

Week 8 Meeting Note

Suggestion from last meeting

- Using GRU
- Using one direction not bidirection
- Train the generator more times
- Make the structure larger

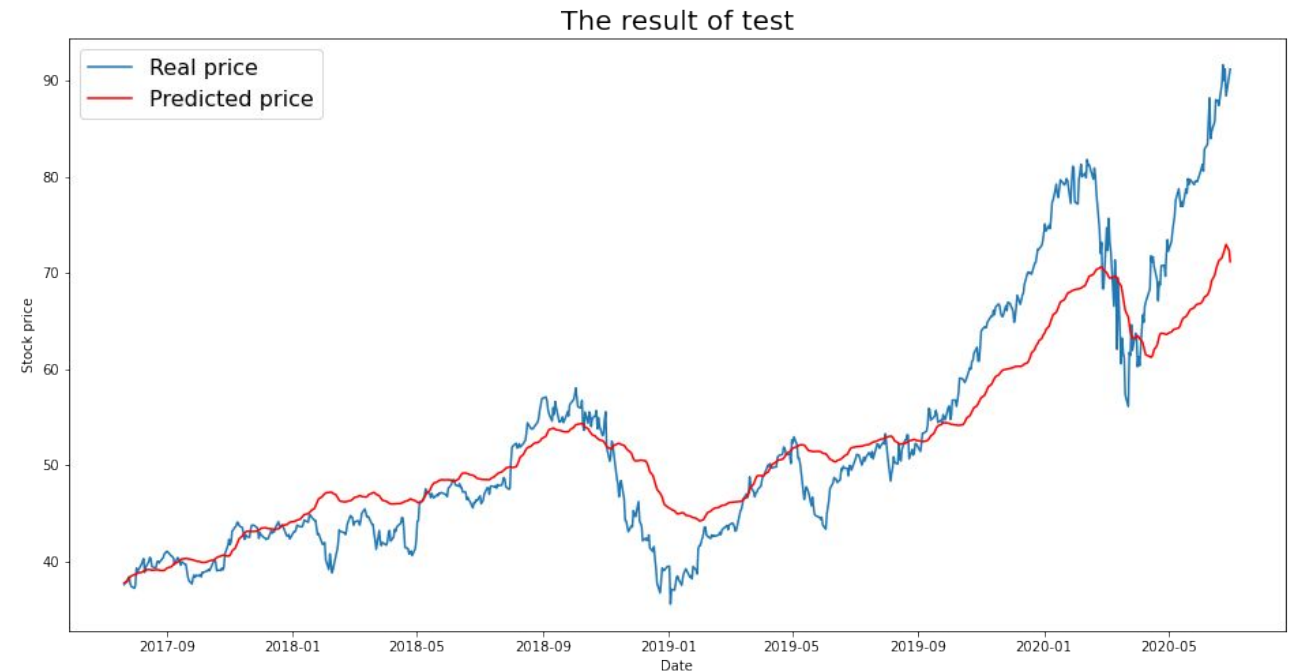
-> For the following model:

We set GRU as a Generator in the GAN

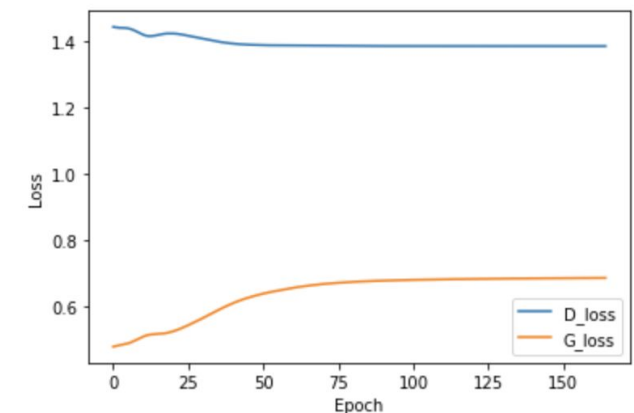
**Input 30, output 3*

Basic GAN (Train data)

Layer (type)	Output Shape	Param #
gru_44 (GRU)	(None, 30, 1024)	3262464
gru_45 (GRU)	(None, 30, 512)	2362368
gru_46 (GRU)	(None, 256)	591360
dense_78 (Dense)	(None, 128)	32896
dense_79 (Dense)	(None, 64)	8256
dense_80 (Dense)	(None, 3)	195
conv1d_36 (Conv1D)	(None, 17, 32)	128
conv1d_37 (Conv1D)	(None, 9, 64)	10304
conv1d_38 (Conv1D)	(None, 5, 128)	41088
flatten_12 (Flatten)	(None, 640)	0
dense_81 (Dense)	(None, 220)	140800
leaky_re_lu_51 (LeakyReLU)	(None, 220)	0
dense_82 (Dense)	(None, 220)	48400
dense_83 (Dense)	(None, 1)	221

