Portfolio Optimization-Based Stock Prediction Using

Long-Short Term Memory Network in

Quantitative Trading

REVIEW

제목

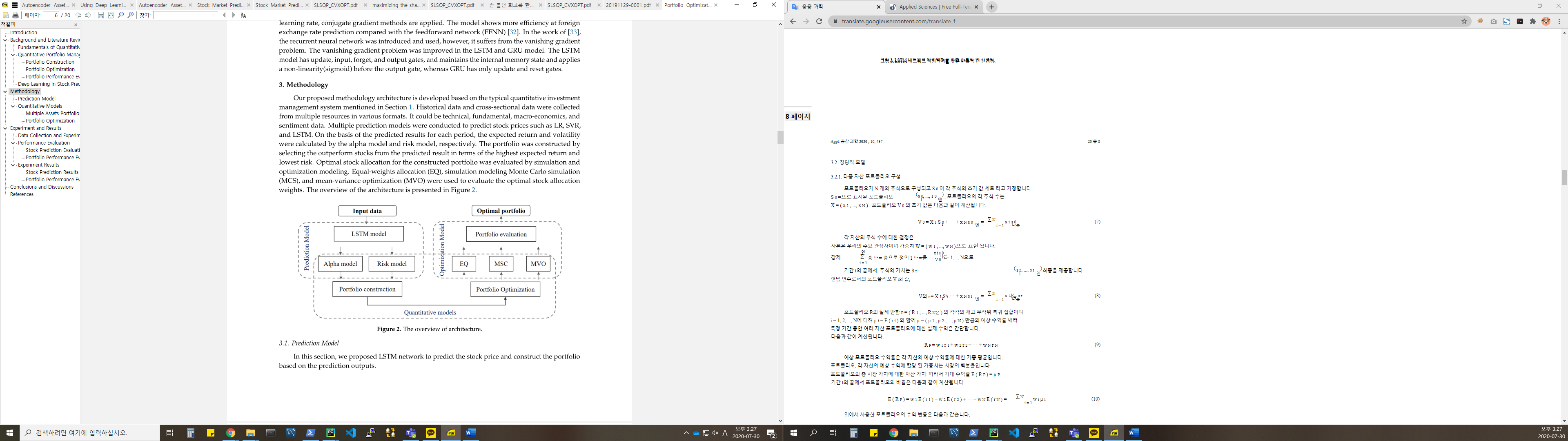
Quantitative Trading에 LSTM 모델을 사용한 포트폴리오 최적화 기반 주식 예측

초록

LSTM

모델에는 업데이트, 입력, 잊어 버림 및 출력 게이트가 있으며 내부 메모리 상태를 유지하고 적용

핵심 아키텍쳐



실험 연구

Data : S & P 500의 2008.01.01 ~ 2018.01.01 (2516)

Open-High-Low-Close & Volume

OS : Ubuntu OS.

HW : 64GB RAM, GeForce GTX 1080 Ti 11176MB GPU, Intel Core i7-7700 (3.60GHz) CPU.

Stack : Python 3.6, Keras, TensorFlow 백엔드

Optimizer : Adam

2-Hidden LSTM Layer  
1~600 (256, 512) Neuron

Train-Test : 80:20

LSTM 3차원적 데이터 입력 가능! Batch 설정 가능!

[data\_size, time\_steps, features] -> input\_shape에는 (time\_steps, features)

Data\_size는 데이터의 크기

Time\_steps은 Period가 된다 과거에 몇 개의 데이터를 볼 것인가

Features는 X의 차원. 속성의 수

Batch를 사용하면 종목의 속성 수를 다 때려넣고 구분할 수도 있을 듯!

Batch는 일괄 처리되는 작업의 양이고 data\_size를 한 번에 처리하는 개수를 의미

Data\_size/batch\_size 일 때 나머지가 없어야 한다!

따라서, batch에서 input\_shape는 (batch\_size, time\_steps, features)

