨어가 구동되는 환경의 성능정보와 구동되는 과정에서 발생하는 각종 로그 및 Metric 정보들을 종합하여 이상 탐지를 할 수 있는 구조로 고도화된 상태 감시를 수행한다면 더욱 효율적으로 전투체계의 이상을 탐지하고 선제적으로 조치할 수 있을 것이다.

본 논문에서는 함정전투체계 내부의 무장 체계 연동 통제 환경의 실시간 상태 감시를 고도화하기 위한 부분에 관한 내용으로 AI 기술을 이용한 이상 탐지 방법에 대해 시험 및 제안을 하고자 한다. 다양한 이상 탐지 방법 중 기존 Rule 기반 방식을 탈피하여 AI의 적용 가능성을 확인하고자 AI 학습 방법의 비지도학습중에 하나인 Long Short-Term Memory(이하 LSTM) AutoEncoder 및 K-Means Clustering 방식을 사용하여 해당 환경의 이상 탐지를 수행할 수 있는 신경망 구성 및 시험을 수행하였다.

본 논문의 구성은 다음과 같이 이루어진다. 2절에서는 전투체계, Machine Learning, LSTM AutoEncoder, ELK에 대한 개념 및 scikit-learn API, K-Means Clustering에 관해 설명한다. 3절에서는 연구 내용에 대한 데이터 수집 및 데이터 전처리 적용 내용, 하이퍼 파라미터 설정 및 모델 훈련에 관해 설명한다. 4절에서는 학습한 모델인 LSTM AutoEncoder 신경망과 K-Means Clustering 알고리즘을 이용한 전투체계 무장 체계 연동 통제 이상 탐지 결과 분석 및 평가에 관해 설명하고, 마지막 5절에서는 전투체계에 AI 적용 가능성 판단 및 추후 연구과제에 관해 설명한다.

II. Preliminaries

1. Related works

1.1 Naval Combat Management System

함정 전투체계는 인간의 두뇌와 같은 역할을 수행한다. 함정에 탑재된 레이더, 소나와 같은 센서, 함포, 유도탄과 같은 무장과 기타 장비들을 통합해 작전에 필요한 모든 정보를 종합적으로 수집, 전술상황평가·지휘결심·위협평가·무기할당·교전 등의 기능을 효율적으로 수행할 수 있도록 지원하는 일련의 자동화된 무기체계를 함정 전투체계라고 한다. 현재 함정 전투체계는 Fig. 2.과 같이 지휘무장통제체계(Combat Fire CommandSystem/CFCS), 센서, 무장 및 데이터링크로 구성되며 함정의 임무와 특성에 따라 다양한 구성이 가능하다. 이 가운데에서 다양한 체계를 통

합해 최대의 성능을 발휘할 수 있도록 하는 핵심적인 역할은 CFCS가 수행한다. CFCS는 데이터를 처리하는 단일기판컴퓨터(Single Board Computer/SBC)가 내장된 정보처리장치(Information Processing Node/IPN), 운용자의 명령을 입력받고 전술 상황을 전시하는 다기능 콘솔(Multi Function Console/MFC), 센서 및 무장을 연동시키는 연동단(Interface Control Unit/ICU), 연동되는 체계 상호간 데이터를 전송하는 전투체계 통합 네트워크, 레이더비디오 및 TV/IR비디오를 분배·전송하는 영상분배장치(Radar Tv Video DistributorUnit/RTVDU) 그리고 각종 전시기 등으로 구성된다[5].

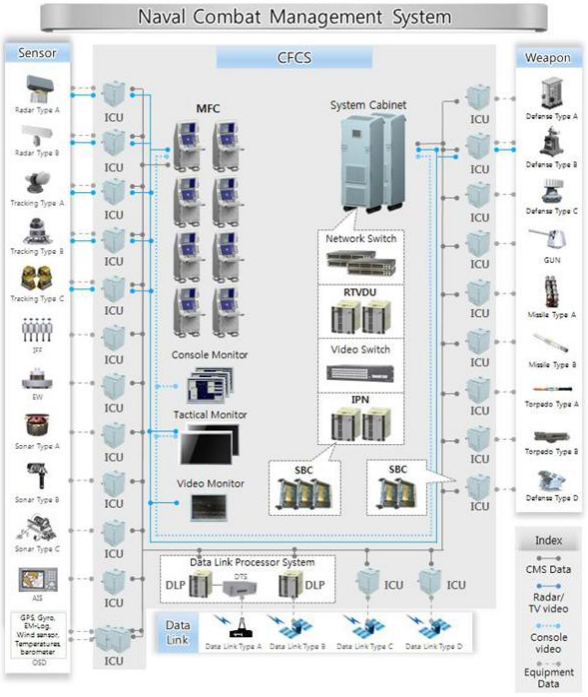

Fig. 2. Naval Combat Management System

1.2 Machine Learning(ML)

Machine Learning은 인공지능의 하위집합 중 하나로, 축적된 데이터를 이용해 컴퓨터가 새로운 규칙을 생성하며 학습하는 분야를 말한다. 기존의 Rule 기반 프로그래밍 방식을 탈피하여, 데이터를 기반으로 학습하고 모델을 만들어 분류, 회귀, 예측, 군집화 등 분야에 사용할 수 있다.

