



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0105977
(43) 공개일자 2024년07월08일

- (51) 국제특허분류(Int. Cl.)
G08G 3/00 (2006.01) G06F 18/15 (2023.01)
G06F 18/214 (2023.01) G06N 20/00 (2019.01)
G06N 3/0442 (2023.01)
- (52) CPC특허분류
G08G 3/00 (2013.01)
G06F 18/15 (2023.01)
- (21) 출원번호 10-2022-0188504
(22) 출원일자 2022년12월29일
심사청구일자 2022년12월29일
- (71) 출원인
(주)세이프텍리서치
대전광역시 유성구 문지로 271-6(문지동)
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
- (72) 발명자
김경섭
서울특별시 강남구 광평로9길 34, 108동 1402호
공인영
대전광역시 유성구 배울2로 134 대덕테크노밸리푸
르지오하임1단지 101동 101호
(뒷면에 계속)
- (74) 대리인
박길환, 임종승, 김갑수

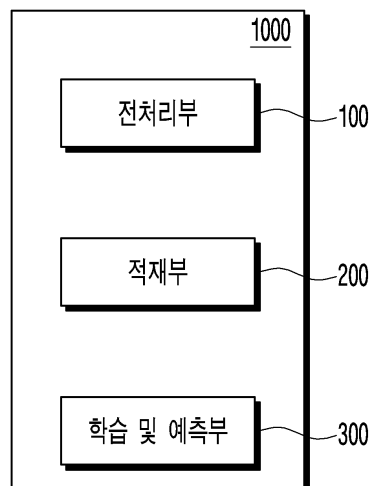
전체 청구항 수 : 총 5 항

(54) 발명의 명칭 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템 및 그 방법

(57) 요약

본 발명은 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템 및 그 방법에 관한 것으로, 해상교통 데이터를 위도, 경도 및 시간의 축을 가진 다차원의 텐서 형태로 적재하고, 적재된 데이터를 이용하여 머신러닝 모델을 학습하고, 학습한 머신러닝 모델을 통해 해상교통 혼잡도를 예측하는 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템 및 그 방법에 관한 것이다. 본 발명에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템은, 해상교통 데이터의 전처리(preprocessing)를 수행하는 전처리부, 상기 전처리부에 의해 처리된 데이터를 기 설정된 시간에 대한 텐서 형태로 적재하는 적재부 및 상기 텐서 형태로 적재된 데이터에 기초하여 머신러닝 모델을 학습하고, 상기 머신러닝 모델을 통해 해상교통 혼잡도를 예측하는 학습 및 예측부를 포함한다.

대표도 - 도1



(52) CPC특허분류

G06F 18/214 (2023.01)

G06N 20/00 (2021.08)

G06N 3/0442 (2023.01)

(72) 발명자

이은규

대전광역시 유성구 노은서로266번길 61, 619호

칸 주나이드

대전 유성구 대학로145번길 55, 206호

김현석

대전광역시 유성구 문지로 300 효성해링턴플레이스
112동 1404호

한재석

대전광역시 유성구 반석로 80, 101동 1102호

이 발명을 지원한 국가연구개발사업

과제고유번호 1345356233

과제번호 1345356233

부처명 교육부

과제관리(전문)기관명 한국연구재단

연구사업명 지자체-대학 협력기반 지역혁신 사업

연구과제명 (대전세종충남지역혁신플랫폼)충남대학교

기 여 율 1/1

과제수행기관명 (대전세종충남지역혁신플랫폼)충남대학교

연구기간 2022.05.01 ~ 2023.03.31

명세서

청구범위

청구항 1

해상교통 데이터의 전처리(preprocessing)를 수행하는 전처리부;

상기 전처리부에 의해 처리된 데이터를 기 설정된 시간에 대한 텐서 형태로 적재하는 적재부; 및

상기 텐서 형태로 적재된 데이터에 기초하여 머신러닝 모델을 학습하고, 상기 머신러닝 모델을 통해 해상교통 혼잡도를 예측하는 학습 및 예측부;

를 포함하는, 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템.

청구항 2

청구항 1에 있어서,

상기 적재부는,

상기 전처리부에 의해 처리된 데이터를 복수 개의 격자에 표현하는 데이터 격자화를 수행하며,

상기 데이터 격자화는,

국립해양조사원의 격자형 해양정보관리체계에 기초하여 6단계의 격자화를 수행하는 것을 특징으로 하는, 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템.

청구항 3

청구항 2에 있어서,

상기 적재부는,

상기 복수 개의 격자에 표현된 데이터를 위도, 경도 및 시간의 축을 기준으로 선박들의 침로(CoG, Course of Ground) 정보, 속도(SoG, Speed of Ground) 정보 및 길이 정보를 텐서 형태로 저장하여 해상교통 데이터를 텐서 형태로 적재하며,

상기 위도의 축 및 상기 경도의 축의 기본 단위는 상기 데이터를 격자화한 크기와 동일하고,

상기 시간의 축은 기 설정된 시간동안 1초 단위로 이루어지는 것을 특징으로 하는, 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템.

청구항 4

청구항 3에 있어서,

상기 적재부는,

상기 기 설정된 시간동안 온전한 데이터를 적재하지 못하는 경우, Kinematic 외삽법 또는 Extended Kalman filter 법을 통해 적재되지 못한 데이터를 예측하는 것을 특징으로 하는, 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템.

청구항 5

해상교통 데이터를 전처리하는, 전처리 단계;

상기 전처리된 해상교통 데이터를 기 설정된 시간에 대한 텐서 형태로 적재하는, 적재 단계; 및

상기 텐서 형태로 적재된 데이터에 기초하여 머신러닝 모델을 학습하고, 상기 머신러닝 모델을 통해 해상교통 혼잡도를 예측하는, 모델 학습 및 혼잡도 예측 단계;

를 포함하는, 다차원 텐서를 이용한 해상교통 혼잡도 예측 방법.

발명의 설명

기술 분야

- [0001] 본 발명은 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템 및 그 방법에 관한 것으로, 더욱 상세하게는 해상교통 데이터를 위도, 경도 및 시간의 축을 가진 다차원의 텐서 형태로 적재하고, 적재된 데이터를 이용하여 머신러닝 모델을 학습하고, 학습한 머신러닝 모델을 통해 해상교통 혼잡도를 예측하는 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템 및 그 방법에 관한 것이다.

배경 기술

- [0002] 해상교통 혼잡도의 예측은 과거 및 현재 해상교통 혼잡도를 파악하는 것에 더불어 선박운항자의 입장에서는 안전한 항해를 담보하기 위한 필수 기술이며, 해역을 관리하는 기관의 입장에서는 해상교통 안전성을 확보하기 위한 기초자료로 사용될 수 있다.
- [0003] 해상교통 혼잡도를 예측하기 위해서는 선박들의 통항에 영향을 주는 선박운항자의 특성, 선박의 조종성능, 자연환경의 특성, 교통환경의 특성, 지원체계의 특성 등 다양한 요소를 고려하여야 하지만 신뢰할 수준의 충분한 자료를 수집하는 것이 어려우며, 종속변수와 독립변수들 간의 의미 있는 관계가 없음에도 불구하고 정확한 예측을 하는 것처럼 보이는 허구성이 나타날 수 있기 때문에 주의하여야 한다.
- [0004] 해상교통 혼잡도의 예측은 과거 해상교통량의 추세를 이용한 장래 해상교통량 추정 방식을 이용할 수 있으며, 물동량 증가율에 따른 해상교통량 추정 방식을 이용할 수도 있다. 이외에도 항만개발계획에 따른 각 선석별 특성을 고려한 해상교통량 추정방식을 이용할 수도 있으며, 최근에는 시계열 데이터를 처리하는 다양한 인공지능 모델을 활용된 방법들도 이용되고 있다.
- [0005] 이러한 예측 모델은 신뢰도 확보가 가장 중요한 부분이라고 할 수 있으며, 예측 모델의 신뢰도는 모델 자체의 설계에 영향을 받기도 하지만 보다 근본적인 것은 입력 변수의 설정 및 구성이라고 할 수 있다.
- [0006] 기존의 해상교통 혼잡도 예측 방법들은 해상교통 데이터에서 해상교통 혼잡도를 계산하고 이를 포함하여 입력 변수를 구성하였다. 하지만 이용 가능한 신뢰도 높은 자료가 해상교통 데이터 밖에 없는 상황에서 이렇게 해상교통 혼잡도를 먼저 계산해버리게 되면, 차원 축소로 인해 발생하는 단점과 같이 해상교통 혼잡도를 예측하기 위한 정보의 소실이 발생할 수 있다.
- [0007] 따라서 본 발명에서는 해상교통 데이터를 다차원 텐서 형태로 정보 소실을 최소화하는 입력 변수를 구성하여 해상교통 혼잡도를 예측하고자 하였다.

선행기술문헌

특허문헌

- [0008] (특허문헌 0001) 대한민국 등록특허공보 제10-2429235호 (2022년08월04일 공고)

발명의 내용

해결하려는 과제

- [0009] 상술한 문제점을 해결하기 위한 본 발명의 목적은 해상교통 혼잡도를 예측함에 있어 정보의 손실 없이 신뢰도 높은 해상교통 혼잡도를 예측할 수 있으며, 데이터가 일부 확보되지 않은 경우에도 데이터를 예측하여 해상교통 혼잡도를 예측할 수 있는 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템 및 그 방법을 제공하는 것이다.

과제의 해결 수단

- [0010] 상술한 목적을 달성하기 위하여, 본 발명의 일 실시예에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템은, 해상교통 데이터의 전처리(preprocessing)를 수행하는 전처리부, 상기 전처리부에 의해 처리된 데이터를 기 설정된 시간에 대한 텐서 형태로 적재하는 적재부 및 상기 텐서 형태로 적재된 데이터에 기초하여 머신러닝

모델을 학습하고, 상기 머신러닝 모델을 통해 해상교통 혼잡도를 예측하는 학습 및 예측부를 포함한다.

- [0011] 상기 적재부는, 상기 전처리부에 의해 처리된 데이터를 복수 개의 격자에 표현하는 데이터 격자화를 수행하며, 상기 데이터 격자화는, 국립해양조사원의 격자형 해양정보관리체계에 기초하여 6단계의 격자화를 수행한다.
- [0012] 상기 적재부는, 상기 복수 개의 격자에 표현된 데이터를 위도, 경도 및 시간의 축을 기준으로 선박들의 침로 (CoG, Course of Ground) 정보, 속도(SoG, Speed of Ground) 정보 및 길이 정보를 텐서 형태로 저장하여 해상교통 데이터를 텐서 형태로 적재하며, 상기 위도의 축 및 상기 경도의 축의 기본 단위는 상기 데이터를 격자화한 크기와 동일하고, 상기 시간의 축은 기 설정된 시간동안 1초 단위로 이루어진다.
- [0013] 상기 적재부는, 상기 기 설정된 시간동안 온전한 데이터를 적재하지 못하는 경우, Kinematic 외삽법 또는 Extended Kalman filter 법을 통해 적재되지 못한 데이터를 예측한다.
- [0014] 상술한 목적을 달성하기 위하여, 본 발명의 일 실시예에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 방법은, 해상교통 데이터를 전처리하는, 전처리 단계, 상기 전처리된 해상교통 데이터를 기 설정된 시간에 대한 텐서 형태로 적재하는, 적재 단계 및 상기 텐서 형태로 적재된 데이터에 기초하여 머신러닝 모델을 학습하고, 상기 머신러닝 모델을 통해 해상교통 혼잡도를 예측하는, 모델 학습 및 혼잡도 예측 단계를 포함한다.

발명의 효과

- [0015] 본 발명인 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템 및 그 방법에 따르면, 해상교통 혼잡도를 예측함에 있어 해상교통 데이터에서 해상교통 혼잡도를 예측하기 위한 정보의 손실 없이 신뢰도 높은 해상교통 혼잡도를 예측할 수 있으며, 데이터가 일부 확보되지 않은 경우에도 데이터를 예측하여 해상교통 혼잡도를 예측할 수 있는 효과가 있다.
- [0016] 본 발명인 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템 및 그 방법을 통해 예측된 해상교통 혼잡도는 선박운항자의 입장에서는 해상교통 상황을 예측하여 안전운항을 하는데 도움을 줄 수 있고, 해역을 관리하는 기관의 입장에서는 해상교통 안전성을 확보하기 위한 기초자료로 사용될 수 있다. 뿐만 아니라 최근 개발되고 있는 자율운항 선박에서 안전운항을 위해 해상교통 상황을 인식하고 예측하는 상황인식 알고리즘의 필수적인 기능으로 사용될 수 있는 효과가 있다.

도면의 간단한 설명

- [0017] 도 1은 본 발명에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템의 주요 구성을 보여주는 블록도이다.
- 도 2는 본 발명에 따른 전처리부의 데이터 보간 작업에 따른 가시화를 예시적으로 보여주는 도면이다.
- 도 3은 국립해양조사원의 격자형 해양정보관리체계를 보여주는 도면이다.
- 도 4는 본 발명에 따른 적재부의 데이터 격자화의 예시를 보여주는 도면이다.
- 도 5는 본 발명에 따른 적재부의 데이터 적재 개념도이다.
- 도 6은 해상교통 데이터를 위경도 좌표계를 기준으로 1초마다 도시한 예시를 보여주는 도면이다.
- 도 7은 도 6의 위경도 좌표계에 도시된 데이터를 누적 도시한 예시를 보여주는 도면이다.
- 도 8은 도 6의 위경도 좌표계에 도시된 데이터를 누적하지 않고 텐서 형태로 적재한 예시를 보여주는 도면이다.
- 도 9는 본 발명에 따른 적재부의 데이터 예측의 개념도를 도시한 도면이다.
- 도 10 및 도 11은 본 발명에 따른 적재부의 데이터 예측 시 예시를 각각 보여주는 도면이다.
- 도 12는 본 발명에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 방법의 각 단계를 대략적으로 도시한 도면이다.
- 도 13은 본 발명에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 방법의 각 단계를 상세하게 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0018] 이하, 본 발명의 일부 실시예들을 예시적인 도면을 통해 상세하게 설명한다. 각 도면의 구성 요소들에 참조 부호를 부가함에 있어서, 동일한 구성 요소들에 대해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의해야 한다.
- [0019] 그리고 본 발명의 실시예를 설명함에 있어, 관련된 공지 구성 또는 기능에 대한 구체적인 설명이 본 발명의 실시예에 대한 이해를 방해한다고 판단되는 경우에는 그 상세한 설명은 생략한다.
- [0020] 또한, 본 발명의 실시예의 구성 요소를 설명하는 데 있어서, 제1, 제2, A, B, (a), (b) 등의 용어를 사용할 수 있다. 이러한 용어는 그 구성 요소를 다른 구성 요소와 구별하기 위한 것일 뿐, 그 용어에 의해 해당 구성 요소의 본질이나 차례 또는 순서 등이 한정되지 않는다.
- [0021] 본 명세서에서 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다. 명세서에서 사용되는 “포함한다” 및/또는 “포함하는”은 언급된 구성요소 외에 하나 이상의 다른 구성요소의 존재 또는 추가를 배제하지 않는다.
- [0022] 이하, 첨부된 도면을 참조하여 본 발명을 보다 상세하게 설명하고자 한다.
- [0024] 도 1은 본 발명에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템(1000)의 주요 구성을 보여주는 블록도이다.
- [0025] 본 발명에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템(1000)(이하, ‘시스템(1000)’이라 함)은 데이터베이스부(미도시)에 저장된 해상교통 데이터의 전처리(preprocessing)를 수행하는 전처리부(100), 전처리부(100)에 의해 처리된 데이터를 기 설정된 시간에 대한 텐서 형태로 적재하고, 적재된 텐서 형태의 해상교통 데이터를 데이터베이스부에 저장하며, 디스플레이장치에 표현하는 적재부(200) 및 텐서 형태로 적재된 데이터에 기초하여 머신러닝 모델을 학습하고, 텐서 형태로 적재된 데이터에 기초하여 학습된 머신러닝 모델을 통해 해상교통 혼잡도를 예측하는 학습 및 예측부(300)를 포함한다.
- [0027] 도 2는 본 발명에 따른 전처리부(100)의 데이터 보간 작업에 따른 가시화를 예시적으로 보여주는 도면이다.
- [0028] 먼저, 본 발명에 따른 전처리부(100)는 해상교통 데이터의 전처리(preprocessing)를 수행한다. 해상교통 데이터는, 일 실시예로서, 한국형 이네비게이션(Korean e-Navigation)의 일환으로 구축된 선박 및 환경데이터 수신 시스템(미도시) 또는 선박 및 환경데이터 수신 시스템이 관리하는 서버(미도시)로부터 DDS(Data Distribution System)를 통해 수신하여 데이터베이스부에 저장될 수 있으며, 선박 및 환경데이터 수신 시스템 또는 선박 및 환경데이터 수신 시스템이 관리하는 서버는 AIS 수신기, VPASS 수신기, LTE-M 수신기, VPASS 수신기, 바다누리 해양정보 서비스 등을 포함한다.
- [0029] 데이터베이스부에 저장된 해상교통 데이터는, AIS 동적정보 및 AIS 정적정보를 포함할 수 있으며, 이에 더하여 정보 수신 시간(Time), 선박식별번호(MMSI), 위도(Latitude), 경도(Longitude), 침로(CoG), 속도(SoG), 선박종류(ShipType), 선박의 길이, Heading, 선박의 폭, 흘수(吃水) 등에 대한 정보를 포함할 수 있다.
- [0030] 전처리부(100)는, 교통 혼잡도를 예측하고자 하는 대상 해역의 해상교통 데이터(항적 데이터)를 데이터베이스부로부터 추출할 수 있다. 전처리부(100)는 데이터베이스부로부터 추출한 해상교통 데이터의 이상치 제거, 결측치 처리 및 데이터 보간(data interpolation) 작업을 포함하는 전처리 작업을 수행한다.
- [0031] 전처리부(100)는, 해상교통 데이터의 이상치 제거와 결측치 처리 시 ‘Recommendation ITU-R M.1371-5’에서 명기하고 있는 유효한 해상교통 데이터 범위를 참고하여 이상치를 제거하고, 결측치를 처리한다.
- [0032] ‘Recommendation ITU-R M.1371-5’에서 명기하고 있는 유효한 해상교통 데이터 범위는 아래 <표 1> 및 <표 2>와 같다.

