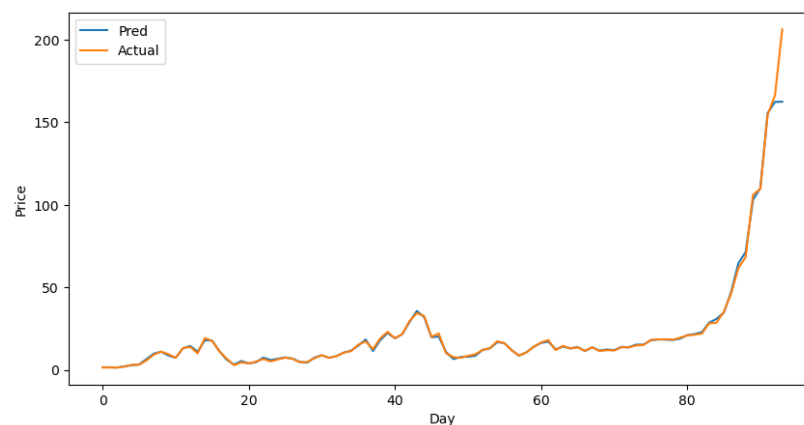


1. 從表格一中可以看到，step 不變、將 Window size 增加到 30，驗證損失與測試損失皆上升許多，而 Window size 不變、將 step 減少至 10，測試損失相較原本的下降一些。如果繼續減少 step 至 5 甚至 3，則觀察到三種損失皆下降非常多，由此可知 step 的值對模型的重要性應該比 Window size 的更高。

	Train loss	Validation loss	Test loss
Original (10, 15)	136.5631	159.3628	376.8634
Window size = 30, step = 15	108.9981	335.9499	440.8775
Window size = 10, step = 10	73.9757	148.4461	297.7568
Window size = 10, step = 5	9.3419	17.0860	22.1838
Window size = 10, step = 3	<b>2.1729</b>	<b>4.9525</b>	<b>nan</b>

表格一

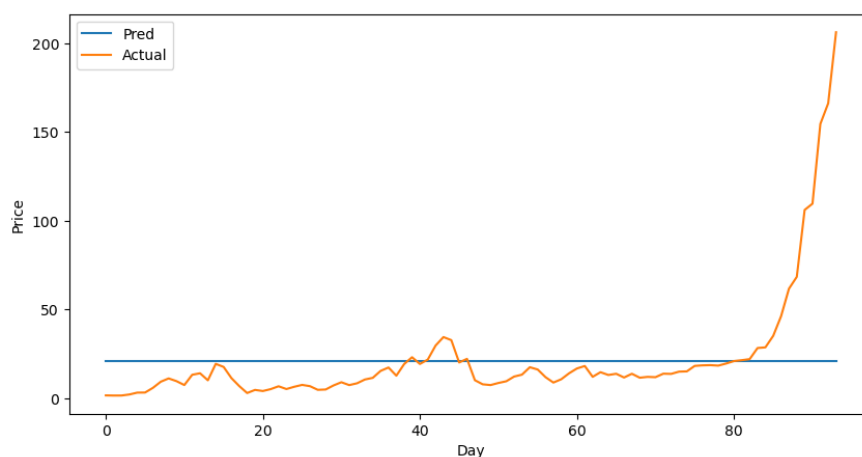


圖一、Window size = 10, step = 3 所得到的預測曲線

2. (a) 根據表格二第二列的結果，可以看到加入'Volume'後的損失變得非常大；圖二的預測呈現一直線，與實際值存在不小的誤差，這可能是由於'Volume'的值域遠大於其他特徵所導致的。  
 (b) 根據表格二的實驗結果，最佳的特徵組合為'Open', 'High', 'Low'，然而實際上沒有使用'Volume'的五個特徵組合所得到的損失之間差異非常小。選擇組合的方式是從原本使用的四個特徵中刪除一個，以此判斷哪個特徵相對較不重要。

Window size = 10, step = 5	Train loss	Validation loss	Test loss
'Open', 'High', 'Low', 'Close'	6.7046	10.1366	18.8439
'Open', 'High', 'Low', 'Close', 'Volume'	969.6949	803.4329	1138.2756
'High', 'Low', 'Close'	<b>6.0424</b>	12.6376	19.7423
'Open', 'High', 'Close'	8.0886	6.5993	20.2029
'Open', 'Low', 'Close'	7.8790	11.6079	23.0873
'Open', 'High', 'Low'	6.9750	<b>1.1934</b>	<b>17.3528</b>