Fig. 3.과 같이 크게 지도학습, 비지도학습, 강화학습으로 정의된다. 비지도학습은 정답 데이터 없이 학습이 이루어지며, 데이터의 패턴을 학습하여 군집화, 차원 축소 등에 이용된다. 지도학습은 입력데이터와 정답(출력) 데이터를 기반으로 학습을 진행하며, 분류, 회귀 등에 이용된다. 강화학습은 환경에서의 상태를 인식 및 행동에 따른 최대 보상에 대해 학습하는 방법이다.

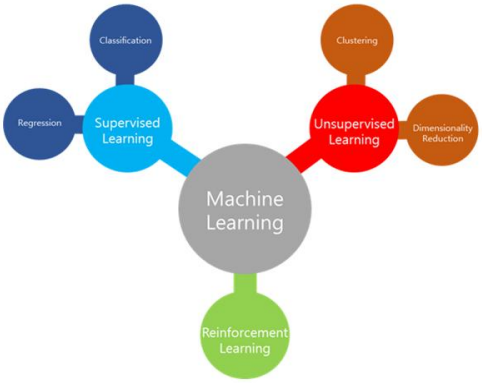

Fig. 3. Machine Learning Algorithm

1.3 LSTM AutoEncoder

LSTM AutoEncoder 신경망은 기존 시계열 데이터를 기반으로 이상 탐지를 수행하는 데 효과적이다. 기존 Machine Learning 시계열 탐지 알고리즘인 RNN(Recurrent Neural Network)의 장기의존성 문제를 해결하기 위해 나온 LSTM을 사용하였으며, 기존 AutoEncoder의 특성인 입력 데이터를 Encoder를 이용해 압축시킨 후, 데이터의 특징을 추출한 뒤 Decoder를 이용해 본래의 입력 형태로 복원시키는 신경망이다. Fig. 4.와 같이 구성되며, 모델이 학습되는 과정으로는 입력데이터와 복원된 데이터의 MSE(Mean Square Error)를 줄여나가며 학습이 수행되게 된다. 이상 탐지 모델을 생성하기 위해서는 위의 입력되는 데이터를 정상적인 시계열 데이터만을 이용해 학습을 수행하여야 한다. 또한, 이렇게 학습된 모델을 이용하여 시계열 데이터를 모델의 입력으로 입력 시 복원된 데이터가 특정 임계치를 초과하는 경우를 이상 상황으로 탐지하게 된다[6].

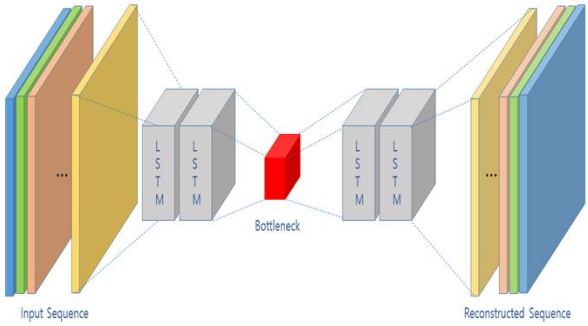

Fig. 4. LSTM AutoEncoder

1.4 ELK Stack

ELK Stack은 Elasticsearch, Logstash, Kibana, Beats 등으로 구성되어 있으며, 어떠한 소스, 형식의 데이터든 안정적이고 보안이 유지되는 방식으로 수집, 분석, 시각화하는 플랫폼이며 구성은 Fig. 5.와 같다[7].

1.4.1 Elasticsearch

Elasticsearch는 Apache Lucene을 기반으로 개발된 분산 검색엔진으로 방대한 양의 데이터를 신속하게 저장하고 처리할 수 있으며, 실시간 검색과 분석을 할 수 있는 검색엔진이다[8].

1.4.2 Logstash

Logstash는 200개 이상의 확장 가능한 플러그인 및 프레임워크가 포함되어 동적 데이터 수집 파이프라인을 구성할 수 있다. 해당 파이프라인을 이용하여 다양한 소스에서 동시에 데이터를 수집하고 변환하여 Elasticsearch로 보낼 수 있다[9].

1.4.3 Kibana

Kibana는 정형데이터 또는 비정형데이터를 시각화하고, 통합된 시각적 UI를 사용하여 Elastic Stack의 모든 기능을 구성 및 관리할 수 있는 도구이다[10].

1.4.4 Beats

Beats는 단말 장치의 데이터를 File, Metric, Traffic, Window Eventlog 등의 데이터를 Elasticsearch 및 Logstash로 전송하는 경량 데이터 수집기로 단말 장치와 시스템에 설치하여 다양한 유형의 데이터를 Logstash 또는 Elasticsearch로 전송하는 데이터 수집기 무료 오픈 소스 플랫폼이다[11].

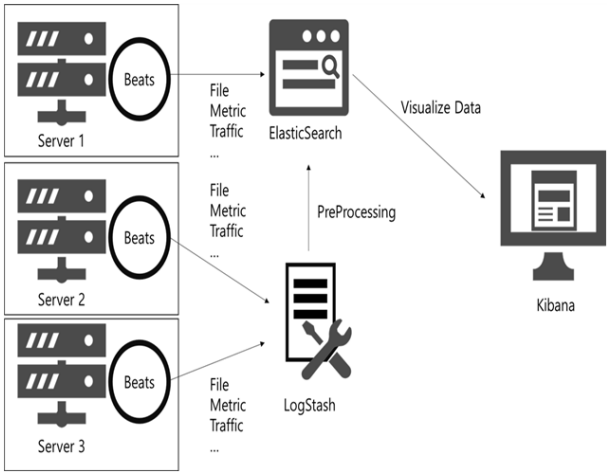

Fig. 5. ELK Stack

1.5 Scikit Learn

scikit-learn은 Machine Learning 및 데이터 분석 작업에 가장 많이 사용되는 Python 라이브러리 중 하나이다. 다양한 Machine Learning 모델을 제공하며 데이터 분석, 데이터 마이닝, 자연어 처리 등 다양한 분야에서 사용된다. 그리고 개발자들이 사용하기 쉽게 API를 제공하며, Machine Learning 및 데이터 분석에 필요한 다른 라이브러리와의 연동, 확장 및 사용이 편리하게 구성되어 있다. 이번 이상 탐지 모델 생성 시에는 수치데이터의 정규화 및 생성된 모델의 성능 측정 부분에서 사용하였다.

