

## 國立清華大學資訊工程系 113 學年度 上 學期專題報告

專題名稱	Transforming the Futures: Using Transformer-Based Models in Trading				
參加競賽或計畫	<input type="checkbox"/> 參加對外競賽	<input type="checkbox"/> 參與其他計畫		<input checked="" type="checkbox"/> 無參加對外競賽或任何計畫	
學號	110062333	110062243	109062234		
姓名	林家逸	許祐銓	李傳中		

### 摘要

本專題主要分析在相同的量化交易策略下，透過加入 Transformer-based 模型預測作為進出場條件對策略的影響。

我們收集了自2011年到2024年8月16號台指期近月的資料作為訓練與回測，其中2011年到2023年作為模型的訓練資料，2023年以後作為模型的測試資料，2022年到2024年作為策略的回測資料。資料會在我們前處理以及計算完各種技術指標後才使用。

在回測過程中，除了透過技術指標以及模型預測作為進出場依據外，我們亦規定所有交易皆為日內交易，所有未完成的買賣皆會在收盤前半小時進行關倉。此外，為確保可以完整回測所有資料，我們建立了若目前總資金回撤大於50%則強制平倉的機制。最後，我們規定初始資金為500,000元，且交易手續費為2.5‰

建立了一個共同的交易策略後，透過三人訓練出的三個不同的模型，我們根據模型的預測，各自在交易策略上新增了不同的進出場條件。結果顯示，不加入任何模型預測的情況下最終資金為850,566.9元；加入模型一號後最終資金為925,718.5元；加入模型二號後最終資金為752,105.9元；加入模型三號後最終資金為935605.6元。

有鑑於此，我們認為加入模型預測作為交易策略中進出場條件的一部份會犧牲一定的品質，而且其整體績效與品質會和模型的好壞掛勾，但是只要策略與模型搭配得當，兩者的結合是可以創造更高的報酬的。

## 一、研究動機與目的

量化交易源於1970年代，當時主要以數學模型以及統計分析為基礎。隨著1980年代計算機的發展，許多金融機構開始使用計算機去輔助交易。近年來因為金融市場全球化、計算機能力提升等因素，量化交易在期貨交易市場更是逐漸成為主要的交易方式。

計算機領域的發展不只推動了量化交易的普及，更是促使了 AI 領域的興起。2022年11月隨著 OpenAI 推出 ChatGPT 後，大量基於 Transformer 的語言模型如雨後春筍般推出。而 Transformer 模型之所以成功，除了它在自然語言處理（NLP）的天然優勢外，因為自注意力機制（Self-Attention Mechanism）的關係，使得它能夠直接關注長序列資料中任意位置的數據，不會像 CNN、RNN、LSTM 等模型的記憶能力會因為序列的增加而衰減。

本專題的目的便是透過利用 Transformer 模型長距依賴優勢去預測未來期貨市場的走向，並透過回測來分析加入模型預測作為交易策略進出場的條件對整體策略的影響。

## 二、相關研究

因為目前相關研究較少，所以本章節會先介紹在此專題所使用到的模型以及技術指標，最後再介紹國內外相關研究。

### （一） Transformer 模型

Transformer 模型是 Google 團隊於2017年提出的模型[3]，模型主要由數層 Encoder 與 Decoder 組成，如下圖所示。

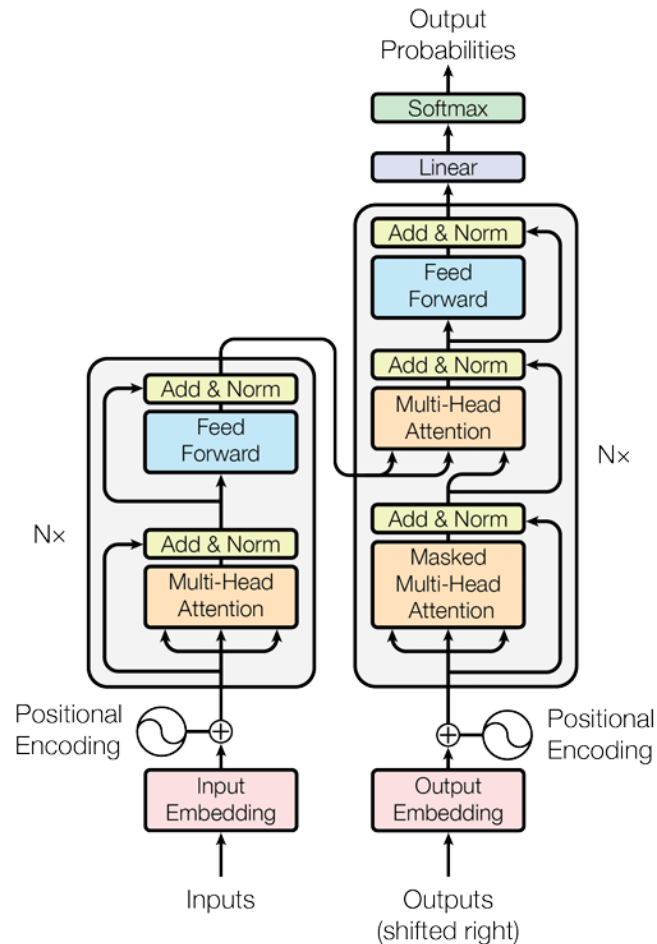


圖1 Transformer 架構圖[3]

其中每層 Encoder 含有兩個子層——Multi-Head Self-Attention 與 Feed-Forward Network，每個子層之間透過 Add & Norm 去作連接，其公式為

$$\text{LayerNorm}(x + \text{SubLayer}(x))$$

每層 Decoder 除了剛剛所提到的兩子層外，兩子層中間還有一個 Multi-Head Self-Attention 作為前一子層與 Encoder 輸出的連接，並且第一個 Multi-Head Self-Attention 為 Masked 過的，確保第 i 個位置的預測結果只會由已知的輸出（第1到第 i-1個位置的預測結果）獲得。此外，在 Embedding 以及所有 Add & Norm 後面還會連接 Dropout 來避免模型過擬合（overfitting）。

