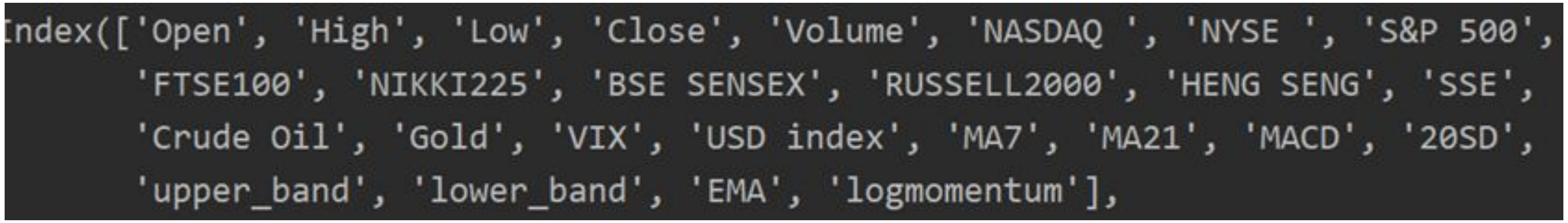
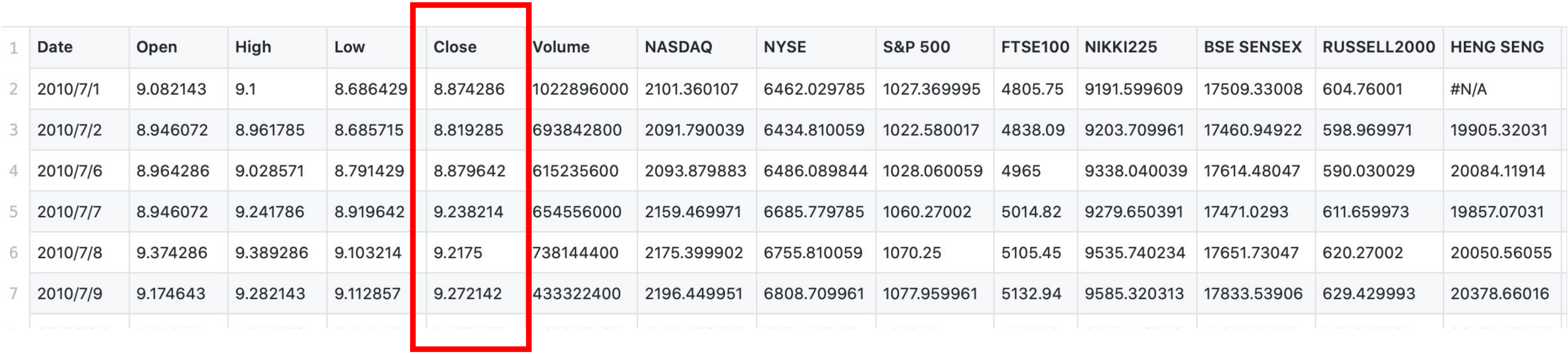
Original Data

- Dataset: 2518 rows * 26 columns



```
Index(['Open', 'High', 'Low', 'Close', 'Volume', 'NASDAQ ', 'NYSE ', 'S&P 500',
       'FTSE100', 'NIKKI225', 'BSE SENSEX', 'RUSSELL2000', 'HENG SENG', 'SSE',
       'Crude Oil', 'Gold', 'VIX', 'USD index', 'MA7', 'MA21', 'MACD', '20SD',
       'upper_band', 'lower_band', 'EMA', 'logmomentum'],
```