1.6 K-Means Clustering

데이터를 K개의 Cluster로 묶는 알고리즘이며 군집분석 방법의 하나다. 데이터를 비슷한 특징을 가진 군집으로 나누는데 나누는 방식은 각 데이터 포인트와 각 그룹의 중심점(Centroid)을 업데이트하는 방식으로 작동한다. 본 논문에서는 K-Means Clustering의 K값을 2개로 지정하여, 2개의 그룹으로 군집화를 수행하는 방식으로 기존의 정상 상황과 비정상 상황의 그룹을 통해 전투체계 무장 체계 연동 통제 이상 탐지 시험을 수행하였다.

III. The Proposed Scheme

본 연구는 다음과 같은 절차로 진행하였다.

1. Data collection, analysis, pre-processing
2. Model Learning and hyperparameter setting
3. Model Learning Result analyze, evaluation

1. Data collection, analysis, pre-processing

실제 함정에서 사용하는 전투체계의 실제 데이터 수집은 보안상 제한되며, 현재는 이러한 로그를 수집 및 분석하는 시스템이 실시간으로 구축되어 있지 않은 상황이어서 함정 전투체계에서 실제 무장 시스템이 가동될 수 있는 환경을 모의하여 로그데이터를 수집하였다. 실제 데이터를 수집하는 모습은 아래 Fig. 6.과 같다. 무장 통제 캐비닛과 정보처리장치, 다기능 콘솔에서 데이터를 수집하였으며, 다기능 콘솔의 경우 window event log를 수집할 수 있는 WinlogBeat, 리눅스에서 구동되는 장치들은 각각 전투체계 로그 및 Syslog를 수집하기 위한 FileBeat, 패킷 데이터를 수집할 수 있는 PacketBeat, 성능관련 데이터를 수집할 수 있는 MetricBeat를 이용하여 로그를 수집하였다.

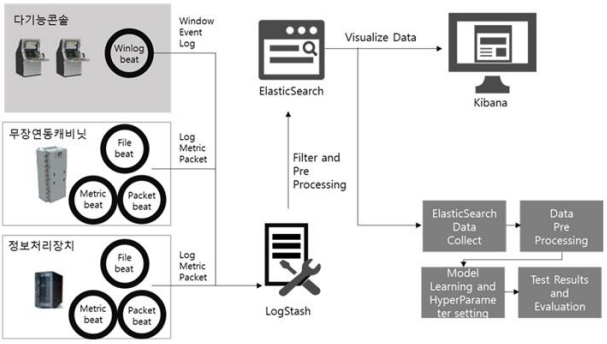

Fig. 6. Data collection flow

약 30일 정도 위의 수집기 역할을 수행하는 Beat들을 이용하여 데이터를 수집 및 LogStash를 거쳐 Elasticsearch에 저장하였다. 저장된 데이터는 분 단위 기준으로 아래 log, Metric, Traffic, Window log의 Count, Traffic, Cpu, Memory 사용량 등 Min, Max, Avg값 컬럼을 학습 데이터의 Feature값들로 총 데이터 51,940건을 추출하였다. 학습을 위한 Feature의 경우 Table 1.과 같이 정의하였다.

Table 1. Features of datasets

feature	Description
log_ count	Number of data collected through FileBeat - range : 0~1.0(Use MinMaxScaler)
metric_ count	Number of data collected through MetricBeat - range : 0~1.0(Use MinMaxScaler)
traffic_ count	Number of data collected through TrafficBeat - range : 0~1.0(Use MinMaxScaler)
winlog_ count	Number of data collected through WinlogBeat - range : 0~1.0(Use MinMaxScaler)
traffic_ byte_ min	Minimum traffic bytes among data collected through MetricBeat(system.network.in.bytes) - range : 0~1.0(Use MinMaxScaler)
traffic_ byte_ max	Maximum traffic bytes among data collected through MetricBeat(system.network.in.bytes) - range : 0~1.0(Use MinMaxScaler)
traffic_ byte_ avg	Traffic bytes average value among data collected through MetricBeat(system.network.in.bytes) - range : 0~1.0(Use MinMaxScaler)
cpu_ usage_ min	Minimum CPU usage rate among data collected through MetricBeat(system.cpu.user.pct) - range : 0~1.0(Use MinMaxScaler)
cpu_ usage_ max	Maximum CPU usage rate among data collected through MetricBeat(system.cpu.user.pct) - range : 0~1.0(Use MinMaxScaler)
cpu_ usage_ avg	Average CPU usage rate among data collected through MetricBeat(system.cpu.user.pct) - range : 0~1.0(Use MinMaxScaler)
memory _usage _min	Minimum memory usage among data collected through MetricBeat(system.process.memory.rss.pct) - range : 0~1.0(Use MinMaxScaler)
memory _usage _max	Maximum memory usage among data collected through MetricBeat(system.process.memory.rss.pct) - range : 0~1.0(Use MinMaxScaler)
memory _usage _avg	Average memory usage among data collected through MetricBeat(system.process.memory.rss.pct) - range : 0~1.0(Use MinMaxScaler)
error_ yn	Not error or error -Range : 0(=noerror), 1(=error)

