ISSN 2383-630X(Print) / ISSN 2383-6296(Online)
Journal of KIISE, Vol. 48, No. 8, pp. 892-904, 2021. 8
https://doi.org/10.5626/JOK.2021.48.8.892

연속적인 시계열 예측을 위한 디노이징 다변량 시계열 모델링

(Denoising Multivariate Time Series Modeling for Multi-step Time Series Prediction)

홍 정 수 † 박 진 욱 † 이 지 은 †
(Jungsoo Hong) (Jinuk Park) (Jieun Lee)

김 경 훈 † 홍 승 균 † 박 상 현 ††
(Kyeonghun Kim) (Seung-Kyun Hong) (Sanghyun Park)

요 약 시계열 예측 연구 분야는 시계열 내의 주기성을 통해 미래의 시점을 예측하는 연구이다. 산업 환경에서는 미래의 연속적인 시점 예측을 통한 의사 결정이 중요하기 때문에 시계열의 연속 예측이 필요하다. 하지만 연속 예측은 이전 시차의 예측 값에 종속적이어서 불안정성이 높기 때문에 전통적인 시계열 예측은 한 시점에 대한 통계적 예측을 한다. 이를 해결하기 위해 본 연구에서는 다변량 시계열에 대해 연속적인 시점을 예측하는 인코더-디코더 기반의 'DTSNet'을 제안한다. DTSNet은 안정적인 예측을 위해 위치 인코딩을 적용한 표현형을 사용하고, 새로운 디노이징 훈련법을 제안한다. 또한, 장기 의존성을 해결하고 복잡한 주기성을 모델링하기 위해 이중 주의 기제 기법을 제안하고, 변수 별 특화 모델링을 위해 멀티 헤드 신경망을 사용한다. 본 모형의 성능 향상을 검증하기 위해 베이스라인 모형들과 비교 분석하고, 구성 요소 및 디노이징 강도 실험 등의 비교 실험을 통해 제안하는 방법론을 입증한다.

키워드: 다변량 시계열 예측, 연속 예측, 디노이징 훈련 기법, 다중 주기성, 주의 기제 기법

Abstract The research field of time series forecasting predicts the future time point using seasonality in time series. In the industrial environment, since decision-making through continuous perspective prediction of the future is important, multi-step time series forecasting is necessary. However, multi-step prediction is highly unstable because of its dependency on predicted value of previous time prediction result. Therefore, the traditional time series forecasting makes a statistical prediction for the single time point. To address this limitation, we propose a novel encoder-decoder

· 이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(IITP-2017-0-00477, (SW스타랩) IoT 환경을 위한 고성능 플래시 메모리 스토리지 기반 인메모리 분산 DBMS 연구개발)과 국토교통부의 스마트시티 혁신인재육성사업의 지원을 받아 수행된 연구임

† 학생회원 : 연세대학교 컴퓨터과학과 학생
jungsoo@yonsei.ac.kr
parkju536@yonsei.ac.kr
jieun199624@yonsei.ac.kr
kkh115505@yonsei.ac.kr
highsk88@gmail.com
†† 종신회원 : 연세대학교 컴퓨터과학과 교수(Yonsei Univ.)
sanghyun@yonsei.ac.kr
(Corresponding author임)

논문접수 : 2020년 12월 17일
(Received 17 December 2020)
논문수정 : 2021년 5월 17일
(Revised 17 May 2021)
심사완료 : 2021년 5월 18일
(Accepted 18 May 2021)

정보과학회논문지 제48권 제8호(2021. 8)

based neural network named 'DTSNet' which predicts multi-step time points for multivariate time series. To stabilize multi-step prediction, we exploit positional encoding to enhance representation for time point and propose a novel denoising training method. Moreover, we propose dual attention to resolve long-term dependencies and modeling complex patterns in time series, and we adopt multi-head strategy at linear projection layer for variable-specific modeling. To verify the performance improvement of our approach, we compare and analyze it with baseline models, and we demonstrate the proposed methods through comparison tests, such as, component ablation study and denoising degree experiment.

Keywords: multivariate time series forecasting, multi-step ahead prediction, denoising training, multiple seasonality, attention mechanism

1. 서 론

시계열 예측 연구 분야(Time Series Forecasting)는 과거 패턴에서 주요한 주기성(Seasonality)을 도출하여 미래의 한 시점(Time Step) 혹은 연속적인 시점을 예측하는 것을 주 목적으로 한다. 특히, 패턴을 분석하고 예측하여 미래에 대한 결정을 하거나, 비정상적인 상황을 감지하기 위하여 시계열 예측을 활용한다. 예를 들면, 전력 소비 규제[1]와 금융 시장[2] 및 산업 목적[3] 등과 같은 많은 응용 분야에 적용될 수 있다.

시계열 예측 연구 분야에서는 일반적으로 시계열 예측을 위해 다중의 반복적인 패턴을 수반하는 복잡한 주기성을 이용한다. 다중 패턴은 단기(Short-term) 패턴과 장기(Long-term) 패턴으로 구분될 수 있다[4]. 본 연구에서는 '단기 패턴'을 하루 내의 반복되는 단기적인 패턴으로, '장기 패턴'은 일주일 내의 일 별 주기로 나타나는 장기적인 패턴으로 정의한다. Fig. 1은 본 연구에서 사용한 'Electricity' 데이터의 예시로, 데이터 시각화를 위하여 시간당 전력 소비량 시계열 데이터를 로그 스케일하여 나타냈다. Fig. 1에서는, 하루 내의 단기 패턴뿐만 아니라 장기 패턴에 해당하는 평일(5일)과 주말(2일)의 패턴이 뚜렷하게 보이는 것을 알 수 있다. 이러한 장기 패턴과 단기 패턴은 서로 상호 연관성을 가지기 때문에, 다중의 주기성이 존재할 경우 패턴 예측이 어려워진다는 한계가 있다[5].

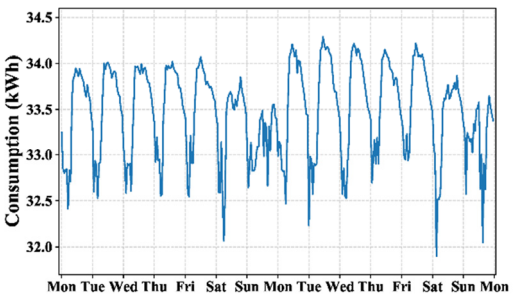


그림 1 전력 소비 환경에서의 실세계 시계열 예시

Fig. 1 Example of real-world time series on electricity consumption

또한, 시계열 예측 연구 분야는 이러한 다중 주기성 문제뿐만 아니라 연속 예측의 어려움에 대한 문제도 가지고 있다. 대부분의 전통적인 시계열은 한 시점만 예측하는 점 예측(One-step ahead prediction) 방법을 사용한다. 한편, 연속 예측(Multi-step ahead prediction)은 다소 긴 시간 범위에서 미래 데이터의 완전한 사이클을 예측하는 방법이다. 연속 예측 방법론은 연속적인 예측 값들을 생성하기 위하여 이전 시점들의 실제 예측 값을 입력 값으로 사용하여 반복적인 예측을 한다. 예를 들어, Table 1에서 x는 입력 값, $\hat{y}$은 예측을 하고자 하는 미래 값이라고 하였을 때, $\hat{y}_{t+2}$를 예측하기 위하여 이전 시차의 예측 값인 $\hat{y}_{t+1}$을 입력 값으로 이용한다. 즉, 이전 시점의 예측 값이 다음 시점의 예측을 위한 입력 값으로 축적되어 사용된다[6]. 실제 산업 환경에서는 연속적인 예측을 통한 미래 전망을 기반으로 의사결정을 수행하기 때문에 점 예측보다 연속 예측이 더 중요하게 활용될 수 있다. 하지만 시계열의 연속 예측은 시계열의 미래 패턴을 정확하게 설명할 수 있어야 하며, 누적된 예측 수행으로 인해 분산이 커지기 때문에 예측이 매우 어렵다는 단점을 가진다. 또한 단순히 실제 시계열을 후행하여 예측하거나, 시계열 패턴 모양 자체를 다르게 예측하는 문제가 있다[7].

