

Deep Learning Research on Vessel Trajectory Prediction Based on AIS Data with Interpolation Techniques

Won-Hee Lee*, Seung-Won Yoon*, Da-Hyun Jang*, Kyu-Chul Lee*

*Student, Dept. of Computer Science, Chung-nam National University, Daejeon, Korea

*Student, Dept. of Computer Science, Chung-nam National University, Daejeon, Korea

*Student, Dept. of Computer Science, Chung-nam National University, Daejeon, Korea

*Professor, Dept. of Computer Science, Chung-nam National University, Daejeon, Korea

[Abstract]

The research on predicting the routes of ships, which constitute the majority of maritime transportation, can detect potential hazards at sea in advance and prevent accidents. Unlike roads, there is no distinct signal system at sea, and traffic management is challenging, making ship route prediction essential for maritime safety. However, the time intervals of the ship route datasets are irregular due to communication disruptions. This study presents a method to adjust the time intervals of data using appropriate interpolation techniques for ship route prediction. Additionally, a deep learning model for predicting ship routes has been developed. This model is an LSTM model that predicts the future GPS coordinates of ships by understanding their movement patterns through real-time route information contained in AIS data. This paper presents a data preprocessing method using linear interpolation and a suitable deep learning model for ship route prediction. The experimental results demonstrate the effectiveness of the proposed method with an MSE of 0.0131 and an Accuracy of 0.9467.

▶ **Key words:** Trajectory Prediction, Interpolation, Deep Learning Model, AIS Data, Vessel

[요 약]

해상 운송의 대부분을 차지하고 있는 선박의 경로를 예측하는 연구는 해상의 위험을 사전에 탐지하여 사고를 예방할 수 있다. 도로와 달리 해상에는 신호체계가 따로 존재하지 않고, 교통 관리가 어렵기에 해상 안정성을 위해 선박 경로 예측은 필수적이다. 그러나 선박의 경로 데이터셋의 시간 간격은 통신 장애로 인해 불규칙하다. 본 연구는 이 문제를 해결하기 위해 선박 경로 예측에 적합한 보간법을 사용하여 데이터의 시간 간격을 조정하는 방법을 제시한다. 또한, 선박의 경로를 예측하기 위한 선박 경로 예측 딥러닝 모델을 개발하였다. 본 연구의 모델은 선박의 실시간 경로 정보를 담고 있는 AIS 데이터를 통해 선박의 이동패턴을 파악하여 이후에 위치할 선박의 GPS 좌표를 예측하는 LSTM 모델이다. 본 논문은 선형 보간법을 사용한 데이터 전처리 방법과 선박 경로 예측에 적합한 딥러닝 모델을 제시하고, 실험을 통해 MSE 0.0131, Accuracy 0.9467로 본 논문에서 제시하는 방법의 예측 성능이 우수함을 나타낸다.

▶ **주제어:** 경로 예측, 보간법, 딥러닝 모델, AIS 데이터, 선박

- First Author: Won-Hee Lee, Corresponding Author: Kyu-Chul Lee
- *Won-Hee Lee (k334lwh@gmail.com), Dept. of Computer Science, Chung-nam National University
- *Seung-Won Yoon (yoonenoch11@gmail.com), Dept. of Computer Science, Chung-nam National University
- *Da-Hyun Jang (dahyunjang11@gmail.com), Dept. of Computer Science, Chung-nam National University
- *Kyu-Chul Lee (kcllee@cnu.ac.kr), Dept. of Computer Science, Chung-nam National University
- Received: 2023. 12. 01, Revised: 2024. 03. 13, Accepted: 2024. 03. 13.

I. Introduction

선박을 이용한 해상 운송은 세계 무역의 최대 90%를 차지하고 있는 중요한 운송수단이다. 많은 화물 운송이 선박을 통해 이루어지고, 선박 관련 기술이 점차 발전하면서 해상에 선박의 밀도가 증가하고 해상 교통이 복잡해졌다 [1]. 이로 인해 해상에서의 교통사고가 빈번해지고, 항해의 위험이 증가하였다. 따라서 해상에서 발생할 수 있는 사고를 미리 탐지하고 예방하기 위해 선박 경로 예측의 중요성이 커지고 있으며, 경로 예측과 관련된 연구들이 활발히 진행되고 있다. 또한, 머신러닝 또는 딥러닝 프레임워크와 같은 인공지능 기술을 이용해 사람의 판단이나 능력을 사용하지 않고 무인으로 선박을 운행하는 자율운행 선박 기술이 발전하고 있는 추세이다. 이 기술은 실시간으로 최적의 경로를 판단하여 선박 운영 비용을 감소시키고 해상 사고를 예방할 수 있다는 장점이 있다. 선박 경로 예측 연구를 통해 자율운행 선박 시장의 발전에 도움이 될 수 있다.

선박 경로 예측을 수행하기 위해 일반적으로 사용되는 AIS(Automatic Identification System)는 일정한 시간 간격으로 선박의 위치 및 운항 정보를 전송(broadcast)하여 주변의 다른 선박이나 육상 기지국에서 선박의 운항 상황을 자동으로 탐지할 수 있는 시스템이다. 송수신되는 정보들은 크게 선박과 관련된 정적정보와 항해함에 따라 변화하는 동적 정보로 구분할 수 있다. 정적정보에는 선박의 고유 식별번호 값인 MMSI 번호, 선박의 종류, 길이, 폭, 너비와 같은 정보를 담고 있고, 동적 정보에는 실시간 선박의 위치(위도, 경도), 방향, 속도 등과 같은 정보를 담고 있다. 이처럼, AIS 데이터는 다량의 선박 궤적 데이터를 얻을 수 있고, 선박의 이동 특성을 반영할 수 있는 추가적인 정보들을 얻을 수 있어 선박 경로 예측에 유용한 데이터이다. 본 연구에서는 GPS 좌표를 사용하여 선박의 경로를 예측하므로 Timestamp, 경도, 위도의 데이터를 사용한다.