Date	Open	High	Low	Close	Volume	NASDAQ	NYSE	S&P 500	FTSE100	NIKKI225	BSE SENSEX	RUSSELL2000	HENG SENG
2010/7/1	9.082143	9.1	8.686429	8.874286	1022896000	2101.360107	6462.029785	1027.369995	4805.75	9191.599609	17509.33008	604.76001	#N/A
2010/7/2	8.946072	8.961785	8.685715	8.819285	693842800	2091.790039	6434.810059	1022.580017	4838.09	9203.709961	17460.94922	598.969971	19905.32031
2010/7/6	8.964286	9.028571	8.791429	8.879642	615235600	2093.879883	6486.089844	1028.060059	4965	9338.040039	17614.48047	590.030029	20084.11914
2010/7/7	8.946072	9.241786	8.919642	9.238214	654556000	2159.469971	6685.779785	1060.27002	5014.82	9279.650391	17471.0293	611.659973	19857.07031
2010/7/8	9.374286	9.389286	9.103214	9.2175	738144400	2175.399902	6755.810059	1070.25	5105.45	9535.740234	17651.73047	620.27002	20050.56055
2010/7/9	9.174643	9.282143	9.112857	9.272142	433322400	2196.449951	6808.709961	1077.959961	5132.94	9585.320313	17833.53906	629.429993	20378.66016

...

⋮

Data preprocessing

- NA value for index value

  x(t) = average(x(t-1)+x(t+1))

- Normalized the data

```
X_scaler = MinMaxScaler(feature_range = (-1,1))
y_scaler = MinMaxScaler(feature_range = (-1,1))
```