속성의 수가 5개인 4개 종목이 입력되었을 때

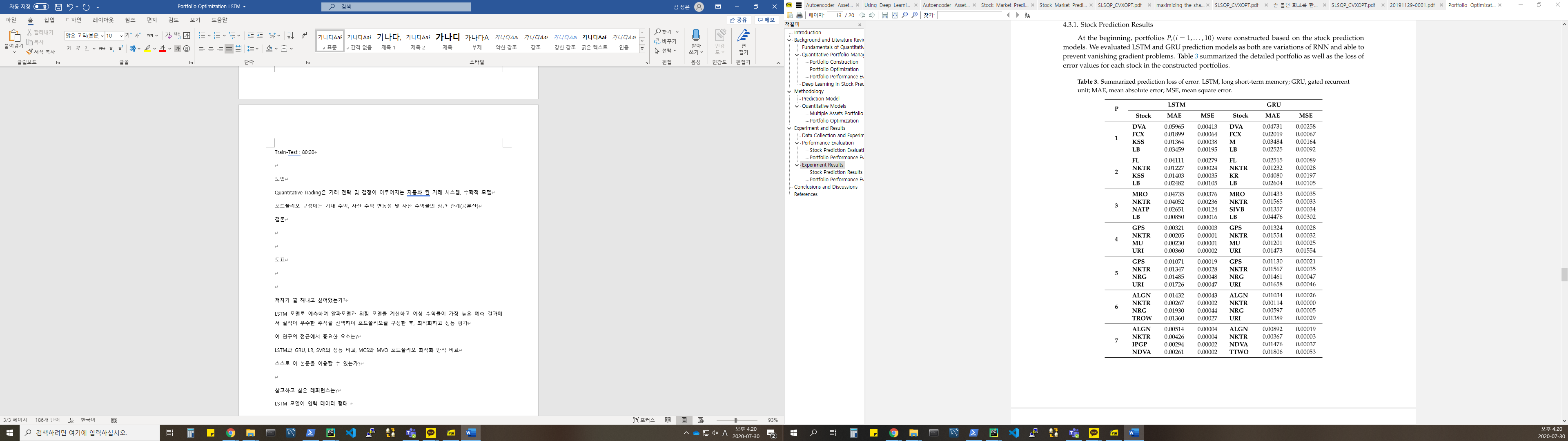
일괄 처리되는 작업의 양은 5가 되어야 하므로 (5, period, 5) 가 되어야 하지 않을까!!