표 1

구분	Number of bits	특성
Navigational status	4	0 = under way using engine, 1 = at anchor, 2 = not under command, 3 = restricted maneuverability, 4 = constrained by her draught, 5 = moored, 6 = aground, 7 = engaged in fishing, 8 = under way sailing,
Rate of turn ROTAIS	8	0 to +126 = turning right at up to 708° /min or higher 0 to -126 = turning left at up to 708° /min or higher ROTAIS = 4.733 SQRT(ROTsensor) ° /min
SOG	10	Speed over ground in 1/10 knot steps (0-102.2 knots) 1023 = not available, 1022 = 102.2 knots or higher
Longitude	28	Longitude in 1/10000 min (±180° , East = positive, West = negative) 181 = not available
Latitude	27	Latitude in 1/10000 min (±90° , North = positive, South = negative) 91 = not available
COG	12	Course over ground in 1/10 = (0-3599). 3600 = not available 3601-4095 should not be used
True heading	9	Degrees (0-359) 511 indicates not available

<표 1 : 동적정보의 유효한 데이터 범위>

표 2

구분	Number of bits	데이터 범위
User ID	30	MMSI number
IMO number	30	0 = not available 0000000001-0000999999 = not used 0001000000-0009999999 = valid IMO number 0010000000-1073741823 = official flag state number
Call sign	42	7 x 6 bit ASCII characters, @@@@@@ = not available
Name	120	Maximum 20 characters 6 bit ASCII “@@@@@@@@@@@@@@@@@@@@” = not available
Type of ship and cargo type	8	0 = not available or no ship 1-99 = as defined in § 3.3.2 100-199 = reserved, for regional use 200-255 = reserved, for future use Not applicable to SAR aircraft
Overall dimension/reference for position	30	Reference point for reported position. Also indicates the dimension of ship (m)

Type of electronic position fixing device	4	0 = undefined (default) 1 = GPS 2 = GLONASS 3 = combined GPS/GLONASS 4 = Loran-C 5 = Chayka 6 = integrated navigation system 7 = surveyed 8 = Galileo, 9-14 = not used 15 = internal GNSS
ETA	20	Estimated time of arrival; MMDDHHMM UTC Bits 19-16: month; 1-12; 0 = not available Bits 15-11: day; 1-31; 0 = not available Bits 10-6: hour; 0-23; 24 = not available Bits 5-0: minute; 0-59; 60 = not available
Maximum present static draught	8	In 1/10 m, 255 = draught 25.5m or greater, 0 = not available
Destination	120	Maximum 20 characters using 6-bit ASCII; @@@@@@@@@@@@@@@@@@@@ = not available

[0038] <표 2 : 정적정보의 유효한 데이터 범위>

[0040] 한편, 이상치는, 데이터의 대부분 값의 범위에서 벗어난 극단적으로 크거나 작은 값을 의미한다. 전처리부(100)는 이상치 제거 시, 예를 들어, IQR(Interquartile Range)을 활용하거나 Z-Score를 활용할 수 있다. 다만 이에 한정되는 것은 아니고, 공지된 다른 방법을 통해서도 이상치 제거가 수행될 수 있다.

[0041] 예를 들어, IQR 방법은 사분위범위수(Interquartile range)를 활용한 방법으로 제3 사분위수(75%)에서 제1 사분위수(25%)를 뺀 값이 IQR이다. IQR방법에 대한 이상치 판단식은 다음과 같다.

[0042] ‘IQR방법 이상치판단 = 값<Q1-1.5xIQR 또는 값>Q3-1.5xIQR’

[0043] 예를 들어, 100개의 데이터가 있다고 가정하고 제일 큰 값에서 점점 작아지는 방향으로 일렬로 세웠을 때, 25번째로 큰 값이 제3분위수(Q3)이다. 그리고, 75번째로 큰 값이 제1분위수(Q1)이며 IQR은 Q3-Q1을 한 값이다. 25번째로 큰 값에 1.5배 IQR을 더한 값, 그리고 75번째로 큰 값에 1.5배 IQR을 뺀 값이 정상적인 데이터로 본다는 의미이다. IQR방법에서는 먼저 Q1, Q3, IQR 값을 구한다. quantile은 분위수를 구하는 함수로 1이 제일 큰 값, 0이 제일 작은 값이다. 이후, 이상치의 경계 값을 구한다. 예를 들어, Q3+1.5IQR 보다 큰 값과 Q1-1.5IQR보다 작은 값을 이상치로 보는 경우, 그 경계 값을 구한다. 이후, 그 경계 값을 벗어난 데이터 값을 이상치로 결정하고, 그 이상치를 제거한다.

[0044] 예를 들어, Z-Score 방법은 데이터 값을 집단의 평균으로 뺀 뒤에 표준편차로 나눈 값의 특정기준을 활용한다. 일반적으로 Z-score 1.5를 사용하며 값이 커질수록 이상치를 판단하는 범위가 넓어진다. 즉, Z-Score가 커지면 이상치가 덜 발생한다. Z-Score를 구하는 방법은 다음과 같다.

[0045] ‘Z-Score=(값-평균)/표준편차’

[0046] 즉, Z-Score 방법에서, 데이터 값에서 평균을 빼주고 표준편차로 나눠 z score를 계산하고, z score가 특정기준을 넘어서면 이상치라고 판단하며, 그 이상치를 제거한다.

[0047] 한편, 결측치란 말 그대로 데이터에 값이 없는 것을 뜻한다. 줄여서 NA라고도 하고, Null 이라는 표현도 쓴다. 이러한 결측치는 데이터 분석하는데 있어 매우 방해가 되며, 예를 들어, 결측치를 모두 제거(Deletion)하면 막대한 데이터 손실을 부를 수 있고, 결측치를 잘못 대체(대치, Imputation))하면 데이터에서 편향이 생길 수 있어, 결측치 처리에 따라 분석결과가 매우 틀어질 수 있다. 결측치 처리는 단순히 빼는 것이 가장 쉽지만, 실제 분석에서는 결측치를 특정값이나 통계량, 즉, 최빈값(mode), 중앙값(median), 평균값(mean), 조건부 대치(Similar case Imputation) 또는 회귀분석을 이용한 대치(Generalized Imputation)를 통해 채워넣는 것이 바람

직할 수 있다. 또는 예측 모델(Prediction model)을 통해 결측치를 처리하는 것이 더 바람직할 수 있다. 예측 모델을 통한 결측치 처리는 결측치들의 특성이 패턴을 가진다고 가정하고 진행한다. 이때 결측값이 없는 컬럼들로 구성된 dataset으로 결측값이 있는 컬럼을 예측하거나, 회귀 분석 기술을 활용하거나, SVM과 같은 기계 학습 방법 또는 이러한 결측치를 채우는 데이터 마이닝 방법과 같은 다양한 통계 방법을 통해 결측치를 처리할 수 있다.

- [0048] 전처리부(100)의, 해상교통 데이터의 데이터 보간은, 선박의 동적정보에 따라 달라지는 해상교통 데이터 주기를 모든 선박에 동일하게 맞추기 위한 것이다. 데이터의 보간 작업은, 실제 해상교통 상황과 유사하도록 모든 선박의 데이터를 동일한 기준으로 적용하기 위해 반드시 수행되는 것이 바람직하다. 데이터 보간 작업에 따른 가시화 예시는 도 2에 도시된 바와 같다.
- [0050] 본 발명에 따른 적재부(200)는, 전처리부(100)에 의해 처리된 데이터를 기 설정된 시간에 대한 텐서 형태로 적재한다. 적재부(200)는, 데이터 격자화, 데이터 적재 및 데이터 예측을 수행한다.
- [0051] 도 3은 국립해양조사원의 격자형 해양정보관리체계를 보여주는 도면이고, 도 4는 본 발명에 따른 적재부(200)의 데이터 격자화의 예시를 보여주는 도면이다.
- [0052] 먼저, 적재부(200)는 전처리부(100)에 의해 전처리된 데이터를 복수 개의 격자에 표현하는 데이터 격자화를 수행한다.
- [0053] 적재부(200)는 데이터 격자화를 위해, 도 3을 참조하면, 국립해양조사원(KHOA)의 격자형 해양정보관리체계에 기초하여, 도 4를 참조하면, 6 단계의 격자화를 수행한다. 적재부(200)는 해상교통 혼잡도 예측을 위해, 6 단계의 격자를 기본으로 사용하며, 필요시 더 많은 단계의 격자를 사용하여 각 격자를 축소할 수 있다. 데이터를 격자화하면 데이터의 누락을 방지하고 정보가 소실되는 것을 최소화할 수 있다. 한편, 적재부(200)는, 한 격자에 복수 개의 데이터가 존재할 경우 복수 개의 데이터의 평균치를 데이터베이스부에 저장한다.
- [0055] 도 5는 본 발명에 따른 적재부(200)의 데이터 적재 개념도이고, 도 6은 해상교통 데이터를 위경도 좌표계를 기준으로 1초마다 도시한 예시를 보여주는 도면이고, 도 7은 도 6의 위경도 좌표계에 도시된 데이터를 누적 도시한 예시를 보여주는 도면이고, 도 8은 도 6의 위경도 좌표계에 도시된 데이터를 누적하지 않고 텐서 형태로 적재한 예시를 보여주는 도면이다.
- [0056] 적재부(200)는, 데이터 격자화를 통해 데이터베이스부에 저장된 복수 개의 데이터를 적재한다. 적재부(200)는, 데이터의 적재 시, 도 5에 도시된 바와 같이, 위도, 경도 및 시간의 축을 기준으로 선박들의 침로(CoG, Course of Ground) 정보, 속도(SoG, Speed of Ground) 정보 및 길이 정보를 텐서 형태로 데이터베이스부에 저장한다. 적재부(200)는 필요시 Heading, 선박의 폭, 흘수 등을 더 저장할 수 있다. 이때 위경도 축의 기본 단위는 데이터를 격자화한 크기와 동일하다. 그리고 시간의 축은 기 설정된 시간동안 1초 단위로 이루어진다. 예를 들어 기 설정된 시간이 1시간일 경우, 적재부(200)는 1초 단위로 0초부터 3599초까지 총 1시간의 데이터를 텐서 형태로 적재한다.
- [0057] 적재부(200)는, 예를 들어, 각 1초의 데이터들이 위경도 좌표계를 기준으로 해상교통 데이터가 도 6과 같이 표현된다면 이를 누적하여 도 7과 같이 항적도의 형태로 데이터를 표현할 수 있으며, 이를 누적하지 않고 텐서 형태로 적재하면 도 8의 형태로 표현할 수 있다. 텐서 형태의 해상교통 데이터는 시간에 따른 선박들의 움직임, 즉 해상교통 상황을 정보의 소실을 최소화하여 표현할 수 있다.
- [0058] 적재부(200)는, 적재된 텐서 형태의 해상교통 데이터를 데이터베이스부에 저장 및/또는 적재된 텐서 형태의 해상교통 데이터를 디스플레이장치에 표현할 수 있다.
- [0060] 도 9는 본 발명에 따른 적재부(200)의 데이터 예측의 개념도를 도시한 도면이고, 도 10 및 도 11은 본 발명에 따른 적재부(200)의 데이터 예측 시 예시를 각각 보여주는 도면이다.
- [0061] 적재부(200)는 기 설정된 시간(예를 들어, 1시간)동안 온전한 데이터를 적재하지 못하는 경우, 적재되지 못한 데이터를 대상으로 데이터의 예측을 수행한다. 적재부(200)는 데이터 예측 시 Kinematic 외삽법 또는 Extended Kalman filter 법을 통해 데이터를 예측하며, 이때 적재부(200)는 도 9에 도시된 바와 같이 t 까지의 실제 데이

터 이후, $t+n$ 까지 데이터를 예측한다.

[0062] 예를 들어 적재부(200)는 Kinematic 외삽법을 통해 아래 수식 1과 같이 선박의 속도 정보 및 방위 정보를 추정하여 선박의 선수방향 속도 및 정횡방향 속도를 예측할 수 있다.

$$\begin{aligned} \psi_{i+n} &= \psi_i + \left(\frac{\psi_i - \psi_{i-1}}{\nabla t} \right) n && \text{여기서, } \psi = \text{Course of Ground} \\ U_{i+n} &= U_i + \left(\frac{U_i - U_{i-1}}{\nabla t} \right) n && U = \text{Speed of Ground} \\ u_{i+n} &= U_{i+n} \cos(\psi_{i+n}) && t = \text{time} \\ v_{i+n} &= U_{i+n} \sin(\psi_{i+n}) && u = \text{Lateral speed} \\ &&& v = \text{Vertical speed} \\ &&& \nabla t = t_i - t_{i-1} \end{aligned}$$

[0064]

[0065] <수식 1>

[0067] 적재부(200)는, Kinematic 외삽법을 통해 선박의 속도 정보를 예측할 때, 선박의 속도 정보를 포함하는 선박 운항 정보가 기 설정된 시간간격보다 큰 간격으로 수신되는 경우, 아래 <수식 2>와 같이 기 수신한 선박의 속도 정보를 기반으로 예측에 제한을 둘 수 있다.

$$U_{i+n} = \begin{cases} \max(U_i, U_{i-1}) & , U_{i+n} > \max(U_i, U_{i-1}) \\ \min(U_i, U_{i-1}) & , U_{i+n} < \min(U_i, U_{i-1}) \\ U_{i+n} & , otherwise \end{cases}$$

[0069]

[0070] <수식 2>

[0072] 이때 적재부(200)는, 기 예측한 선박의 선수방향 속도 및 정횡방향 속도를 기준으로 기 예측된 선박의 위치 데이터에 오차가 있는 경우, 그 오차까지 감안하여 아래 <수식 3>과 같이 상기 선박의 위치 정보를 예측한다.

$$\begin{aligned} \lambda_{i+n} &= \lambda_i + \int_{t_i}^{t_{i+n}} \frac{v_t}{Ne \times \cos(\phi_t)} dt + \Delta \lambda \\ \phi_{i+1} &= \phi_i + \int_{t_i}^{t_{i+n}} \frac{u_t}{Me} dt + \Delta \phi \end{aligned}$$

여기서, $\lambda = \text{Longitude}$

$\phi = \text{Latitude}$

$Ne = \text{The prime vertical radius of curvature}$

$Me = \text{The median radius of curvature}$

$\Delta \lambda = \text{Estimation error of Longitude}$

$\Delta \phi = \text{Estimation error of Latitude}$

[0074]

[0075] <수식 3>

[0077] 또한 적재부(200)는 Extended Kalman filter법을 통해 데이터를 예측할 수 있으며, 이때 간략한 선박 조종 모델인 K-T 모델을 사용하여 다음과 같이 계산한다.