해당 Features들을 이용한 Dataset의 샘플 데이터는 아래 Fig. 7.과 같다.

log_count	metric_count	traffic_count	winlog_count	cpu_usage_min	cpu_usage_avg	cpu_usage_max	traffic_byte_max
3538	196	595	1	0.031310	0.041637	0.055348	1.446615e+09
3552	196	595	1	0.029492	0.042992	0.078073	1.450501e+09
3554	197	595	1	0.029492	0.041343	0.059994	1.454361e+09
3579	197	595	1	0.029492	0.042664	0.049490	1.458242e+09
3493	197	594	1	0.029189	0.041132	0.062519	1.462117e+09

traffic_byte_avg	traffic_byte_min	memory_usage_min	memory_usage_avg	memory_usage_max	error_yn
5.587232e+08	0.0	0.003838	0.021310	0.039491	0
5.602191e+08	0.0	0.003838	0.021214	0.039390	0
5.617237e+08	0.0	0.003838	0.020969	0.039390	0
5.632239e+08	0.0	0.000202	0.020995	0.039390	0
5.647213e+08	0.0	0.003838	0.021055	0.039087	0

Fig. 7. Sample Data

'log_count', 'metric_count', 'traffic_count', 'winlog_count'의 경우는 각 전투체계가 구성되는 시스템에서 각각 수집을 통해 확보된 데이터로서 로그의 건수를 나타내는 값이며, Elasticsearch의 날짜별로 구성되어 인덱스를 분별 기준으로 구성되도록 전처리를 수행하였다. 추가로, 트래픽 데이터는 'traffic_byte'로 시작하며, cpu사용량은 'cpu_usage', 메모리 사용량은 'memory_usage' 명칭을 사용하되 각 값의 max, min, avg와 같이 통계치를 사용하였다. 해당 데이터는 전투체계가 실행되는 시스템별 성능 정보이며, 기존 count 값과 같은 기준으로 분별 합을 통해 데이터를 구성하였다.

구성된 데이터를 Features 별 전처리를 위해선 Scikit-Learn에서 제공하는 Scaler를 사용하였다. 값들이 대부분 수치였기 때문에, 각 Feature 별 수치의 분포를 재정의하여 값을 치환하였다. Scaler의 경우는 MinMaxScaler 을 사용하여 수집된 Feature 별 수치데이터를 최댓값은 1, 최솟값은 0으로 데이터의 범위를 조정하는 정규화 수행을 통해 전처리를 수행하였다. 다음으로, 구성된 학습 데이터를 기반으로 모델 생성에 필요한 데이터를 훈련(training)/검증(validation)/시험(test) 데이터셋으로 구성하였다. 학습에 사용한 데이터는 훈련용 데이터 33,173건, 검증용 데이터 8,294건 활용하였으며, 시험 데이터 셋은 정상/이상 케이스가 혼재된 10,461건을 활용하였으며, 여기서 이상 데이터는 전투체계 무장 체계 연동 통제에 이상이 있는 시점의 데이터를 이상으로 라벨링 하였으며, 상세한 이상 시점으로는 네트워크 오류가 발생하여 연동에 문제가 생긴 시점, 임의로 전투체계 내의 최대부하를 초과

하는 데이터를 이용하여 부하를 주었을 때 연동 오류가 발생하는 시점, 전투체계 무장 체계 연동 관련 응용이 종료되어 관련 트래픽 또는 로그가 발생하지 않는 시점을 이상으로 라벨링 하였으며, 이외에 전투체계 무장 체계 연동 통제가 정상적인 상황을 정상 데이터로 구분하였다. 또한, 전투체계 무장 체계 연동 통제가 수행되는 상황에서 발생하는 시계열 상에서의 데이터들을 통해 과거 데이터를 고려하여 미래 데이터를 예측하기 위한 LSTM과 입력데이터를 압축하여 특징 추출 및 복원을 거치며 입력값과 출력값의 오차를 줄여나가는 AutoEncoder 신경망의 특성을 결합한 LSTM AutoEncoder 신경망을 이용한다면, 정상 상황인 시계열 데이터의 특징을 통해 복원되는 데이터의 오차를 이용하여 해당 환경의 이상 탐지에 적합할 것으로 판단하고 수집된 데이터를 해당 모델을 생성하기 위해 전처리를 수행하였다. 훈련데이터 생성 시에는 정상 데이터만을 학습시킨 후, 이상 데이터를 입력으로 출력을 복원시키는 경우 복원이 잘되지 않고 임계치를 벗어나는 데이터가 발생하는 것을 확인하기 위한 신경망을 구성하여야 하므로 학습과 검증용 데이터 셋은 정상 데이터만을 이용하였다. K-Means Clustering도 비지도 학습으로 데이터의 정답 데이터 없이 K의 개수에 따라 그룹으로 나누기에 모든 학습 데이터를 이용하여 모델을 생성하였다. 두 모델 생성 시 사용한 각 Feature 별 데이터의 분포는 Table2과 같다. 각 Feature 별 평균, 표준편차, 최소, 4분위수(25,50,75%), 최대값 분포이다.