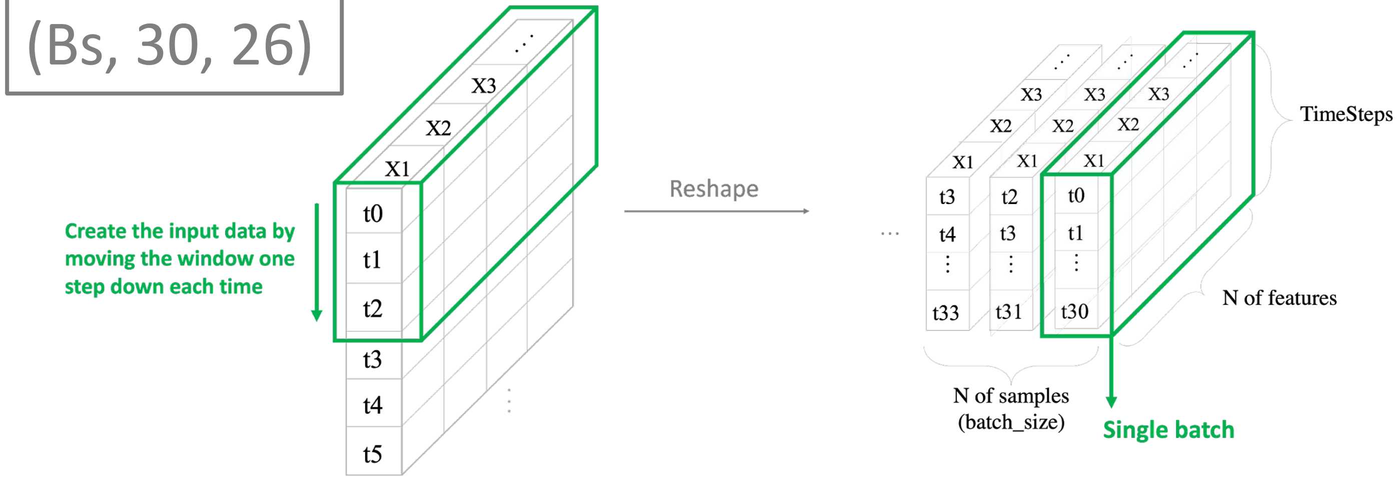
- Train test split

  Train: 1736

  Test:744

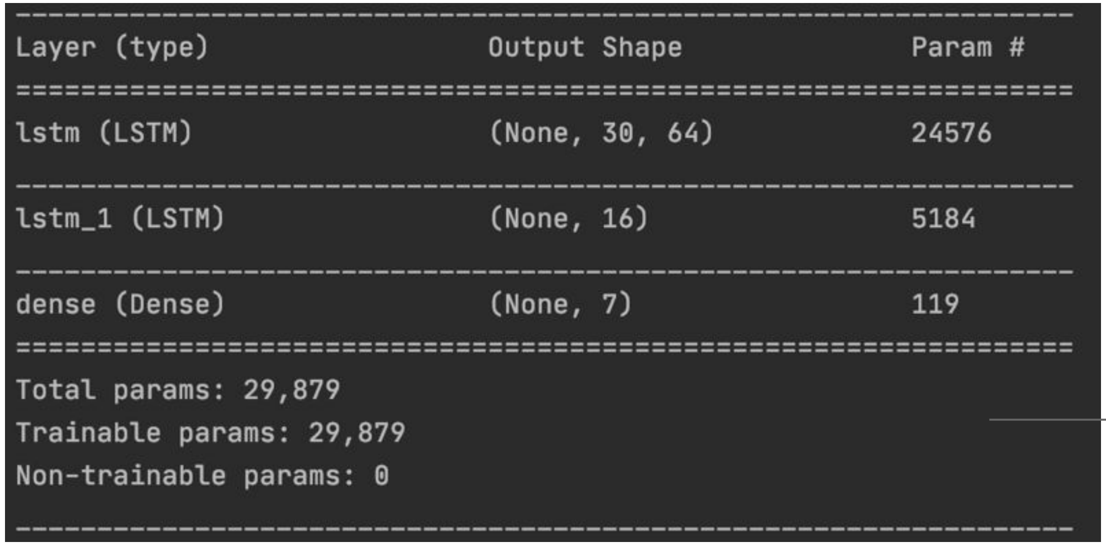
LSTM

Input Data
(Bs, 30, 26)



Create the input data by moving the window one step down each time
X1
X2
X3
t0
t1
t2
t3
t4
t5
Reshape
TimeSteps
X1
X2
X3
t3
t4
t33
t2
t3
t31
t0
t1
t30
N of features
N of samples (batch_size)
Single batch