[0078] 한편, <수식 4>는 K-T 모델을 나타내며 <수식 5>는 K, T 상수를 무차원화하여 나타낸 식에 대한 것이다.

[0080]

$$T \cdot \ddot{\psi} + \dot{\psi} = K \cdot \delta$$

[0081] <수식 4>

$$\frac{L}{U} T' \cdot \ddot{\psi} + \dot{\psi} = \frac{U}{L} K' \cdot \delta$$

where :

$\ddot{\psi}$: Yaw angular acceleration [rad/s²]
 $\dot{\psi}$: Yaw angular velocity [rad/s]
 δ : Rudder angle [rad]
 K : Proportionality constant [1/sec]
 T : Time constant [sec]
 L : Ship Length [m]

[0083]

[0084] <수식 5>

[0086] <수식 5>에서 ψ , $\dot{\psi}$ 을 Extended Kalman Filter의 State로 사용하기 위한 State Equation은 <수식 6>과 같다.

[0088]

$$\begin{aligned} \psi &= \int \dot{\psi} dt \\ \dot{\psi} &= \int \left(-\frac{U}{L} \frac{1}{T'} \dot{\psi} + \frac{U^2}{L^2} \frac{K'}{T'} \delta \right) dt \\ &= \int \left(-\frac{U}{L} a \dot{\psi} + \frac{U^2}{L^2} b \delta \right) dt \quad \left(\because a = \frac{1}{T'}, b = \frac{K'}{T'} \right) \end{aligned}$$

[0089] <수식 6>

[0091] Extended Kalman Filter의 기본 State Space Format은 <수식 7>과 같다.

[0093]

$$\begin{aligned} X_{k+1} &= F_k \cdot X_k + w_k \\ y_{k+1} &= H_k \cdot X_k + v_k \end{aligned}$$

[0094] <수식 7>

[0096] <수식 7>의 X는 State Vector, Y는 Output Vector, F와 H는 System Matrix, w_k 는 Process Noise, v_k 는 Measurement Noise를 말하며, K-T 방정식을 State Space로 나타내면 <수식 8>과 같다.

$$\begin{bmatrix} \psi \\ \dot{\psi} \\ a \\ b \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & dt & -\frac{1}{2} \frac{U}{L} dt^2 \dot{\psi} & \frac{1}{2} \frac{U^2}{L^2} \delta dt^2 \\ 0 & 1 & -\frac{U}{L} \dot{\psi} dt & \frac{U^2}{L^2} \delta dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \psi \\ \dot{\psi} \\ a \\ b \end{bmatrix}_k + w_k$$

$$y_k = [1 \ 0 \ 0 \ 0] X_k + v_k$$

[0098]

[0099] <수식 8>

[0101] Extended Kalman Filter 알고리즘은 다음과 같이 Prediction과 Update를 반복하는 구조이며, 알고리즘의 초기 값 P_0 는 Identity matrix를, X_0 는 경험식을 사용해 초기화한다.

[0102] - Initialize

$$\begin{aligned} K_0' &= 0.428419 + 0.19936 L/B \\ T_0' &= 1.94599 - 0.04104 L/B \\ P_0 &= I_4 \end{aligned}$$

[0103]

[0104] <수식 8-1>

[0106] Extended Kalman Filter를 이용해 u, v, r을 추정하여 WGS84 기준의 Geodetic 좌표계의 위·경도를 계산하는 식은 <수식 9>와 같다.

$$\begin{aligned} \psi_{k+1} &= \psi_k + \int r_k dt \\ U_{k+1} &= u_k \times \cos(\psi_{k+1}) - v_k \times \sin(\psi_{k+1}) \\ V_{k+1} &= u_k \times \sin(\psi_{k+1}) + v_k \times \cos(\psi_{k+1}) \\ \phi_{k+1} &= \phi_k + \int U_{k+1}/M_E dt \\ \lambda_{k+1} &= \lambda_k + \int V_{k+1}/(N_E \times \cos \phi_k) dt \end{aligned}$$

where :

ϕ : Latitude [deg]

λ : Longitude [deg]

M_E : The meridian radius of curvature

$$= \frac{R_{ea}(1-e^2)}{(1-e^2 \sin^2 \phi)^{3/2}}$$

N_E : The prime vertical radius of curvature

$$= \frac{R_{ea}}{\sqrt{1-e^2 \sin^2 \phi}}$$

R_{ea} : semi-major axis

$$= 6,378,137.0m$$

e : The first eccentricity

$$= 0.08181919$$

[0108]

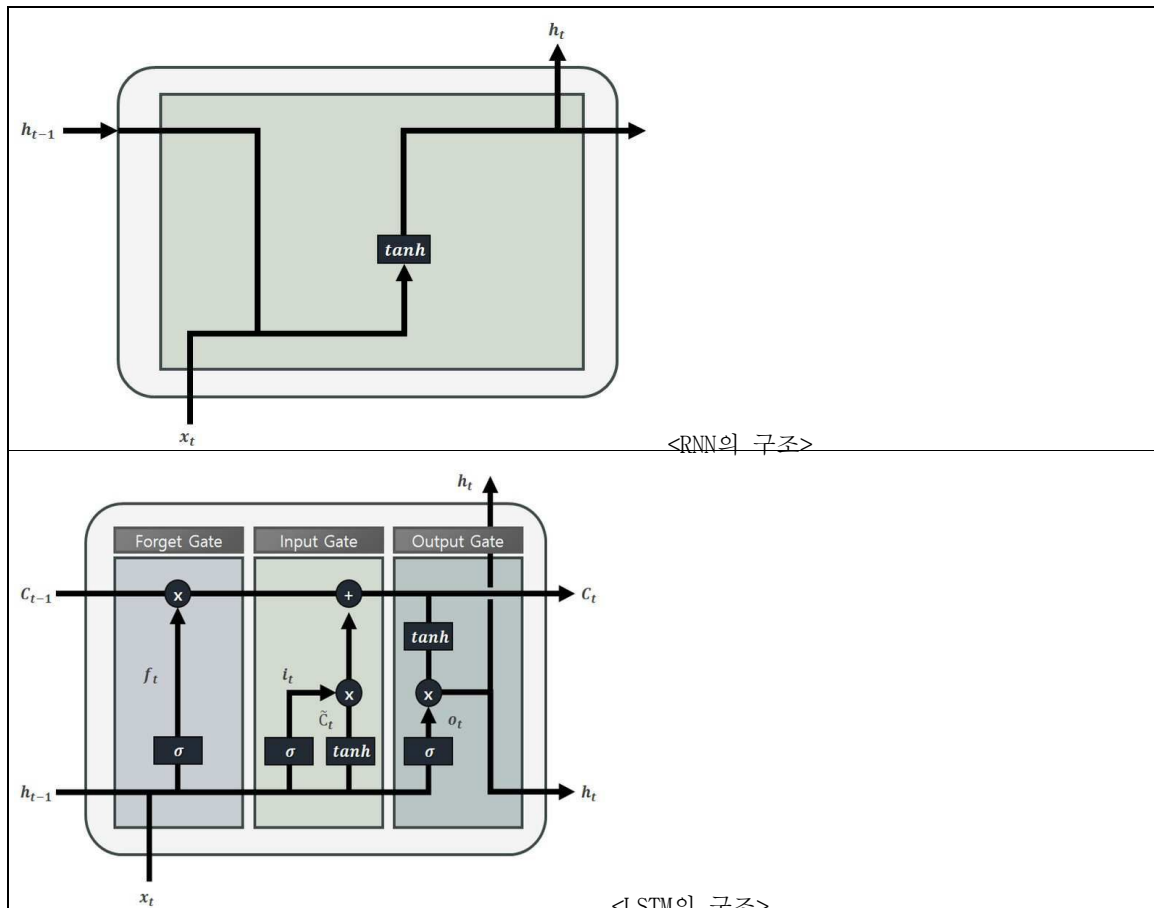
[0109] <수식 9>

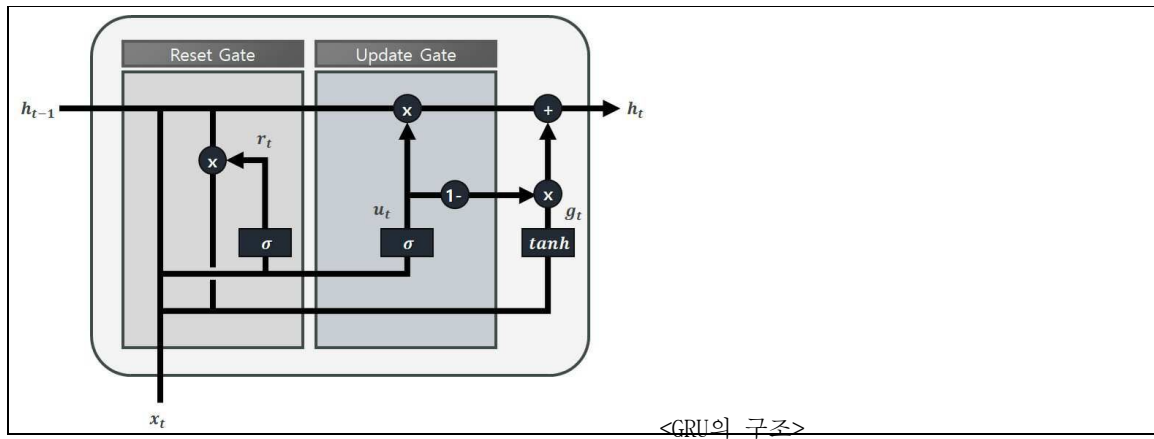
[0111] 적재부(200)는 도 10에 도시된 바와 같이 0초의 해상교통 데이터만 적재 가능한 상황에서는, 불게 표시된 1초에서 3599초의 해상교통 데이터를 예측한다. 또한 적재부(200)는 도 11에 도시된 바와 같이 1초까지의 해상교통 데이터를 적재 가능한 상황에서는 불게 표시된 2초에서 3599초에 대한 예측을 수행한다. 이를 통해 적재부(200)는 온전한 데이터를 적재하지 못하는 경우에도 기 설정된 시간에 대한 텐서 형태의 해상교통 데이터를 확보할 수 있다.

[0113] 본 발명에 따른 학습 및 예측부(300)는, 텐서 형태로 적재된 데이터에 기초하여 머신러닝 모델을 학습하고, 학습된 머신러닝 모델을 통해 해상교통 혼잡도를 예측한다. 예를 들어, 학습 및 예측부(300)는, 적재부(200)가 생성하여 데이터베이스부에 저장한 텐서 형태의 데이터를 학습데이터로 사용하여 머신러닝 방법 등을 통해 향후 1시간에서 24시간 동안의 해상교통 혼잡도를 예측한다.

[0115] 머신러닝 모델은, 표 3을 참조하면, 시계열 데이터를 처리하는 대표적인 모델인 순환 신경망(Recurrent Neural Network, RNN), 장단기 메모리(Long short-Term Memory, LSTM), 게이트 순환신경망(Gate Recurrent Unit, GRU) 등이 있다. RNN은 은닉층에서 활성화 함수를 통해 나온 결과값을 출력층과 은닉층의 다음 입력으로 보내는 특징이 있으며, 시퀀스가 길어질수록 앞의 정보가 뒤로 충분히 전달되지 못하는 현상인 장기 의존성 문제가 있다. LSTM은 RNN의 장기 의존성 문제의 해결책으로 제시되었으며, 3개의 게이트에서 활성화 함수를 통해 과거와 현재의 입력 정보 중에 중요한 정보를 판단하고, 이를 이용하여 은닉층에 있는 셀스테이트(Cell state)의 정보를 제어한다. GRU는 업데이트 게이트와 리셋 게이트의 단순한 구조로 학습할 파라미터를 줄여 학습 속도를 향상하면서 LSTM과 유사한 수준의 성능을 보이는 모델이다.

표 3





- [0118] 도 12는 본 발명에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 방법의 각 단계를 대략적으로 도시한 도면이고, 도 13은 본 발명에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 방법의 각 단계를 상세하게 도시한 도면이다.
- [0119] 본 발명에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 방법(이하, ‘방법’이라 함)은, 도 12를 참조하면, 해상교통 데이터 전처리 단계(S100), 전처리된 데이터를 텐서 형태로 적재하는 적재 단계(S200) 및 적재된 데이터에 기초하여 머신러닝 모델 학습 및 해상교통 혼잡도를 예측하는 모델 학습 및 해상교통 혼잡도 예측 단계(S300)를 포함한다.
- [0121] 단계(S100)은, 전술한 바와 같이, 전처리부(100)가 해상교통 데이터의 전처리를 수행하는 단계이다. 단계(S100)에서 전처리부(100)는 데이터베이스부에서 교통 혼잡도를 예측하고자 하는 대상 해역의 해상교통 데이터를 추출하고, 추출한 해상교통 데이터의 이상치 제거, 결측치 처리 및 데이터 보간 작업을 수행한다.
- [0122] 상세하게는 단계(S100)은, 도 13을 참조하면, 전처리부(100)가 추출한 해상교통 데이터의 이상치를 제거하는 이상치 제거 단계(S101), 결측치가 존재하는 경우 결측치를 제거, 대체 또는 예측 모델을 통해 결측치를 처리하는 결측치 처리 단계(S103) 및 선박의 동적정보에 따라 달라지는 해상교통 데이터 주기를 모든 선박에 동일하게 맞추는 데이터 보간 단계(S105)를 포함한다.
- [0124] 단계(S200)은, 전술한 바와 같이, 적재부(200)가 각 데이터를 텐서 형태로 적재하여 데이터베이스부에 저장하는 단계이다.
- [0125] 상세하게는 단계(S200)은, 데이터 격자화 단계(S201), 데이터 적재 단계(S203), 데이터의 예측이 필요한지 판단하는 단계(S205) 및 데이터의 예측이 필요할 경우 데이터를 예측하는 단계(S207)를 포함한다.
- [0126] 데이터 격자화 단계(S201)에서 적재부(200)는, 전술한 바와 같이, 전처리부(100)에 의해 전처리된 데이터를 복수 개의 격자에 표현하고, 이를 데이터베이스부에 저장한다. 데이터 적재 단계(S203)에서 적재부(200)는, 전술한 바와 같이, 데이터 격자화를 통해 적재된 복수 개의 데이터를 위도, 경도 및 시간의 축을 기준으로 선박들의 침로 정보, 속도 정보 및 길이 정보를 텐서 형태로 적재 및 표현하고, 이를 데이터베이스부에 저장한다. 데이터의 예측이 필요한지 판단하는 단계(S205)에서 적재부(200)는, 전술한 바와 같이, 기 설정된 시간(예를 들어, 1 시간)동안 온전한 데이터가 적재되었는 지 여부를 판단하는 단계이다. 만일 기 설정된 시간동안 온전한 데이터가 적재되지 않은 경우 적재부(200)는, 데이터의 예측이 필요할 경우 데이터를 예측하는 단계(S207)에서, 전술한 바와 같이 적재되지 못한 데이터를 대상으로 데이터의 예측을 수행한다.
- [0128] 단계(S300)은, 전술한 바와 같이, 학습 및 예측부(300)가 텐서 형태로 적재된 데이터에 기초하여 머신러닝 모델을 학습하고, 학습된 머신러닝 모델을 통해 해상교통 혼잡도를 예측하는 단계이다.
- [0129] 상세하게는 단계(S300)은, 적재부(200)에 의해 텐서 형태로 적재된 데이터를 모델을 학습하기 위한 학습 데이터

또는 학습된 모델을 테스트하기 위한 테스트 데이터로 분할하여 출력하는 데이터 분할 단계(S301), 출력된 학습 데이터에 기초하여 모델을 학습하는 단계(S303), 출력된 테스트 데이터에 기초하여 학습된 모델을 이용한 혼잡도를 예측하고, 예측한 혼잡도를 출력하는 단계(S305)를 포함한다.