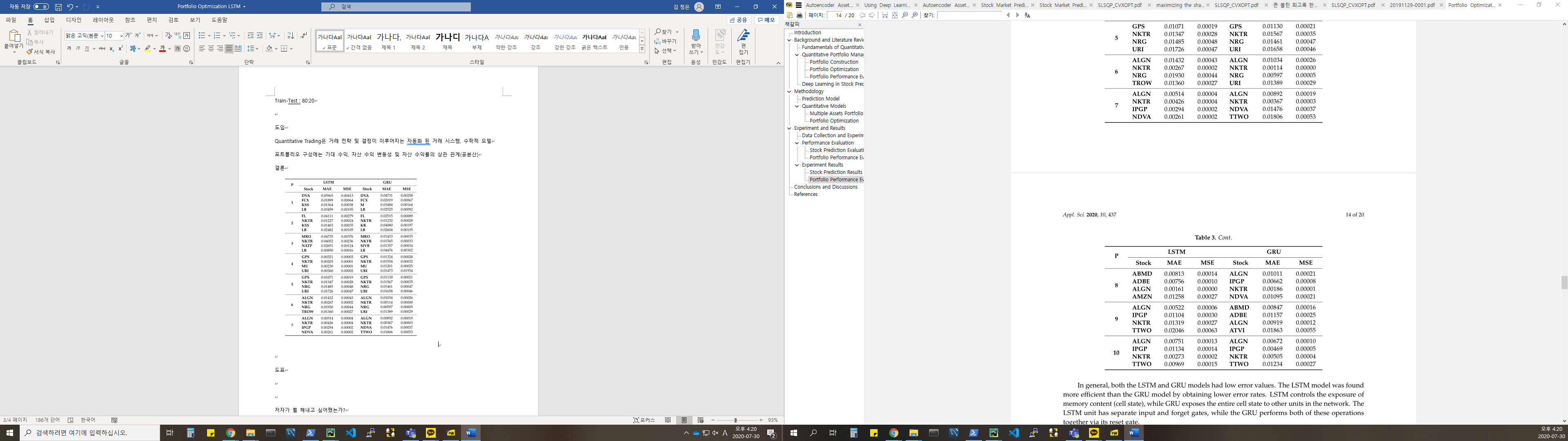
도입

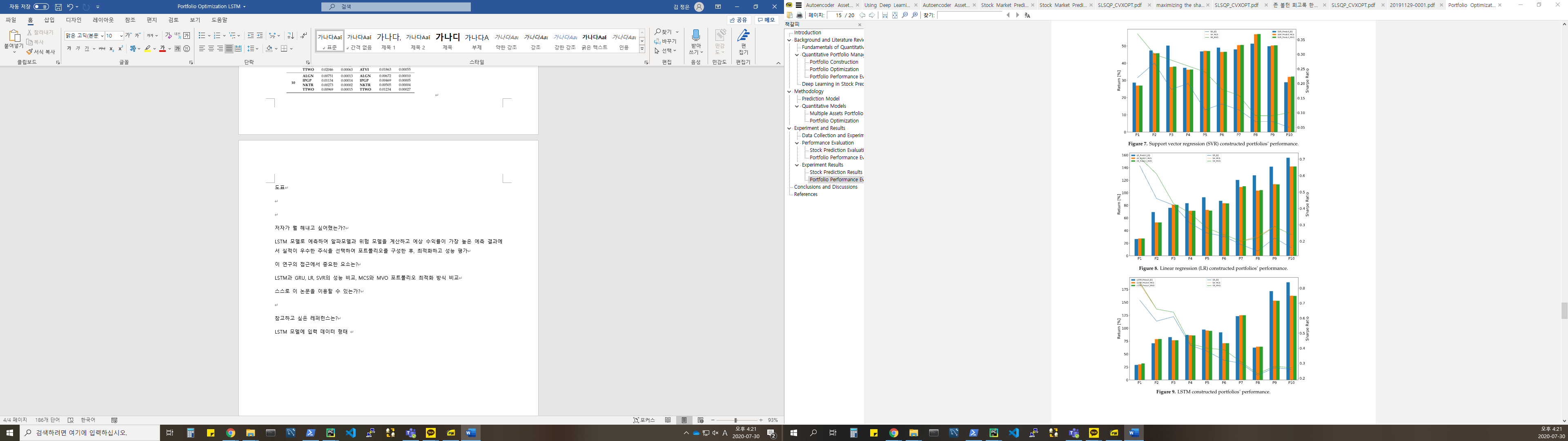
Quantitative Trading은 거래 전략 및 결정이 이루어지는 자동화 된 거래 시스템, 수학적 모델

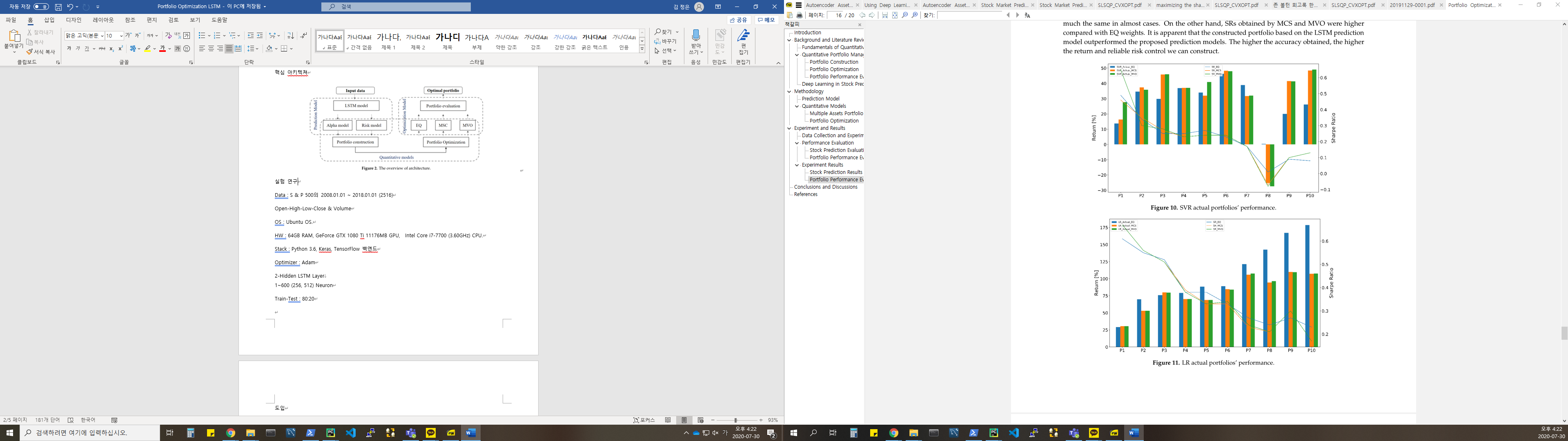
포트폴리오 구성에는 기대 수익, 자산 수익 변동성 및 자산 수익률의 상관 관계(공분산)