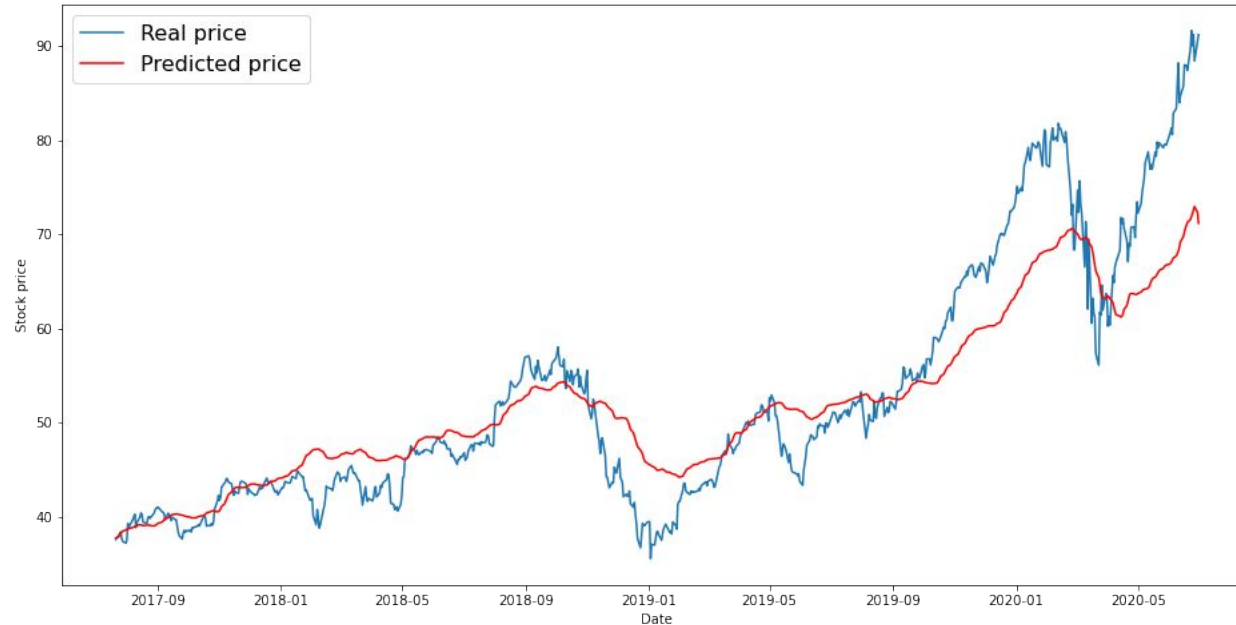


RMSE: 2.07



Basic GAN (Test data)

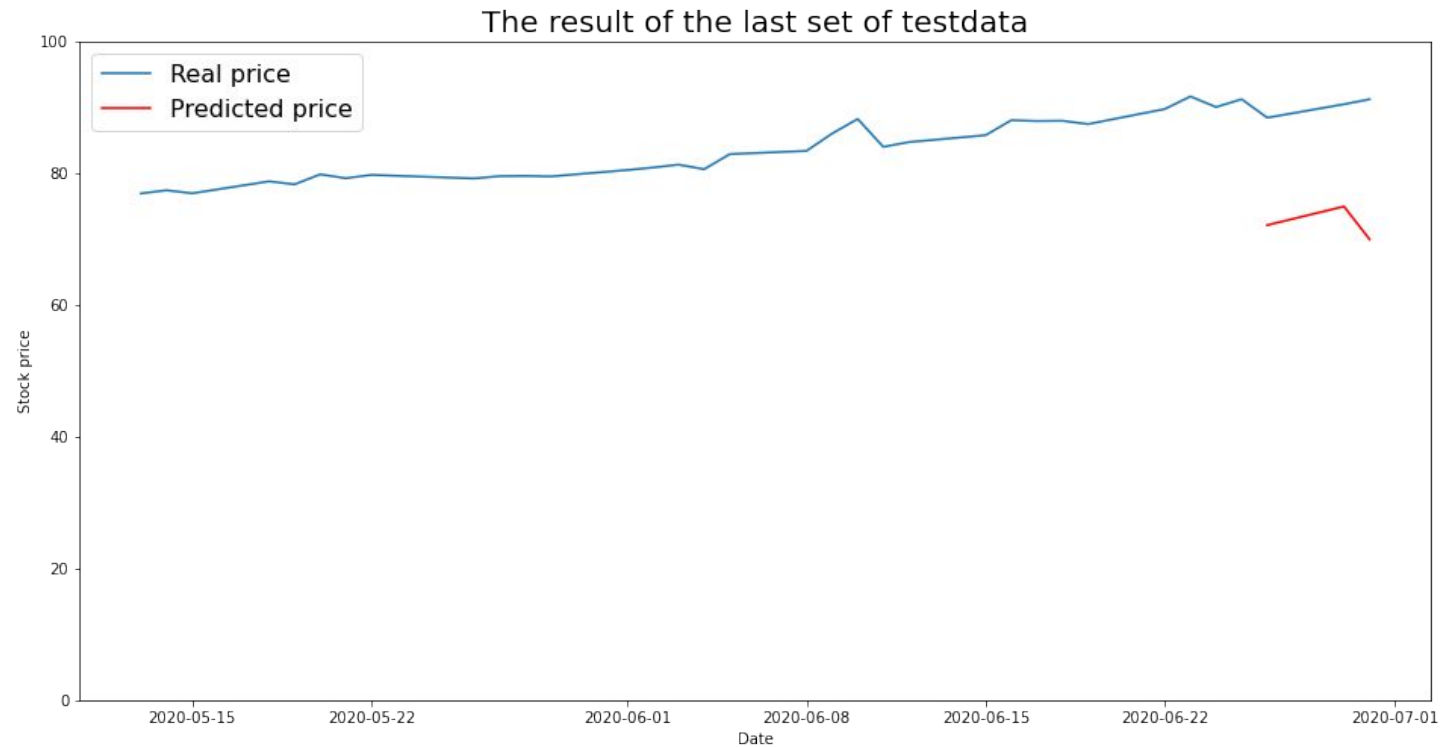
The result of test



RMSE: 5.54

Date				
2020-06-17	72.980716	70.019538		
2020-06-18	67.416058	73.190256	70.291891	
2020-06-19		67.686427	73.400380	70.555103
2020-06-22			67.989491	73.618525
2020-06-23				70.834884
2020-06-23			68.291586	73.854033
2020-06-24				71.122691
2020-06-24			68.593680	74.069502
2020-06-25				71.472792
2020-06-25			68.891683	74.339624
2020-06-26				71.791249
2020-06-26			69.252098	74.641064
2020-06-29				72.098883
2020-06-29			69.600428	74.930731
2020-06-30				69.940701