[0130] 한편, 본 발명에 따른 전처리부(100)는, 전술한 바와 같이, 교통 혼잡도를 예측하고자 하는 대상 해역의 해상교통 데이터를 데이터베이스로부터 추출할 수 있다. 이때의 해상교통 데이터는, AIS 동적정보 및 AIS 정적정보를 포함할 수 있으며, 이에 더하여 정보 수신 시간(Time), 선박식별번호(MMSI), 위도(Latitude), 경도(Longitude), 침로(CoG), 속도(SoG), 선박종류(ShipType), 선박의 길이, Heading, 선박의 폭, 흘수(吃水) 등에 대한 정보를 포함할 수 있다.

[0131] 이와 같은 해상교통 데이터는, 적재부(200)에 의한 데이터 격자화를 통해 위도, 경도 및 시간의 축을 기준으로 적재된다. 특히 해상교통 데이터는 기 설정된 시간동안 일정 시간의 단위로 정리되어 적재된다. 즉 해상교통 데이터, 데이터 격자화된 해상교통 데이터 및/또는 다차원 텐서 형태로 적재된 해상교통 데이터는 시계열 데이터라 할 수 있다.

[0133] 본 명세서에서는 시계열 데이터의 예측을 위한 MLOps(Machine Learning Operations) 워크플로를 기술하고 관리할 수 있는 언어와 플랫폼을 제안한다.

[0134] 시계열 데이터의 예측을 위해서는 개발된 모델을 빠르게 적용하고 변화하는 데이터의 특성에 따라 개선할 수 있는 MLOps 플랫폼이 필요하다. 이에, 최근 데이터 분석에 많이 활용되고 있는 Python 기반의 AMML(AI/ML Modeling Language)을 개발하여 손쉽게 MLOps 워크플로를 구성하고 실행할 수 있도록 제안한다.

[0135] 제안하는 AI MLOps 플랫폼은 AMML을 이용하여 다양한 데이터 소스(R-DB, NoSql DB, Log File 등)에서 시계열 데이터인 해상교통 데이터의 추출, 해상교통 데이터의 전처리 및 해상교통 혼잡도 예측을 수행할 수 있다.

[0136] 기계 학습은 컴퓨터의 시스템이 패턴과 추론에 의존하여 명시적인 지시 없이 업무를 수행하는 데 사용하는 알고리즘과 통계 모델을 개발하는 것을 말한다.

[0137] 기계 학습은 학습 방법에 따라 지도 학습(supervised learning)과 비지도 학습(unsupervised learning)으로 구분된다.

[0138] 지도학습은 입력 데이터와 정답 라벨이 있는 데이터에 대해 모델을 학습하는 것으로 목적 값의 형태에 따라 분류(Classification)와 회귀(Regression)로 나뉘어진다. 대표적인 알고리즘으로는 SVM(support vector machine), SGD(Stochastic Gradient Descent), NN(Nearest Neighbors), GP(Gaussian Processes), NB(Naive Bayes), DT(Decision Trees) 등이 있다.

[0139] 반면 비지도 학습은 정답 라벨이 있는 데이터 없이 입력 데이터만 있는 경우에 학습하는 것이며, 군집화(Clustering), 차원 축소(Dimension reduction)로 나뉘어진다. 대표적인 군집화 알고리즘으로 k-평균(k-means), DBSCAN(density-based spatial clustering of applications with noise), EM(expectation-maximization), OPTICS(Ordering points to identify the clustering structure) 등이 있다.

[0140] 기계 학습 모델의 적절한 적용을 위해서는 개발뿐만 아니라, 예측하지 못한 환경의 변화로 데이터의 통계적 분포가 변화하는 상황이 발생하여 모델의 정확도를 낮아지게 할 수 있기 때문에, 데이터 변화에 유연하게 적응할 수 있도록 학습을 위한 자동화된 데이터의 확보 방안과 학습된 모델을 운영 환경에 지속해서 유지/관리하는 것이 중요하다. 따라서 변화하는 데이터를 운영 환경에서 지속적으로 수집하고 학습하여 모델을 최신화해야 할 필요성이 있다.

[0141] 특히, 시스템(1000)을 모니터링할 때 이상 상황이 발생하기 전에 징후의 변화를 미리 확인하고 향후 모델의 개선을 위해 데이터를 저장하고 관리할 필요가 있다. 일반적으로 시스템(1000)에 오류가 발생하면 이상 상황 해결에 집중하기 때문에, 이상 상황 분석에 필요한 비정상 데이터들은 자동으로 저장되어야 한다. 또한 이렇게 얻은 데이터를 이용하여 학습된 모델을 운영 환경에 빠르게 적용할 수 있어야 한다.

[0142] 시계열 데이터는 일정 기간 동안 수집된 일련의 순차적인 데이터를 의미하며, 추세(Trend)와 주기성(cycle), 계절성(seasonality) 등의 특징이 있다. 이러한 시계열 데이터를 예측하는 것은 시간 범위에 따라, 예측에 필요한 요소에 따라, 데이터의 형태에 따라 사용하는 기법이 달라지기 때문에 매우 어려운 작업이다.

[0143] 시계열 분석 기법으로는 통계적 기법과 딥러닝을 활용한 기법들이 있다. 통계적 기법으로는 회귀 분석,

ARIMA(autoregressive integrated moving average), ETS(Exponential Smoothing) 기법 등이 있으며 딥러닝 기법으로는 RNN(Recurrent Neural Network), LSTM(Long Short-Term Memory), GRU(Gated Recurrent Unit) 등의 다양한 기법이 사용되고 있다.

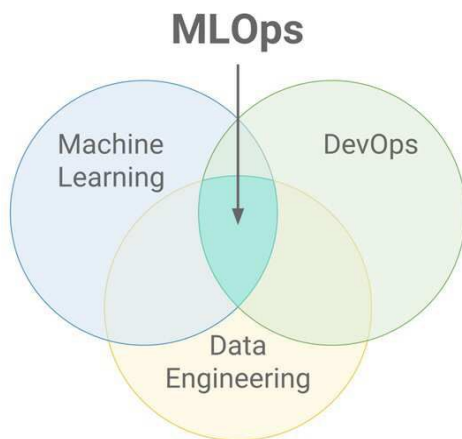
[0145] 본 명세서에서는 시계열 데이터 예측을 위한 MLOps 플랫폼을 제안한다. 세부적으로는 자동화된 운영 데이터 저장과 신속한 모델을 전개를 위한 Python 기반의 워크플로기술 언어 AMML(AI/ML modeling language)과 실행 엔진, 모델 관리 방안에 대해 제안한다.

[0146] 이하에서는 순서를 붙여 제안하는 MLOps(machine learning operations) 플랫폼에 대한 세부 사항, LSTM 모델을 사용한 사례 및 MLOps 플랫폼에 대한 결론 등을 제시한다.

[0148] 1. Related works

[0149] 1.1 MLOps

[0150] MLOps는 기계 학습 시스템을 소프트웨어의 개발과 운영을 구분하지 않고 개발의 생산성과 운영의 안전성을 최적하기 위한 DevOps의 개념을 기계학습 시스템에 적용한 것으로 지속적 통합(CI, continuous integration), 지속적 배포(CD, continuous deploy), 지속적 자동화(CT, continuous training)의 원칙이 있다. <그림 1>은 MLOps의 개념도이다.



[0152]

[0153] <그림 1> MLOps의 개념도 (출처 : Breuel, Cristiano. "ML Ops: Machine Learning as an Engineering Discipline". Towards Data Science. Retrieved 6 July 2021.)

[0155] 기계 학습 시스템은 기존 소프트웨어와 달리 코드와 데이터로 모델을 학습하기 때문에, 예측하지 못한 환경의 변화로 데이터의 통계적 분포가 변화하는 상황에 대응하기 위해 데이터 수집, 통합, 분석, 라벨링, 검증 등의 데이터 엔지니어링 기술이 필수적이다.

[0157] MLOps는 일반적인 단위, 통합 테스트 외에 데이터 검증, 학습된 모델 품질 평가, 모델 검증이 추가로 필요하며, 학습된 모델을 배포하는 것에 추가하여, 모델을 재학습하고 검증하는 과정의 자동화가 필요하다. 알고리즘의 최적화를 통해 최고의 성능을 낼 수 있는 일반적인 소프트웨어 시스템과 달리 기계 학습 시스템은 데이터 환경의 변화만으로도 성능이 저하될 수 있다. 따라서 데이터 환경의 변화를 추적하고 지속적으로 데이터를 수집하여 모델의 재학습을 통해 시스템의 성능을 최적화하는 작업이 필요하다.

[0159] MLOps의 CI는 코드뿐만 아니라 데이터, 데이터 스키마, 모델 등이 모두 테스트되고 검증되어야 하며, CD는 기계

학습 파이프라인 전체를 배포해야 한다. CT는 DevOps와 구분되는 MLOps의 원칙으로 시스템의 지속 가능 여부와 직접적인 연관이 있다.

[0161] MLOps는 자동화 수준에 따라 성숙도 수준을 나타낼 수 있으며, 대표적으로 구글과 마이크로소프트에서 정의한 MLOps 성숙도 수준이 있다.

[0163] 구글의 성숙도 모델은 <그림 2> 내지 <그림 4>와 같이 총 3단계로 구성되며, 각 수준은 다음과 같다.

[0164] 수준 0: 수동 프로세스 (Manual process),

[0165] 수준 1: ML 파이프라인 자동화(MLpipeline automation),

[0166] 수준 2: CI/CD 파이프라인 자동화(CI/CD pipeline automation)

[0168] 마이크로소프트의 성숙도 단계는 <그림 5> 내지 <그림 9>와 같이 총 5 단계로 구성되며, 각 수준은 다음과 같다.

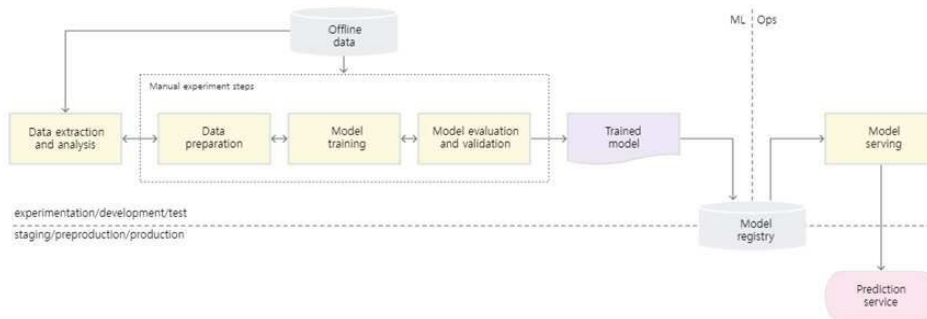
[0169] 수준 0: MLOps 없음(No MLOps),

[0170] 수준 1: DevOps는 있지만 MLOps 없음(DevOps no MLOps),

[0171] 수준 2: 자동화된 학습(Automated Training),

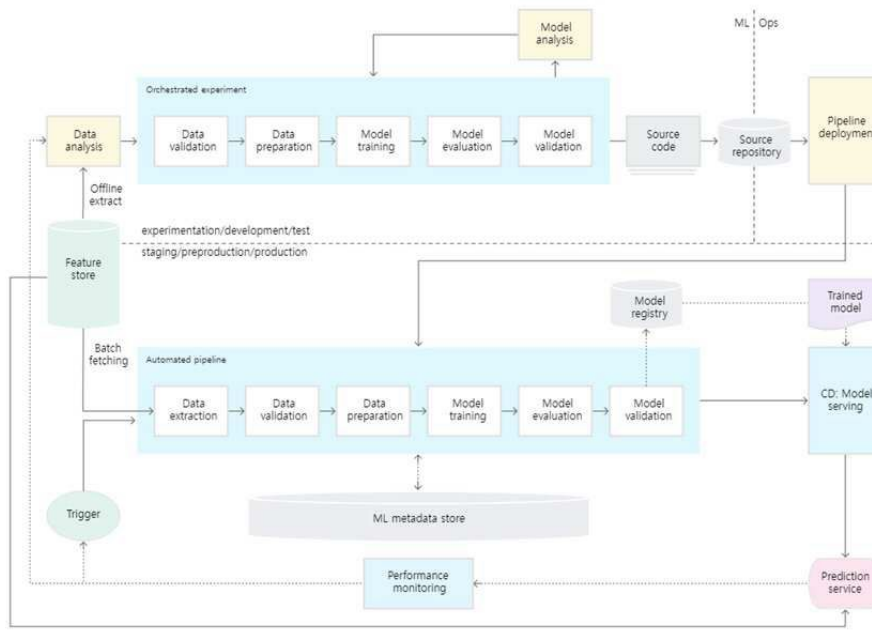
[0172] 수준 3: 자동화된 모델 배포(Automated Model Deployment),

[0173] 수준 4: 전체 MLOps 자동화된 작업(Full MLOps Automated Retraining)

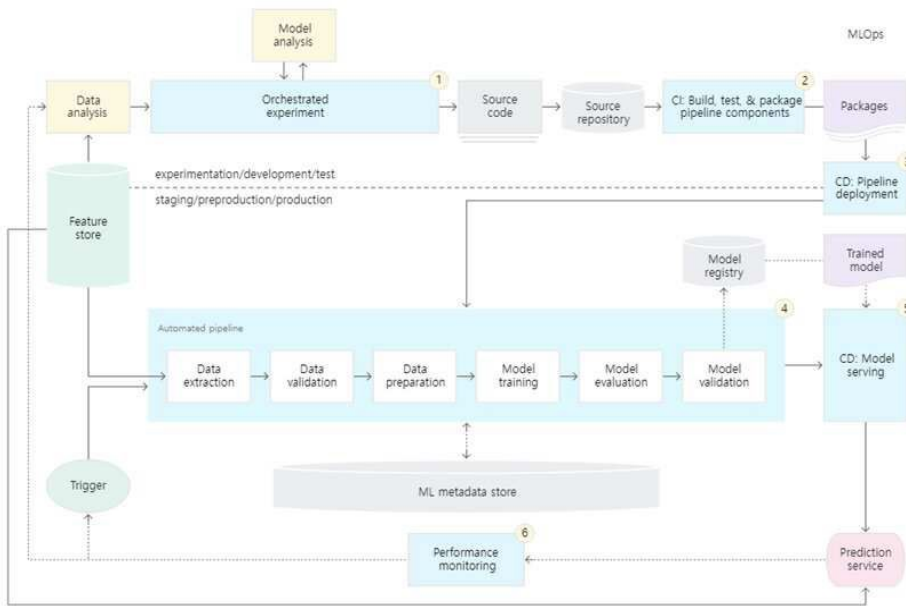


[0175]

[0176] <그림 2> 구글의 성숙도 모델 수준 0: 수동 프로세스 (Manual process)



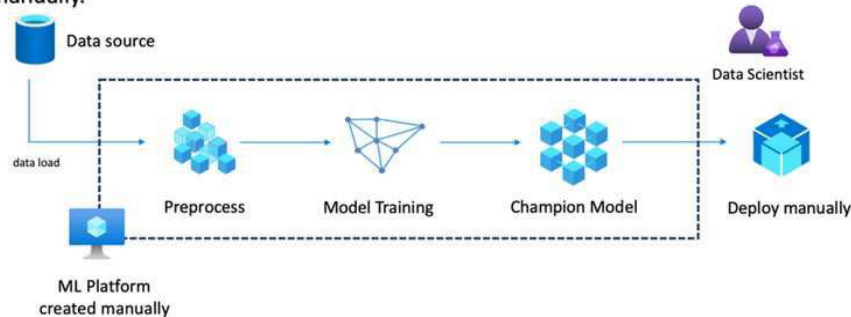
<그림 3> 구글의 성숙도 모델 수준 1: ML 파이프라인 자동화(MLpipeline automation)



<그림 4> 구글의 성숙도 모델 수준 2: CI/CD 파이프라인 자동화(CI/CD pipeline automation)

Level 0 – No MLOps

- Find best model interactively and exploratory.
- Create environment, gather and preprocess data, model training, deploy and test manually.