실시간으로 항해 중인 선박에서 정보를 수신하는 방식인 AIS는 통신 장애에 취약하다. 일정한 간격으로 선박에서 정보를 송신하더라도 통신 상황에 따라 온전히 정보를 수신받지 못하여 저장되지 않는 경우가 있다. 이러한 상황으로 인해 데이터 간의 간격이 불규칙적인 문제가 발생한다. 그러나 본 연구에서 사용하는 딥러닝 모델은 위도와 경도만을 사용하여 학습하므로 모델이 정확한 시간 정보를 학습할 수 없다. 따라서 딥러닝 모델을 통하여 선박 경로를 예측하기 위해서는 일정한 간격을 가진 시계열 데이터를 구축하는 것이 필수적이다. 그러나 기존 선박 경로 예측 연구들은 전처리 방법보다 학습 기법 및 딥러닝 아키텍처 개선 방법에 초점을 두었으며, 전처리 방식은 데이터

스케일링 등의 간단한 전처리 방식만을 채택하거나 보간법을 사용하더라도 명확한 보간 간격 기준을 제시하지 않는다. 본 연구에서는 이를 보완하기 위해 보간법을 사용한 데이터 전처리 방식을 제안하여 AIS 데이터셋의 문제를 해결하고자 한다.

본 연구에서는 효과적인 데이터 전처리를 위해 보간 방법을 기반으로 한 데이터 전처리 방법을 제안한다. 해당 전처리 과정은 전처리를 수행하기에 앞서 보간을 위한 데이터 간격의 기준을 정하고 선형 보간법과 스플라인 보간법을 수행한다. 위 보간법들은 선형적 및 곡선 함수 방식으로 계산하여 값을 추정하는 방법으로 데이터 간의 시간 간격이 기준 시간 간격보다 클 때, 그 간격을 기준 간격에 알맞게 지정하여 보간하기에 적합한 방법이다. 본 연구에서는 선형 보간법[2]과 스플라인 보간법[3]을 적용하여 전처리된 데이터를 사용해 실험을 진행하였고, 결과적으로 본 연구에서 제안한 전처리 방식을 적용한 데이터를 사용했을 때 예측 성능이 더욱 좋다는 것을 제시하였다.

AIS 데이터는 실시간으로 항해 정보를 송수신하기에 시간의 흐름을 갖는 시계열 데이터이다. 시계열 데이터는 시간의 순서가 중요하기 때문에 여러 시계열 예측 딥러닝 모델 중, 시간의 흐름을 저장하고 예측에 반영할 수 있는 LSTM[4](Long Short-Term Model) 기반 모델을 사용하였다. LSTM은 RNN(Recurrent Neural Network) 모델의 한계인, 은닉층의 과거의 정보가 마지막까지 전달되지 못하는 장기 의존성 문제를 해결하여 시계열 데이터 처리에 특화된 모델이다. 또한, 딥러닝 모델 학습을 위한 학습 데이터 단위인 look_back과 먼 지점의 위치까지 예측하는 forward_length를 설정하여 학습을 진행하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 AIS 데이터를 활용한 선박 경로 예측 관련 연구에 대해 설명한다. 3장에서는 선형 보간법, 스플라인 보간법을 사용한 데이터 전처리 과정과 본 연구에서 제안하는 딥러닝 모델을 설명하고 look_back, forward_length, 그리고 두 가지 보간법 유무에 따른 모델의 성능을 제시한다. 마지막으로 4장에서는 결론과 향후 연구 방향에 대해 제시한다.

II. Preliminaries

1. Theoretical background

1.1 Interpolations

AIS 데이터셋은 해상에서 수집되는 데이터이기 때문에 통신 상황이 불안정한 상황에서는 데이터가 일정하게 수집되지 않아 결측값들이 존재하는 단점이 있다. 이는 데이

터들의 시간 간격을 불규칙하게 만들고 딥러닝 학습 성능을 저하할 가능성이 있는 요인 중 하나이다. 결측값으로 인한 모델 성능의 저하를 해결하기 위해 본 연구는 AIS 데이터의 시간 간격을 맞추어 줄 수 있는 선형 보간법과 스플라인 보간법을 사용하였다.

선형 보간법은 알려진 두 점 사이의 값을 추정하기 위하여 직선거리에 따라 선형적으로 계산하는 방법이다. Fig. 1 에서, 주어진 (x_0, y_0) 와 (x_1, y_1) 내의 위치한 (x', y') 의 좌표는 확실하게 알 수 없다. 이를 추정하기 위하여 임의의 x' 좌표를 정하고 비례식을 사용하여 y' 의 좌표를 추정할 수 있다.

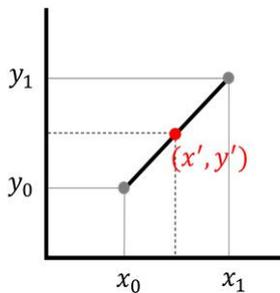


Fig. 1. Linear interpolation

스플라인 보간법은 주어진 점들 사이의 값을 추정하기 위해 곡선의 함수를 계산하는 방법이다. 본 연구에서는 3차 다항식을 사용하는 3차 스플라인 보간법을 사용한다. Fig. 2 에서, (x_{i-1}, y_{i-1}) , (x_i, y_i) , (x_{i+1}, y_{i+1}) , (x_{i+2}, y_{i+2}) 의 좌표는 주어져 있고, 각 인접한 두 점을 통과하는 다항식을 계산하여 3차 다항식인 $f_{i-1}(x)$, $f_i(x)$, $f_{i+1}(x)$ 을 구할 수 있다. 이 다항식들은 각 스플라인 구간의 끝점을 정확하게 통과해야하며 인접하는 모든 다항식들은 해당 지점에서 연속성을 가져야 한다. 이를 통해 곡선이 부드럽게 이어지도록 한다.

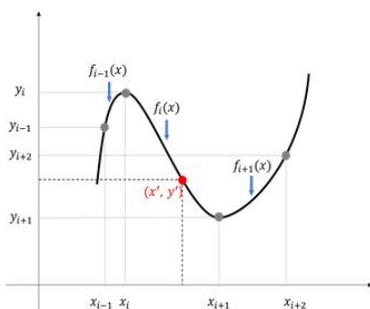


Fig. 2. Spline interpolation

1.2 Deep Learning Model

본 연구에서 사용하는 시계열 데이터는 시간의 경과와 함께 일정한 간격마다 관측값이 기록된 데이터이다. 시계열 데이터는 과거의 데이터에 종속적이기 때문에 시계열 데이터를 예측하기 위해서는 이전 시간 간격들의 정보를 많이 학습할 수 있어야 한다. 따라서 이러한 시계열 데이터 예측에 최적화된 딥러닝 모델인 LSTM(Long Short-Term Model) 모델을 개발하였다. RNN 모델은 시퀀스를 가지고 있는 시계열 데이터를 처리할 때 효과적인 모델이지만 시퀀스의 길이가 길어질수록 중요한 정보를 가진 이전의 데이터를 고려하지 못하게 되어 장기 의존성 문제가 발생한다. 이를 해결하기 위해 Fig. 3 에서 RNN의 hidden state에 cell-state를 추가하여 이전의 데이터를 저장하여 장기 의존성 문제를 개선한 모델인 LSTM 모델을 사용하였다.