Basic GAN - last three days



Date	Predicted_price	Real_price
2020-06-26	72.10	88.41
2020-06-29	74.94	90.45
2020-06-30	69.94	91.20

RMSE: 17.88

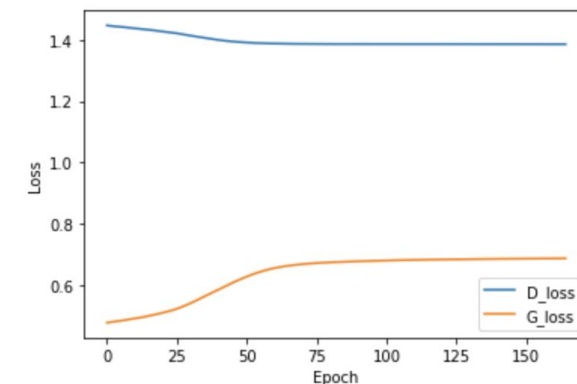
**Input 30, output 3*

Basic GAN - Train G 2 times (Train data)

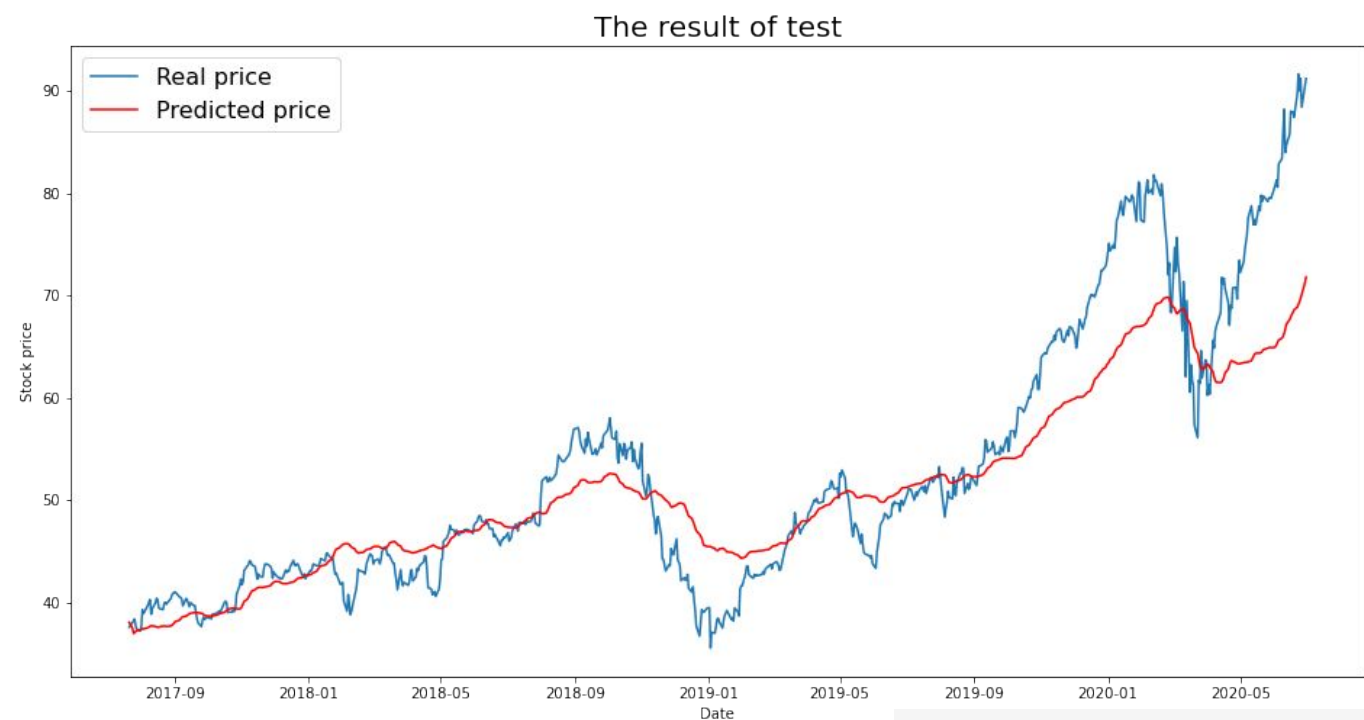
Layer (type)	Output Shape	Param #
gru_37 (GRU)	(None, 30, 1024)	3262464
gru_38 (GRU)	(None, 30, 512)	2362368
gru_39 (GRU)	(None, 256)	591360
dense_143 (Dense)	(None, 128)	32896
dense_144 (Dense)	(None, 64)	8256
dense_145 (Dense)	(None, 32)	2080
dense_146 (Dense)	(None, 3)	99
Layer (type)	Output Shape	Param #
conv1d_84 (Conv1D)	(None, 17, 32)	128
conv1d_85 (Conv1D)	(None, 9, 64)	10304
conv1d_86 (Conv1D)	(None, 5, 128)	41088
flatten_28 (Flatten)	(None, 640)	0
dense_147 (Dense)	(None, 220)	140800
leaky_re_lu_115 (LeakyReLU)	(None, 220)	0
dense_148 (Dense)	(None, 220)	48400
dense_149 (Dense)	(None, 1)	221