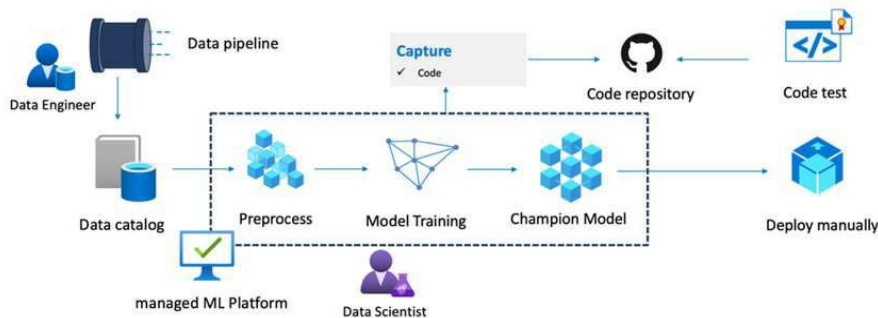


[0184]

[0185] <그림 5> 마이크로소프트의 성숙도 모델 수준 0: MLOps 없음(No MLOps)

Level 1 – DevOps no MLOps

- Create managed ML platform.
- Maintain code test against application and training/inference scripts.



[0187]

[0188] <그림 6> 마이크로소프트의 성숙도 모델 수준 1: DevOps는 있지만 MLOps 없음(DevOps no MLOps)

Level 2 – Automated Training

- Make Code, Data, Model tracked, saved and version controlled.
- Automate training process using pipeline.

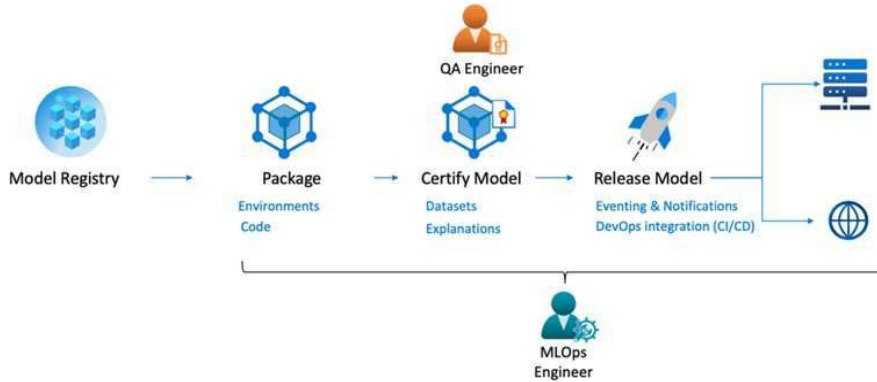


[0190]

[0191] <그림 7> 마이크로소프트의 성숙도 모델 수준 2: 자동화된 학습(Automated Training)

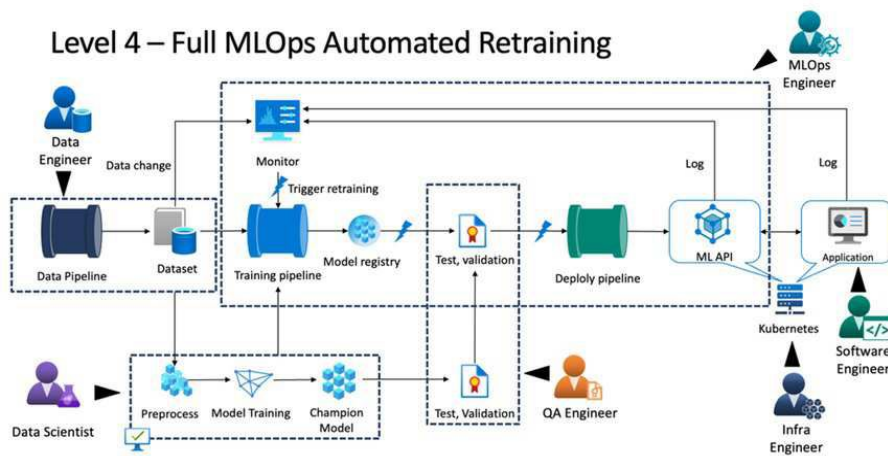
Level 3 – Automated Model Deployment

- Package model, Certify Model and release model are semi-automated.



[0193]

[0194] <그림 8> 마이크로소프트의 성숙도 모델 수준 3: 자동화된 모델 배포(Automated Model Deployment)



[0196]

[0197] <그림 9> 마이크로소프트의 성숙도 모델 수준 4: 전체 MLOps 자동화된 작업(Full MLOps Automated Retraining)

[0199] 대부분의 MLOps 오픈소스는 하나 이상의 핵심 기능에 특화되어 있지만 통합 관리형 오픈소스도 존재한다. <그림 10>은 대표적인 MLOps 오픈소스이다.

Tool	Purpose / Description	Technical Capability	MLOps Process Enabled
	Managing end-end ML workflows on Kubernetes	ML Pipeline Development	- ML Development - Training operationalization
	ML lifecycle management. MLflow shines when it comes to tracking experiments, metadata, and artifacts	Experiment & Metadata Tracking	- Model deployment - Model Management
	Deploying machine learning models at scale on Kubernetes	- Model Deployment - Model Serving	- Model deployment - Prediction serving
	Monitoring metrics collection and alerting toolkit	Monitoring	Continuous monitoring
	Dashboards to query, visualize, and alert on monitoring metrics	Monitoring	Continuous monitoring
	Automating training the deployment pipelines	Automation	Continuous training
	Creating user interface to interact with ML models	Model Serving	Prediction serving (consumption)

<그림 10> MLOps 기술 기능, 프로세스 및 오픈소스 도구

Kubeflow는 기계 학습 모델 개발부터 배포 단계까지 쿠버네티스(Kubernetes) 위에서 모든 작업에 필요한 도구와 환경을 쿠브플로우 컴포넌트로 제공한다. 쿠버플로우 컴포넌트는 Central Dashboard, Notebooks, Training Operators, Experiments(AutoML), KServe, Kubeflow Pipelines(KFP)로 구성되어 있다.

Central Dashboard는 웹브라우저 기반의 대시보드 UI로 Notebooks, Experiments (AutoML), Experiments (KFP) 등의 컴포넌트를 이용할 수 있다. Notebooks은 파이썬 코드를 작성하고 실행할 수 있는 Jupyter Notebook 개발 도구를 제공하며, 데이터 전처리와 분석 등 기계 학습 모델 코드를 개발할 수 있다. Training Operators는 TensorFlow, PyTorch, MXNet 등의 다양한 딥러닝 프레임워크에 대한 분산학습을 지원하며, 쿠버네티스에 배포된 분산 학습 명세서에 따라 워크로드를 수행한다. Experiments(AutoML)는 기계 학습 모델의 성능을 높이기 위해 반복 실험을 자동화하며, 모델의 하이퍼 파라미터를 최적화하는 하이퍼 파라미터 튜닝, 모델의 구조 및 노드 가중치 등 뉴럴 네트워크 아키텍처를 최적화하는 뉴럴 아키텍처 탐색 기능이 있다. KServe는 쿠버네티스에 기계 학습 모델을 배포하고 추론하는 기능을 제공하며 Endpoint, Transformer, Predictor, Explainer로 구성되어 있다. Endpoint는 Predictor에 데이터를 전달하고, Predictor는 데이터를 예측하거나 분류한다. Transformer나 Explainer는 필요에 따라 Predictor와 연결하여 사용하며, Explainer는 데이터를 예측하거나 분류한 결과에 대해 판단 이유를 제시하는 설명 가능한 인공지능(explainable Artificial Intelligence, XAI) 역할을 하고, Transformer는 데이터 전처리, 후처리 기능을 제공한다. Kubeflow Pipelines(KFP)은 머신러닝 워크플로우를 구축하고 배포하기 위한 ML Workflow Orchestration 도구이며, Pipelines과 Pipeline Components를 재사용하여 다양한 실험을 빠르고 쉽게 수행하는 것을 목표로 한다.

MLflow는 모든 ML 라이브러리, 알고리즘, 배포 도구 또는 언어와 함께 작동하도록 설계된 개방형 인터페이스를 장점으로 하는 기계 학습 플랫폼이며, Tracking과 Projects, Models로 구성되어 있다.

Tracking은 기계 학습 코드를 실행할 때 매개변수, 코드 버전, 메트릭 및 출력 파일을 기록하기 위한 API 및 UI이며, 학습 환경과 이에 따른 결과를 로컬 파일 또는 서버에 기록하여 비교 분석할 수 있다. Projects는 데이터 코드 패키징을 위한 표준 형식을 제공하며, 단순히 코드 또는 Git 리포지토리가 있는 디렉터리이며 설명자 파일을 사용하여 해당 종속성 및 코드 실행 방법을 지정한다. Conda 환경을 통해 종속성을 지정할 수 있으며, 프로젝트에는 명명된 매개 변수와 함께 실행을 호출하기 위한 여러 진입점이 있을 수도 있다. Models는 기계 학습 모델을 ‘flavors’ 라는 여러 형식으로 패키징하기 위한 규칙이며, 다양한 모델을 배포하는 데 도움이 되는 여러 도구를 제공한다. 각 모델은 임의의 파일과 Models이 사용할 수 있는 특징을 나열하는 설명자 파일이 포함된 디렉터리로 저장된다.

1.2 ML-Schema

[0211] ML-Schema는 W3C(world wide web consortium) Machine learning schema community group에서 제안한 것으로 기존 기계 학습 온톨로지를 정비하여 기계 학습 알고리즘, 학습 과정, 입출력 데이터 등을 나타내는 데 사용할 수 있는 상호운용성이 높은 온톨로지를 제공하는 것을 목적으로 한다.

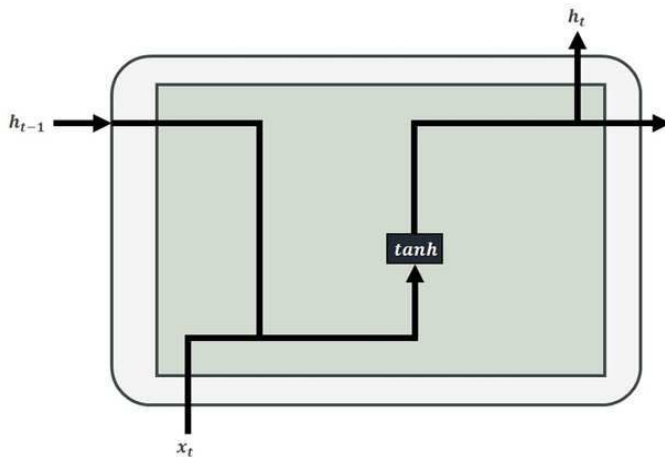
[0212] DL2IA(deep learning description language for image analysis)는 ML-Schema와 PROV-ML기반으로 만들어진 딥러닝 모델을 정의하고 학습할 수 있는 언어이며, XML로 만들어졌다. 하지만 XML은 워크플로와 모델 정보가 모두 포함되어 있어 이해하기 어려우며, 표현이 길어지는 단점이 있다.

[0213] 이러한 단점을 개선하기 위해 개발된 AMML은 워크플로 구성언어와 딥러닝 모델을 정의하는 언어로 구성되어 있으며 YAML을 기반으로 만들어진 언어이다. 이에 본 명세서에서는 간략화된 AMML을 제안한다.

[0215] 2. 시계열 분석을 위한 딥러닝 기법

[0216] 2.1 RNN

[0217] 순환 신경망(Recurrent Neural Network, RNN)은 입력과 출력을 시퀀스 단위로 처리하는 시퀀스 모델로 구조는 그림11과 같다. 일반 인공신경망 모델은 은닉층에서 활성화 함수를 지난 값은 오직 출력층으로 향하지만, RNN은 은닉층에서 활성화 함수를 통해 나온 결과값을 출력층과 은닉층의 다음 입력으로 보내는 특징이 있다. 하지만 단순 RNN 모델은 시퀀스가 길어질수록 앞의 정보가 뒤로 충분히 전달되지 못하는 현상이 발생하는 단점이 있으며, 이를 장기 의존성 문제라고 한다.



[0219]

[0220] <그림 11> RNN의 구조

[0222] 은닉층의 메모리 셀은 은닉 상태 값(h_t)을 계산하기 위해 총 두 개의 가중치를 가진다. <수식 10>의 식(1)과 같이 입력층을 위한 가중치 W_x 와 이전 시점의 은닉 상태 값(h_{t-1})을 위한 가중치 W_h 이다. <수식 10>의 식(2)는 출력층을 표현한 식으로 f 는 비선형 활성화 함수 중 하나이다.

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b) \quad \dots\dots\dots (1)$$

$$y_t = f(W_y h_t + b) \quad \dots\dots\dots (2)$$

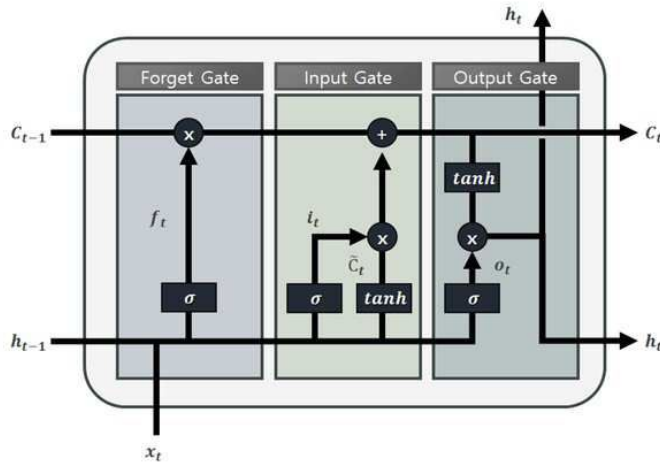
[0224]

[0225] <수식 10>

[0227] 2.2 LSTM

[0228] 장단기 메모리(Long short-Term Memory, LSTM)는 시계열 데이터를 처리하기 위한 인공지능 알고리즘으로써, RNN

의 장기 의존성 문제의 해결책으로 제시되었으며, 그림 12와 같이 입력 게이트(Input Gate), 망각 게이트(Forget Gate), 출력 게이트(Output Gate)의 3개의 게이트로 구성되어 있다. 각 게이트는 시그모이드와 쌍곡탄젠트 함수를 통해 과거와 현재의 입력 정보 중에 중요한 정보를 판단하며, 이를 이용하여 은닉층에 있는 셀스태이트(Cell state)의 정보를 제어한다.



[0230]

[0231] <그림 12> LSTM의 구조

[0233] 망각 게이트는 과거 입력 정보의 중요도를 결정하는 역할을 하며, <수식 11>과 같이 시그모이드 함수를 이용하여 0과 1 사이의 값을 셀스태이트에 보내준다.

[0235]
$$f_t = \sigma(\omega_f[h_{t-1}, x_t])$$

[0236] <수식 11>

[0238] 입력 게이트는 현재 입력된 정보의 중요도를 결정하는 역할을 하며, <수식 12>의 식(1)~(2)와 같이 현재 시점에 입력된 정보(x_t)와 이전 시점 은닉층에 저장된 정보(h_{t-1})를 시그모이드 함수와 쌍곡탄젠트 함수에 넣은 결과를 곱한다.