시계열 예측을 위한 통계적 방법론은 선형 모형을 가지기 때문에 데이터의 선형적 특징을 잘 식별한다는 장점이 있어 널리 쓰여왔다[5]. 하지만 이러한 통계적 방법론은 비선형성을 포착할 수 없으며, 다중 주기성을 포착하기 어렵다는 단점을 가진다. 또한, 통계적 방법론은

표 1 점 예측 방법론과 연속 예측 방법론 예시

Table 1 Example of one-step ahead prediction and multi-step ahead prediction methods

One-step ahead		Multi-step ahead	
Prediction	Inputs	Prediction	Inputs
$\hat{y}_{t+1}$	x_t, x_{t-1}, x_{t-2}	$\hat{y}_{t+1}$	x_t, x_{t-1}, x_{t-2}
$\hat{y}_{t+2}$	x_{t+1}, x_t, x_{t-1}	$\hat{y}_{t+2}$	$\hat{y}_{t+1}, x_t, x_{t-1}$
$\hat{y}_{t+3}$	x_{t+2}, x_{t+1}, x_t	$\hat{y}_{t+3}$	$\hat{y}_{t+2}, \hat{y}_{t+1}, x_t$

수많은 초모수(Hyperparameter)를 필요로 하지만, 이를 위하여 관련 분야 전문가의 지식이 필요하다는 단점을 가진다. 한편, 심층 신경망은 비선형성을 포착하는 데에 특화되어 있다. 순환신경망(Recurrent Neural Network)[8]은 순환 구조로 인해 시계열을 모델링하는데 구조적 이점을 보였으나, 장기 의존성에 한계를 가진다는 문제를 가진다. 합성곱 신경망(Convolutional Neural Network)[9]과 주의 기제(Attention)[10,11]는 순환신경망이 가진 장기 의존성의 한계를 해소할 수 있으나, 장기 주기성을 포착하기 위하여 여러 개의 합성곱 계층이 필요하다는 한계를 갖는다. Seq2Seq(Sequence-to-Sequence)[12]는 다수의 합성곱 계층이 필요한 앞선 방법의 단점을 해소하지만, 연속 예측이 어렵다는 한계를 가진다. 또한, 기존 심층 신경망의 다변량(Multivariate) 모델링은 모든 변수를 한 번에 가지기 때문에, 변수에 특화된 모형을 얻기 어렵다.

본 연구는 앞선 선행 연구들이 가진 한계점을 해결하기 위하여 새로운 방법론인 DTSNet(Denoising Time Series Model using Deep Neural Networks)을 제안한다. 제안하는 모형은 장기 의존성 문제를 해결하고 복잡한 주기성을 모델링하기 위하여 이중 주의 기제를 사용한다. 또한, 멀티 헤드(Multi-head) 신경망을 이용하여 다변량 시계열 데이터를 변수 별 특화된 모형으로 모델링을 하여 예측의 성능을 향상시키기는 방법론을 제안한다. 마지막으로 시계열 연속 예측에 효과적인 디노이징(Denoising) 훈련법과 위치 인코딩을 제안한다.

본 논문의 기여는 다음과 같다. 1) 시계열의 안정적인 연속 예측을 위해 위치 인코딩을 적용한 표현형과 새로운 디노이징 훈련법을 제안한다. 2) 인코더-디코더의 시간 종속성과 디코더 시차간 종속성을 학습할 수 있는 이중 주의 기제 기법을 제안한다. 3) 멀티 헤드 신경망을 통해 변수 별로 특화된 특징을 추출해서 예측하여, 다변량 시계열 예측의 성능을 향상시킨다. 본 논문의 구성은 다음과 같다. 2장에서 선행 연구를 소개하고, 3장에서는 본 연구의 문제를 정의한다. 4장에서는 제안하는 모형의 방법론을 제시하고, 5장에서 실험 환경의 세부사항 및 평가지표를 소개한다. 6장에서 선행 연구 및 본 모형 내의 비교 실험을 보여주고, 7장에서는 결론을 기술한다.

2. 선행 연구

시계열 예측 분야에서 단변량(Univariate) 시계열 예측을 위한 전통적인 시계열 분석 도구 중 하나는 ARIMA (Autoregressive Integrated Moving Average)[5]이다. ARIMA 모형과 그 변형은 자기 회귀(Autoregression)의 선형 조합과 이동 평균(Moving Average)으로 구성되어 있으며, 차분(Differencing)과 같은 시계열의 정상성(Stationarity)을 보장하는 통계 기법이다. 이러한 모형들은 선형 조합으로 이루어져 있기 때문에, 높은 해석력을 가지면서도 연속 예측이 가능하다는 장점이 있다. 그러나 이러한 모형들은 자기 회귀 또는 이동 평균 모델링을 위해 많은 초모수를 필요로 하며, 적절한 초모수를 설정하기 위한 도메인 지식도 필요하기 때문에 고차원 다변량 시계열 분석에는 부적합하다. 다변량 시계열 분석에서 많이 쓰이는 모형인 VAR(Vector Autoregression)[13]은 ARIMA 모형의 일반화로서, 단순하고 해석이 용이하다는 장점을 계승한다. 그럼에도 불구하고, VAR은 비선형성을 모델링하는 능력이 부족하기 때문에 복잡한 패턴을 포착하지 못한다. 초모수가 상대적으로 적은 비모수적 방법인 GP(Gaussian Process)[14] 방법 또한 시계열 예측에 자주 사용되지만 분포 가정과 높은 연산량을 필요로 한다는 단점을 가진다.

Prophet[15]과 TBATS[16]는 시계열 모델링을 통해 연속 예측을 수행할 수 있는 상용 API이다. 먼저, Prophet은 페이스북(Facebook)에서 개발 및 배포하는 시계열 분석 및 예측을 위한 오픈 소스 프레임워크이며, 푸리에 변환(Fourier Transformation)을 이용한 곡선 적합(Curve Fitting)을 사용하여 시계열 모델링 문제를 해결한다. 이를 통해 유연한 모형을 얻을 수 있으며 학습 속도가 빠르다는 장점이 있지만, 데이터에 따른 초모수 설정이 쉽지 않다는 단점이 있다. 또 다른 오픈 소스인 TBATS도 푸리에 변환을 기반으로 하여 ARIMA의 변형인 ARMA 모형을 적합하는 프레임워크이다. 사용자가 임의의 다중 주기성을 초모수로 설정할 수 있으나, 데이터에 대한 사전 도메인 지식이 부족하다면 활용이 어렵다. 또한 이러한 문제점 외에도 시계열의 정상성을 위한 박스-콕스 변환 등에서 강력한 컴퓨팅 성능이 필요하고, 긴 훈련 시간을 요구한다는 단점을 가진다.

최근, 심층 신경망은 시계열의 비선형성을 포착하는 강력한 능력으로 많은 관심을 받고 있다[17]. 특히, 순환신경망은 순환 구조로 인해 시계열을 모델링하는 데 구조적 이점을 보인다. 하지만, 순환신경망과 그 변형 모형들[8,18]은 장기적인 시간 종속성에 한계를 가진다는 단점을 가지고 있다. 이러한 시간 종속성을 향상시키는 동시에 주기성 특징을 모델링하기 위하여 LSTNet이 제안되었다[4]. LSTNet은 합성곱 계층과 순환 계층을 모두 이용하는 모형이다. LSTNet은 합성곱 계층으로 국소 종속성을 포착하고 순환 계층으로 장기 종속성을 포착하여 점 예측을 한다. LSTNet에서 장기 종속성을 포착하기 위해 사용된 Recurrent-Skip은 미리 정의된 초모수를 필요로 하기 때문에 시간이 지남에 따라 주기 길이가 동적인 시계열에서는 작동하기 힘들다는 단점을 가진다. MTNet[19]은 이를 보완하기 위하여 모형에 메

모리 구성 요소를 추가하여 과거의 장기 데이터를 저장해 장기 종속성을 높였다. 하지만 여전히 MTNet도 연속 예측을 할 수 없다는 한계를 가진다.

앞선 선행 심층 신경망 모형들은 모두 점 예측을 하는 모형들이었다. 주식 시장 예측[20]이나 전력 소비 규제[1] 등, 실세계에서는 점 예측 보다는 연속 예측이 더 중요하게 활용되고 있는 분야가 많다. 이러한 연속 예측에서 특히 관심을 끄는 것은 기계 번역에서 큰 성공을 거둔 Seq2Seq 모형[12,21]이다. 하지만, Seq2Seq 예측은 시계열의 변동이 시간의 흐름에 따라 분산이 일정하지 않은 비정상성(Non-stationarity)을 가지기 때문에 이론적 한계가 있다[22]. 이러한 연속 예측의 높은 분산과 불안정한 훈련으로 인하여 모형이 실제 시계열을 단순히 후행 하여 예측하거나, 시계열 모양 자체를 다르게 예측하는 등의 문제가 있었다. 한편, 이를 해결하기 위하여 DILATE[7]가 제안되었다. DILATE에서 제안한 손실 함수는 DTW(Dynamic Time Warping)를 이용하여 시간과 패턴에서 발생하는 손실을 산정한다. 이를 통해 연속 예측을 어느 정도 안정화할 수 있지만, 높은 계산 복잡도로 인해 훈련이 느리다는 단점을 가진다. 이에 따라, 본 연구는 손실 함수를 수정하는 기존의 방법론이 아닌, 모형 자체에서 디노이징을 수행할 수 있게끔 훈련하는 방법론을 제안한다.