表格二

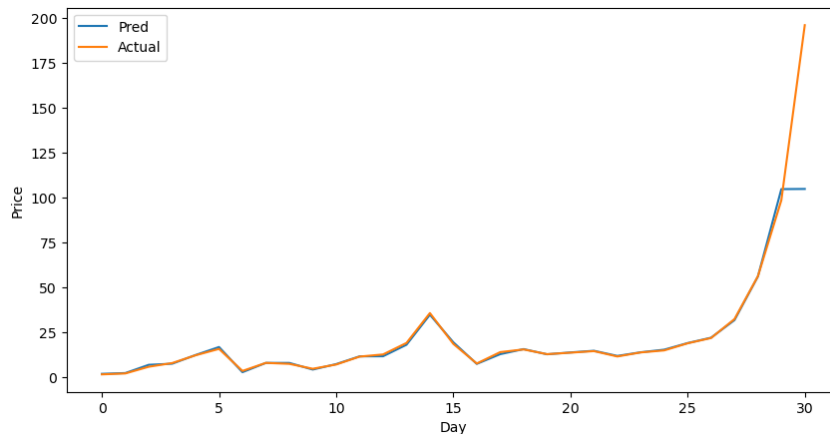


圖二、加入'Volume'的預測結果

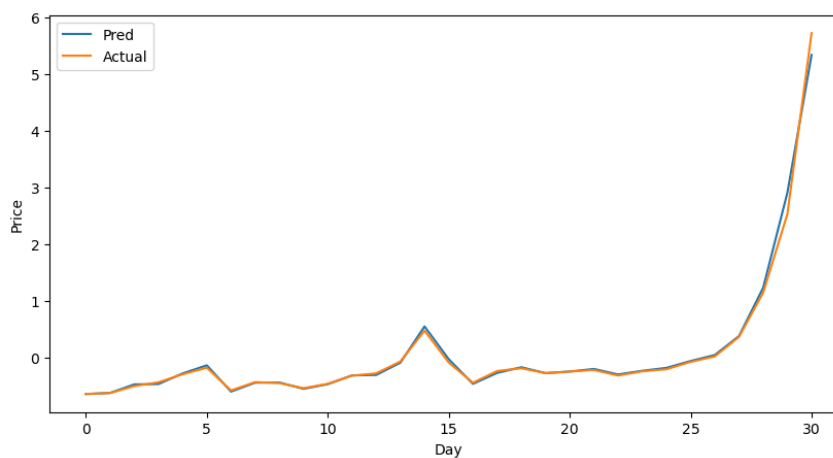
3. 將特徵進行正規化後，由於損失的尺度改變無法直接進行比較，但以圖三和圖四來看，在 30 天附近進行正規化後預測的較好；然而如果特徵有包含'Volume'，進行正規化後的預測效果會遠比沒有正規化的好，因為'Volume'的值域會變得其他特徵差不多。

Window size = 10, step = 15	Train loss	Validation loss	Test loss
w/o normalization	136.5631	159.3628	376.8634
w normalization	0.0022	0.0020	0.0102

表格三



圖三、沒有進行正規化的預測曲線



圖四、正規化之後的預測曲線

4. 我認為應該 window size 應該比 step size 大，因為 step size 比較大會造成中間時段的資訊沒被學到，且 step size 設得太大會減少訓練資料，提升過擬合的風險。
- 5.
6. (a) CNN: window size 通常是由 kernel size 決定，kernel size 越大，每次能看到的感受野也越寬廣。  
 (b) RNN: 經常用來處理時間序列的資料，透過設定 window size 的大小可以決定一次要用多少個時間點來進行預測。  
 (c) Transformer: 傳統自注意力機制的計算時間複雜度與序列長度成平方關係，意味著模型的輸入序列不能太長，為了解決上述問題，在 Longformer 中提出到了 Sliding window attention，設定的 window size  $W$  限制了序列中每個 token 只看左右各  $1/2 * W$  的 token。