因為 Transformer 原先是處理語言翻譯相關的任務，而我們知道這些任務的輸入輸出都是文本或句子，所以在原論文中，所有資料會先進行 Embedding，也就是將字詞轉成向量。由於相同的字詞依據不同順序所組成的句子所代表的意義可能相去不遠，也可能截然不同，因此除了要考慮字詞本身 Embedding 後的向量外，模型也需要考慮字詞在文章或句子中的位置。然而，Transformer 不像 RNN、CNN 具有遞迴（Recurrence）或卷積（Convolution）的結構，所以為了將位置訊息納入考量，資料還須加上依照下列公式所得出的位置編碼（Positional Encoding）。

$$PE_{(pos, 2i)} = \sin(pos / 10000^{\frac{2i}{d_{model}}})$$

$$PE_{(pos, 2i+1)} = \cos(pos / 10000^{\frac{2i}{d_{model}}})$$

公式中的  $d_{model}$  為 Embedding 後的向量維度，同時也是每個子層輸出的維度。

在 Multi-Head Self-Attention 中，輸入有 Query、Key、Value 三個向量。其中 Attention 是指先由 Query 與 Key 算出兩者的關聯度，再將其作為權重與 Value 去計算最終輸出的函數。Attention 有許多種計算方式，在原論文中所採用的為縮放點積注意力（Scaled Dot-Product Attention）。其公式與流程架構圖如下所示。

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Scaled Dot-Product Attention

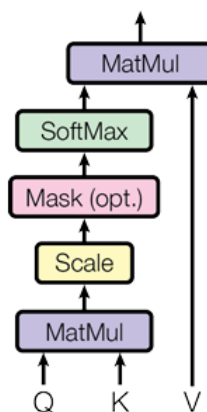


圖2 Scaled Dot-Product Attention 流程架構圖[3]

其中  $d_k$  為 Key 向量的維度。除以  $\sqrt{d_k}$  是為了避免 Dot-Product 的結果趨於兩極化分布，從而導致梯度太小而無法有效更新參數。圖中的 Mask 如同 Decoder 介紹，是為了避免解碼時考慮到未來的資料，實際的實現方式為將  $i+1$  以後的資料全部設為  $-\infty$ ，這樣可以使那部分的資料 softmax 出來的值為 0。至於 Self 在這裡所代表的意思指的是 Query、Key、Value 皆為相同的向量。

原論文的作者們發現與其直接用維度為  $d_{model}$  的 Query、Key、Value 去計算 Self-Attention，先將它們投影成低維度的向量  $h$  次，在每一次計算完後把所

有結果連接（Concatenate）起來，最後再將其投影回維度 $d_{model}$ 有許多優勢。這些優勢包含了可以並行計算、可以模擬 CNN 中多通道輸出等。而投影數次的操作即為 Multi-Head。其公式與流程架構圖如下所示。

$$MultiHead(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W^O$$

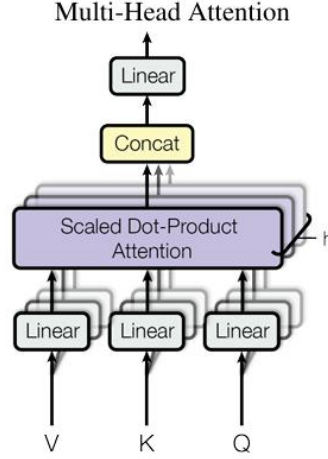


圖3 Multi-Head 流程架構圖[3]

$$where head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

公式中的 $head_i$ 代表每次投影後算出來的 Attention， $W_i^Q$ 、 $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$ 、 $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ 、 $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ 皆為投影參數矩陣。

在 Feed Forward Network 中，輸入的參數只有一向量  $x$ ，而此子層的目的在於透過兩個線性轉換（linear transformation）與一個 ReLU 去更進一步的提取特徵。其公式如下所示。

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

公式中的 $W_1 \in \mathbb{R}^{d_{model} \times d_{ff}}$ 、 $W_2 \in \mathbb{R}^{d_{ff} \times d_{model}}$ 皆為線性轉換矩陣， $b_1 \in \mathbb{R}^{d_{ff}}$ 、 $b_2 \in \mathbb{R}^{d_{model}}$ 可以視為常數，其中 $d_{ff}$ 為此子層中隱藏層的維度。

在預測完前  $i-1$  個字詞後，原論文會透過由先前的預測結果以及目前的輸入去產生一個暫時的輸出，將這個輸出經過線性方程式和 softmax 後即可得到字典中每一個字詞作為第  $i$  個輸出的機率。

## （二） 技術指標

本小節會介紹模型訓練以及交易策略中所使用到的技術指標。在介紹各指標的計算方式時，我們令當前 K 棒的開高低收為 open[0]、high[0]、low[0]、close[0]；先前的第  $i$  根 K 棒的開高收低為 open[-i]、high[-i]、low[-i]、close[-i]；前  $i$  根 K 棒的區間資料為 open[-i:]、high[-i:]、low[-i:]、close[-i:]，其他資訊也是依照此邏輯說明。

## 1. 隨機指標 (KDJ)

KDJ 指標為一個趨勢跟蹤指標，藉由比較收盤價格和價格的波動範圍，預測價格趨勢何時逆轉。「隨機」一詞是指價格在一段時間內相對於其波動範圍的位置。K、D、J 所代表的意義分別為快線、慢線、方向敏感線。K、D 的值是根據 RSV（未成熟隨機值）得出，而 J 值是衡量 K 值與 D 值的背離程度。計算方式有很多種，我們採取的計算方式為以下所示。

$$RSV[0] = (close[0] - l) \div (h - l) \times 100$$

$$K[0] = \alpha_k \cdot \left( \frac{K[-1] + 2 \cdot RSV[0]}{3} \right) + (1 - \alpha_k) \cdot K[-1]$$

$$D[0] = \alpha_d \cdot \left( \frac{D[-1] + 2 \cdot K[0]}{3} \right) + (1 - \alpha_d) \cdot D[-1]$$

$$J[0] = 3 \cdot K[0] - 2 \cdot D[0]$$

其中

$$l = \min(low[-period :])$$

$$h = \max(high[-period :])$$

$$\alpha_k = 2 / (signal_k + 1)$$

$$\alpha_d = 2 / (signal_d + 1)$$

*period*、*signal\_k*、*signal\_d* 為輸入的一部分參數。此外，K[0] 與 D[0] 的公式相當於計算  $EMA(\frac{K[-1] + 2 \cdot RSV[0]}{3}, signal_k)$  與  $EMA(\frac{D[-1] + 2 \cdot K[0]}{3}, signal_d)$ 。

## 2. 布林通道 (Bollinger Bands, BBands)

布林通道結合了移動平均和標準差的概念，其基本的型態是由三條軌道線組成的帶狀通道（中軌和上、下軌各一條）。「中軌」為股價的平均成本，「上軌」和「下軌」可分別視為股價的壓力線和支撐線。其反應的是目前價格的背離程度，因為價格大幅度背離（預設為兩個標準差以上）發生的機率不大，所以當發生時往往會在之後伴隨著回調。其計算方式非常簡單。

$$bband\_mid[0] = SMA(close, period)$$

$$bband\_upper[0] = bband\_mid[0] + mult \times stddev(close, period)$$

$$bband\_lower[0] = bband\_mid[0] - mult \times stddev(close, period)$$

其中

$$SMA(close, period) = \frac{\sum_{i=-period}^0 close[i]}{period}$$

$$stddev(close, period) = \sqrt{\frac{1}{period} \sum_{i=-period}^0 (close[i] - SMA(close, period))^2}$$