3. 문제 정의

본 연구에서는 t시점의 Y_t값을 예측하는 점 예측이 아닌 $\{1, 2, ..., T\}$ 시점을 연속적으로 예측하는 연속 예측에 초점을 두고 있다. 이때, 주어진 시계열 외의 외생변수는 사용하지 않고, 주어진 시계열의 주기성만을 모델링하여 미래 시차를 예측한다. 입력 시계열(X)은 다음과 같이 n 차원을 가지는 L 시차까지의 시계열로 정의한다.

$$X = \{X_1, X_2, \dots, X_L\},\ X \in \mathbb{R}^{n \times L}$$

본 모형은 외생 변수를 사용하지 않기 때문에, 아래와 같이 입력 시계열과 동일한 차원을 가지는 T 길이의 출력 시계열(Y)로 정의한다.

$$Y = \{Y_1, Y_2, \dots, Y_T\}, Y \in \mathbb{R}^{n \times T}$$

4. 제안 방법론

본 연구에서는 연속 예측 시계열 모델링을 위한 인코더-디코더 기반의 새로운 방법론인 DTSNet을 제안한다. 본 연구의 전체적인 구조는 Fig. 2와 같다. 제안하는 방법론은 합성곱 인코더와 이중 주의 기제 기법을 사용한 디코더, 그리고 자기 회귀 레이어(Autoregressive Layer)로 구성된다. 또한 시계열의 시점 위치를 나타낼 수 있는 위치 인코딩(Positional Encoding)과 효과적인 연속 예측을 위한 디노이징 훈련 방법을 제안한다. 다음 각 절에서 구성 요소들을 상세히 다루고, 최종 예측 수행과 훈련을 위한 최적화 방법을 설명한다.

4.1 합성곱 인코더

본 연구에서는 시계열 데이터에 대한 정보를 1차원 합성곱 모형을 통해 추출한다. k 번째 합성곱 필터를 이용한 특징 벡터(Feature Maps)는 다음과 같이 계산된다.

$$e_k = ReLU(W_k * X + b_k) \quad (1)$$

수식 (1)과 같이 입력 시계열 X에 대해 $ReLU$ 활성함수를 사용한 합성곱 인코더를 통해 특징 벡터 e_k를

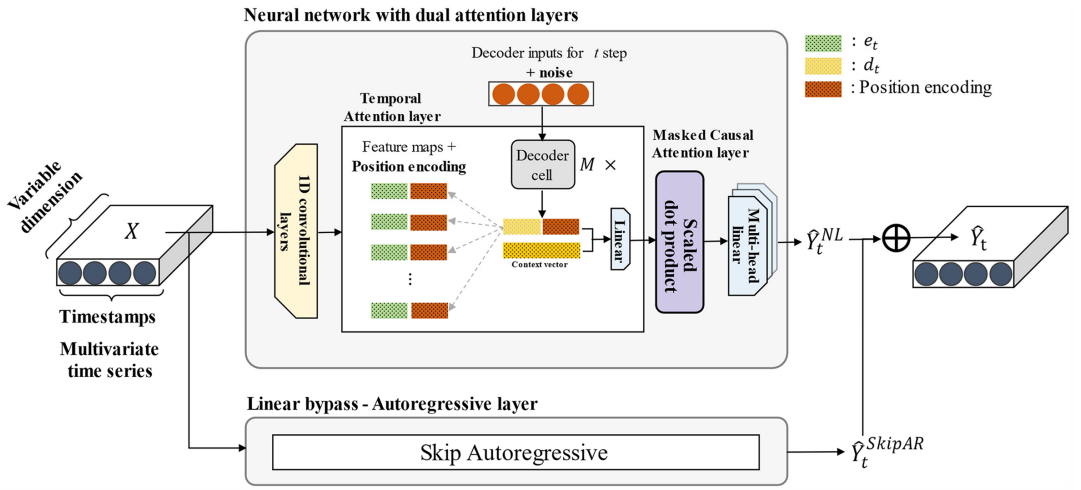


그림 2 제안하는 DTSNet 모형의 구조 개요
Fig. 2 Overview of the proposed method, DTSNet

구한다. 이때, $*$ 연산은 합성곱 연산을 나타내며, W_k는 k번째 필터를 의미한다. 그리고 W_k와 b_k는 학습 가능한 모수이다. 또한, 제로 패딩(Zero Padding)을 사용하여 $e_k \in \mathbb{R}^L$ 차원을 갖는다. 최종적으로 합성곱 인코더의 출력 값은 $E \in \mathbb{R}^{D_k \times L}$ 차원을 갖고, 이때 D_k는 필터의 개수이다.

4.2 이중 주의 기제 기법을 이용한 디코더

이전 예측 값을 입력으로 받아 연속 예측을 수행하기 위한 디코더 셀(Decoder Cell)은 순환 신경망 기반 모형인 GRU(Gated Recurrent Unit)[23]를 사용한다. 이때, 디코더 셀을 M개 스택한다.

디코더는 순환 신경망 모형의 단점인 장기 의존성을 보완하여 복잡한 주기성을 모델링하기 위해, Fig. 3과 같이 두 가지의 주의 기제 기법을 사용한다. (1) 시간 주의 기제 기법(Temporal Attention, TA)을 통해 위치 인코딩을 취한 인코더의 특징 벡터와 노이즈(Noise)가 추가된 디코더의 은닉 상태를 내적하여 새로운 은닉 상태를 구한다. (2) 인과 주의 기제 기법(Masked Causal Attention, CA)을 이용하여 디코더 시차간 정보를 포함하는 결과를 도출한다. 또한 변수 별로 특화된 디코딩을 수행하기 위해 멀티 헤드 완전 연결 신경망을 구축한다.

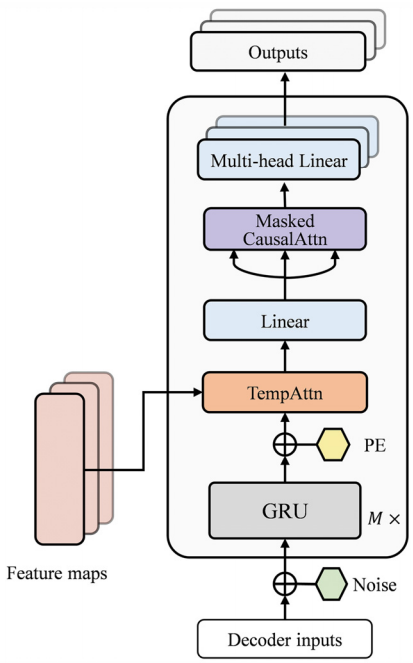


그림 3 이중 주의 기제 기법과 멀티-헤드 완전 연결 신경망을 이용한 디코더 구조

Fig. 3 Decoder architecture using dual attention and multi-head linear

4.2.1 시간 주의 기제 기법

첫 번째로 사용되는 주의 기제 기법인 시간 주의 기제기법은 예측을 수행하는 시점에서 과거 모든 시점의 관측 값과의 중요도를 산정한다. 그리고 산정한 중요도에 따라 과거 정보를 취합한다. 이때, 주의 기제 함수는 내적 주의 기제 기법(Dot-product Attention)을 이용한다. 본 모형의 t번째 디코딩 시차에서, GRU 디코더 셀을 통한 은닉 상태 추출과 시간 주의 기제 기법은 다음 수식을 따라 계산된다.

$$d_t = GRU^{(M)}(d_{t-1}, \hat{y}_{t-1}) \tag{2}$$

$$TempAttn(h_i, h_j) = \text{softmax}(h_i^{\mathrm{T}} h_j) \tag{3}$$

$$\alpha_{t,k} = TempAttn(d_t, e_k) \tag{4}$$

$$c_t = \sum_k \alpha_{t,k} e_k \tag{5}$$

t시차의 GRU 은닉 상태 $d_t \in \mathbb{R}^{D_{model}}$는 수식 (2)와 같이 GRU 디코더 셀을 통해 계산되고, 이때 $GRU^{(M)}$는 M개의 스택된 GRU를 나타낸다. 또한, 주의 기제 점수 α_t는 수식 (3), 수식 (4)와 같이 d_t와 $E = \{e_1, e_2, ..., e_k\}$ 사이의 내적을 이용한 주의 기제 함수 $TempAttn(*)$을 통해 구한다. 이후 컨텍스트 벡터(Context Vector) c_t는 수식 (5)와 같이 인코더 특징맵 E와 t시차의 주의 기제 점수 α_t의 곱으로 구한다. 수식 (6)처럼, 시간 주의 기제의 c_t와 d_t를 사용한 선형 변환 레이어를 통해 h_t를 구한다.

$$h_t = \sigma(W^{TA}[c_t; d_t] + b^{TA}) \tag{6}$$

이때, W^{TA}와 b^{TA}는 학습 가능한 모수이다. [;]는 벡터 연결(Concatenation) 연산, σ는 활성함수를 나타낸다.