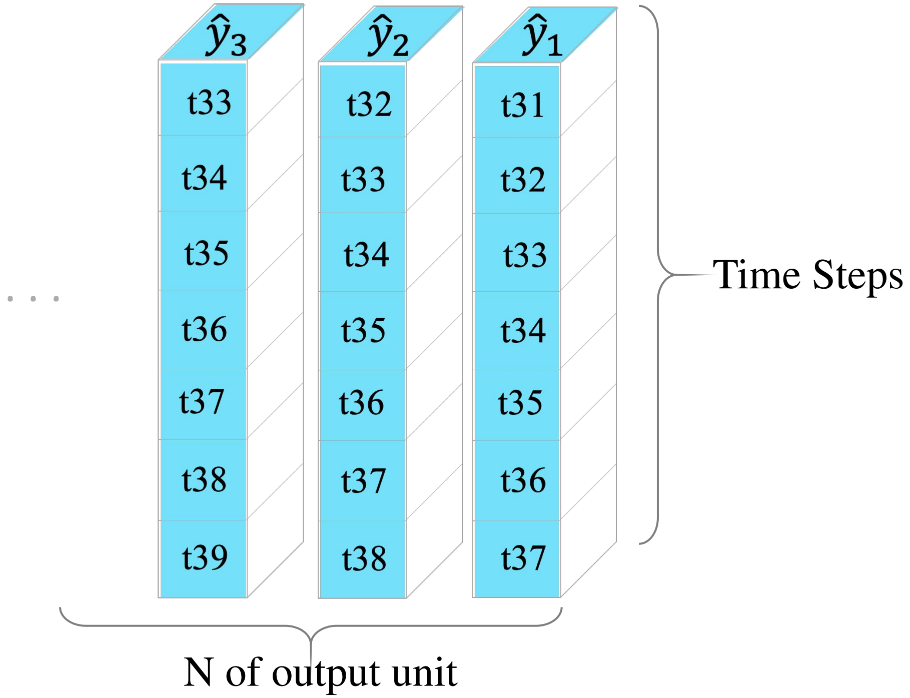
LSTM



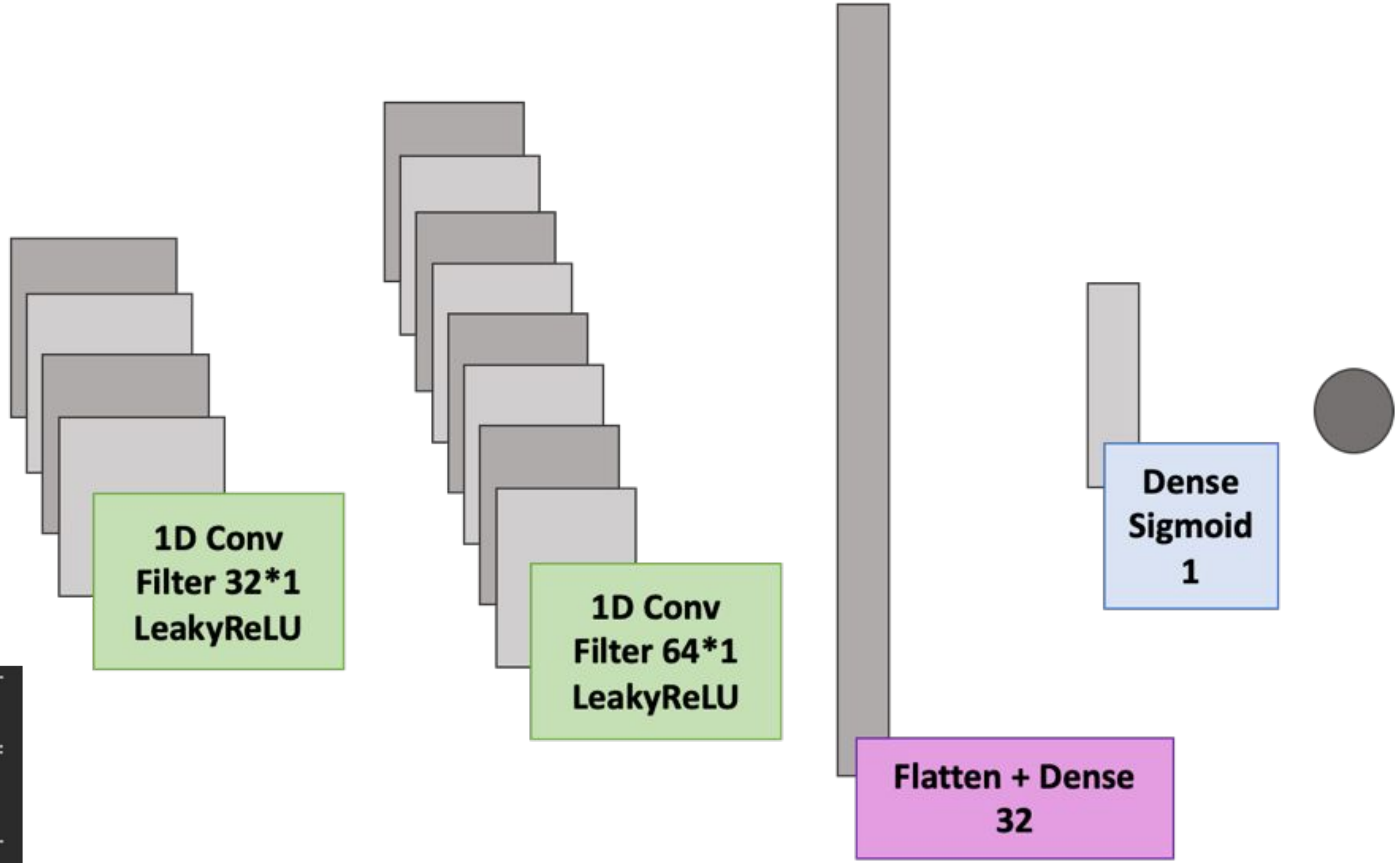
```
----------------------------------------------------------------------
Layer (type)                 Output Shape              Param #
======================================================================
lstm (LSTM)                  (None, 30, 64)            24576

----------------------------------------------------------------------
lstm_1 (LSTM)                (None, 16)                5184

----------------------------------------------------------------------
dense (Dense)                (None, 7)                 119
======================================================================
Total params: 29,879
Trainable params: 29,879
Non-trainable params: 0

----------------------------------------------------------------------
```