*period*、*mult* 為輸入的一部分參數。

### 3. 肯特納通道 (Keltner Channels)

與布林通道類似，肯特納通道結合了移動平均和真實波動幅度 (True Range, TR) 的概念，其基本的型態也是由三條軌道線組成的帶狀通道。其代表的意義也與布林通道相似，皆是判斷目前價格的背離程度，只是判斷的依據不同。其計算方式如下。

$$keltner\_mid[0] = EMA(close, period)$$

$$TR[0] = \max(high[0] - low[0], |high[0] - close[-1]|, |low[0] - close[-1]|)$$

$$TR\_ema[0] = \alpha \cdot TR[0] + (1 - \alpha) \cdot TR\_ema[-1]$$

$$keltner\_upper[0] = keltner\_mid[0] + mult \times TR\_ema[0]$$

$$keltner\_lower[0] = keltner\_mid[0] - mult \times TR\_ema[0]$$

其中

$$\alpha = 2 / (period + 1)$$

$$EMA(close[0], period) = \alpha \cdot close + (1 - \alpha) \cdot EMA(close[-1], period)$$

$period$ 、 $mult$  為輸入的一部分參數。

### 4. 乖離率 (Bias Ratio)

乖離率代表目前價格與移動平均線 (SMA) 的差距，能夠用來衡量短時間內的價格與平均的背離程度。其計算方式非常直觀。

$$bias = \frac{close[0] - SMA(close, period)}{SMA(close, period)} \times 100\%$$

$period$  為輸入的一部分參數。

### 5. 順勢指標 (Commodity Channel Index, CCI)

CCI 是用來測量目前價格趨勢的強弱，其背後的目的希望能更全面的分析一定時間內綜合價格的走勢，因此在計算時不只是以收盤價作為參考，它還進一步使用了最高價、最低價做為計算的一部份。其計算方式如下。

$$TP[0] = \frac{high[0] + low[0] + close[0]}{3}$$

$$TP\_sma = SMA(TP, period)$$

$$TP\_md = \frac{1}{period} \sum_{i=-period}^0 abs(TP[i] - TP\_sma[i])$$

$$CCI = \frac{TP[0] - TP\_sma[0]}{0.015 \times TP\_md}$$

其中

$abs(x) = |x| = \text{absolute value of } x$

$period$  為輸入的一部分參數。

## 6. 蔡金擺動指標 (Chaikin A/D Oscillator, ADOSC)

由累積／派發線 (AD Line) 得來，主要用來衡量長周期與短周期 A/D Line 的差異來確定市場的買賣壓力。背後的原理是透過比對資金流動的情況與目前價格的行為來檢測市場中資金流入和流出的情況，協助進行交易時機的制定。其計算方式如下。

$$A/D[0] = \frac{(close[0] - low[0]) - (high[0] - close[0])}{high[0] - low[0]} \times volume[0] + A/D[-1]$$

$$ADOSC = EMA(A/D, period\_short) - EMA(A/D, period\_long)$$

$period\_short$ 、 $period\_long$  為輸入的一部分參數。

### (三) 國內外相關研究

國內有關 Transformer 預測台股期價格的研究我們只找到了謝傑勝[1]的論文，但是電子全文在2028年5月中之後才開放外部瀏覽。若是將範圍放寬到預測台股價格，則夏鶴芸[2]有使用 Transformer 利用台股歷史股價、台股期、道瓊工業平均指數以及費城半導體指數，並搭配 BERT 預測新聞情緒去預測股價走勢與價格。其餘相關國內研究與謝傑勝的一樣都在未來才開放外部瀏覽。

國外的部分有 Zhaofeng Zhang、Banghao Chen、Shengxin Zhu、Nicolas Langrené 等人收集了中國股市從2010年至2019年4601支股票，超過五百萬筆的 K 棒進行訓練[4]；Lorenzo D. Costa、Alexei M. C. Machado 等人收集了2018年的巴西股市指數 (Ibovespa) 與 LSTM 進行比較[5]。

相關研究目前大多數不是未來才供觀看，就是與 LSTM 等舊模型比較股價預測的準確值，將其用於量化交易策略上的研究目前十分稀少，我們希望此專題能為這方面貢獻一些微薄的經驗。



### 三、研究方法

本章節會先介紹資料集，接著介紹在不加入任何模型預測下的交易策略以及交易規則，之後一一介紹我們所訓練的模型，並在每個模型後面說明我們如何將預測結果作為交易策略的一部份。

#### （一） 資料集

本專題中的資料來源主要有兩部分，一部分是透過「COCO 研究院」網站下載[1]，另一部份是透過永豐金證卷的 API 獲得。其中由「COCO 研究院」下載的台指期時段為2011/01/03至2023/12/08，透過永豐金證卷所獲得的資料時段為2020/03/02至2024/08/16，且兩者資料皆為1分 K。

在訓練模型時，我們會使用「COCO 研究院」2011年至2022年底以及永豐金證卷2023年初至2024年8月中的資料作為訓練、驗證以及測試集。在回測策略時，我們會使用永豐金證卷2022年初至2024年8月中的資料作為回測資料。並且在訓練以及回測時均只使用日盤資料。

此外，因為兩者資料來源皆有缺少部分資料，因此在訓練之前我們會根據不同模型進行不同的資料前處理方式，而回測的資料前處理與模型一號的處理方式相同。

#### （二） 交易策略及規則

##### 1. 交易規則

- （1） 初始資金為500,000元
- （2） 進出場以一口（200點）為單位
- （3） 每次進出場的手續費為2.5‰
- （4） 保證金（margin）為5%
- （5） 槓桿倍數不限，但不得超過20倍
- （6） 所有交易皆為日內交易
- （7） 若（未實現報酬／總資金）< 50%則強平所有倉位

##### 2. 交易策略

我們會使用 KDJ、Bollinger Bands、Keltner Channel 這三個技術指標以及目前 K 棒的狀態作為開關倉的條件，且我們只會在每日9：15到13：15這時間段進出場，若13：15時還持有倉位，則會在下一分鐘以市價去關倉，除此之外，

我們同時間只會持有一個倉位，且倉位大小固定為一口。下表為各技術指標的參數設定。

表1 各技術指標之參數

名稱	參數
KDJ	(period = 30, signal_k = 3, signal_d = 3)
Bollinger Bands	(period = 15, mult = 2)
Keltner Channel	(period = 15, mult = 1.5)

