Deep Mobile Trajectory Prediction with Shift-and-Join and Carry-Ahead: Multi-step User Mobility Prediction

2022-01-12 09:30AM (KST), 2022-01-11 07:30PM (EST)

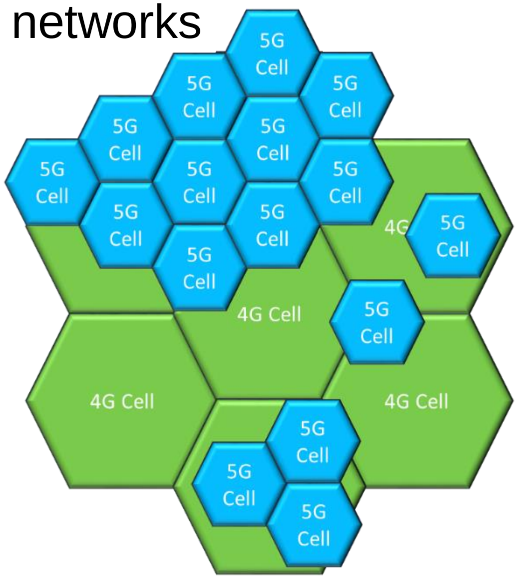
Huigyu Yang
Sungkyunkwan University
huigyu@skku.edu

Presentation Outline

- Introduction
- Multi-step Prediction Approaches
- Proposed Models
  - Step Forward Iteration model
  - Encoder Decoder model
- Dataset
- Experimental results

Introduction (1/2)

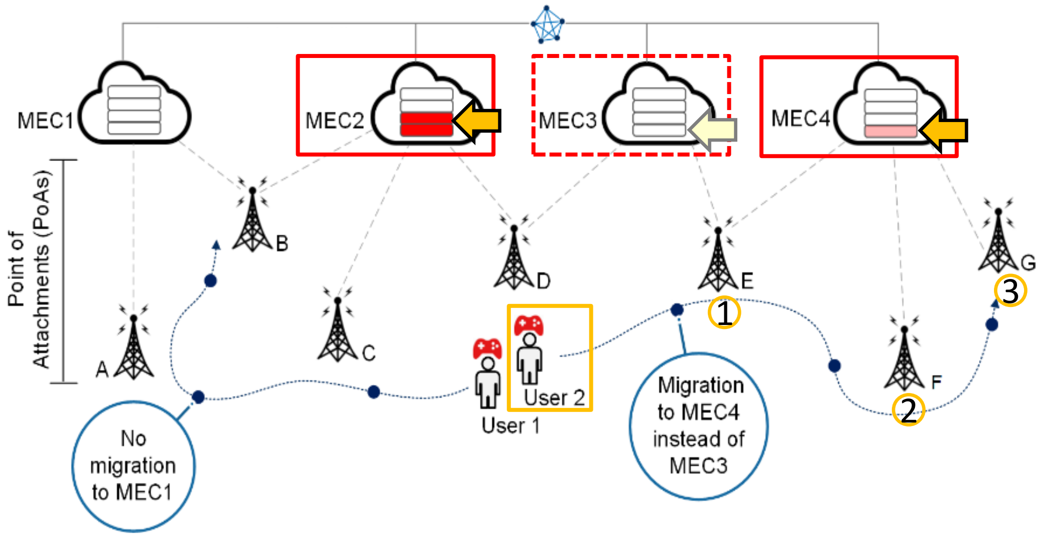
- Mobility
  - It is a characteristic of users or their devices in mobile networks
  - It effects issues in mobile communication systems
    - ★ Handover
    - ★ Dimensioning of signaling network
    - ★ User location updating



- Increasing importance of mobility
  - Impact of user mobility is enlarged with decrease of cell coverage radius
    - ★ Small cell networks becoming trend of 5G mobile system
  - Future applications and services require ultra-low latency mobility

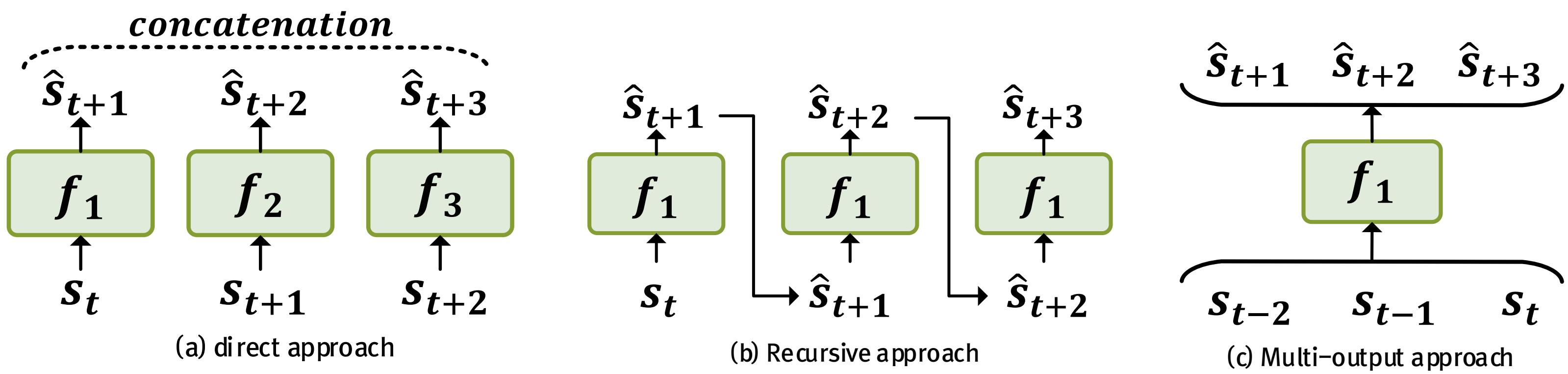
Introduction (2/2)

- Multi-access Edge Computing(MEC) is the key enabler of 5G that moves network functions and services to network edge for reducing transmission delay
- MEC moves services at the network edge to reduce transmission delay between user and services
- The services in MEC servers require to be migrated with user mobility to maintain the service quality in terms of the delay



Multi-step Prediction Approaches (1/4)