RMSE: 1.64



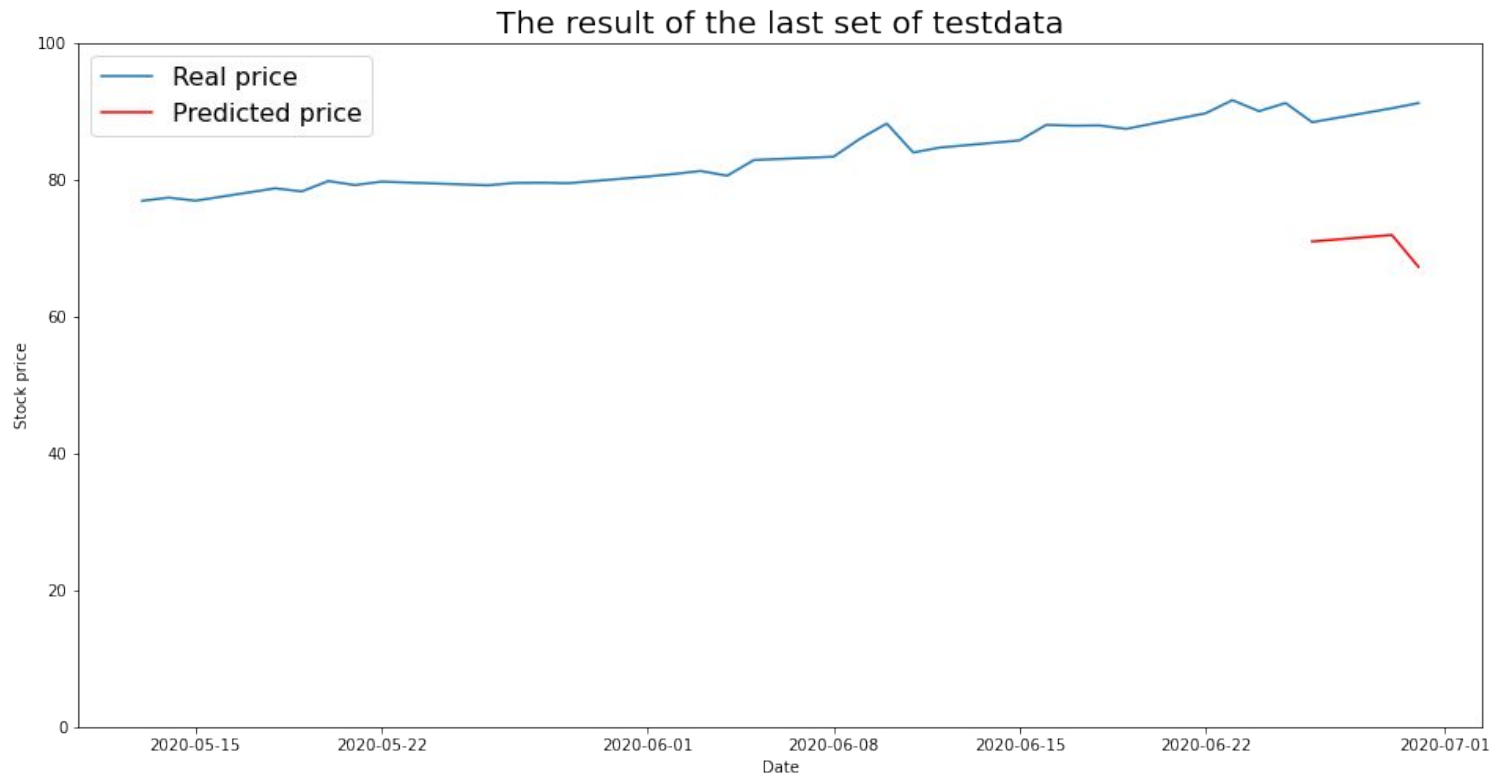
Basic GAN - Train G 2 times (Test data)



RMSE: 5.88

Date				
2020-06-17	69.665108	69.197626		
2020-06-18	65.675759	69.909878	69.420938	
2020-06-19		65.904078	70.156144	69.643179
2020-06-22			66.104476	70.410995
2020-06-23			66.288770	70.676340
2020-06-24				66.473005
2020-06-25				70.954811
2020-06-26				70.378637
2020-06-29				66.636495
2020-06-30				71.281442
				70.674263
				66.823197
				71.615539
				70.964070
				67.033652
				71.925086
				67.258487

Basic GAN - Train G 2 times - last three days



Date	Predicted_price	Real_price
2020-06-26	70.96	88.41
2020-06-29	71.93	90.45
2020-06-30	67.26	91.20

RMSE: 20.17

- For Basic GAN, the training times of G seems has no obvious effect on the result

**Input 30, output 3*

WGAN-GP model

Generator: (train generator 3 times)

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
=====		
gru (GRU)	(None, 30, 512)	844800
gru_1 (GRU)	(None, 30, 256)	591360
gru_2 (GRU)	(None, 30, 128)	148224
gru_3 (GRU)	(None, 64)	37248
dense (Dense)	(None, 32)	2080
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 3)	51
=====		