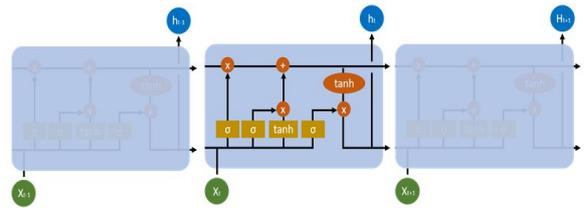


Fig. 3. LSTM Architecture

LSTM 모델의 핵심은 cell-state이다. Fig. 4 처럼, 부수적인 선형 연산들을 거치고 전체 체인을 관통한다. 이러한 구조로 인해 정보들이 지속적으로 다음 단계에 전달되게 된다. LSTM은 이 cell-state를 보호하고 컨트롤하기 위한 3가지 게이트들로 이뤄져 있다. 먼저 cell-state에서 어떤 정보를 버릴지 선택하는 과정인 forget gate가 있다. 이는 시그모이드 활성화 함수에 의해 결정되는데 값이 1이면 모든 정보를 보존, 0이라면 모든 정보를 버리라는 의미이다. 다음은 Fig. 5처럼 input gate를 통해 앞으로 들어오는 새로운 정보들 중 어떤 것을 cell-state에 저장할 것인지를 정하는 것이다. 먼저 시그모이드 활성화 함수가 어떤 값을 업데이트할지 정하고 그 다음 tanh 활성화 함수가 새로운 후보 값들인 C_t 라는 벡터를 만들어 cell-state에 더한다. 이후 Fig. 6 처럼 이전 cell-state(C_{t-1})를 새로운 cell-state(C_t)로 업데이트하는 과정을 거쳐 출력값을 결정하여 output gate를 통해 도출한다.

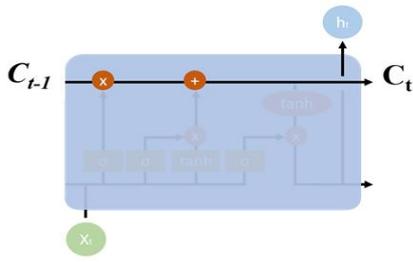


Fig. 4. Cell-state of LSTM

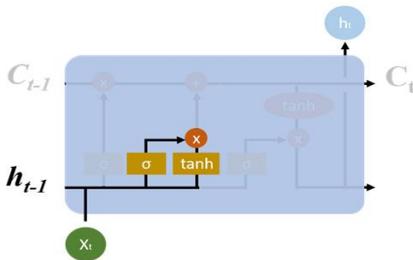


Fig. 5. Input gate of LSTM

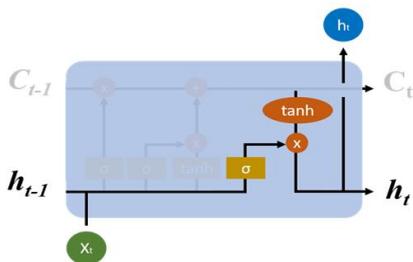


Fig. 6. Output gate of LSTM

2. Related works

경로 예측 연구는 보행자 경로 예측, 차량 경로 예측과 같은 다양한 분야에서 진행되고 있다. 그리고 연구 방법 또한 확률을 기반으로 하는 전통적인 경로 예측 방법에서 딥러닝 기반 예측으로 방법이 발전해왔다.

예측 분야에서도 머신러닝과 딥러닝을 적용한 연구들이 많아졌다. 딥러닝을 활용한 선박 연구 예측들은 주로 AIS 데이터셋을 사용한다. 이 AIS 데이터셋은 시간 간격이 불규칙적인 특징이 있어 이를 보완하기 위해 데이터 전처리 과정에서 보간법을 사용하는 연구들이 제시되었다. 루프 구조를 통해 신경망이 은닉 상태에 과거의 정보를 저장하고 시퀀스를 연산하는 방식인 RNN(Recurrent Neural Network)은 시계열 데이터 예측에 강점을 지니고 있어 경로 예측 분야에 주로 사용되었다. [5]에서는 실시간 선박 경로를 예측하기 위한 딥러닝 프레임워크와 GRU(Gate Recurrent Unit) 모델을 사용하는 방법을 제안하였다. DBSCAN(Density-Based Spatial Clustering of Applications with Noise) 알고리즘을 적용하여 주요 경

로 궤적을 도출한 후, 중복 데이터를 제거하고 들어오는 궤적을 최적화하는 전처리 과정을 거치고 선박 궤적 예측을 위해 순환 신경망을 적용하고 연속적으로 훈련을 시키는 연구이다. [6]는 Bi-GRU(Bidirectional-GRU)를 사용하여 과거의 경로 정보와 미래의 경로 정보를 양방향으로 학습하여 예측하는 방법을 제안하였다. 또한, 스플라인 보간법을 사용하여 데이터의 결측값을 보완하였지만, 보간법을 위한 시간 간격을 지정할 때 원본 데이터의 시간 간격과 흐름을 고려하지 못한 한계점이 있다. 이후 LSTM을 통해 RNN의 단점인 장기 의존성 문제가 일부 해결이 되며 성능이 크게 개선되었고 이를 활용한 연구들이 활발해졌다. [7]에서는 인코더-디코더 RNN에 기반한 선박 경로 예측 모델을 제안하였다. 또한, 선형 보간법을 사용하여 데이터 전처리를 진행하였으나 보간을 진행하기 위한 명확한 시간 간격이 제시되어있지 않았다. [8]에서는 데이터 전처리 방법에서 120초의 시간 간격 기준으로 선형 보간법을 사용했지만 해당 시간 간격 기준에 대한 설명은 나타나 있지 않았다. 또한 선형 보간법 외에도 스플라인 보간법을 사용한 연구들이 진행되었지만[9][10], 보간을 위한 시간 간격이 제시되어 있지 않았다. [11]에서는 3층의 LSTM layer를 활용하여 선박 경로 예측에서 LSTM 모델의 성능이 우수함을 나타내지만, 보간법과 같은 추가적인 데이터 전처리 기법을 사용하지 않았다는 한계점이 존재한다.