4.2.2 인과 주의 기제 기법

두 번째로 사용되는 주의 기제 기법은 예측을 수행하는 시차 간의 의존성을 고려한다. 4.2.1 절에서 시간 주의 기제 기법은 인코더 입력 시계열의 시차와 예측을 수행하는 t시차와의 시간적인 관계를 반영한 벡터이다. 예측하고자 하는 디코더의 출력인 $\{1, 2, ..., T\}$시차에서는 GRU의 순환적인 구조에만 의존하여 직전 시차에서 예측한 정보를 활용한다. 그러나 단일 시점 예측이 아닌 다수의 시차를 연속 예측할 때 장기 의존성 문제가 발생할 수 있다.

이를 보완하기 위하여 자가 주의 기제 기법[24](Self-Attention)을 적용하여 디코더 시차 안의 관계에 대한 정보를 표현한다.

$$S = CausalAttn(q(H), k(H), v(H)) \tag{7}$$

$$CausalAttn(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D_{model}}}\right)V \tag{8}$$

수식 (7)과 수식 (8)과 같이 4.2.1절의 출력물인 $H = \{h_1, h_2, ..., h_T\}$를 사용하는 주의 기제 함수($CausalAttn(*)$)를 통해 각 디코딩 시차간 정보가 반영된 $S \in \mathbb{R}^{D_{model} \times T}$를 얻는다. 이때 $q(\cdot), k(\cdot), v(\cdot)$ 함수는 훈련 가능한 선형 변환을 나타내고, D_{model}는 은닉층의 크기이다.

훈련시에는 주의 기제가 해당 훈련 시차 이후인 예측값에도 적용되는 것을 방지하기 위하여 마스크(Mask)를 적용한다. 즉, 시간의 순행적 인과관계(Causality)를 보장하기 위하여 예측하고자 하는 t 시차에서는 $\{t+1, ..., T\}$ 시차의 은닉값을 마스킹하여 제하고, 오직 $\{1, 2, ..., t\}$ 시차의 은닉값을 사용하여 주의 기제를 수행한다. 테스트 시에는 미래의 예측값이 디코더의 입력값으로 주어지지 않기 때문에 마스크를 사용하지 않는다.

4.2.3 멀티 헤드 완전 연결 신경망

본 모형에서는 이중 주의 기제 기법과 더불어 완전 연결 신경망(Fully Connected Layers)을 통해 비선형 모형의 예측 값을 도출한다. 이때, 변수 별로 특화된 예측 모형을 위하여 은닉층을 여러 개의 헤드($\tilde{h}_{t,i}$)로 나누어 변수 별로 독립적인 완전 연결 신경망을 수행한다.

$$\hat{y}_{t,i} = W^O \tilde{h}_{t,i} + b^O \tag{9}$$

$$\tilde{h}_{t,i} = \sigma\left(W_i^I s_t + b_i^I\right) \tag{10}$$

이때, $i = \{1, 2, ..., n\}$는 변수 차원을 나타내고, 헤드($\tilde{h}_i$)의 은닉크기는 D_{head}로 설정한다. 즉, D_{head} 크기의 완전 연결 신경망(수식 (10))과 뒤따르는 선형 변환인 출력 레이어(수식 (9))를 통해 비선형 모형의 최종 출력값 $\hat{Y}_t^{NL} \in \mathbb{R}^n$를 얻는다. 이때, W와 b는 학습 가능한 모수이다.

4.3 자기 회귀 레이어

본 연구에서 제안한 방법론은 비선형 신경망 모형 기반이기 때문에, 출력 값의 스케일(Scale)은 입력 시계열의 스케일을 적절히 반영하지 못한다. 입력 시계열의 비주기적인 스케일 변화로 인해, 스케일 반영 실패는 예측 모형의 성능을 대폭 악화시킨다[5].

이러한 문제를 해결하고자, 모형의 부족한 선형적 특성과 스케일 반영을 강화하기 위하여 선행 연구들의 선형 가법 모형을 사용한다[25,26]. 본 모형은 하이웨이 네트워크[27]의 본질과 유사하게 최종 예측 값을 선형적 부분과 비선형적 부분으로 분해하여 접근한다. 이때, 선형적 부분에서는 스케일 부족을 해결하고 비선형 부분에서는 다중 주기성을 모델링한다. 선형 모형으로는 자기 회귀 모형을 차용하여 적합한다.

본 연구에서는 일반적인 자기 회귀 모형과 달리, 다중 주기성의 특징을 활용한 스킵 자기 회귀(Skip Autoregression, SkipAR) 모형을 제안한다. 스킵 자기 회귀 모형은 아래의 수식을 따라 변수 별로 독립적으로 학습한다.

$$\hat{y}_t^{SkipAR} = \sum_{k=1}^{q} w_k^{SkipAR} x_{(m \times k)} + b^{SkipAR} \tag{11}$$

수식 (11)과 같이 스킵 자기 회귀 모형은 예측하려는 시점과 동일한 과거의 시점들을 q개 사용하여 선형 적합 한다. 이때, m은 시계열의 세분성에 따른 모수로서, 실험에서는 단위 길이에 대한 시계열 데이터의 개수로 설정하였다(5장 참조).

4.4 최종 예측

DTSNet은 t 시점에서의 예측 값 $\hat{Y}_t$를 구하기 위하여 수식 (12)와 같이 두 개의 출력 값을 가산한다.

$$\hat{Y}_t = \hat{Y}_t^{NL} + \hat{Y}_t^{SkipAR} \tag{12}$$

합성곱 인코더와 이중 주의 기제 기법을 사용하는 디코더 모형의 출력 값인 비선형 요소 $\hat{Y}_t^{NL}$과 스킵 자기 회귀 모형 적합의 출력 값인 $\hat{Y}_t^{SkipAR}$을 통합하여 최종 예측 값 $\hat{Y}_t$을 구한다.

4.5 위치 인코딩

시계열 데이터는 주기성의 영향을 크게 받기 때문에, 모형이 예측을 수행하는 시점에 대한 정보가 있을 때 정확한 예측을 수행할 수 있다. 그러나 단순하게 시간을 정수와 같은 수치 값으로 입력하는 것만으로는, 모형이 예측을 수행하는 시점을 학습하기 어렵다. 본 연구에서는 시간 정보를 추가적으로 제공하여 모형이 주기성을 보다 정확하게 인지하도록 하기 위하여 위치 인코딩을 이용한다.

위치 인코딩은 [24]에서 제안된 방법론으로서, 기계 번역에서 자연어의 상대적, 절대적 위치에 대한 정보를 단어 임베딩에 포함할 수 있다. 이에 착안하여, 본 연구에서는 시계열 모델링을 위해 이용된 각 시차의 고유한 위치 인코딩 적용을 다음과 같이 제안한다.

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/D_{model}}\right), \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/D_{model}}\right) \end{aligned} \tag{13}$$

$$\tilde{E} = E + PE, \tilde{D} = D + PE \tag{14}$$

이때, 위치 인코딩 벡터의 크기는 D_{model}과 같다. i는 위치 인코딩 벡터 내에서의 인덱스를 나타내고, pos는 해당 t 시차의 절대적인 위치를 나타낸다. 즉, 수식 (13)과 같이 t 시차의 PE는 벡터 내 차원 인덱스가 증가함에 따라 사인, 코사인 함수가 교차로 반복되며 생성된다. 최종적으로 PE는 수식 (14)와 같이 시점 정보가 필요한 인코더의 입력값 E와 디코더의 입력값 D에 가산한다.

4.6 디노이징 훈련 기법

기계 학습과 심층 신경망 연구에서 모형을 학습하는 방법론은 모형의 성능에 큰 영향을 끼친다. Seq2Seq와 같이 인코더-디코더 구조를 기반으로 하는 모형은 주로 티쳐 포싱(Teacher Forcing, TF)을 사용한다[28,29].

TF 훈련 기법은 훈련을 할 때에 t 번째의 디코더 입

력 값으로 t-1 번째의 정답 데이터(Ground Truth)를 넣어주는 방법론이다. 그러므로 훈련 시 디코더가 더 정확한 예측을 할 수 있으므로 초기 학습 속도가 빠르다[30]. 그러나 t-1단계에서 출력한 값 기반으로 예측을 하는 훈련이 아니므로, 추론과 학습 단계에서의 차이(Discrepancy)가 존재하여 모형의 성능과 안정성을 떨어뜨리는 노출 편향 문제(Exposure Bias Problem)가 존재한다[31].