### (1) 多頭開倉

我們首先定義「擠壓」狀態：當 Bollinger Bands 的上下軌同時被 Keltner Channel 的上下軌包圍時，我們稱目前價格處在「擠壓」狀態中，其意義為目前無明顯趨勢。



圖4 「擠壓」狀態示意圖

圖中藍色線條為 Keltner Channel，紫色線條為 Bollinger Bands。當紫色線條被藍色線條包圍時（藍色背景的部分）我們稱目前為「擠壓」狀態。請注意這裡以及之後的圖 K 棒顏色為綠漲紅跌！

在「擠壓」狀態中，若滿足以下兩個條件，則我們會說多頭開倉的前提條件已經達成。

- I.  $close[0] < bband\_lower[0]$
- II.  $J[0] < 20$



圖5 多頭進場前提條件示意圖

可以看到黃色線段圈起來的K棒同時滿足「在『擠壓』狀態中」、「收盤價小於布林通道下軌」、「對應的J值小於20」。因此直到下一個「擠壓」狀態前多頭開倉的前提條件皆為已滿足。

在前提條件已經滿足且尚未到達下一個「擠壓」狀態前，若又滿足以下兩個條件，則我們會用市場價進場做多

- I.  $\text{close}[0] > \text{open}[0]$  (K 棒收漲)
- II.  $J[0] > 20$



圖6 多頭進場示意圖

可以看到黃色線段圈起來的 K 棒同時滿足「進場前提條件已達成」、「K 棒本身收漲」、「對應的 J 值大於 20」。因此，我們會在下一分鐘，也就是粉色線段圈起來的 K 棒進場。

開完倉後，我們會設定止損價格以及 J 值關倉的初始條件

- I. 止損價格 (sl\_price) 為  $\min(low[-1], low[-2], \dots, low[-5])$
- II.  $J\_hold\_threshold = 80$

## (2) 多頭關倉

在開完倉後，我們會維護剛剛設定的  $J\_hold\_threshold$ ，若目前的  $J$  值大於  $J\_hold\_threshold + 5$ ，則我們會更新  $J\_hold\_threshold$  為  $\lfloor J[0] \div 5 \rfloor \times 5$ 。

關倉與開倉一樣，必須先滿足前提條件後才會判斷是否關倉。若是在開倉後滿足  $J[0] > 80$ ，則直到關倉前我們都說前提條件已經被滿足。



圖7 多頭出場前提條件示意圖

由圖可知黃色線段圈起來的 K 棒滿足「 $J$  值大於 80」，因此我們直到關倉前都認定關倉的前提條件已被滿足。

在達成前提條件的情況下，若是

- I.  $J[0] \leq J\_hold\_threshold$
- II.  $close[0] < sl\_price$

則我們會在下一分鐘時將其用市價關單出場。



圖8 多頭出場示意圖

可以看到黃色線段圈起來的 K 棒滿足「J 值小於等於 J\_hold\_threshold」，因此我們會在粉色線段圈起來的地方關單出場。

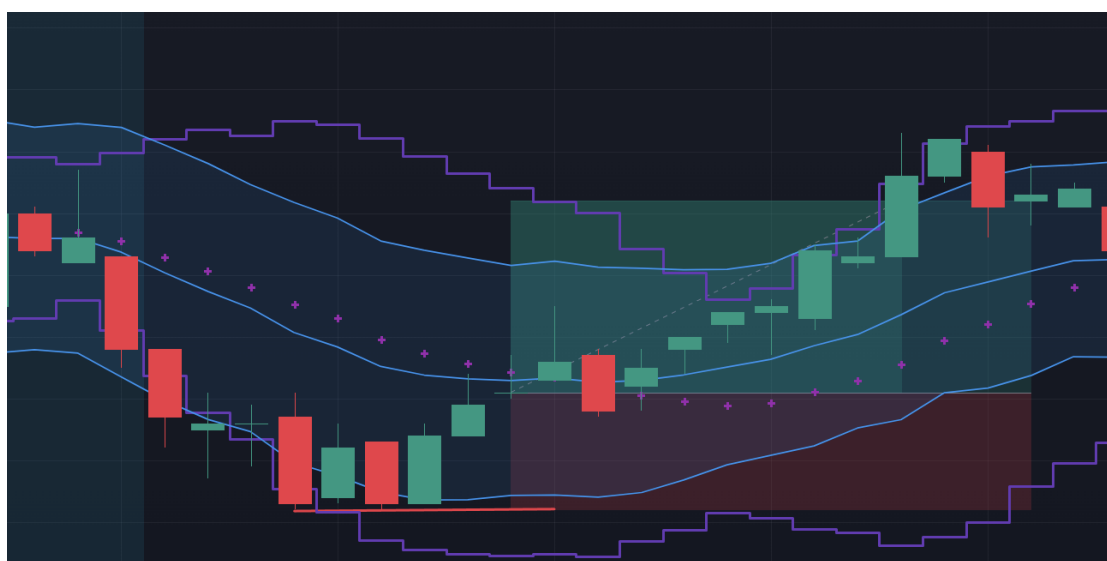


圖9 多頭損益圖範例

### (3) 空頭開倉

空頭的開倉與多頭類似，在「擠壓」狀態中，若滿足以下兩個條件，則我們會說空頭開倉的前提條件已經達成。

- I.  $close[0] > bband\_upper[0]$
- II.  $J[0] > 80$



圖10 空頭進場前提條件示意圖

可以看到黃色線段圈起來的K棒同時滿足「在『擠壓』狀態中」、「收盤價大於布林通道上軌」、「對應的J值大於80」。因此直到下一個「擠壓」狀態前多頭開倉的前提條件皆為已滿足。

在前提條件已經滿足且尚未到達下一個「擠壓」狀態前，若又滿足以下兩個條件，則我們會用市場價進場做空

- I.  $\text{close}[0] < \text{open}[0]$  (K 棒收跌)
- II.  $J[0] < 80$



圖11 空頭進場示意圖

可以看到黃色線段圈起來的 K 棒同時滿足「進場前提條件已達成」、「K 棒本身收跌」、「對應的 J 值小於80」。因此，我們會在下一分鐘，也就是粉色線段圈起來的 K 棒進場。

開完倉後，我們一樣會設定止損價格以及 J 值關倉的初始條件

- I. 止損價格 (sl\_price) 為  $\max(\text{high}[-1], \text{high}[-2], \dots, \text{high}[-5])$
- II. J\_hold\_threshold = 20

#### (4) 空頭關倉

在開完倉後，我們會維護剛剛設定的 J\_hold\_threshold，若目前的 J 值小於 J\_hold\_threshold - 5，則我們會更新 J\_hold\_threshold 為  $\lfloor J[0] \div 5 \rfloor \times 5$ 。