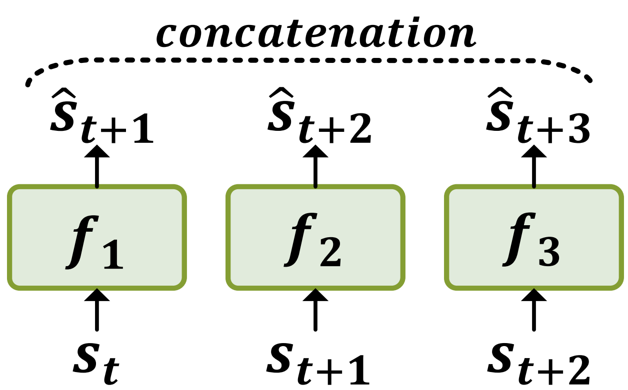
- Three general approaches for multi-step-ahead prediction using Deep Learning (DL)
  - Direct approach
  - Recursive approach
  - Multi output approach



Multi-step Prediction Approaches (2/4)

- Direct approach
  - Multi-step predictions are obtained by **concatenating** prediction of each time-step in this approach
  - It does not suffer from accumulating errors
- Weaknesses
  - Dependencies between two steps are not modeled
  - Larger computation cost
    - ★ It depends on size of prediction time-step
- This approach is not used because of poor performance in modeling dependencies between two different steps

s : input (ground truth)
$\hat{s}$: output (prediction)
t : time-step

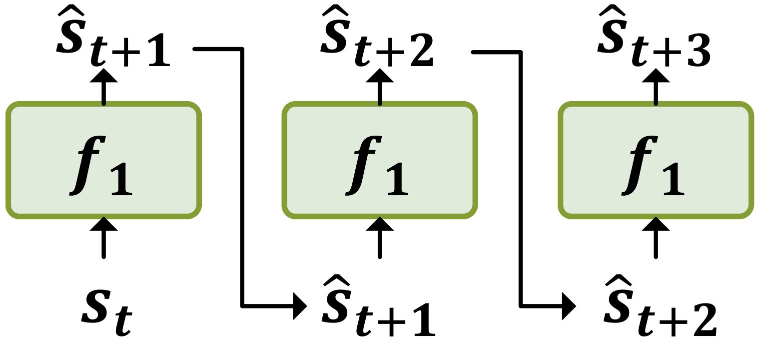


(a) direct approach

Multi-step Prediction Approaches (3/4)

- Recursive approach
  - It predicts multi-step by **repeatedly passing** its predictions
  - Prediction at one time-step is used as next time-step input
- Weaknesses
  - Mismatch between optimization of single-step prediction and multi-step prediction
  - Accumulating errors

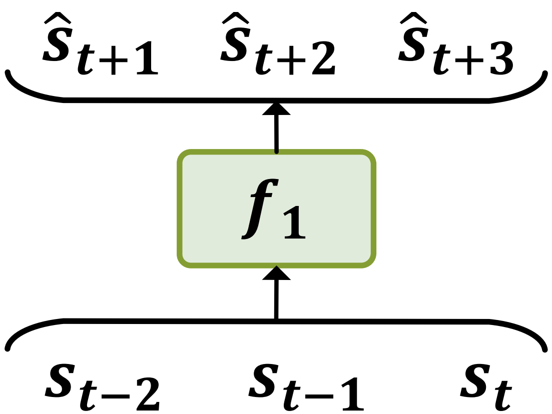
s : input (ground truth)
$\hat{s}$: output (prediction)
t : time-step



(b) Recursive approach

Multi-step Prediction Approaches (4/4)

- Multi output approach
  - Contrary to recursive model, It uses actual observed data rather than predicted data
  - Model that is able to predict simultaneously is required
  - It does not suffer from accumulating errors
  - It can learn dependency between inputs and outputs
- Weaknesses
  - It is more complex than recursive model
  - It requires more training data to avoid over-fitting