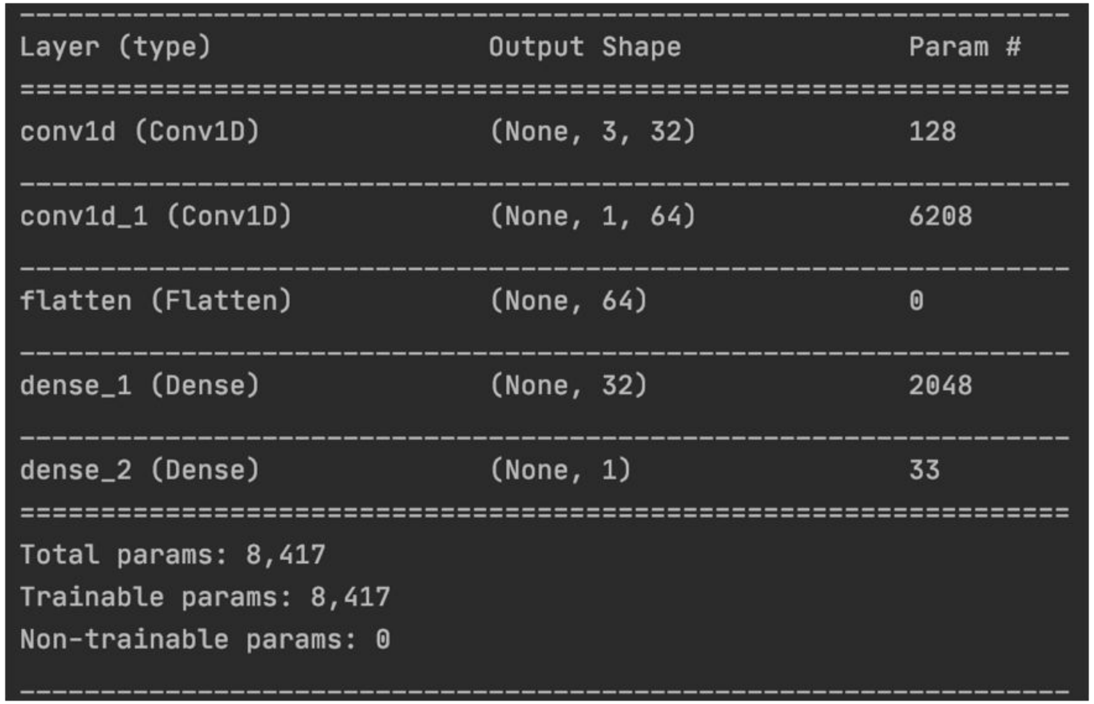
Output Data
(Bs, 7, 1)