본 연구에서는 위의 관련 연구들을 보완하여 보간법을 진행할 때 적용할 보간 시간 간격과 해당 간격을 선정한 기준을 제시한다. 또한 선형 보간법과 스플라인 보간법을 적용하는 방법을 명확하게 설명한다. 마지막으로 시계열 예측 모델들 중, RNN의 단점을 보완한 LSTM 모델을 활용하여 선박 경로 예측을 진행한다.

III. The Proposed Scheme

1. Dataset

1.1 Dataset Description

본 연구는 Fig. 7 에 나타난 것처럼 2022년 1월 6일부터 2022년 1월 15일까지 south China sea를 향한 AIS 데이터를 활용하여 연구를 진행하였다[12]. 해당 지역은 세계 해운의 3분의 1이 통과하여 선박 이동 양이 많고, 지역 특성 상 섬들과 암초들이 많아 해상사고가 빈번한 곳이다. 따라서 해당 지역의 선박 경로 예측이 어려운 점과 더불어 데이터 내에서는 시간 간격이 불규칙적이고 두 데이터 간의 시간 간격이 최대 36분 이상 차이 나는 시계열 데

이터가 존재하기 때문에 본 연구에서는 해당 데이터셋을 선정하여 제안하는 데이터 전처리 방법의 성능을 비교하였다. 해당 데이터셋에는 MMSI(선박 식별번호), timestamp, latitude, longitude, speed 등의 다양한 선박에 관련된 세부정보들이 존재한다. 본 연구에서는 MMSI, timestamp, latitude, longitude의 4가지 정보만 사용하여 실험을 진행하였다. 원본 데이터를 MMSI 별로 데이터를 분할한 후, 시간 순서대로 정렬하는 전처리 과정을 통해 MMSI 마다 시간순으로 나열된 시계열 데이터를 생성하였다. 해당 모델 학습을 위해 선정된 데이터는 2,318개의 GPS 좌표(위도, 경도)들로 이루어진 선박 경로 데이터셋이다.

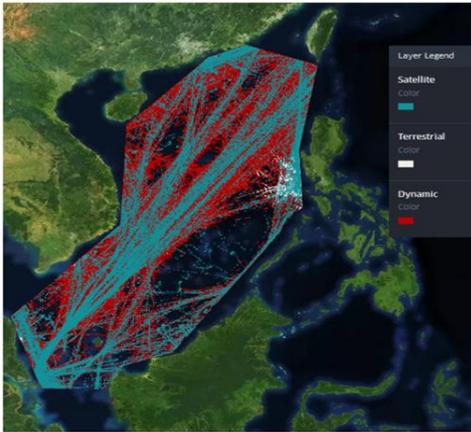


Fig. 7. AIS dataset of South China Sea

1.2 Data Preprocessing

보간법을 적용하기 위해 먼저 데이터 간의 시간 간격을 결정할 필요가 있다. 본 연구에서 사용한 데이터셋의 원본 데이터는 시간 간격이 분 단위로 이루어져 있어 간격이 매우 크고 데이터 간의 시간 간격이 불규칙적이다. 이는 앞서 설명했듯이, 딥러닝 모델이 학습 정보를 정확하게 학습하는 데 영향을 미칠 수 있다. 따라서 원본 데이터들의 시간 간격의 평균을 측정하였고, 평균 시간 간격인 5분을 기준으로 보간법을 적용하는 시간 간격을 설정하였다. 해당 기준을 통해 각 데이터 간의 시간 간격에 따라 보간할 데이터를 개수를 설정하였다. 다음은 5분의 기준 간격을 사용하여 원본 데이터에 선형 보간법을 적용하는 과정을 순서대로 나타낸 것이다. 1) 연속된 두 timestamp 데이터 간의 시간 간격이 5분 이하일 경우에는 보간법을 적용하지 않고 그대로 사용한다. 2) 데이터의 간격이 5분~10분 간격이면 선형 보간법을 사용하여 보간한 값을 한 개 추가한다. 3) 10분~15분 간격일 경우에는 2개의 보간한 값을 추가한다. 4) 15분 이상일 경우에는 3개의 보간한 값을 추가

한다. Fig. 8에서 보이듯이, 20분 이상의 데이터 개수는 다른 시간 간격 데이터들의 수에 비해 매우 적기 때문에 따로 처리하지 않는다. 결과적으로 2,318개의 원본 데이터에서 3,487개의 데이터를 구축하였다. 추가적으로 해당 방법으로 전처리를 진행한 결과를 시각화하여 Fig 9. 10. 11에 나타내었다. Fig. 9는 전처리를 진행하기 전 데이터를 시각화한 모습이다. x축은 timestamp를 나타내고 y축은 latitude를 나타낸다. longitude와 latitude를 따로 나누어 보간을 진행하였기 때문에 latitude를 사용한 예시를 대표적으로 제시하였다. Fig. 10은 선형 보간법을 적용한 후의 데이터를 시각화한 것이고, Fig. 11은 스플라인 보간법을 적용한 후의 시각화 결과이다. 본 시각화를 통해 두 가지 보간법 모두 기존 데이터의 형태를 잘 유지하면서 보간이 완료되었음을 확인할 수 있었다.