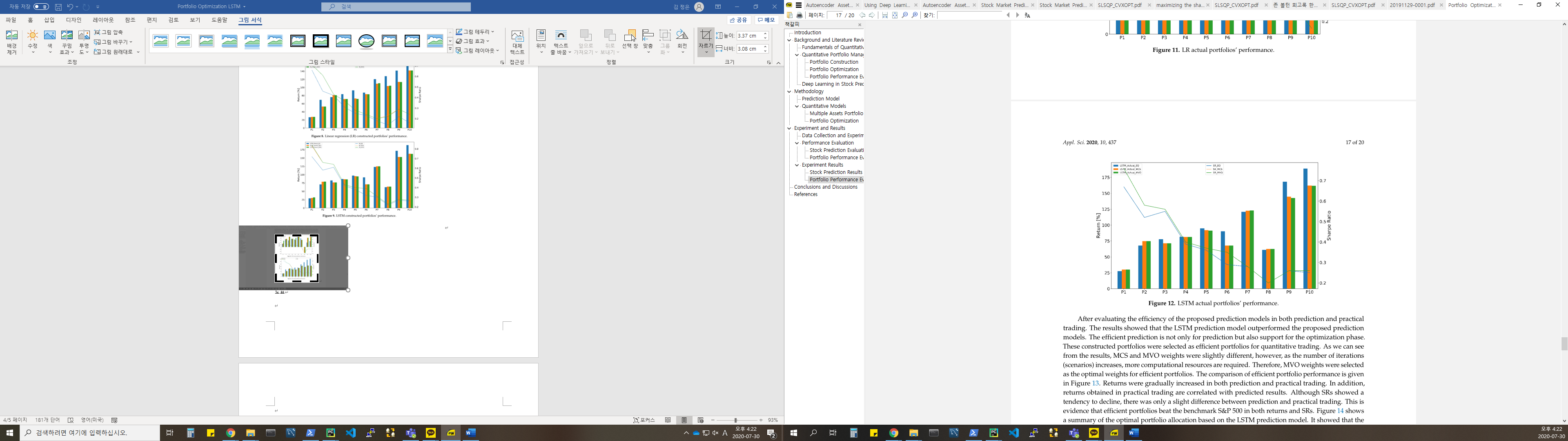
결론

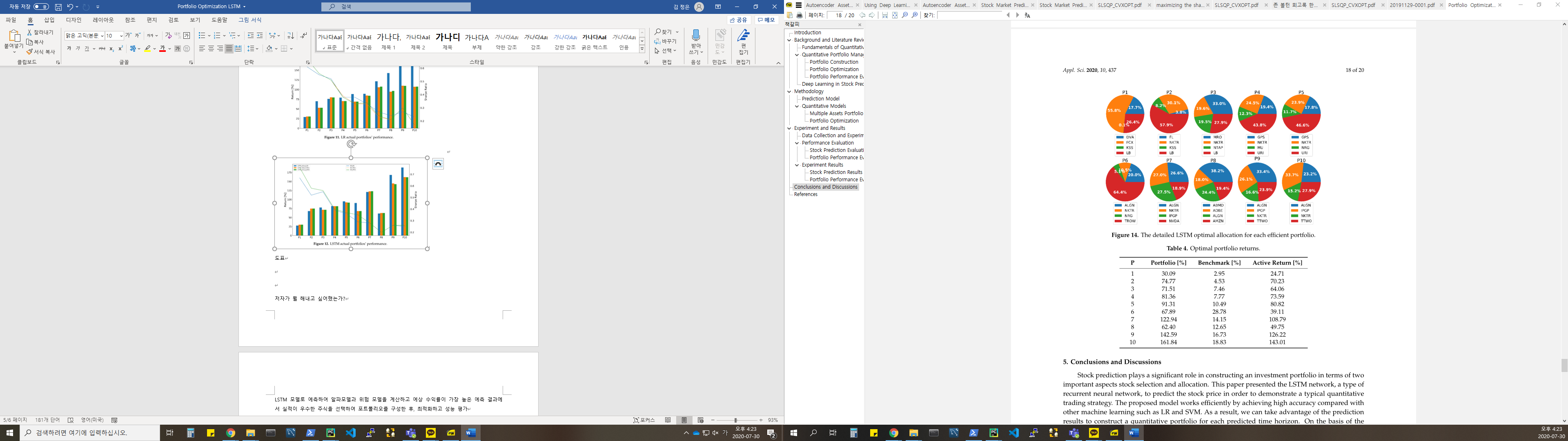












도표

저자가 뭘 해내고 싶어했는가?

LSTM 모델로 예측하여 알파모델과 위험 모델을 계산하고 예상 수익률이 가장 높은 예측 결과에서 실적이 우수한 주식을 선택하여 포트폴리오를 구성한 후, 최적화하고 성능 평가

이 연구의 접근에서 중요한 요소는?

LSTM과 GRU, LR, SVR의 성능 비교, MCS와 MVO 포트폴리오 최적화 방식 비교

스스로 이 논문을 이용할 수 있는가?

LSTM 모델의 입력 데이터 형태를 알 수 없음, 알게 되면 가능할 듯

LSTM 모델의 입력에 다차원 shape가 가능한가?

참고하고 싶은 레퍼런스는?

LSTM 모델에 입력 데이터 형태