關倉與開倉一樣，必須先滿足前提條件後才會判斷是否關倉。若是在開倉後滿足  $J[0] < 20$ ，則直到關倉前我們都說前提條件已經被滿足。



圖12 空頭出場前提條件示意圖

由圖可知黃色線段圈起來的 K 棒滿足「J 值小於 20」，因此我們直到關倉前都認定關倉的前提條件已被滿足

在達成前提條件的情況下，若是

- I.  $J[0] \geq J\_hold\_threshold$
- II.  $close[0] > sl\_price$

則我們會在下一分鐘時將其用市價關單出場。



圖14 空頭出場示意圖

可以看到黃色線段圈起來的 K 棒滿足「J 值大於等於 J\_hold\_threshold」，因此我們會在粉色線段圈起來的地方關單出場。



圖13 空頭損益圖範例

### （三） 模型一號

#### 1. 資料前處理

在此模型中，我們將缺失的資料分為兩種：「頭／尾」與「日內」。「頭／尾」的定義為缺失資料時段的起點為8：46或是終點為13：45，也就是日盤的頭尾；「日內」則指非「頭／尾」的缺失資料。根據不同種類的資料我們有不同的



處理方式。

若缺失的資料屬於「頭／尾」，那麼我們會找到最近的 K 棒，並利用此 K 棒的資料將缺少的部分補齊，且此時的交易量我們會設為0。例如缺失的資料時段為8：46至8：55，則我們會將該時段的 Open、High、Low 以及 Close 皆設為8：56時的資訊，並且8：46至8：55該時段的交易量為0。

若缺失的資料屬於「日內」，則我們希望會缺少的資料會根據最近的前後兩筆資料形成一線性關係，且上一根 K 棒的收盤價為下一根 K 棒的開盤價。例如缺失的資料時段為12：01至12：10，且12：00與12：11的價格為下表

表2 缺少的資料時段中最近的前後兩筆資料

時間	Open	High	Low	Close	Volume
12：00	10010	10020	10005	10015	200
12：11	10030	10040	10025	10035	300

我們令較早的 K 棒為 Head，較晚的 K 棒為 Tail（例子中12：00為 Head、12：11為 Tail）則我們會根據這兩根 K 棒的價格計算出五個差值：較晚 K 棒的開盤價與較早 K 棒的收盤價（ $\text{Open}[\text{Tail}] - \text{Close}[\text{Head}]$ ，entity\_delta）、較晚 K 棒與較早 K 棒的最高價（ $\text{High}[\text{Tail}] - \text{High}[\text{Head}]$ ，high\_delta）、較晚 K 棒與較早 K 棒的最低價（ $\text{Low}[\text{Tail}] - \text{Low}[\text{Head}]$ ，low\_delta）、較晚 K 棒與較早 K 棒的交易量（ $\text{Volume}[\text{Tail}] - \text{Volume}[\text{Head}]$ ，volume\_delta），以及總共缺少了多長時間的資料（ $\text{Tail} - \text{Head} - 1$ ，time\_delta）。如表3所示。

表3 上述之各差值

time_delta	entity_delta	high_delta	low_delta	volume_delta
10	15	20	20	100

接著，我們以12：00為起點，依照目前經過了 i 分鐘去補齊資料，例如對於12：01而言 i 為1。補齊方式如下表

表4 缺少的資料時段中第 i 筆資料的補齊方式

Open	$\text{Close}[i - 1]$
------	-----------------------

<b>High</b>	$High[Head] + (high\_delta \div time\_delta) \times i$
<b>Low</b>	$Low[Head] + (low\_delta \div time\_delta) \times i$
<b>Close</b>	$Close[Head] + (close\_delta \div time\_delta) \times i$
<b>Volume</b>	$Volume[Head] + (volume\_delta \div time\_delta) \times i$

補齊完缺失的資料後，我們會接著計算表5中的技術指標以及其他表中所提及的數據作為模型的輸入，對於特定的技術指標，我們會同時計算長周期與短周期的值，並用 long、short 去區分。

表5 模型的輸入參數及說明

參數名稱	說明
TXF_open	開盤價
TXF_high	最高價
TXF_low	最低價
TXF_close	收盤價
TXF_volume	交易量
bias	乖離率。bias_long 的 period = 9，bias_short 的 period = 3
adosc	蔡金擺動指標。period_long = 15、period_short = 6
bband	布林通道。period = 20、mult = 2
keltner	肯特納通道。period = 20、mult = 1.5
J	KDJ 指標。period = 25、signal_k = 3、signal_d = 3

cci	順勢指標。cci_long 的 period = 21，cci_short 的 period = 14
kbar_diff	最高價－最低價
solid_kbar_diff	最高價－最低價
upper_shadow	最高價－max(開盤價, 收盤價)
lower_shadow	min(開盤價, 收盤價)－最低價