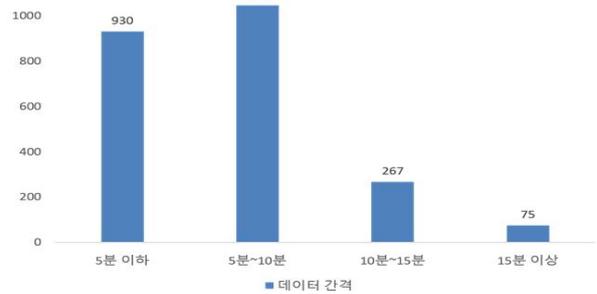


Fig. 8. Time interval distribution of the raw data

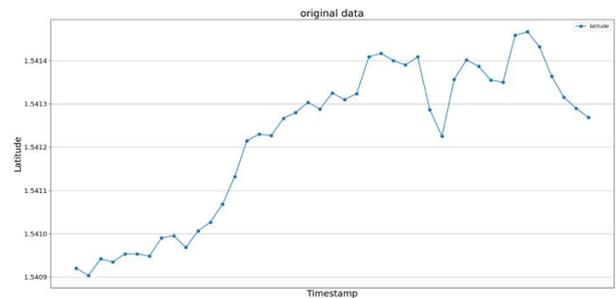


Fig. 9. Visualizing data before preprocessing

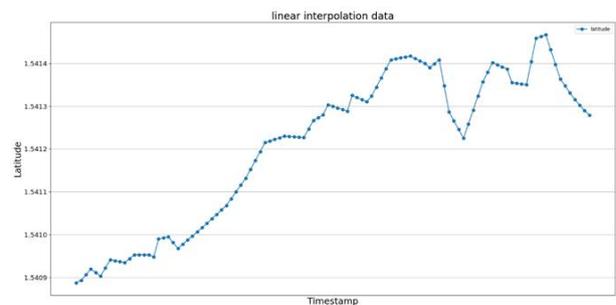


Fig. 10. After using Linear interpolation

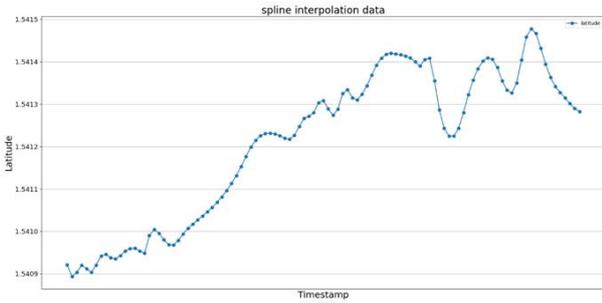


Fig. 11. After using Spline interpolation

1.3 Data Labeling

본 연구에서 개발한 선박 경로 예측 모델은 AIS 데이터의 GPS 좌표 (위도, 경도)를 입력으로 받아 다음 시간의 좌표 (위도, 경도)를 예측하는 모델이다. 훈련 시, 한 번에 모든 훈련 데이터를 통해 학습하는 것이 아닌 일정한 시간 간격을 지정해 해당 간격만큼 훈련 데이터를 학습한 후 그 다음 위치를 예측하기 위해 Look_back의 개념을 활용한 학습 방식을 사용한다. Look_back은 훈련 데이터를 특정 수만큼 묶어 하나의 훈련 데이터를 만들기 위한 단위를 의미한다. Forward_length는 훈련 데이터의 정답 위치를 특정 거리 만큼 떨어진 위치로 설정하겠다는 의미이다. 학습된 지점 바로 다음 위치를 예측하는 것이 아닌, 학습된 지점 이후 더 먼 위치를 예측하는 딥러닝 모델을 생성하기 위해 해당 개념을 적용하였다. 훈련 데이터 길이의 단위가 되는 Look_back 크기만큼의 경로 데이터를 X로 설정하고, 마지막 X부터 Forward_length 이후의 GPS 좌표를 Y로 설정하였다. Fig. 12에 이러한 과정을 거쳐 딥러닝 모델의 훈련 데이터 셋을 구성한 것을 나타냈다. 딥러닝 모델 학습이 된 이후에는 성능 평가를 위해 테스트 데이터를 생성하였는데 이 또한 훈련 데이터 생성과 동일한 방식으로 생성하였다. 테스트 데이터는 Fig. 13처럼 학습된 모델이 테스트 데이터셋의 Look_back 길이만큼의 X 데이터를 보고 Forward_length 이후의 위치를 예측하도록 하였다. 본 연구에서는 훈련 데이터와 검증 데이터, 그리고 테스트 데이터를 6:2:2 비율로 나누어 사용하였다.

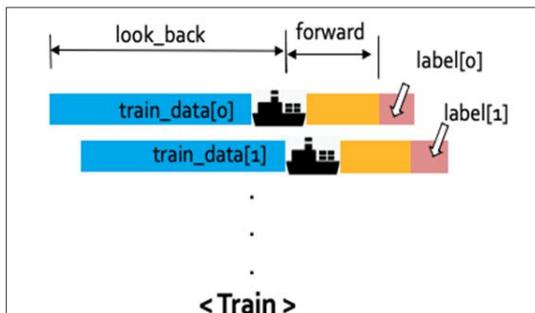


Fig. 12. Train data structure

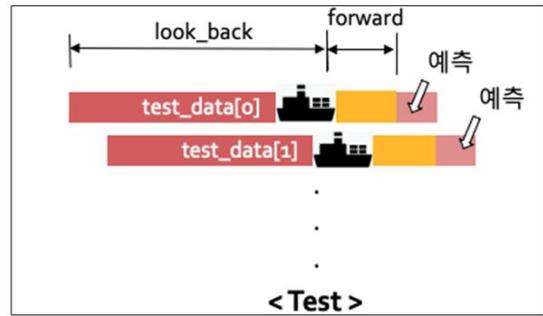


Fig. 13. Test data structure

1.4 Data Scaling

원본 데이터를 그대로 딥러닝 모델이 학습하도록 하면 학습이 느리거나 문제가 발생하게 된다. 그래서 딥러닝 모델에 데이터를 입력하기 전에 데이터 스케일링을 사용하여, 본래 데이터의 분포를 유지하면서 오버플로우나 언더플로우를 방지하고 최적화 과정에서의 안정성 및 학습 수렴 속도를 향상시킨다. 데이터 스케일링은 크게 4가지의 종류가 있다. 평균과 표준 편차를 사용하는 StandardScaler, 최대, 최소값이 각각 1과 0이 되도록 하는 MinMaxScaler[13], 최대절대값과 0이 각각 1과 0이 되도록 하는 MaxAbsScaler, 중앙값(Median)과 IQR(Interquartile range)를 사용하여 이상치(Outlier)의 영향을 최소화하는 RobustScaler[14]가 있다. 본 연구에서는 이상치가 없는 정제된 데이터를 활용하여 모델을 학습하기에 입력데이터를 0과 1 사이의 값이 되도록 하는 MinMaxScaler를 사용하여 스케일링 하였다.

2. Deep Learning Model

본 연구진은 LSTM 모델을 사용하여 선박 경로 예측 딥러닝 모델을 개발하였다. 5개의 LSTM layer를 쌓아 만든 모델을 사용하여 선박 경로 예측을 진행하였다(Fig. 14).