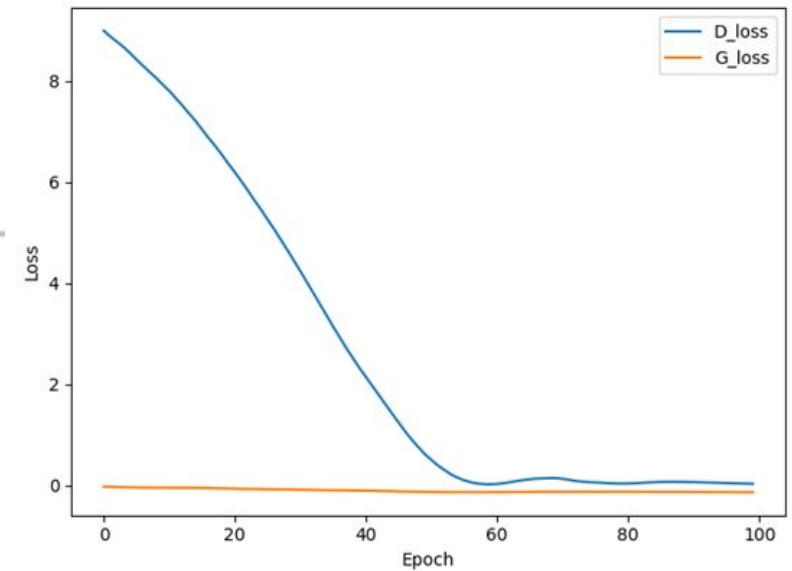
Discriminator:

Layer (type)	Output Shape	Param #
=====		
conv1d (Conv1D)	(None, 16, 32)	128
conv1d_1 (Conv1D)	(None, 7, 64)	6208
conv1d_2 (Conv1D)	(None, 3, 128)	24704
flatten (Flatten)	(None, 384)	0
dense_4 (Dense)	(None, 220)	84480
leaky_re_lu_3 (LeakyReLU)	(None, 220)	0
dense_5 (Dense)	(None, 220)	48400
re_lu (ReLU)	(None, 220)	0
dense_6 (Dense)	(None, 1)	221
=====		
Total params: 164,141		

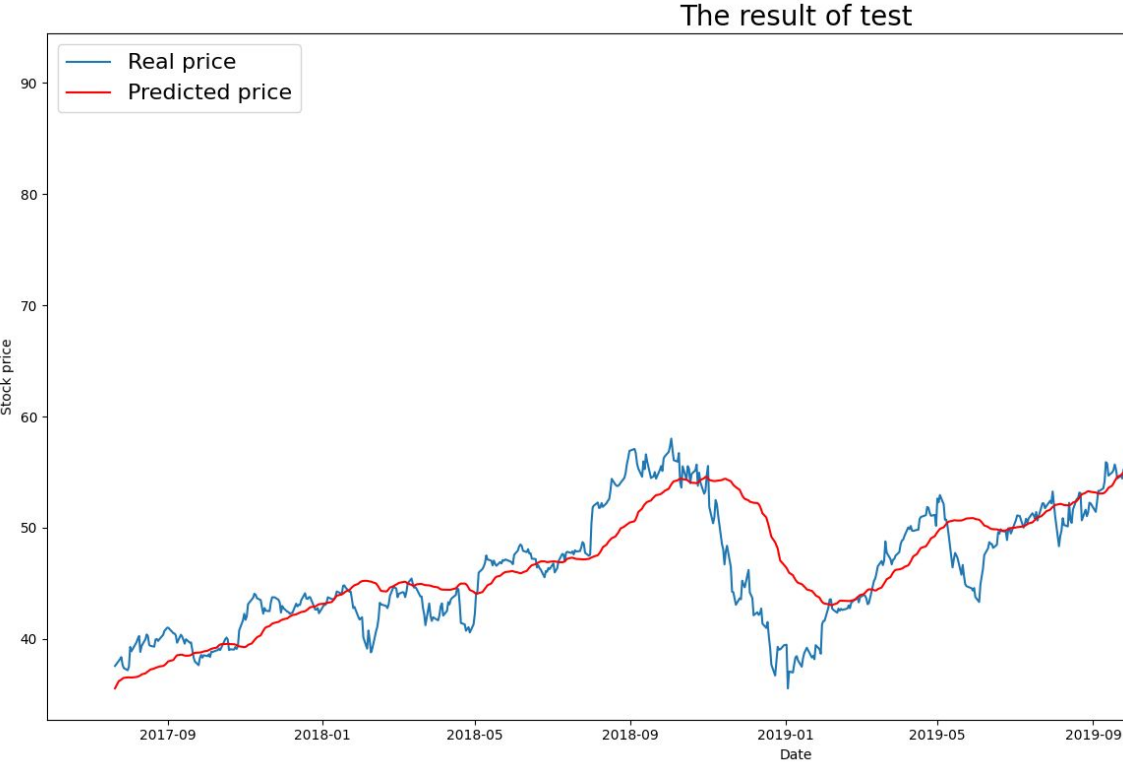
WGAN-GP (Train data)