[0240]
$$i_t = \sigma(\omega_i[h_{t-1}, x_t]) \dots\dots\dots (1)$$

[0241]
$$\tilde{c}_t = \tanh(\omega_c[h_{t-1}, x_t]) \dots\dots\dots (2)$$

[0241] <수식 12>

[0243] 셀스태이트의 업데이트는 망각 게이트와 입력 게이트에서 결정한 과거와 현재의 입력 정보 중에 중요도에 따라 <수식 13>과 같이 수행하며, 이는 출력값에 반영된다.

[0245]
$$c_t = f_t c_{t-1} + i_t \tilde{c}_t$$

[0246] <수식 13>

[0248] 출력 게이트는 출력값(h_t)을 결정하는 역할이며, <수식 14>의 식(1)~식(2)와 같이 시그모이드 함수를 이용한다.

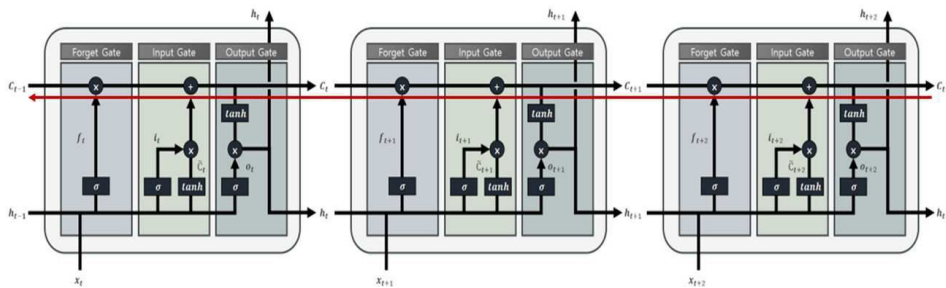
$$o_t = \sigma(\omega_o[h_{t-1}, x_t]) \dots\dots\dots (1)$$

$$h_t = o_t \tanh(C_t) \dots\dots\dots (2)$$

[0250]

[0251] <수식 14>

[0253] LSTM은 <그림 13>과 같이 셀스테이트가 학습되는 과정에서 노드와 노드만을 이용하므로 기울기 소실로 발생하는 장기 의존성 문제를 해결할 수 있다.



[0255]

[0256] <그림 13>

[0258] 3. 데이터 직렬화 양식

[0259] 3.1 XML(eXtensible Markup Language)

[0260] XML은 사람이 읽을 수 있고 기계가 읽을 수 있는 형식으로 문서를 인코딩하기 위한 일련의 규칙을 정의하여 임의의 데이터를 저장, 전송 및 재구성하기 위한 마크업 언어 및 파일 형식이다.

[0261] XML의 디자인은 목표는 단순성, 일반성 및 인터넷 전반의 유용성을 강조하며, 주요 목적은 임의의 데이터를 저장, 전송 및 재구성하는 것이다.

[0262] XML은 정보에 레이블을 지정하고 분류하고 구조적으로 구성하고 있으며, XML 태그는 데이터 구조를 나타낸다. XML 태그는 메타데이터를 포함하며, XML 표준이 지정하는 방식으로 인코딩된 데이터가 있다. 추가 XML 스키마(XSD)는 XML 해석 및 유효성 검사에 필요한 메타데이터를 정의한다.

[0263] 한편, XML은 장황함, 복잡성, 중복성에 대한 단점이 있다. XML의 기본 트리 모델을 프로그래밍 언어 또는 데이터베이스의 유형 시스템에 매핑하는 것은 어려울 수 있으며, 특히 XML이 기본 설계 목표가 아닌 응용 프로그램 간에 고도로 구조화된 데이터를 교환하는 데 사용되는 경우에는 매우 복잡하다. 따라서 JSON 및 YAML은 고도로 구조화된 데이터를 나타내는 데 중점을 둔 간단한 대안으로 자주 제안된다.

[0265] 3.2 JSON(JavaScript Object Notation)

[0266] JSON은 속성-값 쌍 및 배열로 구성된 데이터 개체를 사람이 읽을 수 있는 텍스트를 사용하여 저장하고 전송하는 개방형 표준 파일 형식 및 데이터 교환 형식이며, 서버와 웹 애플리케이션을 포함하여 전자 데이터 교환에서 다양한 용도로 사용되는 공통 데이터 형식이다.

[0267] JSON은 언어 독립적인 데이터 형식이며, JavaScript에서 파생되었지만 많은 최신 프로그래밍 언어에는 JSON 형

식 데이터를 생성하고 구문 분석하는 코드가 포함되어 있다.

[0268] XML 속성은 단일 값만 가질 수 있으며 각 속성은 각 요소에 최대 한 번 나타날 수 있는 반면, JSON은 태그 쌍을 사용하는 형태에서 간결하게 표현될 수 있다. JSON은 일반적인 상황에서 생성, 읽기 및 디코딩을 광범위하게 지원하므로 XML에 대한 대안으로 알려져있다.

[0270] 3.3 YAML(YAML ain' t markup language)

[0271] YAML은 사람이 쉽게 읽을 수 있는 데이터 직렬화 양식으로, 2001년에 클라크 에반스가 고안했고, Ingy dot Net 및 Oren Ben-Kiki와 함께 디자인했다.

[0272] YAML은 모든 데이터를 리스트, 해쉬, 스칼라 데이터의 조합으로 적절히 표현할 수 있도록 개발되었으며, 문법은 상대적으로 이해하기 쉽고, 가독성이 좋도록 디자인되어 고급 컴퓨터 언어에 적합하다.

[0273] 자바스크립트, 오브젝티브-C, 펄, PHP, 파이썬 등에서 YAML을 이용할 수 있으며, 주석 스타일을 제외하고 JSON과 매우 비슷하다. YAML 1.2 버전부터는 JSON의 완벽한 상위 호환 버전이다.

[0274] YAML은 가독성을 가장 중시하고 있으며, <그림 14>는 XML, JSON, YAML을 가독성을 비교한 것이다.

Xml	Json	Yaml
<pre><?xml version="1.0" encoding="UTF-8" ?> <root> <apiVersion>v1</apiVersion> <kind>Pod</kind> <metadata> <name>hello-pod</name> <labels> <app>hello</app> </labels> </metadata> <spec> <containers> <name>hello-container</name> <image>tmkub/hello</image> <ports> <containerPort>8000</containerPort> </ports> </containers> </spec> </root></pre>	<pre>{ "apiVersion": "v1", "kind": "Pod", "metadata": { "name": "hello-pod", "labels": { "app": "hello" } }, "spec": { "containers": [{ "name": "hello-container", "image": "tmkub/hello", "ports": [{ "containerPort": 8000 }] }] } }</pre>	<pre>apiVersion: v1 kind: Pod metadata: name: hello-pod labels: app: hello spec: containers: - name: hello-container image: tmkub/hello ports: - containerPort: 8000</pre>

[0276]

[0277] <그림 14> XML, JSON, YAML의 가독성 비교

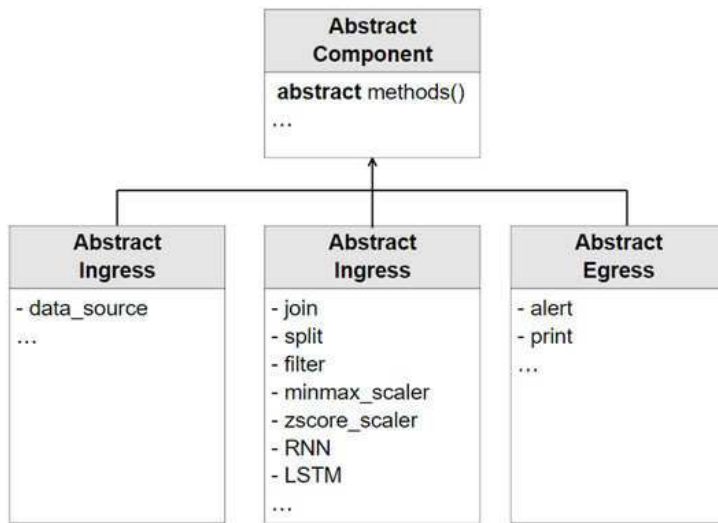
[0279] 4. AMML for workflow

[0280] AMML은 워크플로 구성언어와 딥러닝 모델을 정의하는 언어로 이루어져 있어 MLOps의 워크플로를 유연하게 구성한다. 컴포넌트는 AbstractComponent를 최상위 추상클래스로 하여 AbstractIngress, AbstractProgress, AbstractEgress로 구성되어 있다.

[0281] AbstractIngress는 데이터 소스로부터 데이터를 가져와서 다음 컴포넌트로 전달하며, AbstractIngress를 상속받아 새로운 형태의 데이터 소스를 만들 수 있다.

[0282] AbstractProgress는 이전 컴포넌트로부터 데이터를 받아 join, split, filter, minmax scaler, zscore, LSTM 등 데이터 필터링 및 변환 작업들과 예측 및 학습을 수행하며 처리된 데이터를 다음 컴포넌트로 전달한다.

[0283] AbstractEgress는 print, alert 등 음 컴포넌트로 전달할 데이터가 없는 경우, 이전 컴포넌트에서 입력받은 데이터를 처리하고 종료하는 컴포넌트이다. 아래 <그림 15>는 컴포넌트 클래스 다이어그램을 나타낸다.

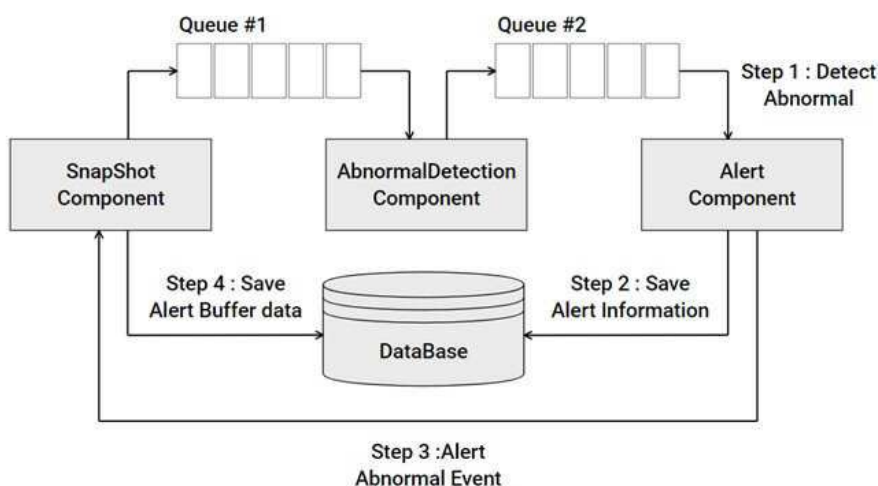


<그림 15> 컴포넌트 클래스 다이어그램

각 컴포넌트는 데이터 소스로부터 python에서 기본적으로 지원하는 key-value 형식으로 구성된 사전(dictionary) 형태의 데이터를 queue을 통해 수발신하며, 각 컴포넌트는 멀티프로세스의 형태로 독립적으로 구동된다. 데이터 소스는 RDB, NoSql 및 로그 파일 등 사전 형태로 표현할 수 있는 다양한 데이터 소스를 구현하여 사용할 수 있다.

워크플로 구성언어는 기계 학습 시스템의 필수 기능 중 하나인 모델의 이상을 탐지하고 이상 데이터를 자동적으로 수집할 수 있도록 하는 스냅샷 생성 컴포넌트를 제공한다.

스냅샷 생성 컴포넌트는 내부의 버퍼에 입력된 데이터를 저장하고 있으며, 모델의 이상을 탐지하는 경우, 데이터베이스에 버퍼의 데이터를 저장하여, 모델 재학습에 활용할 수 있도록 한다. 아래 <그림 16>은 스냅샷 프로세스를 나타낸다.

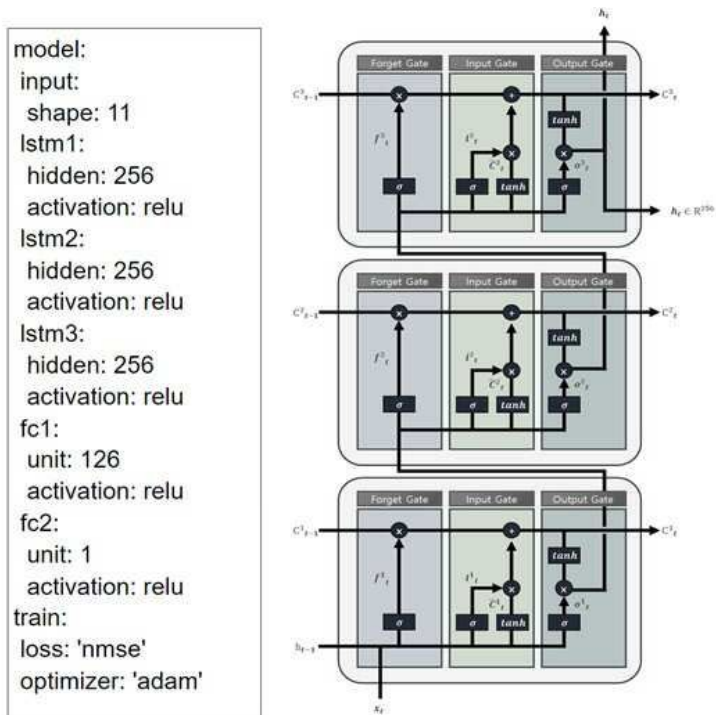


<그림 16> 스냅샷 프로세스

5. AMML for model

기계 학습 모델의 정의 구조는 모델 구조와 학습 구조로 구분하며, YAML을 이용하여 정의한다. <그림 17>은 3장

의 LSTM Layer로 구성된 모델의 정의를 나타내며, 11차원의 입력 데이터를 받아 256차원의 은닉층을 생성하는 구조이다.



<그림 17> LSTM Example of AMML for model

모델 구조는 기계 학습 모델의 레이어의 파라미터 및 활성화 함수를 설정하여 정의한다.

학습 구조는 기계 학습을 훈련하기 위해 필요한 loss 함수 및 optimizer 등의 파라미터를 설정하여 정의한다.

이렇게 정의된 기계 학습 모델의 정의 구조는 데이터베이스부에 저장되며 AbstractProgress의 LSTM 컴포넌트는 데이터베이스부에서 정의 구조를 로드하고, AbstractIngress로부터 입력데이터를 전달받아 학습을 수행한다.

6. Dataset

시계열 모델 데이터 분석 테스트 및 검증을 위해 해상교통조사로 취득한 AIS Data를 활용하였다. AIS Data는 선박 운항 모니터링 및 안전 항행 상황 파악 등 선박 운항에 필요한 중요한 데이터이며, 고유식별번호(Maritime Mobile Service Identity, MMSI), 선명, 선종, 제원 등을 담은 정적 정보와 시간(UTC), 선박의 위치, 침로, 속력 등의 동적 정보를 포함한다.