본 연구에서 제시하는 딥러닝 모델의 최적화 함수는 아담(Adam) 최적화 함수를 사용하였다[15]. 딥러닝에서 사용하는 최적화 기법은 손실함수(Loss Function)의 값을 최소화하는 파라미터를 구하는 기법이다. 이는 학습 속도를 빠르고 안정적이게 하는 장점이 있다. 아담 최적화 함수는 그래디언트가 커져도 스텝사이즈는 어느 정도 정해져 있어 어떤 목적함수를 사용한다고 하더라도 안정적인 최적화를 위한 하강이 가능하도록 하기에 대부분의 딥러닝 연구들에서 활용되고 있다.

손실 함수는 실제 GPS 좌표와 모델이 예측한 좌표의 거리를 최소화 하기위해 평균제곱오차(MSE)를 사용하였다 [16].

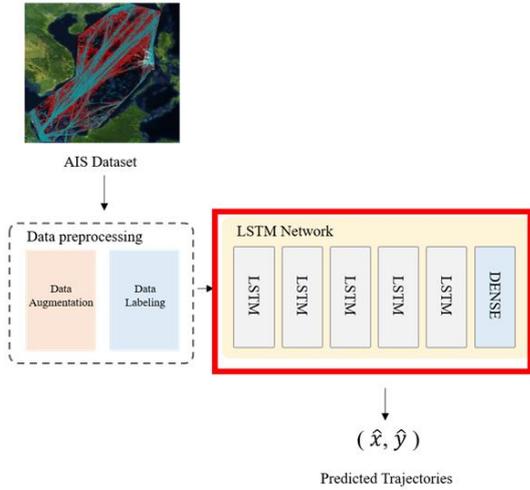


Fig. 14. Model architecture

3. Evaluation

3.1 Prediction results using AIS data with linear interpolation

데이터 전처리 과정에서 제시한 선형 보간법과 스플라인 보간법을 각각 데이터셋에 적용하여 실험을 진행하였다. 먼저 선형 보간법을 적용하기 전과 후의 성능표를 각각 제시하였다. Table 1과 Table 2는 선형 보간법을 적용하기 전 데이터를 학습한 딥러닝 모델의 성능을 나타내고, Table 3과 Table 4는 선형 보간법을 적용한 데이터를 학습한 딥러닝 모델의 성능을 제시하고 있다. 또한 최적의 Look_back 길이를 찾기 위해 Look_back 단위를 10, 30, 50으로 설정하여 실험을 진행하였고, 해당 결과는 Table 1과 Table 3으로 나타났다. Table 2와 Table 4에서는 Forward_length에 따른 성능을 제시했다. 성능 지표로는 MSE와 Accuracy를 사용하였다. MSE는 본 모델이 예측한 위치의 좌표와 실제 위치 좌표의 차이를 나타내는 지표이다. Accuracy는 MSE를 기반으로 측정하여 제시하였다.

선형 보간법을 적용하기 전의 데이터를 사용한 선박 경로 예측 모델의 최고 성능은 Look_back의 길이가 30일 때 MSE가 0.220임을 Table 1에서 확인하였다. 선형 보간법을 적용한 데이터를 사용한 모델의 최고 성능은 Look_back의 길이가 30일 때 MSE가 0.0132임을 Table 3에서 확인하였다. 이를 통해 선형 보간법을 사용한 데이터를 사용하여 딥러닝 모델을 학습시켰을 때, 성능이 크게 상승함을 확인할 수 있었고, Look_back이 30일 때 최적의 성능임을 확인하였다.

Forward_length를 통해 바로 다음 시점의 위치뿐 아니라 더 먼 경로의 위치 또한 예측을 할 수 있는지 성능을

제시하기 위해 추가적인 실험을 진행하였다. 해당 결과는 Table 2와 Table 4에 나타내었다. 보간법을 적용하지 않은 데이터를 사용한 성능을 나타낸 Table 2에서는 Forward_length가 5일 때 MSE가 0.0230이지만, 선형 보간법을 적용한 데이터를 사용한 성능을 나타낸 Table 4에서는 Forward_length가 5일 때 MSE는 0.0141로 선형 보간법을 적용한 후에 유의미한 성능 향상이 있음을 확인하였다. Forward_length가 길어질수록 예측을 해야 하는 길이가 멀어지기에 성능이 감소하는 것은 감안해야 한다. 그러나 Forward_length가 길어져도 성능이 크게 감소하지 않았음을 확인하였으며, 본 연구에서 제안하는 모델이 더 먼 곳의 위치까지 예측이 가능함을 확인할 수 있다.

최종적으로, Look_back 별로 데이터를 선형 보간하기 전과 후의 성능을 Table 5에서 비교하여 나타내었다. 해당 성능표에서 본 연구의 선박 경로 예측 모델의 최고 성능은 Look_back이 30일 때, MSE가 0.0132임을 제시하였다. 해당 결과는 선형 보간법의 사용하기 전보다 성능이 향상되었음을 나타낸다. 또한, Look_back의 길이가 30보다 길어지면 MSE가 높아지고 학습 시간이 증가함을 확인하였다. Table 4에서는 Forward_length가 커질수록 예측하는 지점이 더 멀어지기에 성능이 저하되었으나, 여전히 낮은 MSE 값을 보이며 우수한 예측 성능 모델임을 확인할 수 있었다. 이를 통해 선박의 경로 예측을 위해서는 많은 양의 이전 경로 데이터를 학습하기보다는 바로 이전의 30개 데이터를 학습함으로써 다음 위치를 예측하는 것이 가장 효율적임을 확인하였다.