Reshape



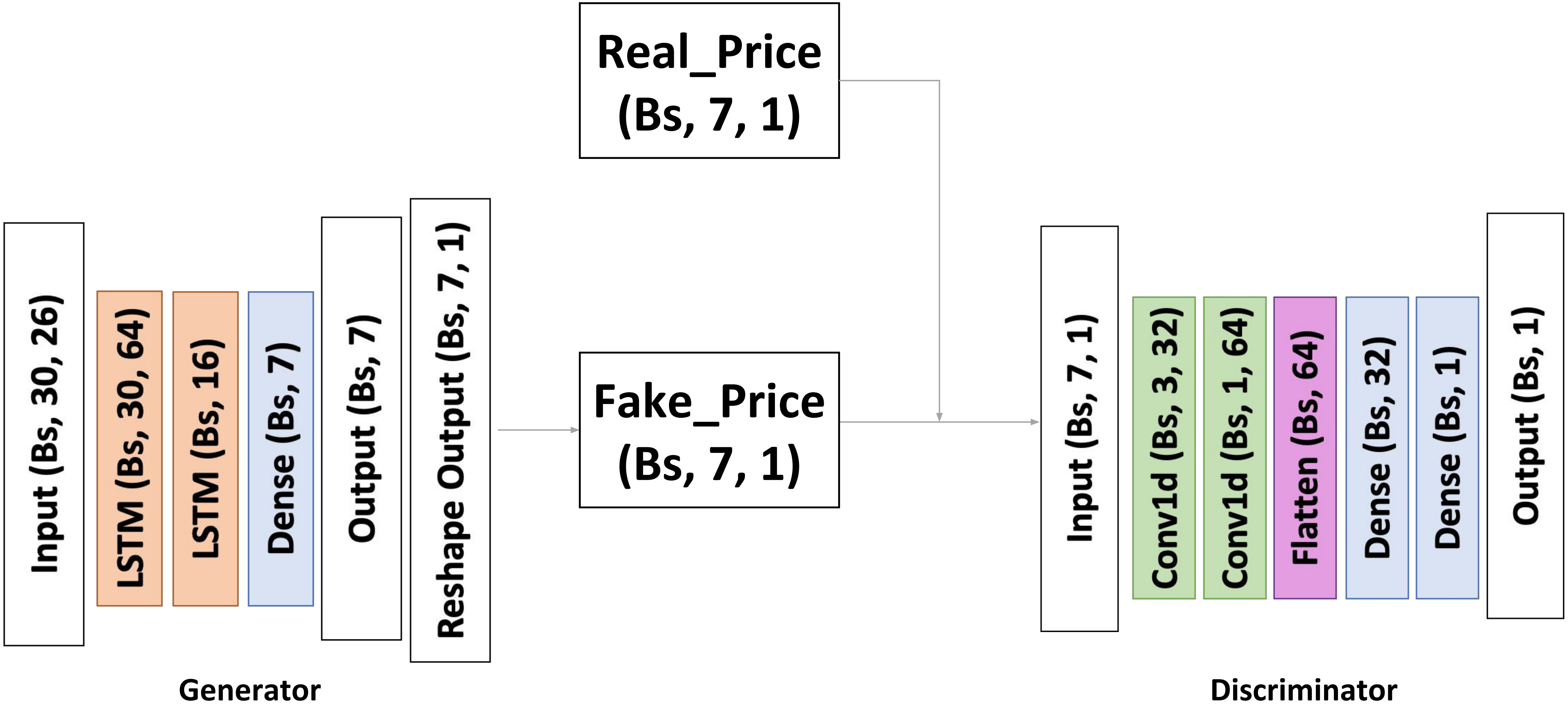
CNN





```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 3, 32)             128
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 1, 64)             6208
_________________________________________________________________
flatten (Flatten)            (None, 64)                0
_________________________________________________________________
dense_1 (Dense)              (None, 32)                2048
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 33
=================================================================
Total params: 8,417
Trainable params: 8,417
Non-trainable params: 0
_________________________________________________________________
```

GAN



Input (Bs, 30, 26)

LSTM (Bs, 30, 64)

LSTM (Bs, 16)

Dense (Bs, 7)

Output (Bs, 7)

Reshape Output (Bs, 7, 1)

Generator

Real_Price
(Bs, 7, 1)

Fake_Price
(Bs, 7, 1)

Input (Bs, 7, 1)

Conv1d (Bs, 3, 32)

Conv1d (Bs, 1, 64)

Flatten (Bs, 64)

Dense (Bs, 32)

Dense (Bs, 1)

Output (Bs, 1)

Discriminator

GAN Math

x : Input for generator
y : Real price from original data
$G\left(x^i\right)$: Generated price (fake price)

- *Learning D*

Maximize the objective function:

the smaller the better

$$\hat{V} = \boxed{\frac{1}{m}\sum_{i=1}^{m}\log D\left(y^i\right)} + \sum_{i=1}^{m}(1-\boxed{\log D(G\left(x^i\right)}))$$

the bigger the better

- *Learning G*

Minimize the objective function:

$$\hat{V} = \frac{1}{m}\sum_{i=1}^{m}(1-\boxed{\log D(G\left(x^i\right)}))$$

the bigger the better

Next step

- Feature Engineering:

1. Add one more features: Add “News” feature through NLP
2. Do the feature selection: XGBoost

- GAN model improvement