Table 2. Features Data Statistics

Statistics	log_count	metric_ count	traffic_ count	winlog_ count
mean	3827.322	411.2592	5417.743	6.504467
std	546.3588	194.7839	4896.027	3.459508
min	1	189	594	1
25%	3553	203	605	3
50%	4086	588	701	7
75%	4156.25	597	10478	10
max	4948	626	10679	15
Statistics	cpu_usage_ min	cpu_usage_ avg	cpu_usage_ max	
mean	0.036441	0.197051	0.300838	
std	0.007007	0.179992	0.279222	
min	0	0.020958	0.031613	
25%	0.031415	0.044911	0.06307	
50%	0.035964	0.051146	0.075649	
75%	0.041288	0.409562	0.624739	
max	0.124786	0.458458	0.72856	
Statistics	traffic_byte_ max	traffic_byte_ avg	traffic_byte_ min	
mean	8225652725	4424865795	0	
std	5459028282	3312211820	0	
min	2244194.75	722106.9067	0	
25%	3823629942	1513453707	0	
50%	7003346352	3189192982	0	
75%	12024576051	6983470969	0	
max	21835578875	12147099569	0	
Statistics	memory_ usage_min	memory_ usage_avg	memory_ usage_max	
mean	0.001615	0.016934	0.034338	
std	0.001836	0.008283	0.01641	
min	0	0	0	
25%	0	0.015851	0.036288	
50%	0.000214	0.021104	0.041128	
75%	0.003876	0.022251	0.043179	
max	0.004256	0.025809	0.052976	

2. Model Learning and HyperParameter setting

실험에 사용된 신경망을 구성하기 위한 환경으로는 python 3.12.5, tensorflow 2.17.0로 지정하여 시험을 수행하였다. LSTM AutoEncoder의 경우 LSTM Encoder와 LSTM Decoder로 구성되어 있으며, 은닉층은 각 64개, 32개의 쌍으로 대칭되는 구조로 구성되어 있다. LSTM Encoder에서 압축된 것을 Decoder로 재구성하여 나타낸 결과의 오차를 줄여나가면서 학습이 구성되도록 RepeatVector Layer(입력을 TimeSquence 값인 5 만큼 반복함)를 구성하였다. 실제 학습을 진행하기 위해선, LSTM sequence를 5로 지정하였으며, 따라서 각 은닉층 및 RepeatVector의 shape는 Fig. 8.의 LSTM AutoEncoder 신경망 구성과 같은 형태로 지정하였다. 학습을 위한 하이퍼 파라미터(모델 생성 시 설정값)의 경우, 활성화 함수(모델 생성 시 입력신호를 출력 신호로 변환하는 함수)로는 relu를 사용하였으며 Learning rate의 경우는 0.001, epochs는 200회, batch_size의 경우 128로 지정하여 학습을 수행하였다. 학습을 위해 필요한 하이퍼 파라미터 지정의 경우 epochs, batch_size, learning rate, 각 은닉층별 개수, 활성화 함수에 변화를 주어 설정해보며 학습이 잘되는 train data 및 valid data 검증 시 loss가 잘 감소하는 모델을 찾을 수 있었으며, History를 기록하여 모델의 학습이 진행되는 내용을 저장하였다.

Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 5, 64)	19,968
lstm_1 (LSTM)	(None, 32)	12,416
repeat_vector (RepeatVector)	(None, 5, 32)	0
lstm_2 (LSTM)	(None, 5, 32)	8,320
lstm_3 (LSTM)	(None, 5, 64)	24,832
time_distributed (TimeDistributed)	(None, 5, 13)	845

Total params: 66,381 (259.30 KB)
Trainable params: 66,381 (259.30 KB)
Non-trainable params: 0 (0.00 B)

Fig. 8. LSTM AutoEncoder structure

K-Means Clustering 알고리즘을 이용하여 이상 탐지를 수행하기 위해서는 군집화하기 위한 그룹의 개수 K를 2로 지정하였다. 모델 학습을 수행하기 위한 데이터의 구조는 13개의 feature로 구성되었으며, LSTM sequence 값을 토대로 전처리하는 부분을 제외하여 학습 데이터를 구성하였다. 모델의 학습 시 사용하는 Parameter의 경우 다른 cluster 중심으로 알고리즘을 수행하는 횟수 값인 n_init는 'auto', 알고리즘의 최대 반복 횟수 max_iter는 '300' 이외 나머지 Parameter는 기본값으로 지정하여 모델을 생성하였다.

IV. Test Results and Evaluation

1. Model Learning Result

1.1 LSTM AutoEncoder Anomaly detection

함정전투체계 무장 체계 연동 통제에서 수집된 정상 데이터를 LSTM Sequence 5로 전처리한 훈련용 데이터 33,173건, 검증용 데이터 8,294건 활용하여 LSTM AutoEncoder 신경망의 학습을 수행하였다. 학습 Epoch 별 손실 그래프는 Fig. 9.의 그래프와 같다. 손실 그래프의 Y축은 손실로서 입력데이터와 재구축 데이터 간의 MSE(Mean Square Error)이며 파란색 선과 주황색 선은 각각 훈련 및 검증의 손실 값을 나타낸다, 학습 시작 이후 감소하기 시작하여, epoch이 25회 정도 되었을 때 재구축 오차가 낮은 수치로 수렴하고 있다.