Table 1. Deep learning model performance table by Look_back length without using interpolation (Forward_length = 0)

Look_back	Training Time (sec)	Accuracy	MSE
10	147.8583	0.7654	0.0225
30	265.8635	0.7800	0.0220
50	563.9958	0.7791	0.0223

Table 2. Deep learning model performance table by Forward_length without using interpolation (Look_back = 30)

Forward_length	Training Time (sec)	Accuracy	MSE
5	95.8122	0.7539	0.0230
10	101.7155	0.7307	0.0250
15	98.2854	0.7116	0.0231

Table 3. Deep learning model performance table by Look_back length using linear interpolation (Forward_length = 0)

Look_back	Training Time (sec)	Accuracy	MSE
10	86.5236	0.9249	0.0137
30	214.9492	0.9468	0.0132
50	330.9600	0.9443	0.0136

Table 4. Deep learning model performance table by Forward_length using linear interpolation (Look_back = 30)

Forward_length	Training Time (sec)	Accuracy	MSE
5	214.4679	0.9462	0.0141
10	203.8411	0.9456	0.0146
15	204.6000	0.9449	0.0161

Table 5. Performance table for different Look_back settings using w/o linear interpolation (Model = LSTM, Look_back = 30, Forward_length = 0)

Look_back	Raw dataset MSE	Linear interpolated dataset MSE
10	0.0225	0.0137
30	0.0230	0.0132
50	0.0223	0.0136

3.2 Prediction results using AIS data with spline interpolation

스플라인 보간법을 적용한 데이터를 사용한 모델의 최고 성능은 Look_back의 길이가 30일 때 MSE가 0.0131임을 Table 6에 나타냈다. 선형 보간법과 동일하게 스플라인 보간법을 사용한 데이터를 사용하여 딥러닝 모델을 학습하는 것도 성능 향상에 도움이 됨을 확인할 수 있었다. 추가로 스플라인 보간법의 적용 유무에 따른 성능 차이를 Forward_length에 따른 성능을 비교도 진행하였다. Forward_length가 5일 때 보간법을 적용하지 않은 데이터를 사용한 딥러닝 모델의 성능은 MSE가 0.0230지만, 스플라인 보간법을 적용한 데이터를 사용한 성능은 MSE가 0.0131로 스플라인 보간법을 적용한 후에 유의미한 성능 향상이 있음을 확인하였다.

스플라인 보간법을 적용하기 전과 후의 MSE 값을 비교하여 Table 8에 나타내었다. 스플라인 보간법을 적용한 데이터셋을 사용하여 딥러닝 모델을 학습했을 때, MSE 값이 감소하며 더욱 좋은 성능을 냄을 확인하였다.

Table 6. Deep learning model performance table by Look_back length using spline interpolation (Forward_length = 0)

Look_back	Training Time (sec)	Accuracy	MSE
10	227.0118	0.9380	0.0132
30	261.3782	0.9467	0.0131
50	427.5600	0.9441	0.0134

Table 7. Deep learning model performance table by Forward_length using Spline Interpolation (Look_back = 30)

Forward_length	Training Time (sec)	Accuracy	MSE
5	656.9697	0.9461	0.0134
10	616.6960	0.9454	0.0139
15	615.3431	0.9448	0.0149

Table 8. Performance table for different Look_back settings using w/o spline interpolation (Model = LSTM, Look_back = 30, Forward_length = 0)

Look_back	Raw dataset MSE	Spline interpolated dataset MSE
10	0.0225	0.0132
30	0.0220	0.0131
50	0.0223	0.0134

3.3 Performance comparison of models on AIS data using interpolation

추가적으로 LSTM 모델 외에 시계열 데이터 예측에 강점을 지니고 있는 RNN과 GRU 모델을 사용하여 성능 비교 실험을 진행하였다.

선형 보간법과 스플라인 보간법의 적용 여부에 대한 성능 비교 실험이므로 앞선 실험에서 성능이 가장 좋았던 Look_back 30, Forward_length 0으로 지정하여 실험을 진행하였다.

RNN 모델에서는 보간법을 사용하기 전 MSE가 0.0324에서 선형 보간법을 적용한 후에 0.0286, 스플라인 보간법을 적용한 후엔 0.0243으로 MSE가 감소함을 확인하였고 스플라인 보간법을 적용했을 때의 성능이 선형 보간법을 적용했을 때 보다 더 우수함을 확인하였다.

GRU 모델에서는 MSE 0.0322에서 선형 보간법을 적용 후 0.0309, 스플라인 보간법 적용 후 0.0255로 나타났다. 보간법들을 적용한 후 MSE 값이 감소하였고, RNN과 동일하게 스플라인 보간법의 성능이 더욱 우수함을 확인했다.

마지막으로 본 연구에서 제시하는 LSTM 기반 모델을 사용하여 선박 경로 예측을 진행하였을 때도 동일하게 기존 MSE 값 0.0230에서 선형 보간법을 적용했을 때 MSE의 값 0.0132, 스플라인 보간법을 적용했을 때 MSE 값 0.0131로 나타나며 보간법들의 적용이 성능 향상에 크게 기여함을 확인하였다.

본 실험을 통해 다른 시계열 데이터 예측에 강점을 지닌 모델들도 보간법을 적용하지 않았을 때 보다 선형 보간법과 스플라인 보간법을 적용했을 때 더욱 좋은 성능을 얻을 수 있었다. 또한, 모든 모델들에서 스플라인 보간법을 적용한 데이터셋을 사용했을 때의 성능이 가장 우수하였으며 스플라인 보간법을 적용하고 LSTM 기반 모델을 사용하여 선박 경로 예측을 진행했을 때, MSE가 0.0131로 가장 우수함을 Table 9에서 확인하였다.

Table 9. Comparison between additional deep learning models and LSTM model based on the application of interpolation methods (Look_back = 30, Forward_length = 0)

Model	Training Time (sec)	Accuracy	MSE
RNN	117.8025	0.6974	0.0324
RNN +Linear	215.2910	0.7806	0.0286
RNN +Spline	276.6028	0.7870	0.0243
GRU	244.6923	0.7782	0.0322
GRU +Linear	509.6406	0.8036	0.0309
GRU +Spline	640.270	0.8379	0.0255
LSTM	265.8635	0.7800	0.0230
LSTM +Linear	214.9492	0.9468	0.0132
LSTM +Spline	261.3782	0.9467	0.0131

IV. Conclusions

본 논문에서는 선박의 경로 예측을 위해 시계열 데이터인 AIS 데이터를 활용한 딥러닝 LSTM 모델을 개발하였고, 최적의 성능을 내는 파라미터를 제시하였다. 선박의 경로 예측은 다른 이동체의 경로 예측과 달리 통신 상황이 불안정하다는 특수한 특징이 있다. 그리하여 보간법이 필수적이다. 본 연구에서는 선형 보간법과 스플라인 보간법을 사용해 데이터를 전처리하는 방법을 제시하였으며, 해

당 방법이 선박 경로 예측 성능을 향상시키는데 적합함을 확인하였다. 추가로 기존 연구들에서는 명확하게 제시하지 않는, 보간법을 적용하기 위해 필요한 보간 시간 간격을 5분으로 제시하였고, 시간 간격들의 평균을 구해 5분이라는 간격을 선정하였음을 나타냈다. 또한, 본 모델의 시계열 데이터 예측에 영향을 끼치는 Look_back과 Forward_length를 사용하여 각 데이터에 따른 성능을 나타내었다. 최종적으로 본 연구의 전처리 방식과 선박 경로 예측 모델이 보다 우수한 성능을 보임을 제시하였다.