반대로 TF 미사용 훈련 기법(w/o TF)은 훈련시에도 t 단계의 출력을 얻기 위하여 입력 값으로 t-1단계에서 예측한 값($\hat{Y}_{t-1}$)을 사용한다[32]. $\hat{Y}_{t-1}$이 크게 잘못되었다면 초기 학습 속도의 저하를 일으킬 수 있어 계산량이 높다는 단점이 있다. 하지만 이러한 훈련 기법은 학습과 추론의 차이가 없으므로 노출 편향 문제가 없고, 따라서 모형의 안정성이 상대적으로 높다. 또한 훈련 기법의 특성상, t-1 시차의 잘못된 예측 값을 어느정도 감안하여 t 시차의 예측 값을 올바르게 예측하여 훈련할 수 있다.

본 연구에서는 가우시안 노이즈를 이용하여 TF 기법과 w/o TF 기법의 장점들을 모두 활용할 수 있는 디노이징 훈련 기법(Denoising Training Method)을 제안한다.

$$\tilde{Y}_{t-1} = Y_{t-1} + \gamma \times scale(Y) \times \varepsilon \tag{15}$$

$$\varepsilon \sim N(0,1) \tag{16}$$

수식 (15)와 같이, 가우시안 노이즈 ε와 노이즈 반영 정도를 결정하는 초모수 γ와 Y의 표준 편차와 곱하여 노이즈를 산정한다. 이후 노이즈를 가산한 $\hat{Y}_{t-1}$을 디코더의 입력 값으로 설정한다. γ가 높을수록 정답 데이터에 더 많은 노이즈가 더해진다. 결과적으로 Y_{t-1}을 통해 TF 기법의 빠른 학습 효과를 목표로 한다. 하지만 일반적인 TF와 달리, 적은 노이즈 ε를 추가한 $\hat{Y}_{t-1}$을 사용함으로써 노출 편향 문제를 제거하고, 이전 시차의 잘못된 예측을 보정하는 훈련을 수행함으로써 모형의 성능과 안정성을 높인다. 한편, 모형에 더해지는 노이즈는 훈련 시에만 사용한다.

훈련 시 손실 함수는 수식 (17)의 절대 오차를 나타내는 MAE 손실 함수를 사용한다. 절대 오차는 시계열 분석에서 빈번하게 발생하는 이상치에 둔감하게 반응한다는 장점이 있다[33]. 최적화는 Adam 기법[34]을 사용하여 훈련한다.

$$L_1 = \frac{1}{|T||n|}\sum_t \sum_n |Y - \hat{Y}| \tag{17}$$

5. 실 험

5.1 데이터셋

본 연구에서는 공개적으로 사용 가능한 다변량 시계열 벤치마크 데이터셋 3가지를 사용한다. 주어진 변량 외의 외생변수는 사용하지 않는다. 이때, 1일 내에 발생하는 시계열 데이터의 개수를 *단위 길이* m으로 정의한다. 선행 연구와의 비교 실험을 위해 모든 데이터셋은 [5]에서 전처리된 자료를 사용하였고, 데이터셋에 대한 통계는 Table 2에 정리되어 있다.

표 2 공개 데이터셋 통계
Table 2 Statistics of public datasets

Datasets	# of Time Stamps	# of Variables	Granularity
Solar-energy	52,560	137	10 minutes
Electricity	26,304	321	1 hour
Traffic	17,544	862	1 hour

- Solar-energy: 2006년 앨라배마 주(Alabama State)의 137개 공장에서 10분마다 태양광 생산을 기록한 것으로 구성되어 있으며, m = 144 이다.
- Electricity: 321개의 지점에 대하여 2012년부터 2014년까지 15분마다 전력 소비량을 kWh 단위로 기록하였다. 시간당 소비량으로 데이터를 변환하였으며, 이에 따른 m = 24 이다.
- Traffic: 캘리포니아(California) 교통부에서 제공한 48개월(2015-2016) 동안의 시간별 고속도로 점유율(0~1) 데이터셋으로, m = 24 이다.

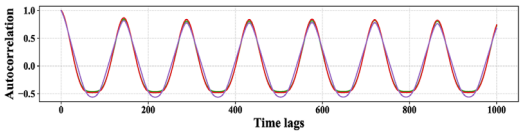
본 실험에서는 주어진 시계열 데이터셋의 마지막 31일을 테스트셋으로 활용한다. 그리고 테스트셋을 제외한 데이터셋에서 랜덤하게 80%를 훈련 데이터셋으로, 20%를 검증 데이터셋으로 사용한다.

시계열 분석에서 가장 중요한 것은 주기성을 찾는 것이다. 이에 따라, 본 데이터셋의 각 변수에 대해서 자기상관함수(Autocorrelation Function, ACF)를 계산하여 시계열의 주기성을 찾는다. 자기상관함수는 시계열 데이터의 자기상관성을 파악하기 위한 함수로, 현재 시차와 다른(지연된) 시차 사이의 상관 관계를 나타낸다. 자기상관함수는 다음과 같이 구한다[6].

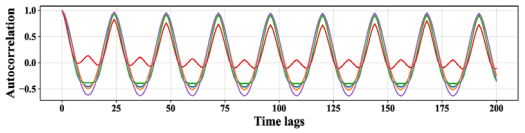
$$\rho(\tau) = \frac{E[(X_t - \mu)(X_{t+\tau} - \mu)]}{\sigma^2} \tag{18}$$

이때, τ는 지연된 시간의 시차를 의미하고, μ와 σ^2는 각각 주어진 시계열의 평균과 분산을 의미한다.

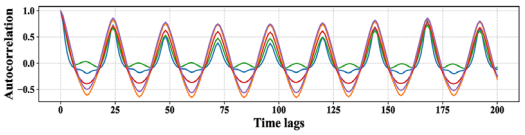
Fig. 4는 각 데이터셋에서 랜덤하게 선택된 5개의 변수에 대한 ACF를 표현한 그림이다. 이때, τ는 Solar-energy, Electricity, Traffic에서 각각 1000/200/200까지의 지연된 시차를 계산한다. Solar-energy 데이터셋의 시간 단위는 10분이므로, 장기 종속성을 조사하기 위하여 더 긴 타임 스탬프를 계산한다. 그림에서 볼 수 있듯이, 데이터셋 모두 높은 자기 상관이 있는 반복 패턴이 존재한다. 또한 가장 큰 자기상관계수 값 외에도, 상



(a) ACF plot for *Solar-energy* ($\tau = 1000$)



(b) ACF plot for *Electricity* ($\tau = 200$)



(c) ACF plot for *Traffic* ($\tau = 200$)

그림 4 모든 데이터셋에 대한 자기상관 도표
Fig. 4 Autocorrelation plots for all datasets

관계수가 매우 높은 시차들이 주기적으로 발생한다. 24시간 동안 매일 나타나는 단기 패턴과 7일마다 반복적으로 나타나는 장기 패턴을 찾을 수 있다.

5.2 비교 모형

본 연구에서 제안하는 DTSNet과의 비교 실험을 위하여 널리 사용되는 기계 학습 모형 4개와 심층 학습 모형 3개, 총 7개의 모형을 사용한다.

- AR은 전통적으로 시계열 분석에서 사용되는 단순한 단변량 선형 회귀 모형이다[6].
- GP는 분포 가정을 통해 시계열 분석을 하는 단변량 모형이다[14].
- Prophet은 푸리에 변환을 이용한 곡선 적합으로 시계열 모델링하는 단변량 모형이다[15].
- TBATS는 푸리에 변환을 기반으로 ARMA 모형을 적합하는 단변량 프레임워크이다[16].

AR, GP, Prophet 및 TBATS와 같은 단변량 시계열 모형들은 각 변수에 대해 모형을 생성하여 독립적으로 훈련한다.

- Seq2Seq-w/o-attn은 GRU를 사용한 일반적인 인코더-디코더 구조이다.
- Seq2Seq-w/-attn은 위의 인코더-디코더 구조와 주의 기제 기법을 사용한다.
- LSTNet-rec는 점 예측을 수행하는 기존의 LSTNet[5]을 연속 예측을 수행하도록 변형한 다변량 시계열 모형이다.

Seq2Seq-w/o-attn, Seq2Seq-w-attn 및 LSTNet-rec와 같은 다변량 시계열 모형은 변수를 하나로 묶어 모형을 생성해 훈련한다. 또한 LSTNet-rec의 경우, 점 예측을 통해 훈련한 후, 연속 예측을 수행한다.

5.3 실험 세부 사항

본 절에서는 DTSNet과 비교 모형들의 초모수 설정에 대해 논의한다. 모든 모형에서 사용되는 입력 시계열의 길이(L)는 *단위 길이*(m) $\times 7$를 *최대 길이*로 제한하지만, 비교 모형들의 제약 사항에 따라 상이하다. 연속적으로 예측하고자 하는 출력 시계열의 길이(T)는 *단위 길이*(m)로 동일하다.