RMSE: 1.89



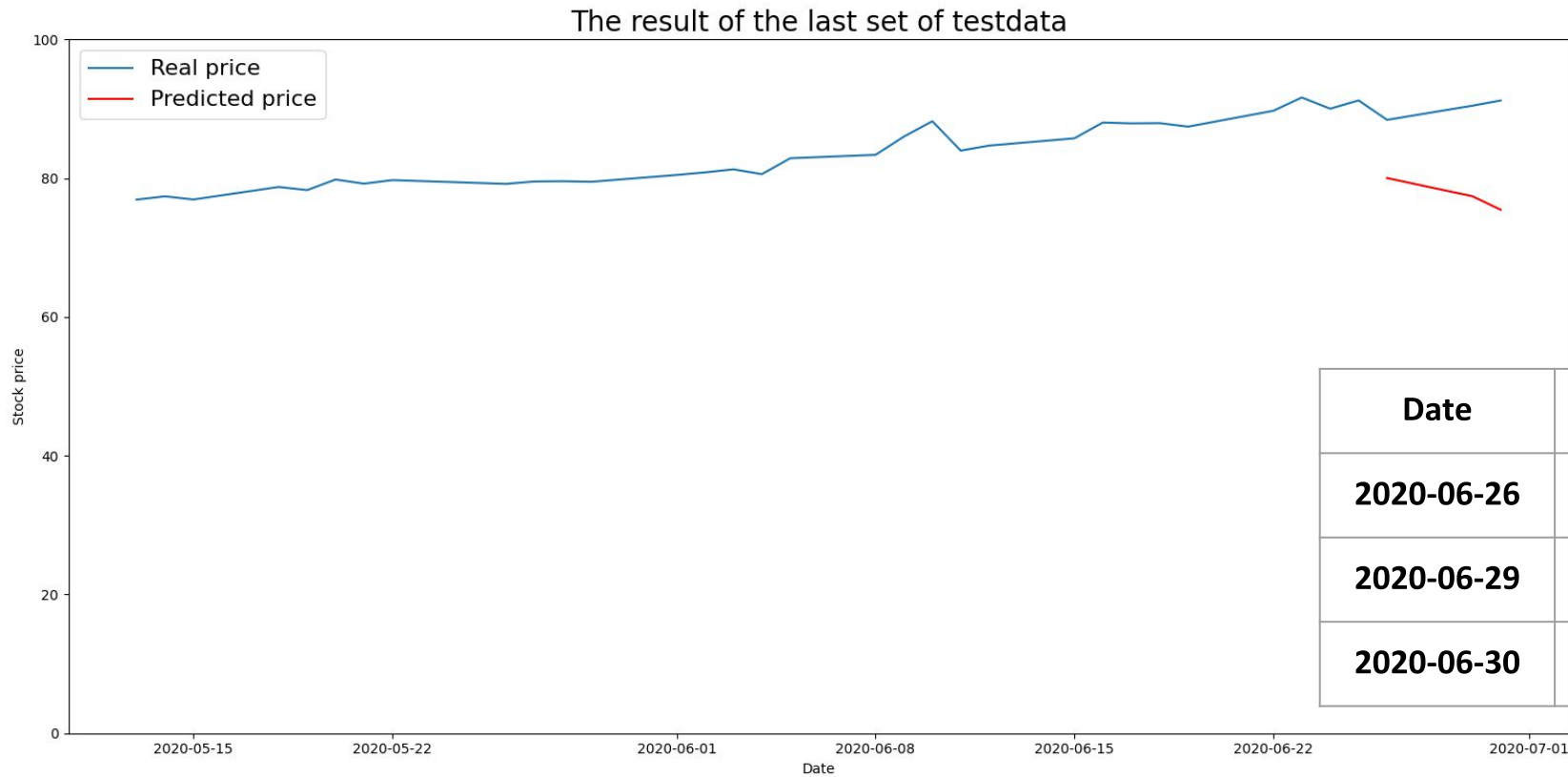
WGAN-GP (Test data)



RMSE: 4.77

	predicted_	predicted_	predicted_	predicted_	predicted_	predicted_	predicted_
2017/7/21	38.55335						
2017/7/24	37.94096	38.59683					
2017/7/25	35.83798	38.01058	38.75824				
2017/7/26		35.908	38.16607	38.90875			
2017/7/27			35.97938	38.31526	38.99351		
2017/7/28				36.06492	38.41359	39.04776	
2017/7/31					36.15733	38.48089	39.04621
2017/8/1						36.23452	38.49506
2017/8/2							36.29041

WGAN-GP - last three days



RMSE: 12.75

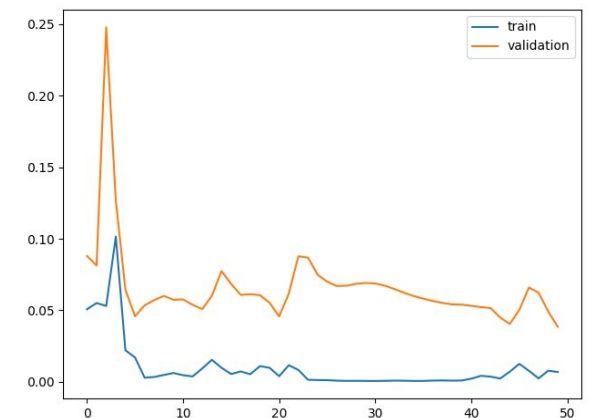
**Input 30, output 3*

Baseline - Basic LSTM (Train data)

Layer (type)	Output Shape	Param #
bidirectional (Bidirectional (None, 128))		51712
dense (Dense)	(None, 3)	387