본 연구를 통해 선박의 위험상황을 미리 예측하여 해상 사고를 예방할 수 있다. 또한 무인 선박의 경로를 예측하고 최적의 항해를 도울 수 있는 연구이다. 향후에는 선박 뿐만 아니라 다양한 이동수단의 이동패턴을 파악하여 경로를 예측하는 경로 예측 모델 개발도 진행하고자 한다.

ACKNOWLEDGEMENT

This research was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-004)

REFERENCES

- [1] E. Smirni, and G. Ciardo, "Workload-Aware Load Balancing for Cluster Web Servers," IEEE Trans. on Parallel and Distributed Systems, Vol. 16, No. 3, pp. 219-232, March 2005. DOI: 10.1109/TPDS.2005.38
- [2] T. Blu, P. Thevenaz and M. Unser, "Linear interpolation revitalized," in IEEE Transactions on Image Processing, vol. 13, no. 5, pp. 710-719, May 2004, doi: 10.1109/TIP.2004.826093.
- [3] S. A. Dyer and J. S. Dyer, "Cubic-spline interpolation. 1," in IEEE Instrumentation & Measurement Magazine, vol. 4, no. 1, pp. 44-46, March 2001, doi: 10.1109/5289.911175.
- [4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [5] Suo, Y.; Chen, W.; Claramunt, C.; Yang, S. "A Ship Trajectory Prediction Framework Based on a Recurrent Neural Network." Sensors 2020, 20, 5133. <https://doi.org/10.3390/s20185133>
- [6] C. Wang, H. Ren and H. Li, "Vessel trajectory prediction based on AIS data and bidirectional GRU," 2020 International Conference on Computer Vision, Image and Deep Learning

- (CVIDL), Chongqing, China, pp. 260-264, 2020 doi: 10.1109/CVIDL51233.2020.00-89.
- [7] S. Capobianco, L. M. Millefiori, N. Forti, P. Braca and P. Willett, "Deep Learning Methods for Vessel Trajectory Prediction Based on Recurrent Neural Networks," in IEEE Transactions on Aerospace and Electronic Systems, vol. 57, no. 6, pp. 4329-4346, Dec. 2021, doi: 10.1109/TAES.2021.3096873.
- [8] K. Donandt, K. Böttger and D. Söffker, "Short-term Inland Vessel Trajectory Prediction with Encoder-Decoder Models," 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, pp. 974-979, 2022, doi: 10.1109/ITSC55140.2022.9922148.
- [9] C. Wang, H. Ren and H. Li, "Vessel trajectory prediction based on AIS data and bidirectional GRU," 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), Chongqing, China, pp. 260-264, 2020, doi: 10.1109/CVIDL51233.2020.00-89.
- [10] Murray, Brian, and Lokukaluge Prasad Perera. "An AIS-based deep learning framework for regional ship behavior prediction." Reliability Engineering & System Safety 215 ,2021. <https://doi.org/10.1016/j.res.2021.107819>
- [11] Jonghee Kim, Chanho Jung, Dokeun Kang, & Chang Jin Lee (2020). "A New Vessel Path Prediction Method using Long Short-term Memory." The transactions of The Korean Institute of Electrical Engineers, 69(7), 1131-1134, doi : 10.5370/KIEE.2020.69.7.1131
- [12] Yang, Sun, et al. "An approach to ship behavior prediction based on AIS and RNN optimization model." International Journal of Transportation Engineering and Technology 6.1 (2020): 16. <https://doi.org/10.11648/j.ijtet.20200601.13>
- [13] Shaheen, H., Agarwal, S., Ranjan, P. "Ensemble Maximum Likelihood Estimation Based Logistic MinMaxScaler Binary PSO for Feature Selection.", In: Sharma, T.K., Ahn, C.W., Verma, O.P., Panigrahi, B.K. (eds) Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing, vol 1380. Springer, Singapore, 2022 https://doi.org/10.1007/978-981-16-1740-9_58
- [14] H. Qian, Q. Wen, L. Sun, J. Gu, Q. Niu and Z. Tang, "RobustScaler: QoS-Aware Autoscaling for Complex Workloads," 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, pp. 2762-2775, 2022. doi: 10.1109/ICDE53745.2022.00252.
- [15] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, pp. 1-2, 2018, doi: 10.1109/IWQoS.2018.8624183.
- [16] Kalyan Das. Jiming Jiang. J. N. K. Rao. "Mean squared error of empirical predictor." Ann. Statist. 32 (2) 818 - 840, April 2004. <https://doi.org/10.1214/009053604000000201>

Authors



and Trajectory Prediction.

Won-Hee Lee received the B.S. and M.S. degrees in Computer Science and Engineering from Chung-nam National University, Korea, in 2022 and 2024. She is interested in Deep Learning, Machine Learning,



Computer Science, Chung-nam National University. He is interested in Deep Learning, Machine Learning, and Trajectory Prediction.

Seung-Won Yoon received the B.S. degrees in Computer Science and Engineering from Chung-nam National University, Korea, in 2018. Seung-Won Yoon is currently studying for an Intergrated PhD in the Department of



Deep Learning, Machine Learning, and Trajectory Prediction.

Da-Hyun Jang received the B.S. degrees in Computer Science and Engineering from Chung-nam National University, Korea, in 2023. Da-Hyun Jang is currently studying for a M.S degree in the Department of Computer



of Computer Science at Chung-nam National University, Daejeon, Korea, in 1989. He is currently a Professor in the Department of Computer Science, Chung-nam National University. He is interested in Database, Artificial Intelligence and Big Data.

Kyu-Chul Lee received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Seoul National University, Korea, in 1984, 1986 and 1989, respectively. Dr. Lee joined the faculty of the Department