5.3.1 기계 학습 모형

AR 학습에서 과거 관측치 기간을 의미하는 최대 시차를 위의 *최대 길이*로 설정한다. 즉, m $\times 7$개의 X를 사용하여 적합한다. GP는 타 모형 실험과 다르게 예측을 하기 위한 날짜와 가장 근접한 7일치의 데이터를 적합한다. 이는 GP가 비모수 커널 기반의 확률적 모델로 분포 가정을 통해 학습하기 때문이다.

Prophet 학습에서는 *최대 길이*를 윈도우로 사용하고, 주기성 정보를 추가하여 예측한다. 일일 주기성, 주간 주기성, 월별 주기성을 사용한다. 각 주기성은 푸리에 급수를 이용하여 패턴의 근사치를 찾는다. 일일 주기성의 푸리에 급수 차수는 50, 주간 주기성은 20, 월별 주기성은 10으로 부여한다. 예측 단계에서는 상한 값을 1, 하한 값을 0으로 설정하고 진행한다.

TBATS 학습 또한 입력 윈도우 길이를 *최대 길이*로 사용하고, 주기성 기간은 $(m, m \times 7)$이다. AIC를 기준으로 박스-콕스 변환을 적용할지 결정하여 예측한다.

5.3.2 심층 학습 모형

DTSNet은 입력 윈도우를 *최대 길이*만큼 사용한다. 또한 인코더 합성곱의 커널 개수 D_k와 디코더의 은닉 크기 D_{model}은 동일한 크기로 설정하되, $\{2^7, 2^8, ..., 2^{10}\}$ 중 검증 데이터셋을 기준으로 그리드 탐색(Grid Search)한다. 인코더의 커널 사이즈는 5로 설정하고, 독립 완전 연결 신경망의 $D_{head} = 8$로 설정한다. Regularization을 위한 드롭 아웃(Dropout) 비율은 0.5로 설정하고, 학습률은 0.001을 사용한다.

Seq2Seq의 윈도우의 크기는 $\{m, m \times 5, m \times 7\}$중 검증 데이터셋을 기준으로 선택한다. 인코더와 디코더 GRU의 은닉 크기는 100으로 설정한다. 초모수 설정은 Seq2Seq-w/o-attn과 Seq2Seq-w/-attn에 동일하게 사용되며, Seq2Seq-w/-attn은 인코더 은닉층과 디코더 은닉층에 내적 주의 기제 기법을 추가하여 훈련한다.

LSTNet[5]은 가능한 원 논문의 초모수를 최대한 차용하여 훈련하는 데이터에 맞게 수정한다. 아래 주어진 모든 그리드에서 검증 데이터셋을 기준으로 그리드 탐색을 수행한다. 입력 시계열 길이(L)의 그리드는 $\{M \times 2^3, M \times 2^4, ..., M \times 2^{10}\}$이다. CNN 레이어와 RNN 레이어의

그리드는 {50, 100, 200}이다. RNN-Skip 레이어의 은닉 층은 Electricity와 Traffic은 10, Solar-energy는 {20, 50, 100} 중 검증 데이터셋을 기준으로 선택한다. RNN-Skip 계층의 Skip률은 Solar-energy는 $\{2, 2^2, ..., 2^6\}$ 중에 탐색하며, Electricity와 Traffic은 24를 사용한다. LSTNet-rec는 점 예측을 수행하는 것으로 훈련을 진행하지만 평가는 연속 예측을 수행한다.

5.4 평가 지표

본 연구에서는 시계열 예측에 널리 사용되는 3가지 평가 지표인 RRSE(Root Relative Squared Error), 정규화 RMSE (Normalized Root Mean Square Error), 경험적 CORR(Empirical Correlation Coefficient)를 이용하였다. 각 평가 지표는 다음과 같이 공식화할 수 있다.

$$RRSE = \sqrt{\frac{\Sigma_t(y_t - \hat{y}_t)^2}{\sum_t(y_t - mean(Y))^2}} \quad (19)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_t(y_t - \hat{y}_t)^2} \quad (20)$$

$$NRMSE = \frac{RMSE}{mean(Y)} \quad (21)$$

$$CORR = \frac{1}{n}\sum \frac{\sum_t(y_t - mean(Y))(\hat{y}_t - mean(\hat{Y}))}{\sqrt{\sum_t(y_t - mean(Y))^2}\sqrt{(\hat{y}_t - mean(\hat{Y}))^2}} \quad (22)$$

이때, Y와 $\hat{Y}$는 각각 참 값과 예측 값을 나타낸다.

다변량 시계열은 각기 다른 스케일을 가지는 시계열들로 이루어져 있기 때문에, 각 시계열들의 실측된 값은 범위의 차이가 크다. 즉, 동일한 데이터셋일지라도, 0과 1K의 범위를 가지는 변수가 있는 반면, 0과 100K의 범위를 가지는 변수가 동시에 존재할 수 있다. 또한, 서로 다른 데이터셋을 비교하고자 하는 경우, 동일한 RMSE를 사용하여 비교하기에는 어려움이 있다. 예를 들어, Electricity 데이터셋은 단위가 kWh이기 때문에 RMSE가 1보다 큰 값으로 계산될 수 있지만, Traffic 데이터셋은 도로의 점유율(Ratio)이므로 RMSE가 1보다 클 수 없다. 따라서, 본 연구에서는 변수들의 상이한 범위 값을 고려하여 평가하기 위하여, 스케일된 RMSE인 RRSE(수식 (19))를 이용한다. RRSE는 데이터 규모의 영향을 받지 않는 정규화 된 평가 지표이다. 또한 일반적으로 시계열에서 널리 사용되는 지표인 RMSE(수식 (20))는 데이터의 스케일을 고려하지 않기 때문에, 다변량 시계열 예측에서는 실제 값의 평균을 나누어 데이터의 스케일을 고려한 NRMSE(수식 (21))를 활용한다. 또한, 경험적 CORR(수식 (22))는 모든 변수의 시차들에 대한 상관 계수의 평균으로 볼 수 있다. 예측 성능의 해석은, 오차 수치를 의미하는 RRSE와 NRMSE는 평가 수치가 낮을수록 유의미한 결과 값을 생성한 것으로 평가한다. 또한 변수 사이의 상관관계의 정도를 나타내는 수치인 CORR은 평가 수치가 높을수록 상관성이 높음을 뜻한다.

6. 결과 및 토의

본 연구에서 제안하는 DTSNet의 우수성과 그 구성 요소들의 효과를 다양한 실험 결과를 통해 검증한다. 먼저, 6.1절에서 기계 학습 및 심층 학습 기법 총 7개의 모형과 DTSNet과의 비교 실험을 통하여 DTSNet의 우수성을 입증한다. 그리고 6.2절에서 DTSNet 내의 각 구성 요소에 대한 심층적인 분석을 수행함으로써 구성 요소별 세부적인 효과를 확인한다.

6.1 메인 결과

제안한 방법론의 성능을 검증하기 위하여, 동일한 데이터셋을 이용해 DTSNet과 시계열 예측을 위해 널리 사용되는 비교 모형 7가지에 대한 비교 실험을 수행한다. 테스트는 본 모형과 동일하게 1일 동안 발생하는 관측치를 연속적으로 예측하는 과정을 반복하여 연속적인 1달 예측을 진행하였다.

Table 3은 전체 모형들에 대한 실험 결과를 나타내고 있다. 실험 결과에서 DTSNet은 모든 기계 학습 및 심층 학습 모형들보다 대부분의 지표에서 높은 성능을 보이고 있다. 이는 제안한 방법론이 여러 개의 복잡한 주

표 3 선행 연구와의 성능 비교. 각 경우의 최고 성능은 굵게 표시됨

Table 3 Evaluation results of all methods on the test set. The best performance is highlighted in bold in each case

	Solar-energy			Electricity			Traffic		
	RRSE	CORR	NRMSE	RRSE	CORR	NRMSE	RRSE	CORR	NRMSE
AR	**0.5375**	0.8440	**0.9792**	0.4554	0.8000	0.7986	0.7010	0.7485	0.6419
GP	0.9899	0.3539	1.8037	0.9433	0.3608	1.6542	0.9650	0.3020	0.8842
Prophet	0.5886	0.8265	1.0725	0.3295	**0.8516**	0.5779	0.6485	0.8046	0.5938
TBATS	0.5968	0.8132	1.0874	0.3724	0.7826	0.6530	0.9078	0.6504	0.8313
S2S-w/-attn	1.2273	0.2326	2.2361	0.5023	0.7357	0.8808	0.7679	0.7424	0.7031
S2S-w/o-attn	1.6416	-0.1176	2.9909	0.6647	0.7116	1.1655	0.7844	0.7260	0.7182
LSTNet-rec	1.1583	0.1176	2.1104	1.0036	0.5390	1.7599	0.9297	0.5603	0.8513
DTSNet	0.5388	**0.8479**	0.9817	**0.2992**	0.8355	**0.5247**	**0.5830**	**0.8346**	**0.5376**

기성을 가지는 데이터셋에서 효과적으로 시계열을 모델링한다는 것을 의미한다.