## 2. 模型架構

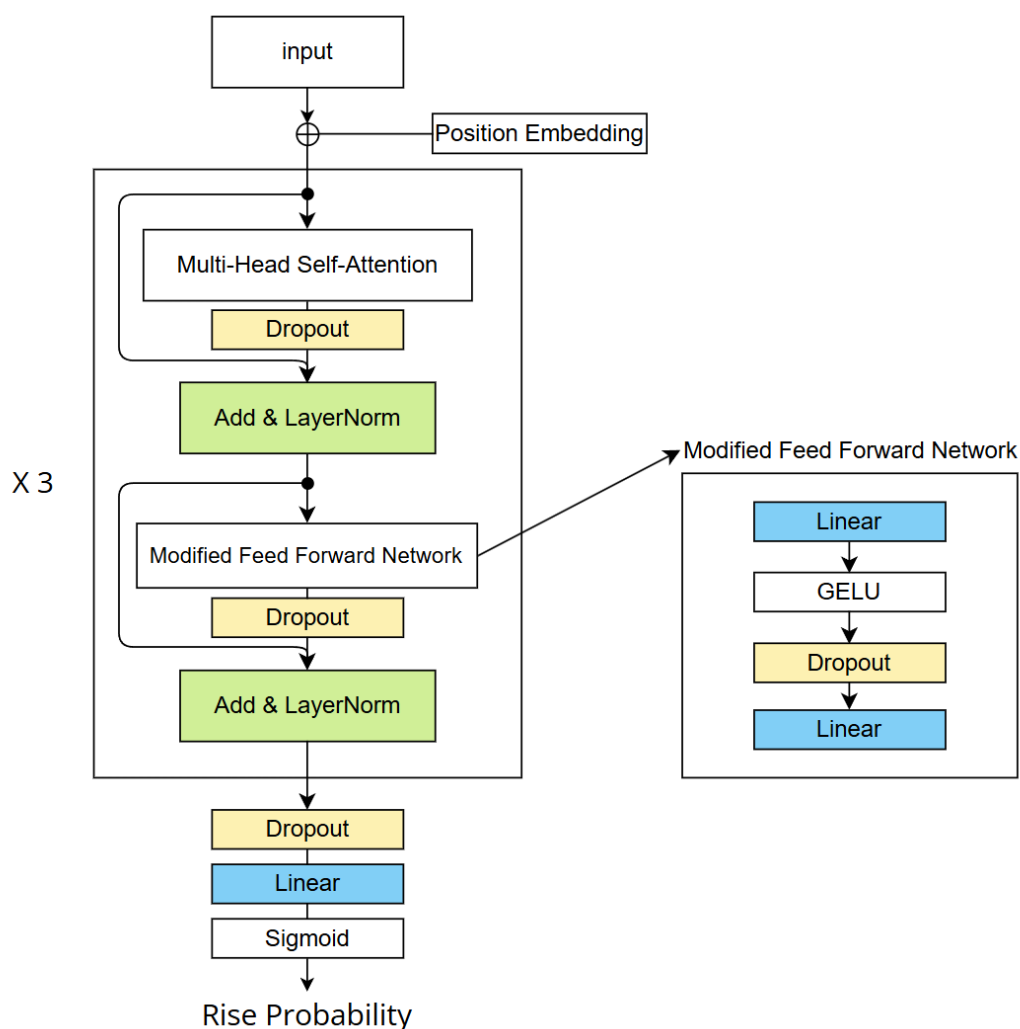


圖15 模型一號架構圖

本模型以 Transformer Encoder 為主，輸入為表5所提的參數，每次會提供前兩個小時的資料進行預測。由於我們相信歷史會一再重複，所以我們認為類似

的 K 線走勢會形成接近的結果，因此在將資料進行 Position Encoding 之前我們會將這兩小時的資料都做最小最大值標準化（MinMax Standardization）以保存參數各自的線性關係的同時將數值範圍縮小至[0, 1]。資料前處理結束後，圖15 的 Transformer-Based 模型一號會去預測15分鐘後收盤價漲的機率。表6為各超參數的值。

表6 模型一號超參數

名稱	值
SEQ_LEN	120
D_MODEL	21
NUM_HEADS	3
DROPOUT_RATE	0.2
BATCH_SIZE	64
EPOCHS	20

訓練時採用 K-Fold Validation 加上 Early Stopper 的方式進行訓練，使用 AdamW 作為優化器（Optimizer）、BCELoss 作為損失函數（Loss Function）進行評估，並使用 pytorch 內建的 CosineAnnealingLR 在每個 Epoch 結束時調整 AdamW 的學習率。

### 3. 應用方式

在訓練及驗證模型時，我們發現對於預測結果來說，不論是數量還是正確率皆是「跌或是保持不變」這個分類比較高。因此，我們在加入此模型到交易策略時只考慮對空頭的部分進行修改。

因為模型需要前兩個小時的資料才能進行預測，所以我們只會在每日的 11：15過後才會開始使用模型進行預測。當前時間尚未到達時我們會採用原先的交易策略。11：15開始我們每分鐘會預測出一個結果並更新一個長度為15的 list。當空單的前提條件滿足後，除了原先的進場條件外，若是 list 中結果為「跌或是保持不變」的數量佔了超過2/3，則我們也會進行開單。

#### （四） 模型二號

##### 1. 資料前處理

計算所有欄位中的缺失值數量，若非0便使用呼叫函數線性插值 `interpolate(method='linear')` 補全數據。線性插值方法會根據缺失值的前後數據推斷出合理的數值來填補。頭尾缺失將交易量設為0，並將其他資料根據最近的上次資料補齊。

##### 2. 模型架構

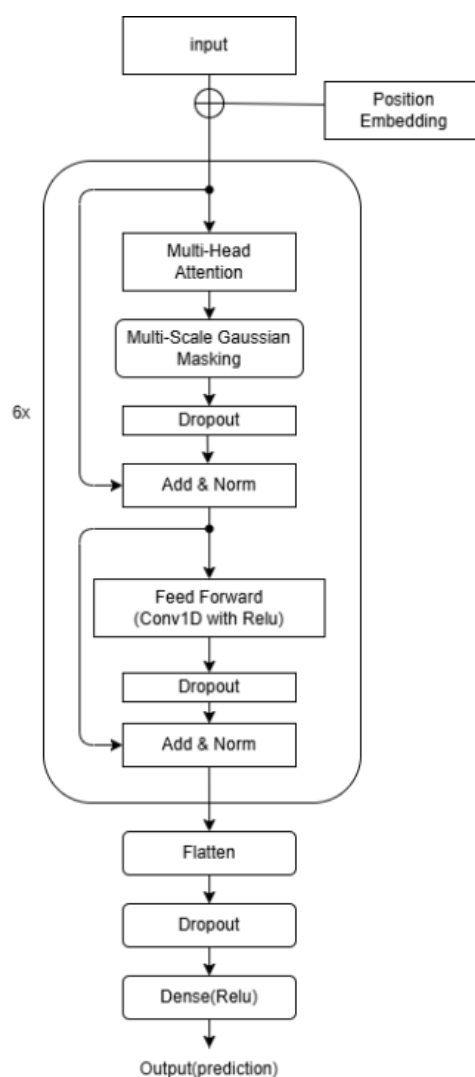


圖16 模型二號架構圖

本模型以 Transformer Encoder 為主，透過增加 Multi-Scale Gaussian Prior 的 mask 來強化 Transformer 中的注意力機制對不同尺度（例如短期和長期趨勢）的捕捉[6]。透過不同的「窗口大小」（如5、10、20、40）控制範圍和影響力。

### (1) Multi-Scale Gaussian Prior Mask

我們會先透過下列公式算出不同的 Mask

$$\left[B_h^{(G)}\right]_{i,j} = \begin{cases} \exp\left(-\frac{(j-i)^2}{2\sigma_h^2}\right) & \text{if } j \leq i \\ 0 & \text{Otherwise} \end{cases}$$

其中 $\sigma_h$ 為 window size， $i$ 、 $j$  表示序列中的位置。當  $j \leq i$  時，Mask 的值代表位置  $i$  與位置  $j$  的 gaussian prior 相關強度。當  $j > i$  時，位置  $j$  對位置  $i$  不產生影響，因此這時候的 Mask 值為0。這些 gaussian mask 最終會與 attention matrix 疊加，以在注意力計算中加入 gaussian prior 相關強度。每個不同尺度的 mask（如 5、10、20、40 大小）都會生成一組不同的偏置矩陣。這些偏置矩陣堆疊在一起並與原本的 attention matrix 相加，然後取平均，並最後使用 softmax 函數來標準化這些 attention 分數。