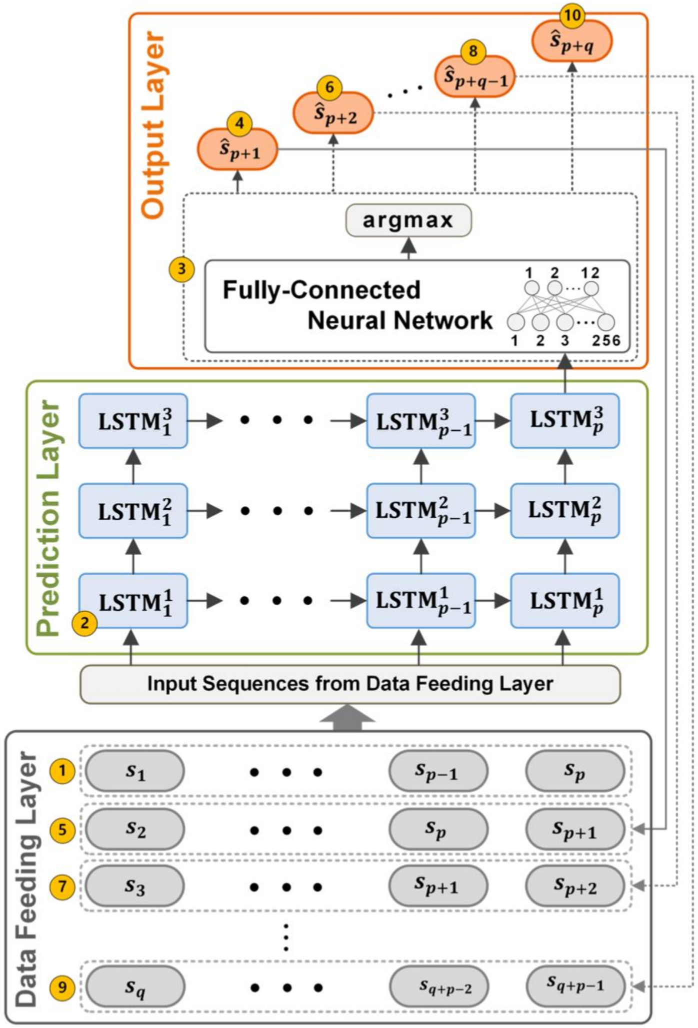


s : input (ground truth)
$\hat{s}$: output (prediction)
t : time-step

(c) Multi−output approach

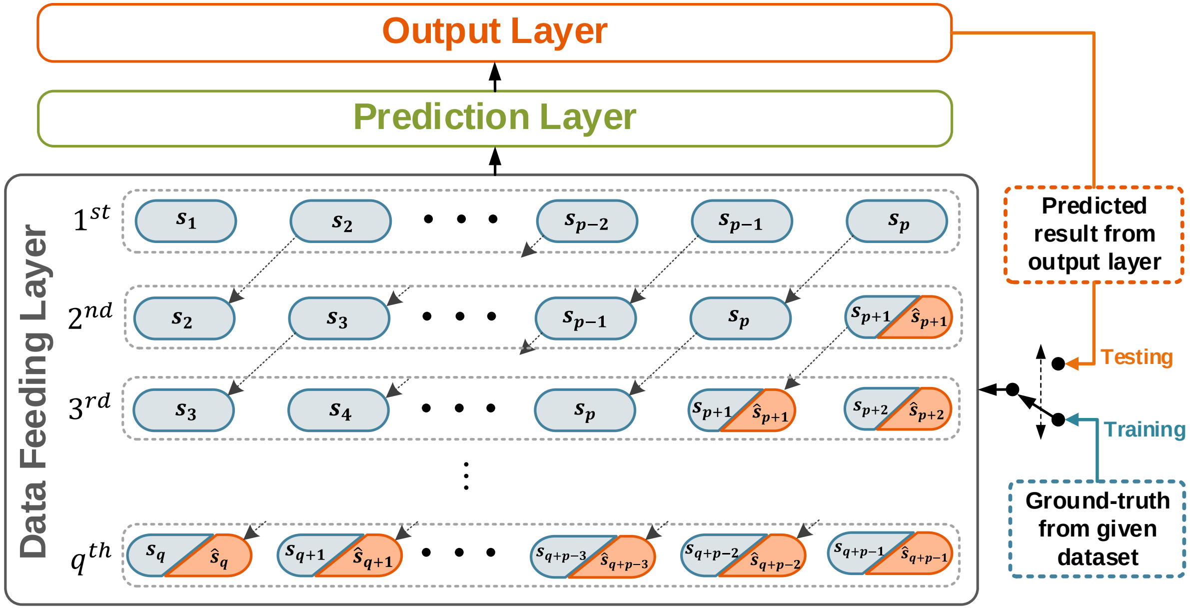
Proposed Models (1/6)
Step Forward Iteration Model (1/3)

- SFI model consist of Data Feeding Layer, Prediction Layer, and Output Layer
- Prediction Layer has three sublayers of LSTM cells, and Output Layer has on Fully-Connected(FC) neural network with argmax function
- Step Forward Iteration (SFI) model is an evolution of our previous single-step prediction model
- Compare to the single-step model, Data Feeding Layer is added that feeds data to the model recursively



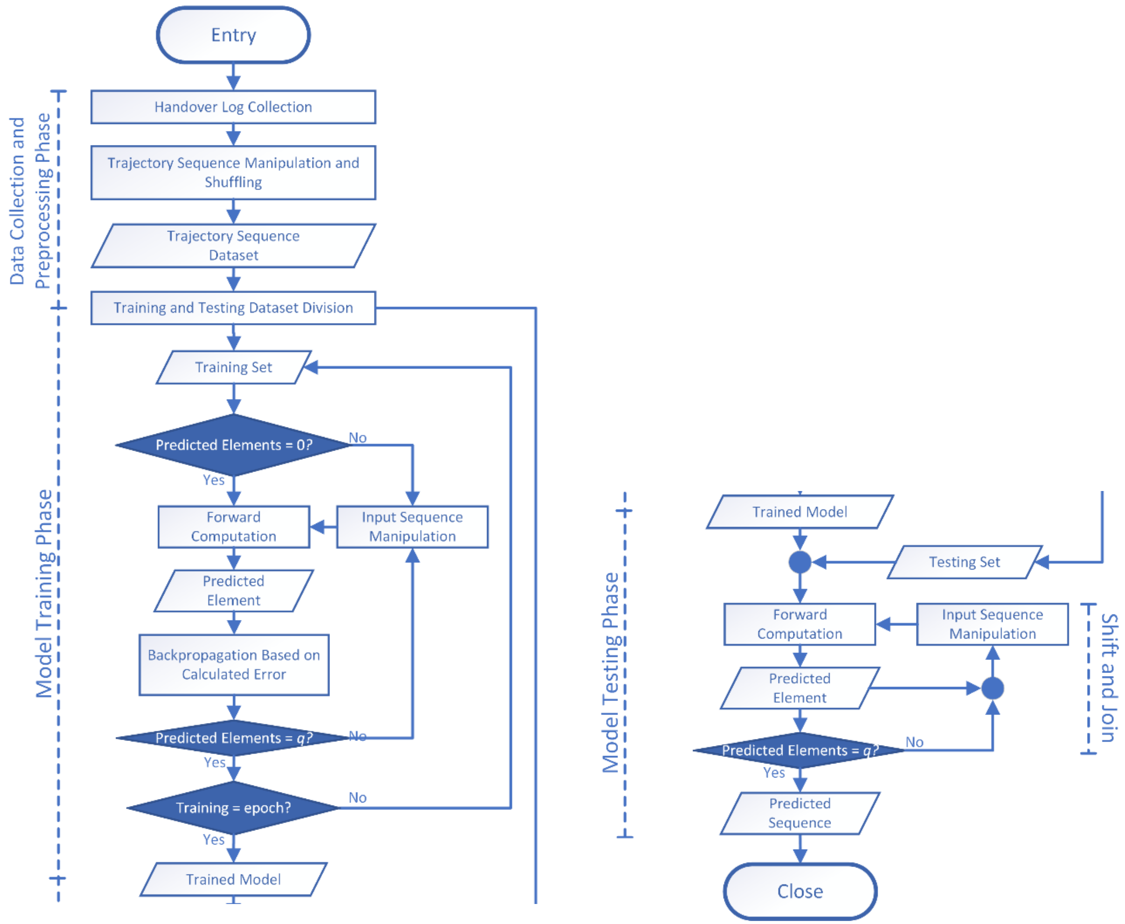
Proposed Models (2/6)
Step Forward Iteration Model (2/3)