또한, 기존의 디코딩을 수행하는 심층 학습 방법론에 비해, DTSNet의 성능이 뛰어나게 우수함을 알 수 있다. 특히, Table 3을 보면 DTSNet을 제외한 모든 심층 학습 방법론이 기계 학습 방법론보다 성능이 월등히 낮다. 이러한 결과를 분석해보기 위하여, LSTNet 모형에 대한 추가 비교를 진행하였다.

Table 4는 LSTNet 모형의 예측 방법에 따른 평가 지표를 나타내고 있다. 이때, 표에서 'LSTNet'은 바로 다음 시차만을 예측하는 점 예측을 수행하고, 'LSTNet-rec'는 바로 직전에 예측한 값인 $\hat{Y}_{t-1}$을 다시 사용하여 $\hat{Y}_t$를 예측하는 연속적인 예측을 수행한다.

Table 4의 결과에서, 점 예측에 대한 결과 수치는 연속 예측보다 월등히 우수함을 알 수 있다. 더욱이 본 연구가 제안하는 방법론인 DTSNet의 연속 예측 결과보다 LSTNet의 점 예측 결과의 성능이 더 높은 것을 알 수 있다. 연속 예측을 위한 LSTNet-rec가 점 예측을 위한 LSTNet보다 성능이 좋지 못한 이유는 LSTNet과 LSTNet-rec 모형 모두 심층 학습 기법에서 특징 추출을 기반으로 시계열을 모델링하는 방법론은 유효하지만, LSTNet-rec의 연속적으로 예측을 수행해야하는 디코딩 단계가 불안정 하기 때문이다. 이는 이전 시차에서 잘못된 예측값에 대한 분산이 누적되어 증가하기 때문에 디코딩이 불안한 것으로 해석할 수 있다. 즉, 심층 학습을 이용해서 연속 예측을 수행하는 것이 점 예측을 수행하는 것보다 매우 어렵다는 것을 증명한다. 또한 Seq2Seq 모형의 경우에도 기존 인코더-디코더 기반의 방법론에서 디코더 셀을 이용한 연속적인 예측을 모두 사용하기 때문에, 인코더에서 특징 벡터를 적절히 추출할 수 있더라도 연속적인 디코딩에서 어려움을 나타내는 것으로 해석할 수 있다.

한편, 본 연구에서 제안하는 방법론에서는 이러한 기존의 심층 학습 모형들의 한계를 극복하고 연속적인 예측을 안정화하는 디노이징 학습 기법과 연속 예측에 최적화된 모형을 구축하여 우수한 성능을 이끌어 낸다. 본 모형은 기존의 기계학습 모형과 심층 학습 모형과 비교하여 뛰어난 성능을 보였다. 특히, 기존의 심층 학습 모형들은 연속 예측에서 큰 성능 저하를 보이지만, DTSNet은 심층 학습 기법에서의 시계열 연속 예측의 한계를 극복하여 우수한 성능을 보였다.

6.2 비교 실험

6.2.1 각 구성 요소들에 대한 효과

본 연구에서 제안하는 모형의 핵심 구성 요소인 위치 인코딩(PE), 인과 주의 기제 기법(CA), 멀티-헤드 완전 연결 신경망(MultiHead), Skip AR의 성능을 검증하기 위하여 구성 요소를 하나씩 제거하며 비교 실험을 하고자 한다.

Table 5와 같이, 본 모형에서 Skip AR을 제거할 경우 모든 지표에서 가장 큰 성능 저하를 확인하였다. AR 요소는 선형 자기회귀 모형의 효과인 입력 시계열의 스케일을 보정해주는 역할이기 때문에, 스케일이 보정되지 않을 경우 에러가 가장 높다고 할 수 있다.

인과 주의 기제 기법(CA) 또한 본 모형에서 제거시 현저한 성능 하락을 보인다. 이는 연속적인 디코딩 예측에서, 시차에 제한되지 않는 주의 기제 방법인 인과 주의 기제가 GRU 디코더 셀과 더불어 장기 의존 문제를 해결한 것으로 해석할 수 있다. 이러한 보완적 정보 호

표 4 LSTNet, LSTNet-rec, DTSNet 간의 비교. LSTNet은 점 예측, DTSNet과 LSTNet-rec는 연속 예측
Table 4 Comparison of LSTNet, LSTNet-rec, and DTSNet. Note that LSTNet follows one-step forecasting, while DTSNet and LSTNet-rec use multi-step forecasting

	Solar-energy			Electricity			Traffic		
	RRSE	CORR	NRMSE	RRSE	CORR	NRMSE	RRSE	CORR	NRMSE
LSTNet	0.2010	0.9819	0.3325	0.0917	0.9155	0.1381	0.4995	0.8520	0.4453
LSTNet-rec	1.1583	0.1176	1.1298	1.0036	0.5390	0.3977	0.9297	0.5603	0.8583
DTSNet	0.5388	0.8479	0.9817	0.2992	0.8355	0.5247	0.5830	0.8346	0.5376

표 5 각 구성 요소에 대한 절제 연구. 각 경우의 최고 성능은 굵게 표시됨
Table 5 Ablation studies for each component. Note that the best scores are highlighted in bold

	Solar-energy			Electricity			Traffic		
	RRSE	CORR	NRMSE	RRSE	CORR	NRMSE	RRSE	CORR	NRMSE
DTSNet	**0.5388**	**0.8479**	**0.9817**	**0.2992**	**0.8355**	**0.5247**	**0.5830**	**0.8346**	**0.5339**
- PE	0.6093	0.8344	1.1102	0.3068	0.8347	0.5380	0.6411	0.7989	0.5870
- CA	0.7920	0.6418	1.4431	0.3167	0.8269	0.5554	0.6122	0.8194	0.5606
- MultiHead	**0.5388**	0.8478	0.9816	0.3314	0.8028	0.5812	0.6062	0.8237	0.5551
- SkipAR	1.1629	0.0085	2.1188	0.4640	0.7510	0.8137	0.9686	0.5100	0.8869

표 6 제안된 디노이징 방법론의 효과. 각 경우의 최고 성능은 굵게 표시됨
Table 6 Effect of proposed denoising training method. Note that the best scores are highlighted in bold

	Solar-energy			Electricity			Traffic		
	RRSE	CORR	NRMSE	RRSE	CORR	NRMSE	RRSE	CORR	NRMSE
$\gamma = 0.00$ (TF)	0.6038	0.8334	1.1001	0.3156	0.8237	0.5534	0.6072	0.8230	0.5560
$\gamma = 0.03$	0.6041	0.8356	1.1006	0.3037	0.8258	0.5326	0.5936	0.8294	0.5436
$\gamma = 0.05$	0.6016	0.8281	1.0961	0.3088	0.8212	0.5416	0.5871	0.8321	0.5376
$\gamma = 0.07$	**0.5388**	**0.8479**	**0.9817**	**0.2992**	**0.8355**	**0.5247**	**0.5830**	**0.8346**	**0.5339**
$\gamma = 0.10$	0.5390	0.8477	0.9821	0.3358	0.8197	0.5889	0.5971	0.8268	0.5467

름을 통하여 높은 성능을 이끌어 낸 것이다. 또한 시계열 데이터의 각 시차에 대한 고유한 위치를 인코딩을 추가하는 PE와 변수 별 예측을 위해 여러 개의 헤드를 사용한 완전 연결 신경망 역시 시계열 예측에 긍정적인 영향을 주는 것을 확인할 수 있다.

6.2.2 노이즈 강도에 대한 효과

본 연구에서는 가우시안 노이즈를 이용한 디노이징 훈련 기법을 제안하여 티쳐 포싱 훈련 기법(TF)과 티쳐 포싱을 사용하지 않는 훈련 기법(w/o TF)의 장점을 모두 활용한다. 가우시안 노이즈 ε는 수식 (15)와 같이 노이즈 강도(γ)를 통해 조정된다. 디노이징 훈련 기법에 대한 효과와 최적의 노이즈 강도를 탐색하기 위해 γ에 따른 비교 실험을 진행한다.