資料前處理後，圖16的模型二號會使用過去64分鐘的 Open、High、Low、Close、Volume 去預測10分鐘後的收盤價，並且在輸入之前會使用區間內各自的平均值以及標準差做局部的標準化。表7為各超參數的值。

表7 模型二號超參數

名稱	值
SEQ_LEN	64
D_MODEL	24
NUM_HEADS	3
DROPOUT_RATE	0.1
BATCH_SIZE	64
EPOCHS	25

訓練時加上 Early Stopper 進行訓練，使用 Adam 作為 Optimizer、MSE 作為 Loss Function 進行評估，並使用 pytorch 內建的 LambdaLR 在每個 Epoch 結束時

調整 Adam 的學習率。此外，我們還使用混和精度訓練配合 GradScaler 來自動增量，確保數值穩定，可以加速模型的訓練速度。

### 3. 應用方式

我們會先比較當前收盤價與預測的價格，若預測價格高於當前價格，則返回 True（表示可能上漲），否則返回 False（表示可能下跌）並將回傳值加到策略當中。之後，在本來的交易策略上，我們將交易策略修改為

- (1) 在持有多頭倉位的時候，若是
  - I.  $J[0] \leq J\_hold\_threshold$
  - II. 信號為 sell
  - III. 模型預測價格下跌(返回值為 false)

則將倉位平倉。

- (2) 在持有空頭倉位的時候，若是
  - I.  $J[0] \geq J\_hold\_threshold$
  - II. 信號為 buy
  - III. 模型預測價格上漲(返回值為 true)

則將倉位平倉。

#### (五) 模型三號

##### 1. 資料前處理

首先，我們將資料依日期進行排序，並計算每筆資料的時間差。若時間差異不為1分鐘，即表示該處出現缺失資料。我們透過以下方式進行補齊：找出缺失的資料位置：利用 `find_missing_rows` 函數，確認缺失位置及持續時間。每當時間間隔超過1分鐘，我們計算出該區間內缺少的分鐘數，並記錄缺失的起始時間與數量。

填補缺失資料：在 `fill_missing_values` 函數中，我們基於缺失位置與數量，生成相對應的空白資料，並使用鄰近的數值進行前向填補（`forward fill`）與後向填補（`backward fill`）。最後再進行線性插值，以確保缺失的價格數據平滑過渡。另外，對於成交量欄位，我們利用價格區間與平均成交量的比率來估計填補，以便維持數據的一致性。

## 2. 模型架構

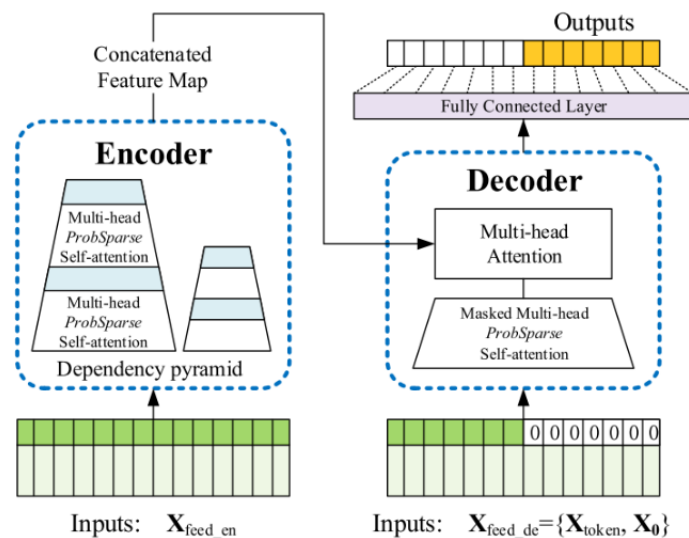


圖17 模型三號架構圖

本模型基於 Informer[7]，使用 Transformer 架構的 Encoder-Decoder 機制進行時間序列預測。Informer 模型在應對長期序列數據的同時，通過 ProbSparse Self-Attention 和 Distilling Operation 技術有效降低了計算成本，因此特別適合應用於需要處理大量數據且具長期依賴的金融商品價格預測任務中。

### (1) ProbSparse Self-Attention

傳統的自注意力機制計算複雜度為  $O(L^2 \cdot d)$ ，其中  $L$  是序列長度， $d$  是特徵維度。當序列長度增加時，這會導致高昂的計算成本。為了解決這個問題，Informer 引入了 ProbSparse Self-Attention。該技術通過概率分佈選擇最關鍵的注意力項，而非每個項目都進行計算。這種方式根據 Kullback-Leibler 散度選擇那些信息量大的位置，從而顯著降低計算量，使得計算複雜度下降至  $O(L \cdot \log L)$ ，大幅提升效率。

### (2) Distilling Operation

Distilling Operation 是一種多分辨率學習方法，用於分層提取不同時間尺度的信息。透過逐層壓縮原始時間序列數據，模型能夠捕捉短期和長期的趨勢。具體而言，底層的 Distilling Operation 層主要提取短期波動的信息，而更高層次的操作則捕捉長期趨勢。這樣的多尺度處理對於處理多尺度依賴的金融數據（例如短期波動和長期趨勢並存的市場數據）特別有效。Distilling Operation 還能進一步壓縮數據，減少模型的計算負擔，增強模型的泛化能力。

此外，我們在 Position Embedding 之前也會做 MinMax Standardization 以減少模



型的訓練難度。資料前處理結束後圖17的模型三號會使用目前的 Open、High、Low、Close、Volume 去預測未來一段時間內每分鐘的收盤價。表8為各超參數的值。

表8 模型三號超參數

名稱	值
SEQ_LEN	120
ENCODER_NUM	4
LABEL_LEN	60
DECODER_NUM	3
PRED_LEN	60
D_MODEL	512
NUM_HEADS	8
DROPOUT_RATE	0.1
BATCH_SIZE	64
EPOCHS	20

其中 LABEL\_LEN 是指 Decoder 在 decode 時會參考前60分鐘的值，PRED\_LEN 是指模型最終會預測出未來一小時內每分鐘的價格。

訓練時加入 Early Stopper 進行訓練，使用 AdamW 作為 Optimizer、MSE 作為 Loss Function 進行評估，並使用 pytorch 內建的 CosineAnnealingLR 在每個 Epoch 結束時調整 AdamW 的學習率。

### 3. 應用方式

在原有以 KDJ 與 keltner\_bands 為交易主軸的策略上，我們加入模型預測的價格作為輔助開關倉的最後一道濾網，用 predicted\_1min 與 predicted\_30min 值

的高低決定是否執行該筆交易。以多頭倉位為例，若原先策略有做多的訊號，且此時  $\text{predicted\_30min} > \text{predicted\_1min}$ ，倉位才會順利開啟，空頭的部分反之亦然。透過多一層濾網的檢視，可以更好的避免掉一下開倉的雜訊。