<그림 18> 및 <그림 19>는 AIS Data의 동적정보 정적정보 컬럼구조와 샘플데이터를 나타낸다.

	r_MMSI	r_Lat	r_Lon	r_Cog	r_Sog	r_Hdg	r_Time	r_StartPoint
	필터	필터	필터	필터	필터	필터	필터	필터
1	440132430	34.871752	127.782352	156.3	0.0	0	2020-07-01 09:40:54	1
2	440133050	34.906577	127.671182	109.8	0.1	37	2020-07-01 09:40:54	1
3	440133640	34.8614	127.753625	63.3	0.0	0	2020-07-01 09:40:54	1
4	440066360	34.905203	127.669607	218.2	0.1	216	2020-07-01 09:40:54	1
5	440009400	34.905533	127.669082	0.0	0.0	331	2020-07-01 09:40:54	1
6	440301490	34.894558	127.6596	233.1	5.3	288	2020-07-01 09:40:55	1
7	356417000	34.902498	127.758692	0.0	0.0	103	2020-07-01 09:40:55	1
8	440222930	34.864243	127.733418	197.7	0.1	177	2020-07-01 09:40:55	1
9	563007500	34.86543	127.779328	286.6	9.2	287	2020-07-01 09:40:55	1
10	440008790	34.754688	127.758953	52.3	0.0	0	2020-07-01 09:40:55	1

[0309]

[0310]

<그림 18> AIS Data의 동적정보 컬럼구조 및 샘플데이터

	r_MMSI	r_IMO	r_CallSign	r_Name	r_CargoType	r_AntA	r_AntB	r_AntC	r_AntD	r_EPFD	r_Dest
	필터	필터	필터	필터	필터	필터	필터	필터	필터	필터	필터
1	440196110	0	190015	NO.19 YUSUNG	80	24	8	4	4	1	
2	440147680	0	325YMH	25 YONG MA HO	52	0	0	0	0	1	
3	636016402	9655975	D5FZ8	CLEAROCEAN MARIA	80	147	36	8	24	1	MIZUSHIMA
4	440222930	0	154002	DO YANG NO.11	52	15	20	5	5	15	
5	440080030	0		NO.1 WOORI	80	20	10	2	4	15	INCHEON
6	440068620	0		HWAHAK BANGJE 2	0	0	0	0	0	15	YEOSU.KOR
7	440314690	0	652	JIN NAM	52	10	29	4	4	1	
8	538005095	9654567	V7AY7	ARDMORE ENDURANCE	80	144	39	9	23	0	HK HKG
9	564970000	9694476	9V2551	BW BOBCAT	80	149	34	6	26	1	JPW KWS >
10	440137000	9673111	DSHP9	KWANG JIN NO.17	52	14	20	5	5	1	YEOSU

[0312]

[0313]

<그림 19> AIS Data의 정적정보 컬럼구조 및 샘플데이터

[0315]

7. 해상교통 혼잡도 예측을 위한 AMML model

[0316]

해상교통 혼잡도 예측을 위한 AMML 워크플로는 학습 워크플로 및 예측 워크플로로 구성하였다. <그림 20>은 학습 워크플로의 구조를 나타낸다.

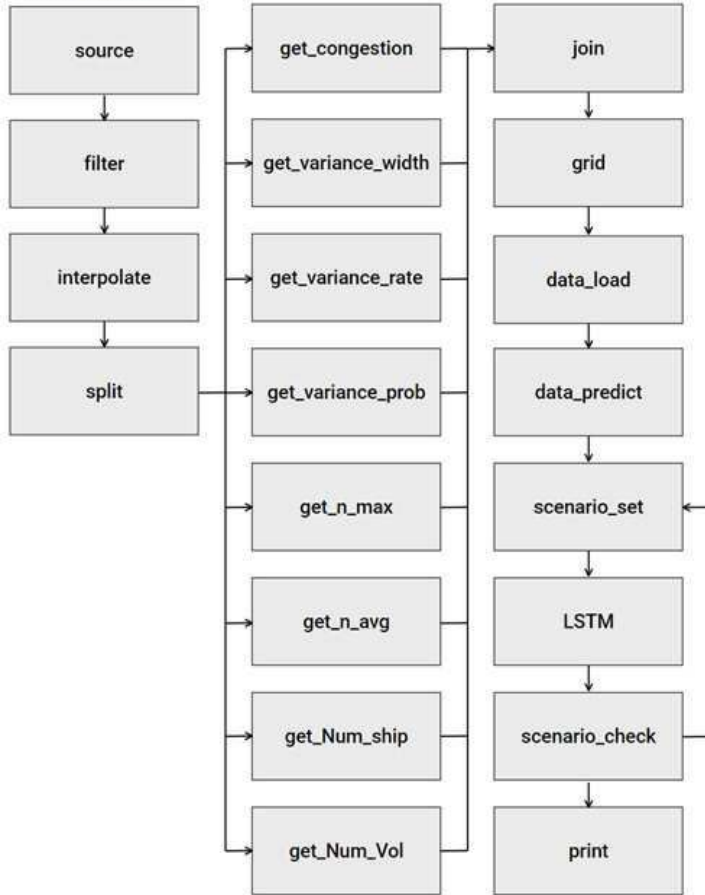
[0317]

학습 워크플로는 'AIS_con'의 데이터를 로드하여 filter 및 interpolate를 통해 전처리를 작업을 수행하고 split을 통해 데이터를 나눠서 feature를 계산한다. feature의 계산은 feature_engineering을 통해서 하며, feature_engineering에는 get_congestion, get_variance_width, get_variance_rate, get_variance_prob, get_n_max, get_n_avg, get_Num_ship, get_Num_Vol이 있으며, 각각 혼잡도, 변동폭, 변동율, 변동범위에서 움직일 확률, n일에 대한 최대값 및 평균값, 선박 척수, 선박 교통량을 구하는 컴포넌트이다.

AMML for Maritime Congestion Training Model

```
source id = "AIS_con" | filter | interpolate | split | feature_engineering | join | grid |
data_load | data_predict | scenario_set | train_LSTM
model="AIS_con:scenario##", epoch=500 | scenario_check | print
```

AMML Diagram



<그림 20> 해상교통 혼잡도 예측을 위한 학습 워크플로

각 feature의 계산이 완료되면 join을 통해서 feature를 취합하고 grid를 통해 데이터의 격자화를 수행한다. 이후 data_load를 통해 텐서 형태로 적재하고 적재되지 못한 데이터를 대상으로 data_predict를 통해 예측한다. scenario_set은 설정된 feature의 조합에 따른 시나리오에 따라 data_predict를 통해 온전한 텐서 형태의 데이터를 LSTM 컴포넌트에 입력하여 모델을 학습한다. scenario_check은 시나리오 종료 여부를 판단하여 설정된 시나리오에 따라 학습을 반복할지 print로 결과를 저장할지 결정한다. 학습된 모델 및 모델 메타 정보는 데이터베이스에 저장하여 추후 해상교통 혼잡도 예측에 사용할 수 있도록 한다.

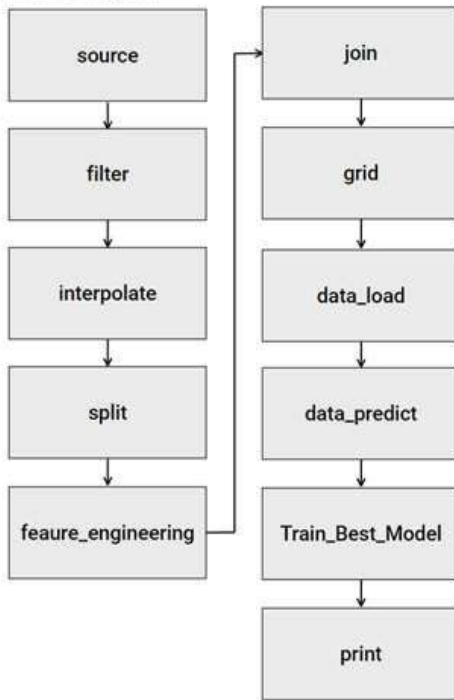
예측 워크플로는 <그림 21>과 같이 구성하였다.

예측 워크플로에서도 학습 워크플로와 마찬가지로 데이터 전처리, 데이터 격자화, 데이터 적재 및 필요시 예측을 수행한다. 이후 Train_Best_Model은 학습 워크플로에서 시나리오에 따라 학습한 모델 중 가장 성능이 좋은 모델을 요청하여 print를 통해 결과를 출력한다.

AMML for Maritime Congestion Prediction Model

```
source id = "AIS_con" | filter | interpolate | split |
feature_engineering | join | grid | data_load |
data_predict | Train_best_Model, out_cols=
[predict_congestion] | print
```

AMML Diagram



[0327]

[0328] <그림 21> 해상교통 혼잡도 예측을 위한 예측 워크플로

[0330] 8. LSTM model training and results

[0331] 해상교통 혼잡도 예측을 위하여 AMML로 만들어진 학습 및 예측 워크플로를 수행하였다. 시나리오는 Feature의 조합에 따라 <표 4>와 같이 구성하였으며, <표 5> 및 <표 6>은 시나리오에 따른 학습 결과이다. 시나리오 구성에 따른 모델 Train / Validate 결과, Congestion만을 이용한 (a)시나리오에서 Valid loss가 0.2717 나왔으나, 변동폭, 변동율, 변동범위 내에서 움직일 확률을 추가한 (b)시나리오의 경우, 0.2182로 모델의 성능이 개선되었다.

[0332] 혼잡도와 혼잡도의 3, 6, 12, 24일 평균값을 사용한 (c)시나리오와 3, 6, 12, 24일 최대값을 사용한 (d)시나리오에서는 각각 0.1490과 0.1164의 Valid loss로 앞의 두 시나리오보다 모델의 성능이 좋았으며, 특히 (d)시나리오의 경우, 상대적으로 높은 혼잡도를 잘 예측하는 경향을 보였다.

[0333] 혼잡도와 선박의 척수 및 교통량을 이용한 (e)시나리오도 Valid loss가 0.2047로 (b)시나리오와 마찬가지로 모델의 성능이 개선되는 모습을 보였으며, (f)시나리오에서 다른 Gate Line의 혼잡도를 제외한 모든 feature를 이용할 때 Valid loss 0.0842로 높은 모델의 성능을 보였다.

[0334] 다른 Gate Line의 혼잡도를 이용한 (g)시나리오에서는 Valid loss 0.1520으로, (b)시나리오와 (e)시나리오 보다 모델의 성능이 더욱 개선되는 것을 보였으며, 모든 feature를 이용한 (h)시나리오에서 Valid loss 0.0835로 모델의 성능이 가장 높은 것으로 나타났다.

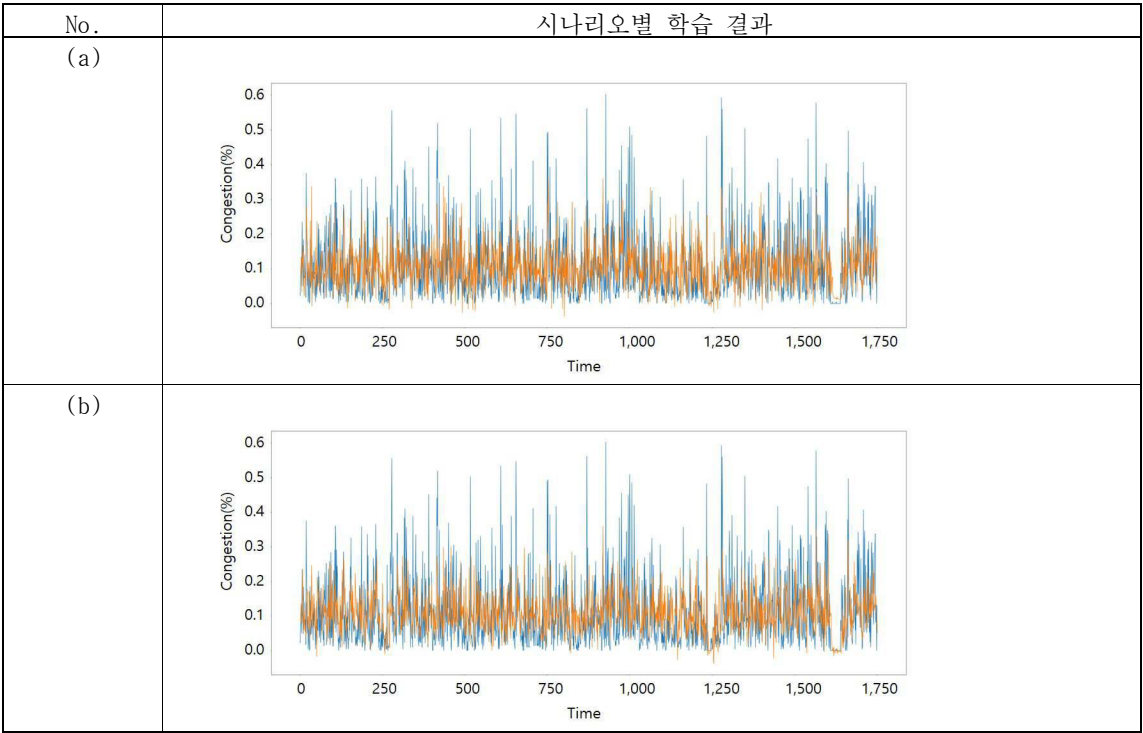
표 4

No.	Input		Output
	Trend	Prediction	
(a)	Congestion	—	Congestion
(b)	Congestion	—	
	Variance width		
	Variance rate		
(c)	Variance Prob.	—	
	Congestion		
(d)	3, 6, 12, 24 avg	—	
	Congestion		
(e)	3, 6, 12, 24 max	Num of ship Traffic Vol	
	Congestion		
(f)	Congestion	Num of ship Traffic Vol	
	Variance width		
	Variance rate		
	Variance Prob.		
	3, 6, 12, 24 avg		
	3, 6, 12, 24 max		
(g)	—	Congestion(B gate) Congestion(C gate)	
(h)	Congestion	Number of ship Traffic Volume Congestion(B gate) Congestion(C gate)	
	Variance width		
	Variance rate		
	Variance probability		
	3, 6, 12, 24 avg		
	3, 6, 12, 24 max		

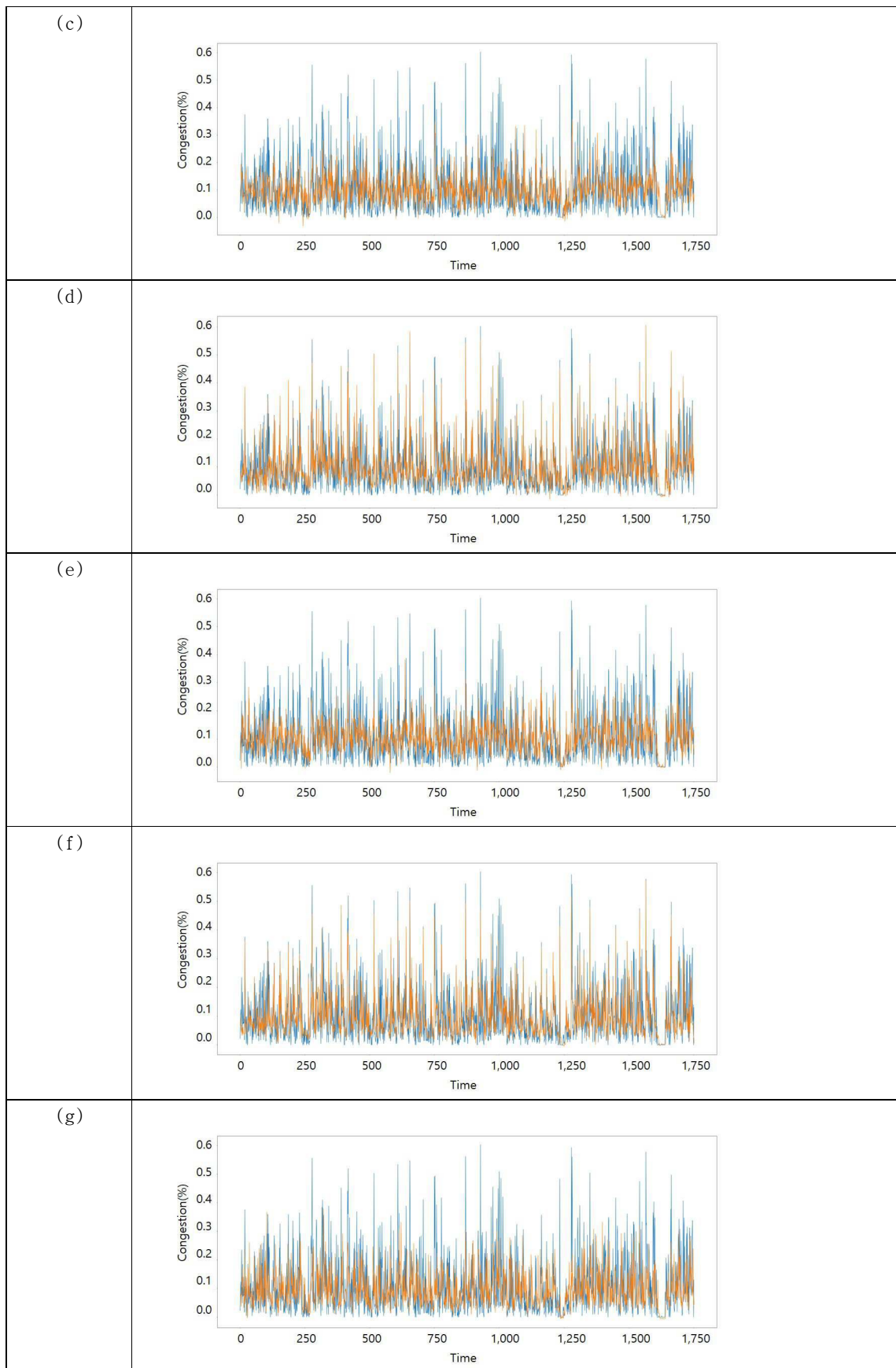
[0335]