Table 6은 최적의 노이즈 강도를 찾기 위한 노이즈 강도에 따른 비교 실험($\gamma = 0.00, 0.03, 0.05, 0.07, 0.10$) 결과를 나타낸다. $\gamma = 0.00$의 경우 노이즈 강도를 주지 않은 경우로서, TF 훈련 기법을 사용한 것과 동일하다. 주어진 시계열 데이터에 존재하는 변동성에 따라 최적의 노이즈 강도가 상이할 수 있지만, 우리는 경험적으로 3개의 시계열 데이터셋에 대해 $\gamma = 0.07$가 본 모형에 가장 적합함을 확인하였다. 디노이징 훈련 기법은 TF 훈련 기법을 기반으로 하기 때문에 노이즈가 적을 경우 TF 훈련 결과와 유사하거나 오히려 성능이 좋지 않다. 또한 노이즈가 너무 클 경우(0.10), 훈련을 악화시키는 영향이 있다. 하지만 시계열의 스케일에 비례하는 적절한 노이즈를 더할 경우, 노출 편향 문제를 완화하여 디노이징 훈련 기법의 장점인 안정적인 연속 예측을 수행할 수 있는 것을 확인할 수 있다.

7. 결 론

시계열 예측 연구 분야는 과거의 관측치로부터 주기성을 도출하여 미래의 시점을 예측하는 연구이다. 시계열 예측 연구 분야에서 시계열 예측은 다중 패턴 문제와 연속 예측의 어려움이라는 한계를 가지고 있다. 본 연구에서는 이를 인코더-디코더 간 시간 주의 기제 및 디코더 내의 인과 주의 기제 기법으로 다중 패턴 문제를 해결하고, 멀티 헤드를 이용한 독립적인 디코더 모듈을 통해 변수 별 특화된 모형을 수립한다. 또한 위치 인코딩과 디노이징 훈련법으로 연속 예측에 최적화된 DTSNet을 제안한다.

본 연구의 검증을 위하여, 시계열 분석용 공개 데이터셋 3가지를 사용하여 실험을 진행하였다. DTSNet은 실험 결과에서 시계열 분석에서 널리 쓰이는 기계 학습과 심층 학습 기법을 포함한 선행 연구들보다 높은 성능을 나타낸다. 또한 구성 요소 제거 실험과 디노이징 훈련 기법 실험 등 심층적인 비교 실험을 통해 제안하는 방법론의 우수성을 입증하였다.

DTSNet은 디노이징 훈련 기법 및 위치 인코딩과 이중 주의 기제 기법, 변수 별 세부 모형화를 통해 복잡한 주기성을 가지는 다변량 시계열의 연속 예측을 효과적으로 수행한다. 후속 연구로는 상대적으로 다량의 변량을 지닌 고차원 다변량 시계열 데이터의 변수를 효과적으로 모델링하는 연구를 진행할 수 있다. 또한 역할이 유사한 변수를 그룹화하여 모델링을 진행하는 것도 흥미로운 연구 주제가 될 것으로 사료된다.

References

[1] J. Zheng, C. Xu, Z. Zhang, and X. Li, "Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network," *2017 51st Annual Conference on Information Sciences and Systems (CISS)*, pp. 1-6, 2017.

[2] Y. Wu, J. M. H. Lobato, and Z. Ghahramani, "Dynamic covariance models for multivariate financial time series," *Proc. of the 30th International Conference on International Conference on Machine Learning*, Vol. 28 of *ICML'13*, pp. 558-566, 2013.

[3] Z. Liu and M. Hauskrecht, "A regularized linear dynamical system framework for multivariate time series analysis," *Proc. of the TwentyNinth AAAI Conference on Artificial Intelligence, AAAI'15*, pp. 1798-1804, 2015.

[4] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," *The 41st International*

ACM SIGIR Conference on Research amp; Development in Information Retrieval, SIGIR'18, pp. 95-104, 2018.

[5] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control. 5th Ed.*, John Wiley and Sons, 2015.

[6] V. M. Landassuri-Moreno, C. L. Bustillo-Hernández, J. J. CarbajalHernández, and L. P. S. Fernández, "Single-step-ahead and multi-step-ahead prediction with evolutionary artificial neural networks," *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 65-72, 2013.

[7] V. LE GUEN and N. THOME, "Shape and time distortion loss for training deep time series forecasting models," *Advances in Neural Information Processing Systems*, Vol. 32, pp. 4189-4201, 2019.

[8] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *Trans. Neur. Netw.*, Vol. 5, pp. 240-254, Mar. 1994.

[9] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," *arXiv preprint arXiv: 1703.04691*, 2018.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *3rd International Conference on Learning Representations, Conference Track Proc., ICLR'15*, 2016.

[11] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *Conference on Empirical Methods in Natural Language Processing, EMNLP'15*, pp. 1412-1421, 2015.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems*, Vol. 27, pp. 3104-3112, 2014.

[13] I. Melnyk and A. Banerjee, "Estimating structured vector autoregressive models," *Proc. of the 33rd International Conference on International Conference on Machine Learning*, Vol. 48 of *ICML'16*, pp. 830-839, 2016.

[14] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," Philosophical Transactions of the Royal Society A: Mathematical, *Physical and Engineering Sciences*, 2013.

[15] T. SJ and L. B., "Forecasting at scale," *The American Statistician*, Vol. 72, No. 1, pp. 37-45, 2018.

[16] A. M. D. Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American Statistical Association*, Vol. 106, No. 496, pp. 1513-1527, 2011.

[17] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: a generic approach for extreme condition traffic forecasting," *Proc. of the 17th SIAM International Conference on Data Mining, (SDM)*, pp. 777-785, 2017.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, Vol. 9, pp. 1735-1780, Nov. 1997.

[19] Y.-Y. Chang, F.-Y. Sun, Y.-H. Wu, and S.-D. Lin, "A memory-network based solution for multivariate time-series forecasting," *arXiv preprint arXiv: 1809.02105*, 2018.

[20] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," *Proc. of the 24th International Conference on Artificial Intelligence, IJCAI'15*, pp. 2327-2333, 2015.

[21] M. A. Zaytar and C. E. Amrani, "Sequence to sequence weather forecasting with long short-term memory recurrent neural networks," *International Journal of Computer Applications*, Vol. 143, pp. 7-11, Jun. 2016.

[22] Z. Mariet and V. Kuznetsov, "Foundations of sequence-to-sequence modeling for time series," *Proc. of Machine Learning Research*, Vol. 89, pp. 408-417, Apr. 2019.

[23] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724-1734, Oct. 2014.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, Vol. 30, pp. 5998-6008, 2017.

[25] S. Huang, D. Wang, X. Wu, and A. Tang, "Dsanet: Dual self-attention network for multivariate time series forecasting," *Proc. of the 28th ACM International Conference on Information and Knowledge Management, CIKM'19*, pp. 2129-2132, 2019.

[26] G. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, Vol. 50, pp. 159-175, 2003.

[27] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," *Advances in Neural Information Processing Systems*, Vol. 28, pp. 2377-2385, 2015.

[28] Y. Keneshloo, T. Shi, N. Ramakrishnan, and C. K. Reddy, "Deep reinforcement learning for sequence-to-sequence models," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 31, No. 7, pp. 2469-2489, 2020.

[29] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, Vol. 1, No. 2, pp.

270-280, 1989.
[30] T. Mihaylova and A. F. T. Martins, "Scheduled sampling for transformers," *Proc. of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 351-356, Jul. 2019.
[31] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *4th International Conference on Learning Representations, Conference Track Proc., ICLR'16*, 2016.
[32] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," *Advances in Neural Information Processing Systems*, Vol. 28, pp. 1171-1179, 2015.
[33] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance," *Climate Research*, Vol. 30, No. 1, pp. 79-82, 2005.
[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, Conference Track Proc., ICLR'15*, 2015.

홍 정 수
2019년 이화여자대학교 컴퓨터공학과(학사). 2019년~현재 연세대학교 컴퓨터과학과 석사과정. 관심분야는 빅데이터마이닝 & 기계 학습

박 진 욱
정보과학회논문지
제 48 권 제 3 호 참조

이 지 은
정보과학회논문지
제 48 권 제 3 호 참조

김 경 훈
2020년 경희대학교 응용수학과, 컴퓨터공학과(학사). 2020년~현재 연세대학교 컴퓨터과학과 석사과정. 관심분야는 기계학습 & 자연언어처리

홍 승 균
1993년 연세대학교 물리학과(학사). 2018년 연세대학교 공학대학원 컴퓨터공학과(석사). 1993년~1995년 POSCO, 1995년~2020년 SKTelecom. 2021년~현재 연세대학교 컴퓨터과학과 박사과정. SK Innovation MySUNI 전문교수단 AI/DT 담당교수. 관심분야는 빅데이터, 시계열데이터 비정상 감지 모델

박 상 현
정보과학회논문지
제 48 권 제 3 호 참조