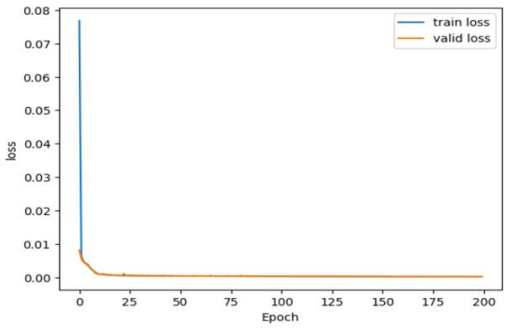

Fig. 9. LSTM AutoEncoder learning results

Fig. 10.은 훈련 수행 중에 입력된 값의 결과로 출력되는 재구성 값과 입력된 값 비교로 발생하는 재구성 손실 분포가 나타나 있다. 해당 그래프를 기준으로 정상과 비정상 데이터 간의 분류를 수행할 수 있는 임계치 설정에 도움을 준다. 즉, 재구축 오류가 점점 줄어들게 되는 마지막 값을 임계치로 설정을 수행하며 정상 데이터와 비정상 데이터를 통하여 생성된 이상 탐지 모델을 시험할 수 있다.

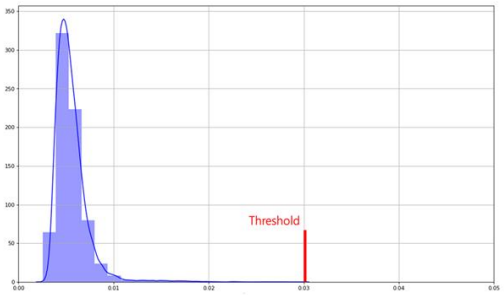

Fig. 10. Distribution of reconstruction loss errors during training

Fig. 11.과 Fig. 12.는 정상/비정상 케이스가 혼재된 시험 데이터셋 10,461개 기반으로 Reconstruction error 분포도(입력값 기준으로 출력과 입력 간의 오차)를 나타냈다. 기존에 정해둔 임계치인 0.03와 0.05 기준으로 비정상 케이스와 정상 케이스간의 구분에서의 모델 평가는 2.Model evaluation에서 수행하였다.

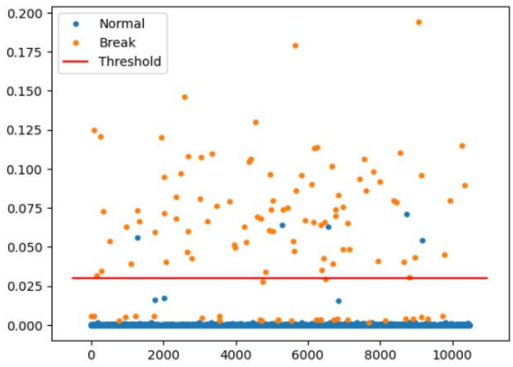

Fig. 11. Test dataset reconstruction error distribution (threshold 0.03)

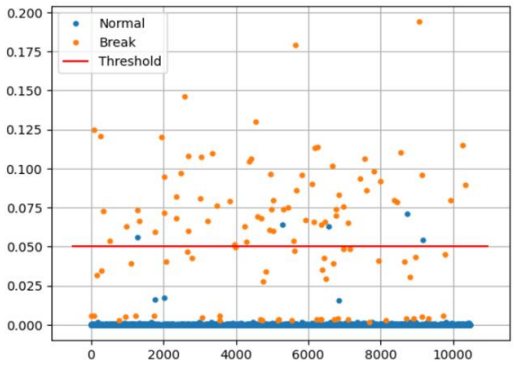

Fig. 12. Test dataset reconstruction error distribution (threshold 0.05)

1.2 K-Means Clustering Anomaly detection

K-Means Clustering 모델 생성 및 이상 탐지 결과는 Fig. 13.과 같다. 주황색 네모 모양이 비정상 결과로 예측되었으며, 파란색 동그라미 모양이 정상 결과로 예측되었다. 2차원 그래프로 Clustering 결과를 나타내고자 차원 축소를 수행하여 분포도를 나타내었으며 x 좌표와 y 좌표는 PCA(Principle Component Analysis) 기법을 이용하여 13개의 Feature를 2개로 변경하여 2차원 그래프로 나타낼 수 있도록 차원을 축소하였다. 그래프로 표시된 결과로 어느 정도 구분은 되나 높은 확률로 정상과 이상을 구분하지 못하는 결과를 확인하였다. 따라서, 그래프를 통한 이상 탐지 확인에서도 LSTM AutoEncoder 이상탐지 모델이 K-Means Clustering 이상 탐지 모델보다 성능이 높다는 것을 추측할 수 있다.