- Data Feeding Layer manipulates the elements of input sequences to predict step-forwarded result
- In training and testing phase, different source of data is used for input sequence manipulating
  - Ground-truth values are only used in training phase to train the model efficiently



Proposed Models (3/6)
Step Forward Iteration Model (3/3)

- The operational flowchart of proposed SFI model



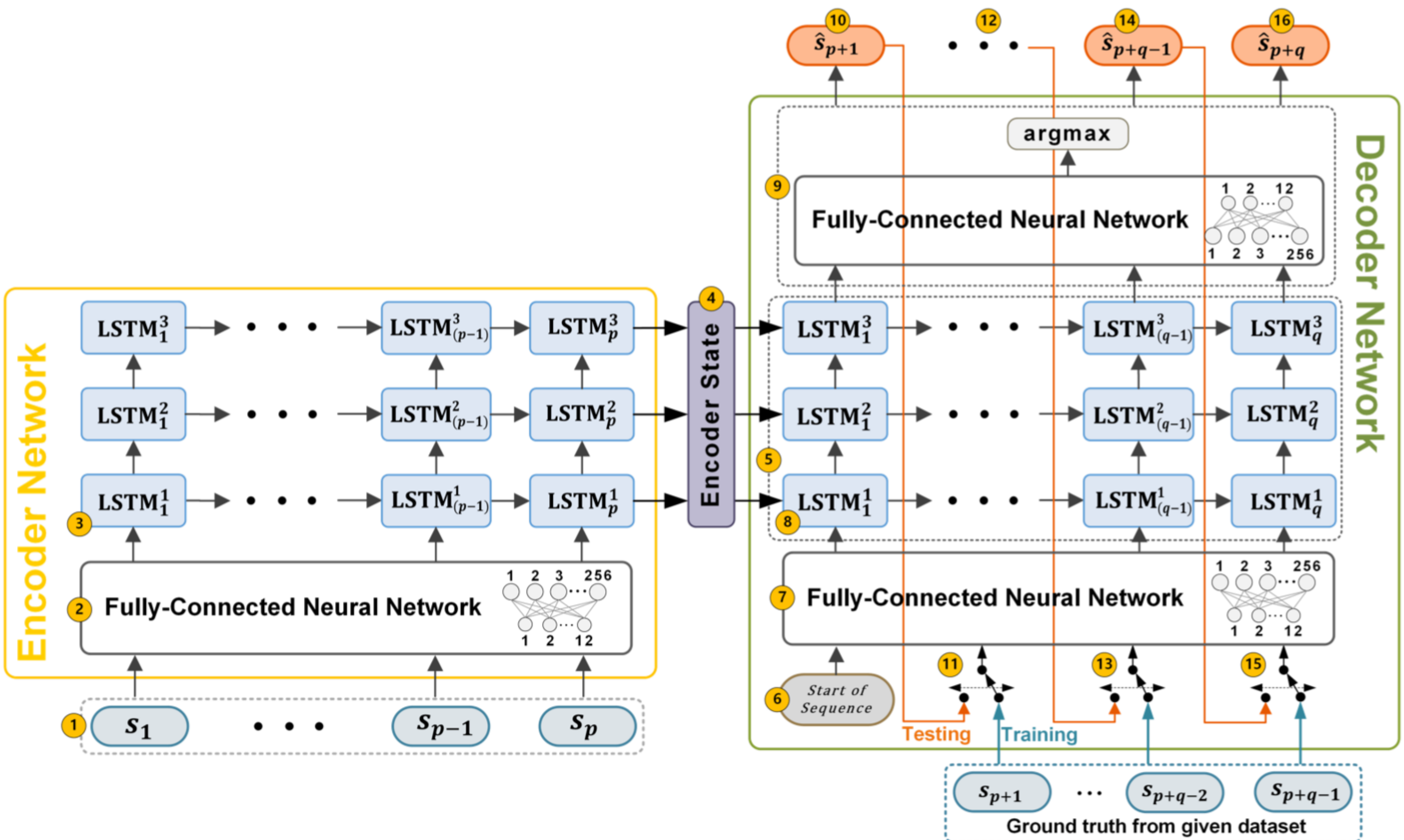
Proposed Models (4/6)
Encoder Decoder Model (1/3)

- ED model consists of Encoder network and Decoder network
- The role of each network is processing the input sequences and generating output sequences, respectively

- Encoder network converts input sequence to fixed length vector which is called Encoder state
- Encoder state is used as initial values in Decoder network, and this enables Decoder network to output historical data-based prediction
- Decoder network generate output sequences by recursively computing elements of the sequences