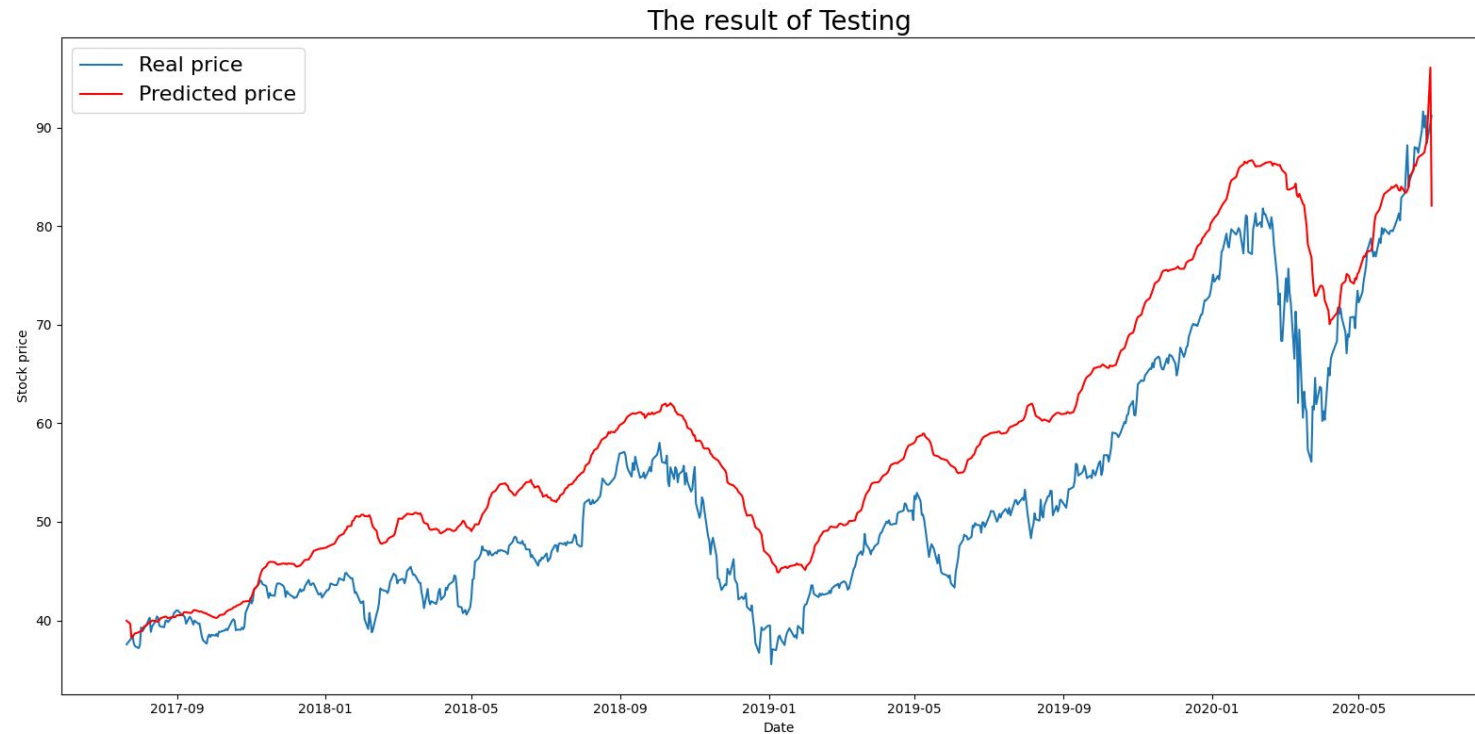


RMSE: 1.52



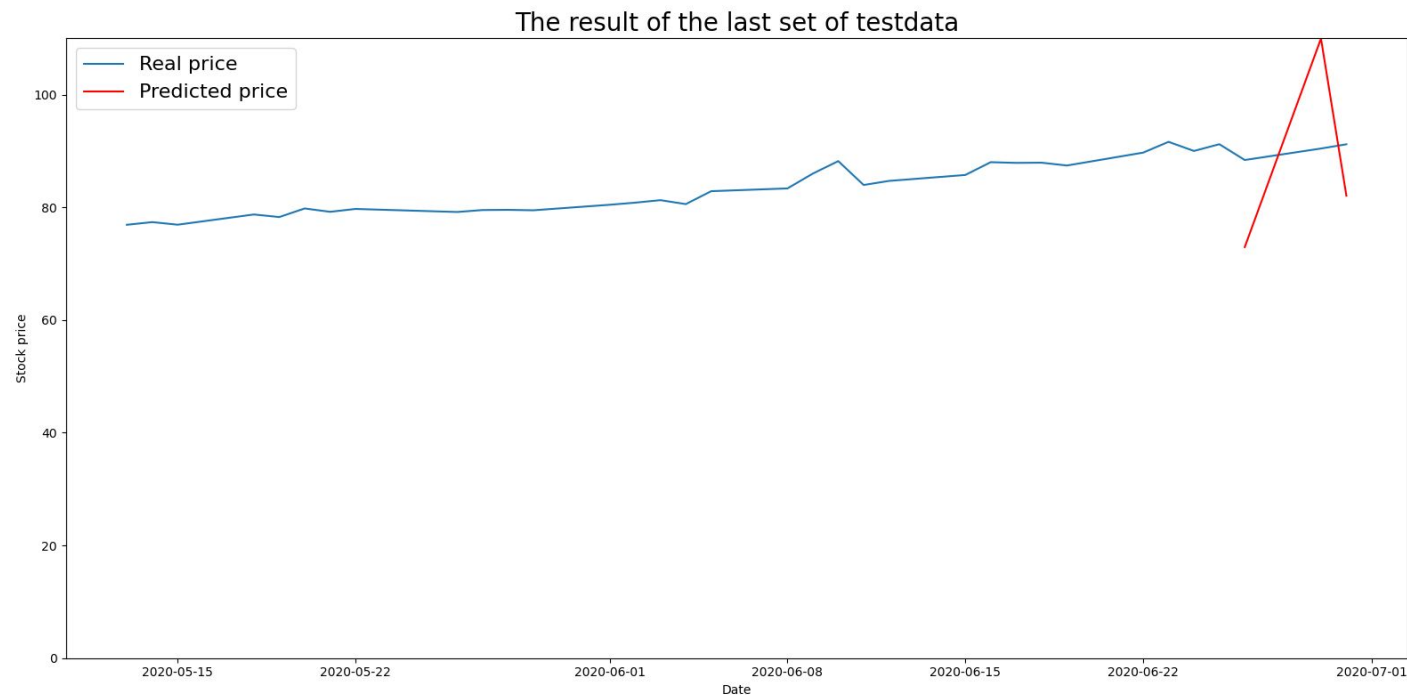
**Input 30, output 3*

Baseline - Basic LSTM (Test data)