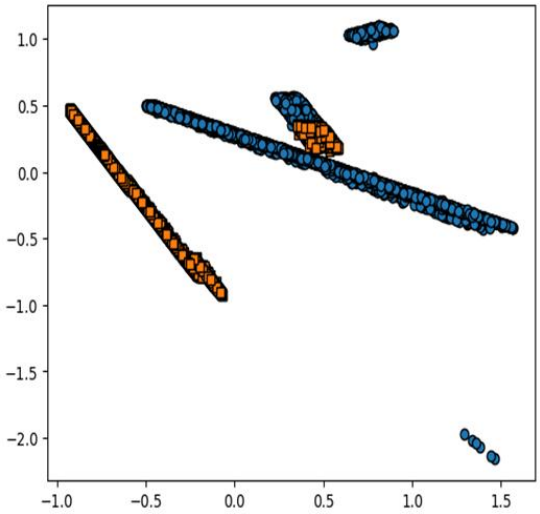

Fig. 13. K-Means Clustering Results

2. Model evaluation

생성된 LSTM AutoEncoder 이상 탐지 모델 및 K-Means Clustering 이상 탐지 모델의 성능 측정은 아래와 같이 confusion matrix와 정확도(Accuracy), 정밀도(Precision), 재현율(Recall) 및 F1-Score 지표를 활용하여 수행하였다. confusion matrix는 훈련을 통해 생성된 모델을 통해 예측을 수행하였을 경우 예측된 값과 실제 값을 비교하여 성능을 나타내기 위한 Fig. 14.과 같은 형태이다. 시험 데이터 기준으로 TP(True Positive)는 실제 True인 값을 True라고 예측(정답), TN(True Negative)은 실제 False인 값을 False라고 예측(정답), FP(False Positive)는 실제 False인 값을 True라고 예측(오답), FN(False Negative)은 실제 True 값을 False라고 예측(오답)한 데이터의 개수이다.

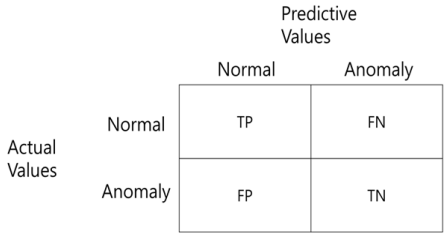

Fig. 14. confusion matrix

생성된 모델의 성능을 확인하기 위한 성능지표들은 (1)~(4)와 같이 식으로 정리할 수 있다. confusion matrix에 있는 TP, TN, FP, FN 값들을 토대로 연산을 수행할 수 있다. Accuracy는 전체 데이터 중에 Normal 또는 Anomaly 값을 정상적으로 분류한 수치이다. Precision은 Normal이라고 예측한 값 중 실제 Normal인 값의 비율이다. Recall은 실제 Normal 값 중 Normal이라고 예측한 비율이다. F1-Score는 Precision과 Recall의 조화 평균 값이다.

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

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

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

$$\text{F1-Score} = 2 \times \frac{\text{Precisioin} \times \text{Recall}}{\text{Precisioin} + \text{Recall}} \quad (4)$$

2.1 LSTM AutoEncoder confusion matrix

그림 Fig.15.과 Fig.16.는 각각 임계치 0.03과 0.05의 경우 생성되는 confusion matrix 값이다. 실제와 예측 시의 Anmoaly 값을 탐지하는 데에 각 임계치 별로 차이가 있으며, 임계치가 0.03의 경우가 0.05의 경우보다 이상 탐지에서 더 좋은 성능을 보였다.

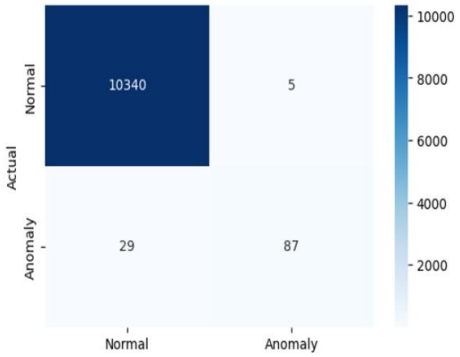

Fig. 15. LSTM AutoEncoderE confusion matrix (threshold 0.03)

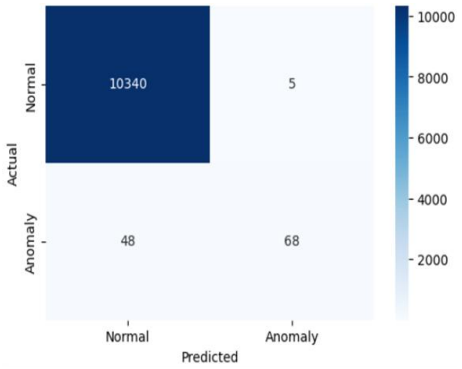

Fig. 16. LSTM AutoEncoder confusion matrix (threshold 0.05)

2.2 K-Means Clustering confusion matrix

그림 Fig.17.는 K-Means Clustering 이상 탐지 모델을 사용하여 시험 데이터 셋을 예측한 결과와 기존 결과를 이용한 confusion matrix이다. 실제 정상값을 Anomaly라고 예측한 오답이 LSTM AutoEncoder 모델과 비교하면 상당히 많은 부분을 차지하고 있다.

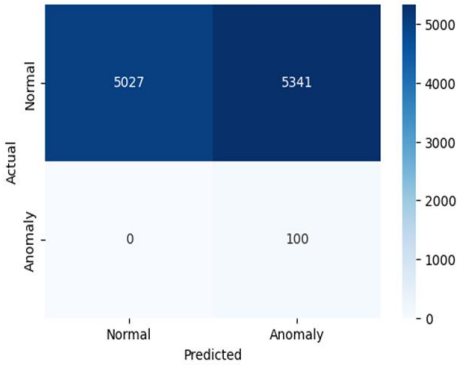

Fig. 17. K-Means Clustering confusion matrix

2.3 Model performance evaluation results

식(1)~(4)를 이용하여 LSTM AutoEncoder 이상 탐지 모델의 임계치 0.02부터 0.06을 기준으로 성능지표를 구한 내용은 Table3과 같다. 또한, K-Means Clustering 이상 탐지 모델의 K값을 2로 설정하였을 때의 성능지표는 Table4와 같다. 전투체계 무장 체계 연동 통제에 적용하고자 시험을 진행하였던 LSTM AutoEncoder 이상 탐지 모델과 K-Means Clustering 이상 탐지 모델의 성능 비교 시 LSTM AutoEncoder 이상 탐지 모델이 성능이 더 뛰어나다는 것을 확인할 수 있었다. 각 성능 지표상 Recall 데이터는 다른 데이터에 비해 낮게 나왔지만, 실제 전투체계 무장 체계 연동 통제 관련 이상 탐지 시 해당 모델을 사용한다면 이상 탐지를 수행 및 식별하는 데 효과적임을 볼 수 있다.