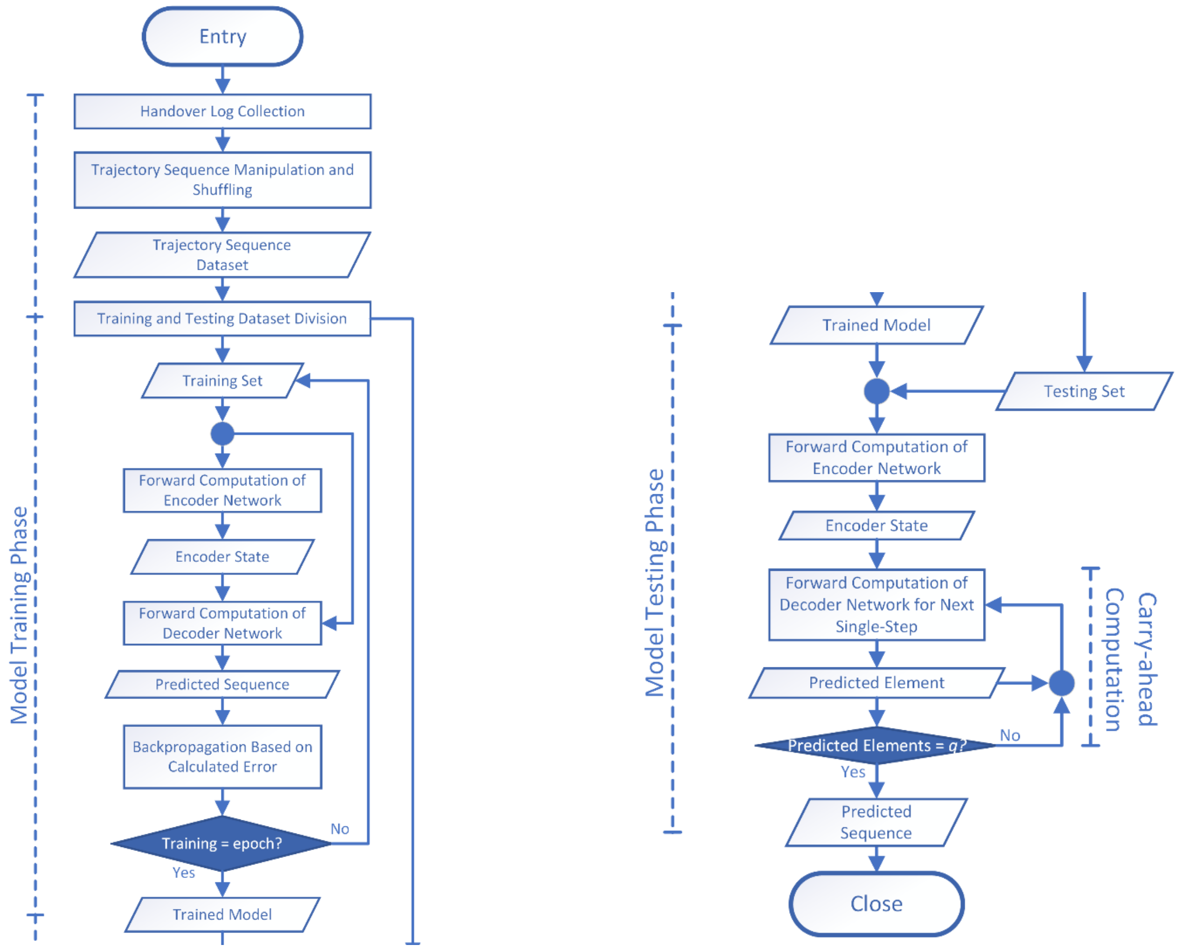
- Because of recursive computations in Decoder network, ground-truth values are used in training phase

Proposed Models (5/6)
Encoder Decoder Model (2/3)



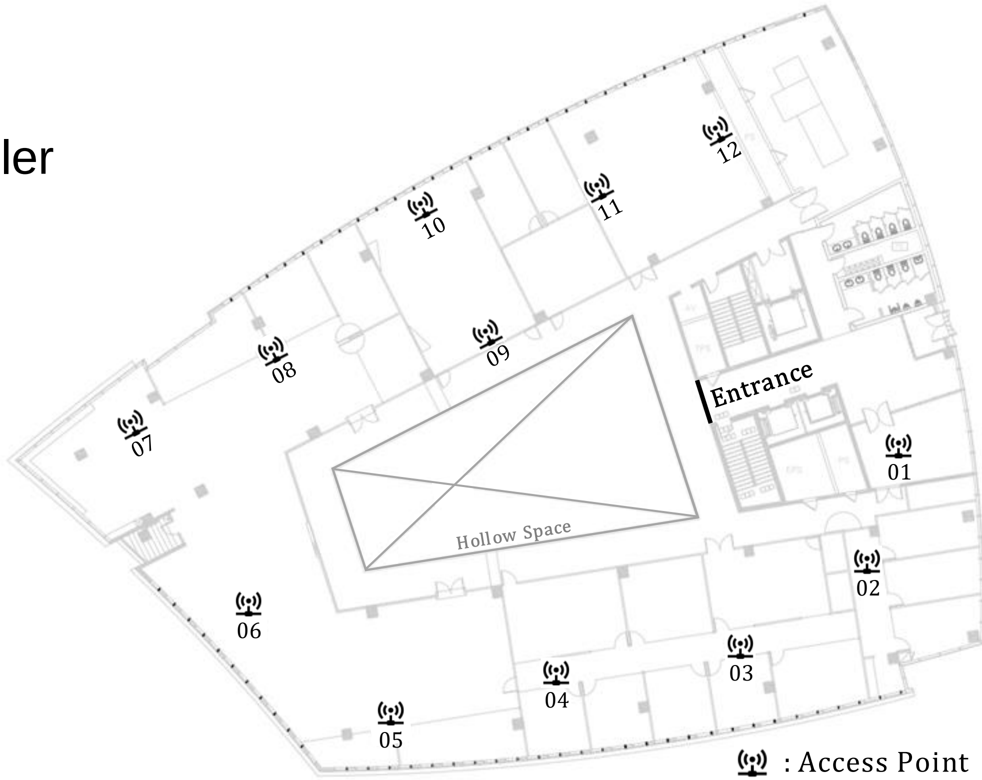
Proposed Models (6/6)
Encoder Decoder Model (3/3)