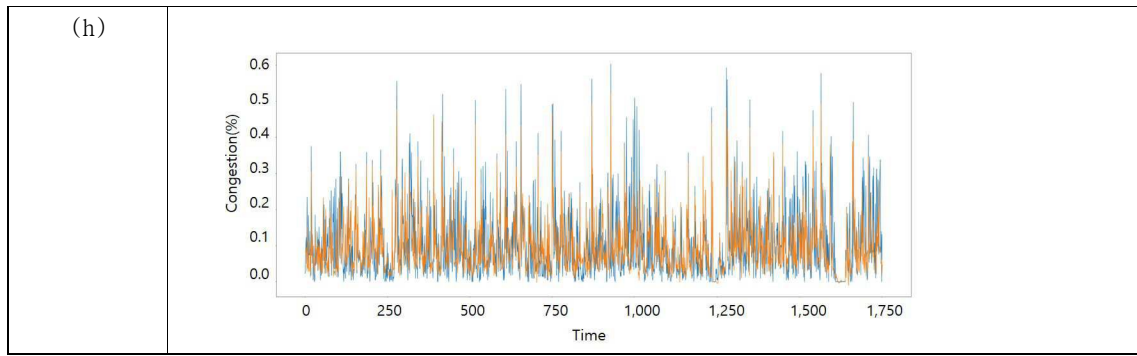
[0336] <표 4> 학습 및 실험 시나리오 구성

표 5



[0337]





<표 5> 시나리오별 학습 결과

표 6

No.	Train loss	Valid loss
(a)	0.2094	0.2717
(b)	0.1574	0.2182
(c)	0.1205	0.1490
(d)	0.0943	0.1164
(e)	0.1491	0.2047
(f)	0.0647	0.0842
(g)	0.1116	0.1520
(h)	0.0677	0.0835

<표 6> 시나리오별 Train loss 및 Valid loss

최대한 간결하게 워크플로 파이프라인을 구성하고 설정을 최소화하여 사용성을 높인 AMML을 이용하여 다차원 텐서를 이용한 해상교통 혼잡도 예측 모델을 개발하고 적용하였다. 제안하는 시스템(1000)은 데이터 전처리 및 수집, AI 모델 학습, AMML을 이용한 운영 환경 배포 등을 지원한다.

9. Conclusions

기계 학습 시스템은 모델을 개발하고 적용하여, 변화하는 데이터 환경에 따라 지속적으로 모델을 학습 및 개선하는 것을 말한다. MLOps는 이러한 과정들을 자동화하여 업무 효율성을 높이고 신뢰도 높은 기계 학습 모델을 개발 및 적용할 수 있다.

본 명세서에서는 AMML을 사용한 MLOps를 제시하여 전술한 전처리부(100)의 해상교통 데이터의 이상치 제거 및 결측치 처리, 적재부(200)의 데이터 예측 및 적재, 학습 및 예측부(300)를 통해 텐서 형태로 적재된 해상교통 데이터를 기반으로 해상교통 혼잡도를 예측하였다.

본 발명에 따른 데이터베이스부는 데이터베이스 관리 시스템(이하, DBMS)을 포함할 수 있다. DBMS는 다수의 사용자들이 데이터베이스부 내의 데이터에 접근할 수 있도록 해주는 소프트웨어 도구의 집합이다. DBMS는 IMS, CODASYL DB, DB2, ORACLE, INFORMIX, SYBASE, INGRES, MS-SQL, Objectivity, O2, Versant, Ontos, Gemstone, Unisql, Object Store, Starburst, Postgres, Tibero, MySQL 또는 MS-access등을 포함할 수 있다. DBMS는 특정 명령어의 입력에 따라 해당하는 특정 데이터에 접근하는 것이 가능하다.

본 명세서에서 전처리부(100), 적재부(200), 그리고 학습 및 예측부(300)는 메모리에 저장된 연속된 수행과정들을 실행하는 프로세서들일 수 있다. 또는, 프로세서에 의해 구동되고 제어되는 소프트웨어 모듈들로서 동작할 수 있다. 나아가, 프로세서는 하드웨어 장치일 수 있다.

참고로, 본 발명의 일 실시예에 따른 다차원 텐서를 이용한 해상교통 혼잡도 예측 방법은 다양한 컴퓨터 수단을

통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독가능매체에 기록될 수 있다. 상기 컴퓨터 판독가능매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용가능한 것일 수도 있다. 컴퓨터 판독가능매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광매체, 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함될 수 있다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급언어코드를 포함한다. 상술한 하드웨어 장치는 본 발명의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0351] 본 명세서에 있어서 '부(部)'란, 하드웨어에 의해 실현되는 유닛(unit), 소프트웨어에 의해 실현되는 유닛, 양방을 이용하여 실현되는 유닛을 포함한다. 또한, 1개의 유닛이 2개 이상의 하드웨어를 이용하여 실현되어도 되고, 2개 이상의 유닛이 1개의 하드웨어에 의해 실현되어도 된다.

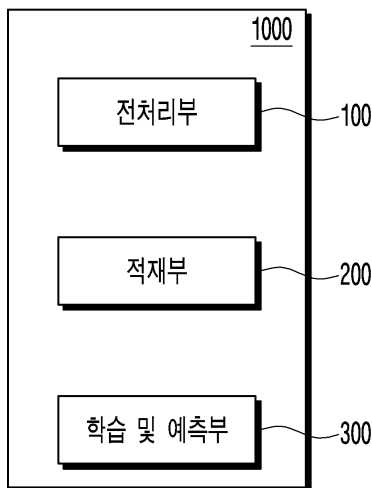
[0352] 본 발명의 보호범위가 이상에서 명시적으로 설명한 실시예의 기재와 표현에 제한되는 것은 아니다. 또한, 본 발명이 속하는 기술분야에서 자명한 변경이나 치환으로 말미암아 본 발명이 보호범위가 제한될 수도 없음을 다시 한 번 첨언한다.

부호의 설명

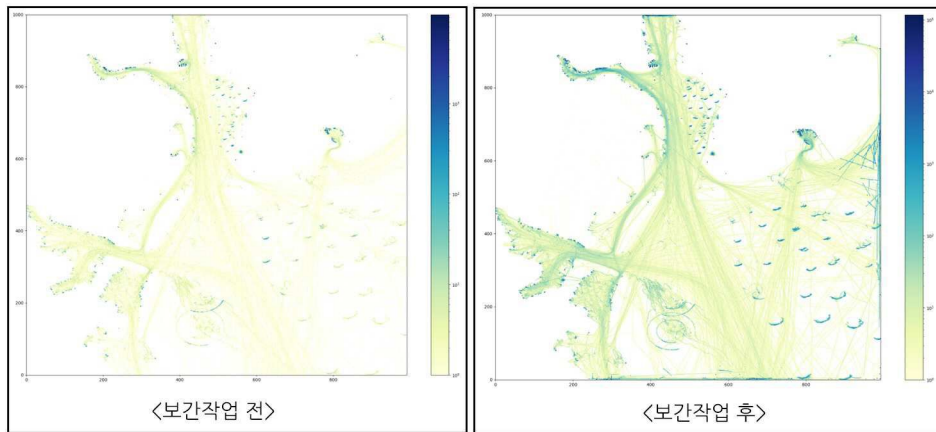
- [0353] 100 : 전처리부
200 : 적재부
300 : 학습 및 예측부
1000 : 다차원 텐서를 이용한 해상교통 혼잡도 예측 시스템

도면

도면1



도면2



도면3

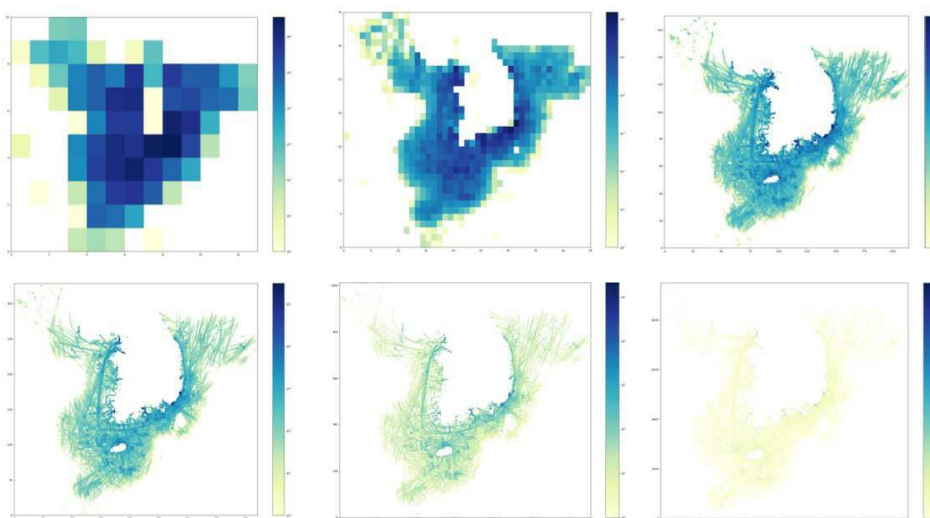
격자의 단계별 크기

항목	단계	Interval		참조대상	활용
		도	km		
수평 해상도 (6단계)	1	1"	약 100km	전자해도 KR3	서비스용
	2	15'	약 25km	전자해도 KR5 측량원도 1/5만	
	3	3'	약 5km	전자해도 KR6	
	4	1'30"	약 2.5km	측량원도 1/5,000	
	5	30"	약 1km	수치조류도	
	6	3"	약 100m	조석보정체계, 국가지점번호 등	연구용

※ 격자번호의 명명규칙은 전자해도 명명규칙을 준용하여 부여되었음
 ※ 연구용데이터는 사용 승인된 사용자에게 제공됩니다.

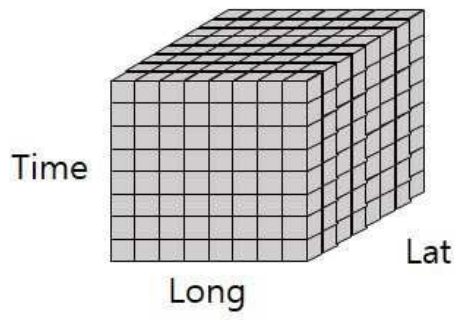
〈국립해양조사원의 격자형 해양정보관리체계〉

도면4



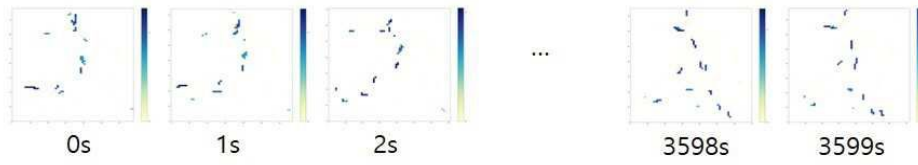
〈데이터 격자화 예시〉

도면5



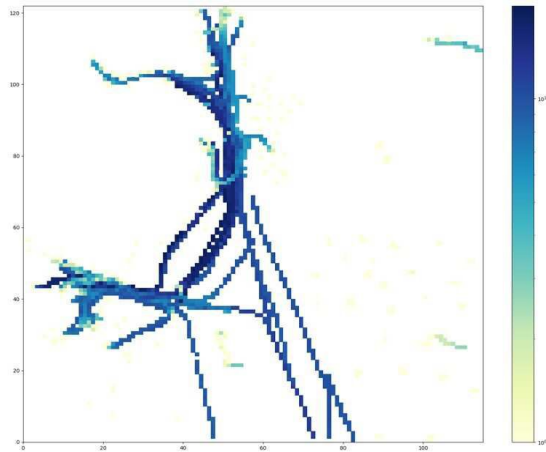
〈데이터 적재 개념도〉

도면6



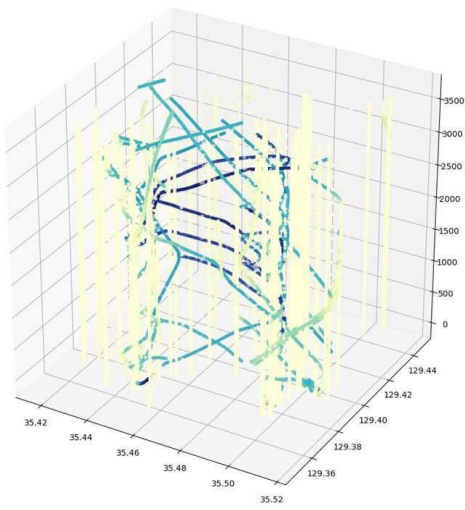
〈각 1초의 데이터 예시〉

도면7



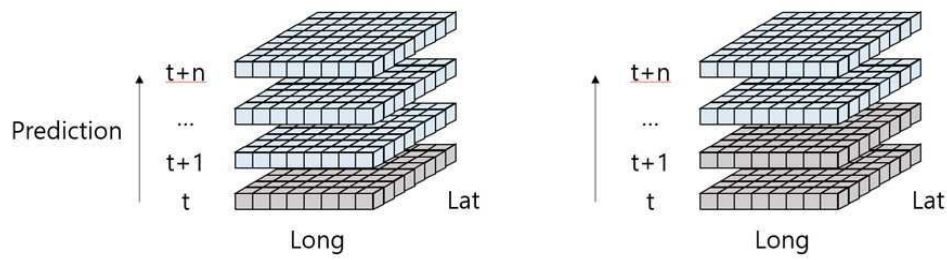
〈각 1초의 데이터 누적 예시〉

도면8



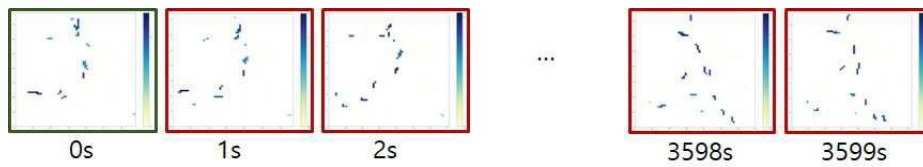
〈텐서 형태 데이터 적재 예시〉

도면9



〈데이터 예측의 개념도(t까지 실제 데이터 이후, t+1까지 예측)〉

도면10



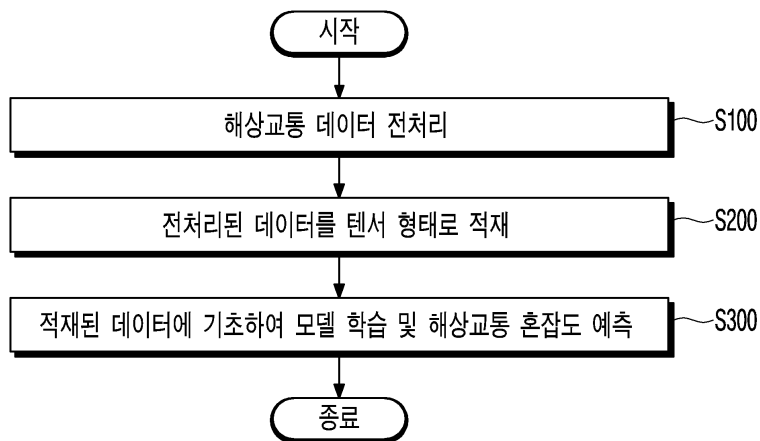
〈데이터 예측 예시(1)〉

도면11



〈데이터 예측 예시 (2)〉

도면12



도면13