Table 3. Performance LSTM AutoEncoder Model

Threshold	Accuracy	Precision	Recall	F1-Score
0.02	0.996	0.972	0.883	0.923
0.03	0.996	0.971	0.874	0.917
0.04	0.996	0.968	0.844	0.897
0.05	0.994	0.963	0.792	0.858
0.06	0.994	0.974	0.767	0.841

Table 4. Performance K-Means Clustering Model

K	Accuracy	Precision	Recall	F1-Score
2	0.489	0.509	0.742	0.344

IV. Conclusions

본 논문에서 AI를 이용한 함정 전투체계의 무장 체계 연동 통제를 위한 소프트웨어와 하드웨어가 구동되는 환경의 성능정보와 구동되는 과정에서 발생하는 각종 로그 및 Metric 정보들을 실시간으로 수집 및 종합하여 AI 기반 이상 탐지 모델을 생성하여 성능 측정 및 이상 탐지를 통해 전투체계의 연속성에 대해 확보하는 부분에 관해 연구하였다. 생성된 모델의 성능 평가를 통해 전투체계 무장 체계 연동 통제 시스템 성능 분석 및 이상 상황을 탐지하기 위한 실시간 감시방안에 대해 AI 기술을 이용 시에 약 Accuracy 0.99, Precision 0.97, Recall 0.87, F1-Score 0.92의 높은 점수로 시스템의 이상 탐지를 수행할 수 있다는 부분에 대한 적용 가능성을 확인하였다.

추후, 전투체계 전체 노드의 실시간 로그 수집을 통해 LSTM AutoEncoder 신경망을 이용한 이상 탐지 수행하는 시스템을 설계하여 전투체계의 연속성을 확보하는 방안에 관해 연구할 필요가 있다. 또한, 현재 수행한 시험의 경우 전투체계가 구동되는 시스템의 환경적인 요소에 대한 데이터만을 수집하였지만, 전투체계 운용 시 더욱 다양한 데이터들의 수집 파이프라인을 통해 데이터 수집체계를 만들고 이렇게 수집한 데이터를 기반으로 다양한 모델 생성 및 실시간 서비스화를 통하여 전투체계 성능 및 기능의 고도화 방안에 관해 연구할 필요가 있다.

REFERENCES

[1] Department of Defense Editorial Board, “Announcement of Defence Innovation 4.0 Framework Plan: : Redesigning Defence at the Second Army Level, Fostering AI Science and Technology Force”. Defense & Technology, Vol. 530, pp.10-11, April. 2023.

[2] Department of Defense Editorial Board, “Report on the implementation plan for the detailed implementation of the National Defense Policy Direction : Establishment of a three-axis system, strengthening of ROK-US joint exercises, and creation of a defense AI center”. Defense & Technology, Vol. 522, pp.11-13, Aug. 2022.

[3] Joon-Ho Lee, Ki-Hyun Jung, Kee-Young Yo, “Hybrid Information Hiding Method Based on the Characteristics of Military Images on Naval Combat System” Journal of Korea Multimedia Society Vol. 19, No. 9, pp.1669-1678 September 2016. DOI : 10.9717/kmms.2016.19.9.1669

[4] J. G. Lee, “A Study on the Standard Architecture of Weapon Control Software on Naval Combat System,” Journal of The Korea Society of Computer and Information Vol. 26 No. 11, pp. 101-110, November 2021. DOI : 10.9708/jksci.2021.26.11.101

[5] Sang-Min Kwon, Seung-Mo Jung, "Virtualization based high efficiency naval combat management system design and performance analysis," Journal of the Korea Society of Computer and Information Vol. 23, No. 11, pp. 9-15, 2018. DOI : 10.9708/jksci.2018.23.11.009

[6] Jeong-Ho Lee, Dong-hyeok Im, Tae-Hyun Kim, Man-Jung Kim, Seong-Jin Park, Oh-Seok Yang, Jeong-Hyun Baek, "Implementation of Smart Farm Data Anomaly Detection Using LSTM Autoencoder" Journal of Knowledge Information Technology and Systems(JKITS), Vol. 18, No. 3, pp. 587~596, June 2023. DOI : 10.34163/jkits.2023.18.3.008

[7] Elastic Stack, *https://www.elastic.co/kr/elastic-stack*

[8] Elastic Search, *https://www.elastic.co/kr/elasticsearch*

[9] Logstash, https://*www.elastic.co/kr/logstash*

[10] Kibana, *https://www.elastic.co/kr/kibana*

[11] Beats, *https://www.elastic.co/kr/beats*

Authors

Hyun-Ho Na received the B.S degrees in Computer Science and Engineering from Chungnam National University, Korea, in 2015. he is currently working in Hanwha Systems Co. from 2021.

he is interested in Weapon Control Software of the Naval Combat Management System. Artificial Intelligence and so on