■ The operational flowchart of proposed ED model



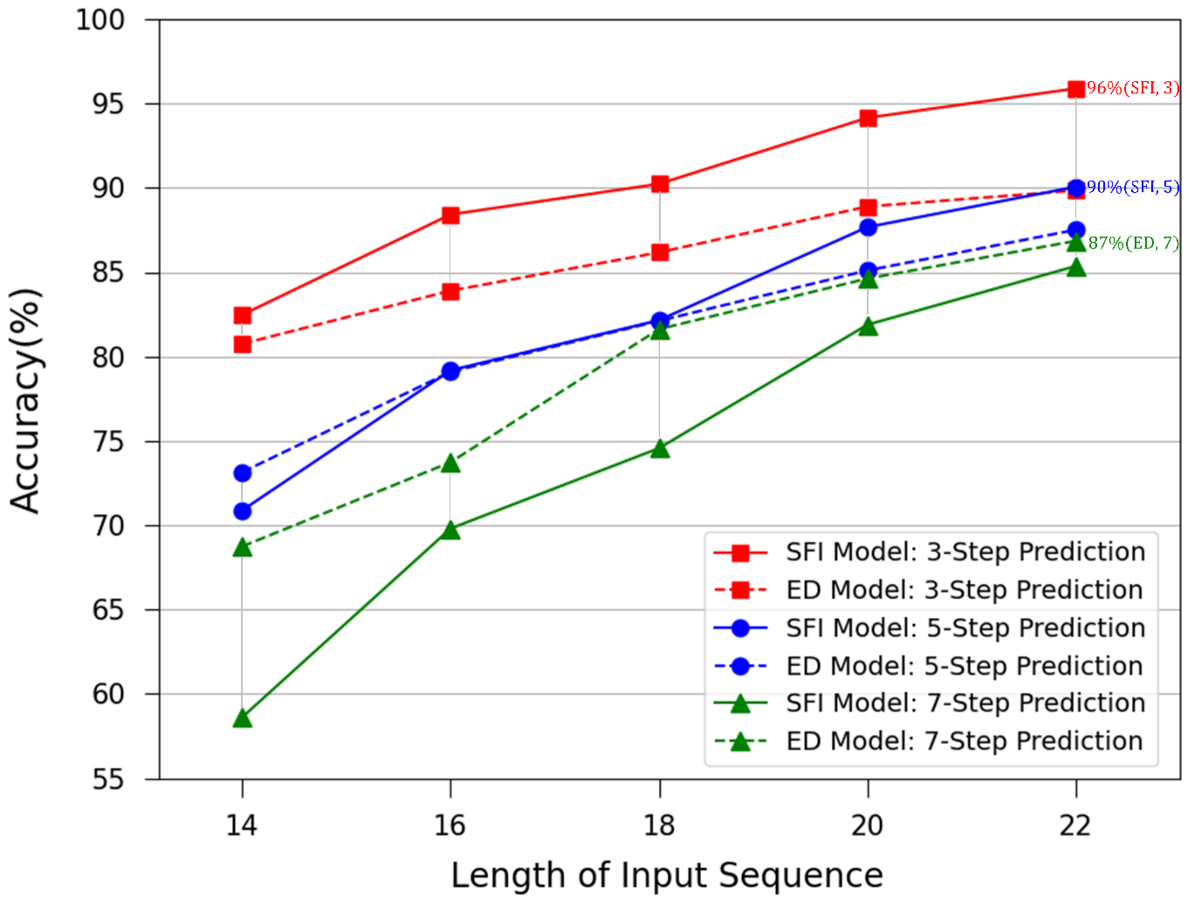
Dataset

- Our mobility dataset is collected from wireless network
  - Intelligent ICT Convergence Research Center
- Cisco Access Point (AP)
  - 12 APs with proprietary controller
  - It has logging functionality
- Roaming Log Message
  - Time of occurrence
  - Terminal id
  - Source AP number
  - Destination AP number



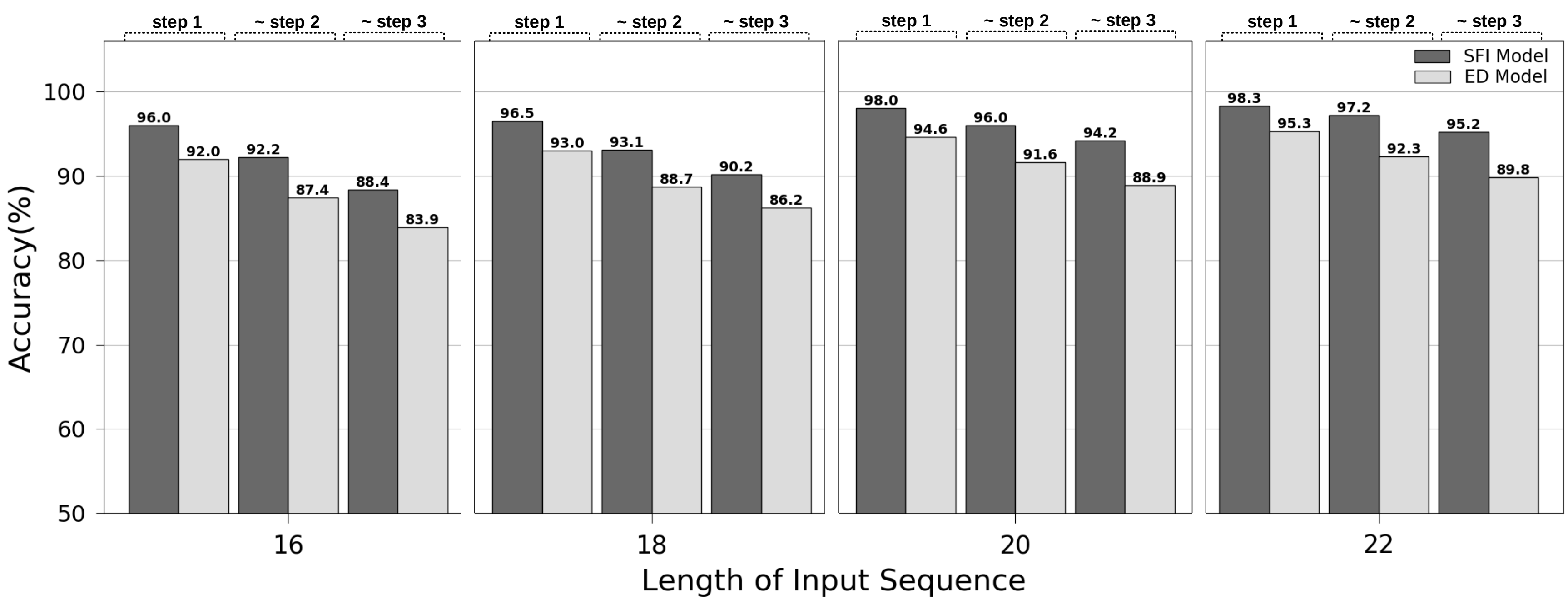
Experimental Results (1/6)

■ The overall accuracy of each model in different prediction steps



Experimental Results (2/6)