RMSE: 6.60

Basic LSTM - last three days



Date	Predicted_price	Real_price
2020-06-26	103.33	88.41
2020-06-29	94.21	90.45
2020-06-30	105.05	91.20

RMSE: 11.95

Model Comparison

Train dataset

	Basic LSTM	Basic GAN	Basic GAN - 2G	WGAN-GP
RMSE	1.52	2.07	1.64	1.89

Test dataset

	Basic LSTM	Basic GAN	Basic GAN - 2G	WGAN-GP
RMSE	6.60	5.54	5.88	4.77

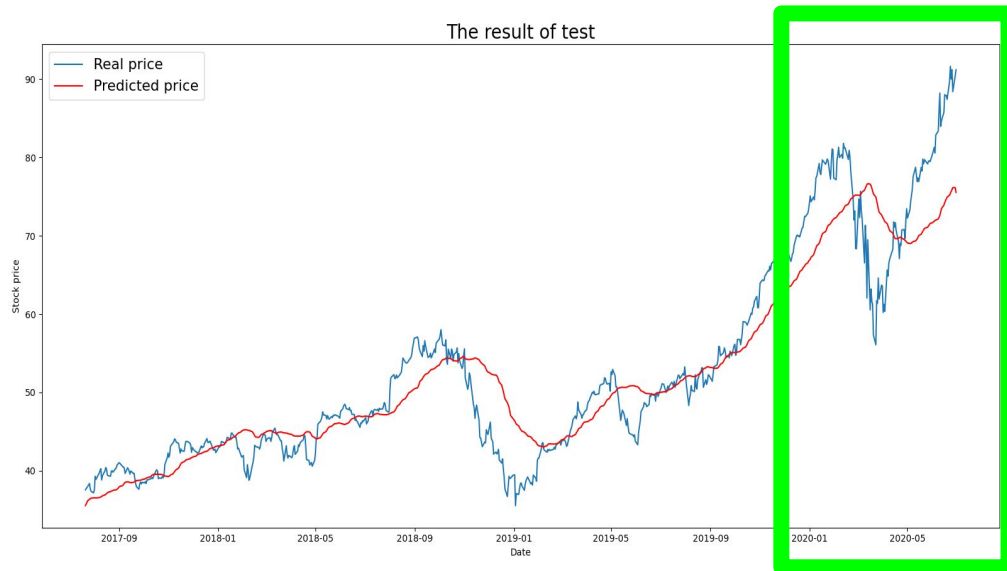
The last three days

	Basic LSTM	Basic GAN	Basic GAN - 2G	WGAN-GP
RMSE	11.95	17.88	20.17	12.75

Phenomenon of the last three days prediction

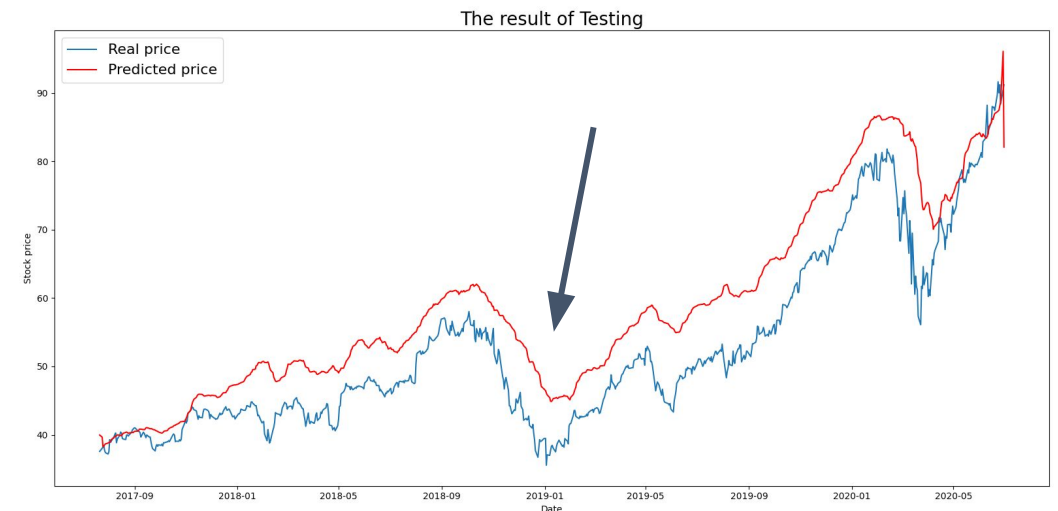
- The prediction of the last three days through **GAN** is very inaccurate

-> Might cause from unpredictable sharp drop and growth which due to the COVID-19



- The prediction of the last three days through **Basic LSTM** is more accurate

-> As can be seen from the plot, the prediction of LSTM is always much higher than the real price, which makes the prediction of the recent period more accurate.



Problem

- GAN model did not perform better than Baseline model
model still need to be improved

Question

- WGAN-GP best model (train RMSE 1.89, test RMSE 4.77)
when train RMSE smaller than 1.89 test RMSE will increase,
overfitting?

Thank you