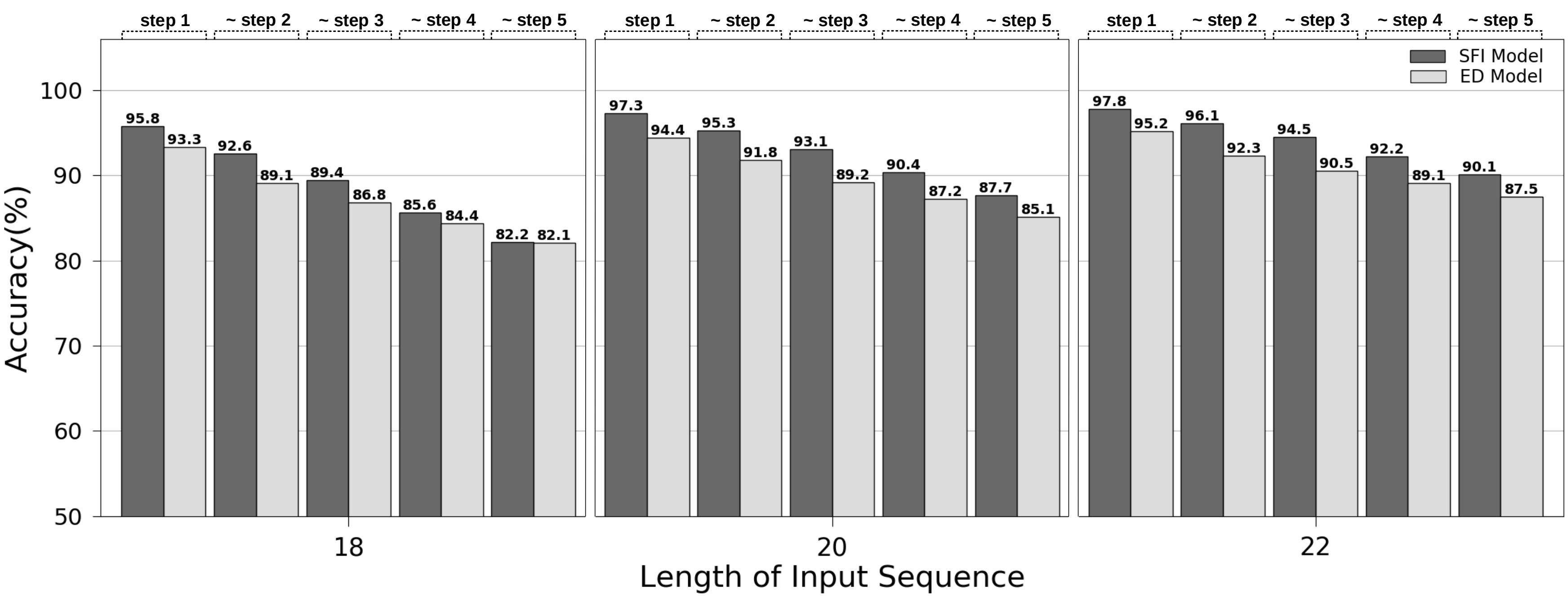
- The accuracy of individual step in each model with different length of input sequence and prediction steps



(a) 3-step prediction

Experimental Results (3/6)

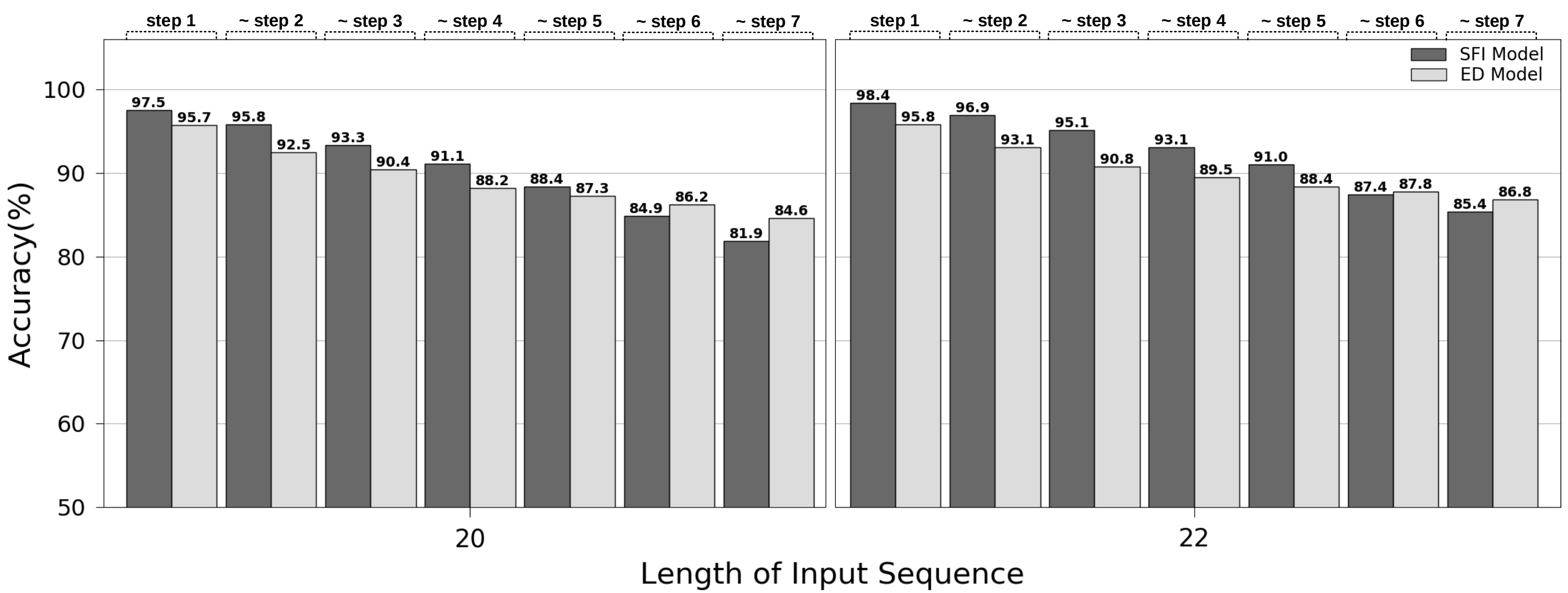
- The accuracy of individual step in each model with different length of input sequence and prediction steps



(b) 5-step prediction

Experimental Results (4/6)

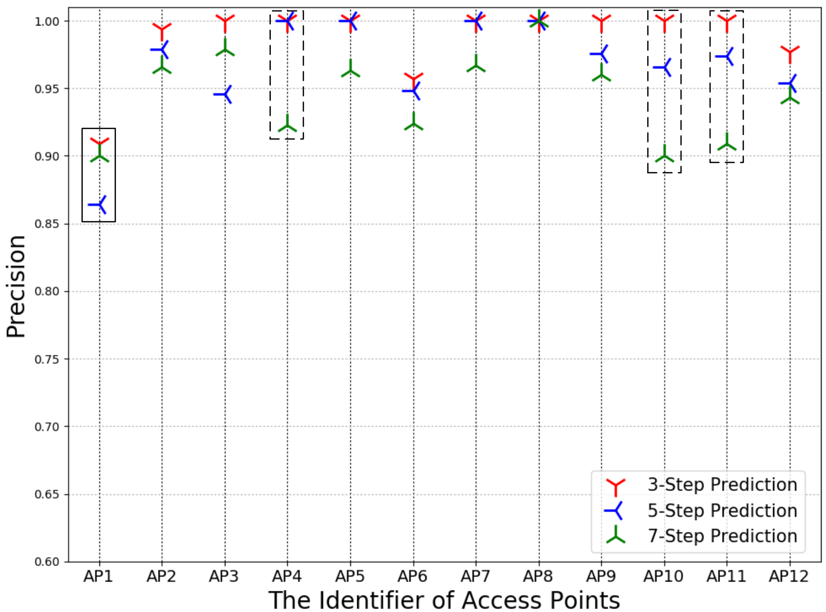
- The accuracy of individual step in each model with different length of input sequence and prediction steps



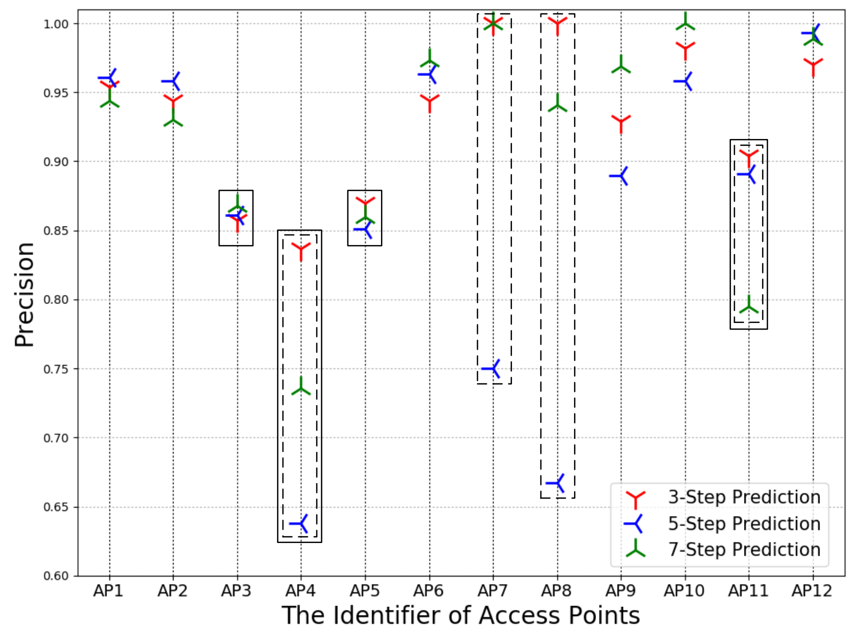
(c) 7-step prediction

Experimental Results (5/6)

- AP precision of each model in different prediction steps



(a) AP precision in SFI model



(b) AP precision in ED model

Experimental Results (6/6)

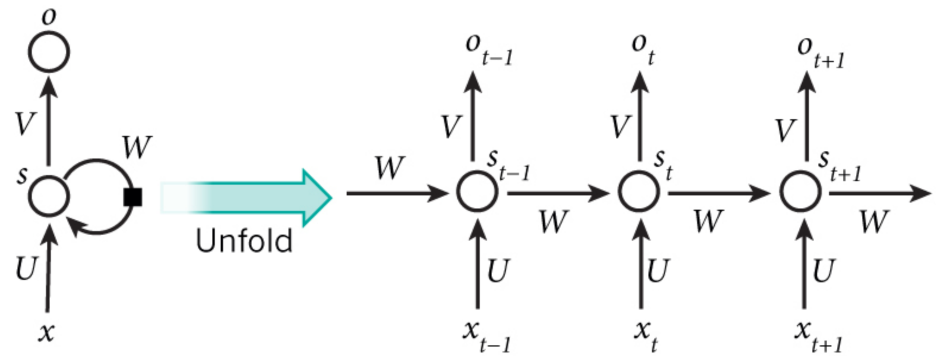
- Training and testing time comparison of two model in different prediction steps

The number of steps in predicted sequence	Training Time (sec)		Total Testing Time (sec)		Testing Time per Sequence (sec)	
	SFI Model	ED Model	SFI Model	ED Model	SFI Model	ED Model
3	151.85	67.63	0.64	137.23	0.0007	0.1471
5	133.43	64.54	0.72	177.27	0.0009	0.2098
7	126.70	62.25	0.79	210.85	0.0011	0.2731

Appendix: Long-Short-Term Memory (1/2)

- Recurrent Neural Networks (RNN)
  - It shows good performance at handling sequential data

- The structure of RNN
  - X_t: input of time step t
  - S_t: hidden state of time step t
  - O_t: output value of time step t



- Back-Propagation in RNN for training
  - Main difference is gradient of weights are added after each time step
  - RNN uses Back-Propagation Through Time (BPTT) algorithm

Appendix
Long-Short-Term Memory (2/2)

- Long-Short Term Memory (LSTM)
  - It solves gradient vanishing problem by adding cell state

- Structure of LSTM
  - x_t represents Inputs
  - c_t represents Cell state
    - c_{t-1} represents previous cell state
  - h_t represents Hidden state
    - h_{t-1} represents previous hidden state
  - f_t represents Forget gate
  - i_t represents Input gate
  - o_t represents Output gate