#### 四、研究成果

本章節會先介紹在不考慮任何模型下，單純使用交易策略的績效。接著再一一展示加入各模型的預測作為進出場條件後的績效。

##### 1. 純交易策略

表9 純交易策略的評估表

評估指標	數值
最終資金	850566.9
多單次數	總共1202／贏642／輸560
空單次數	總共1129／贏600／輸529
勝率	總共53.28％／多53.41％／空53.14％
夏普比率	0.866358
最大回撤	0.273％
SQN	1.55544

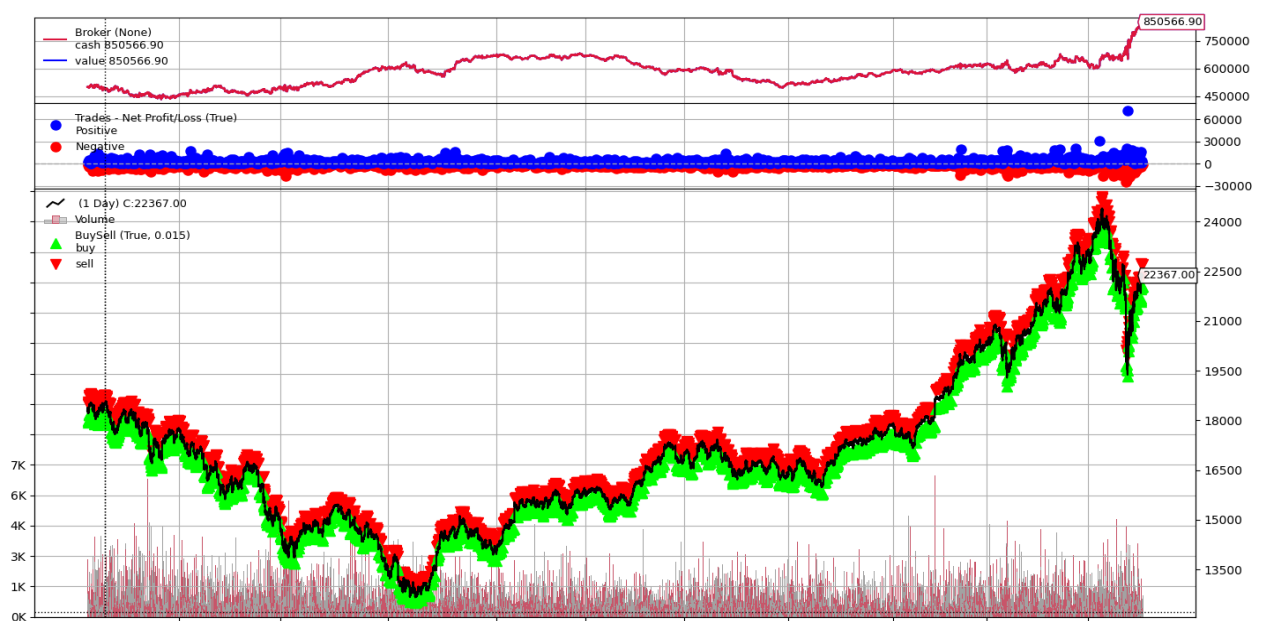


圖18 純交易策略的績效圖

## 2. 加入模型一號

表10 加入模型一號後的交易策略評估表

評估指標	數值
最終資金	925718.5
多單次數	總共900／贏496／輸404
空單次數	總共7915／贏3121／輸4794
勝率	總共41.03％／多55.11％／空39.43％
夏普比率	0.8542385
最大回撤	0.4529％
SQN	1.50011

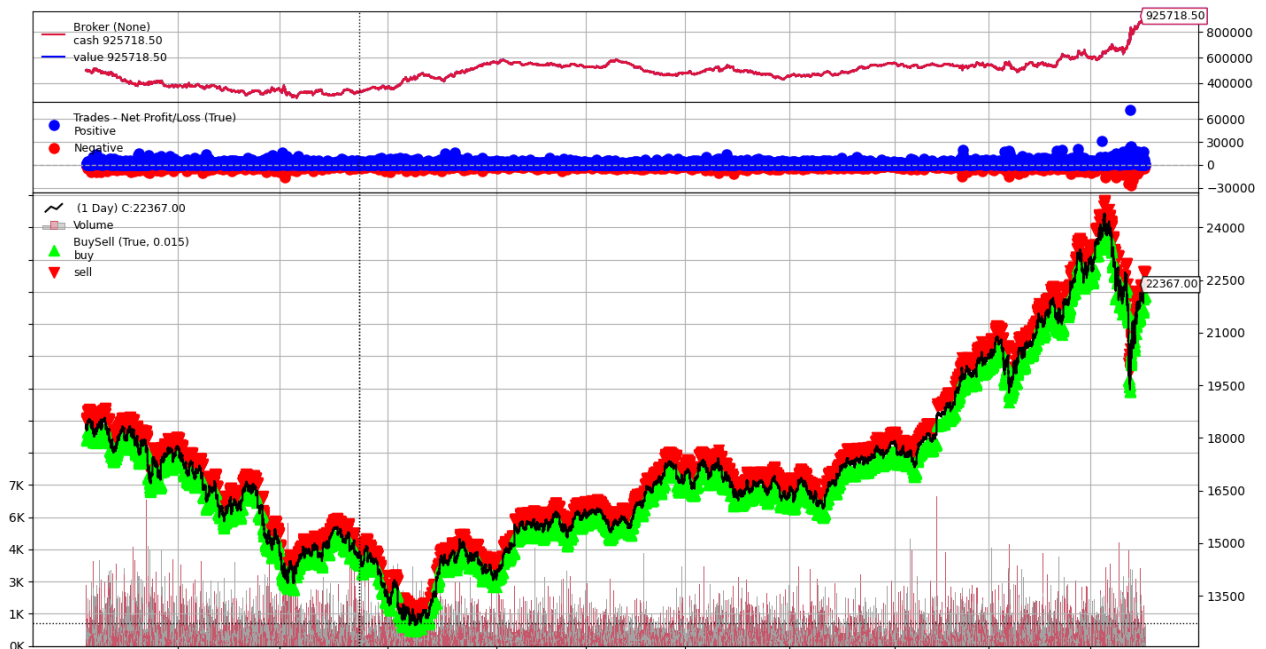


圖19 加入模型一號後的交易策略績效圖

### 3. 加入模型二號

表11 加入模型二號的交易策略評估表

評估指標	數值
最終資金	752105.9
多單次數	總共0／贏0／輸0
空單次數	總共941／贏281／輸660
勝率	總共29.86％／多0％／空29.86％
夏普比率	0.698528
最大回撤	0.3657％
SQN	0.925456

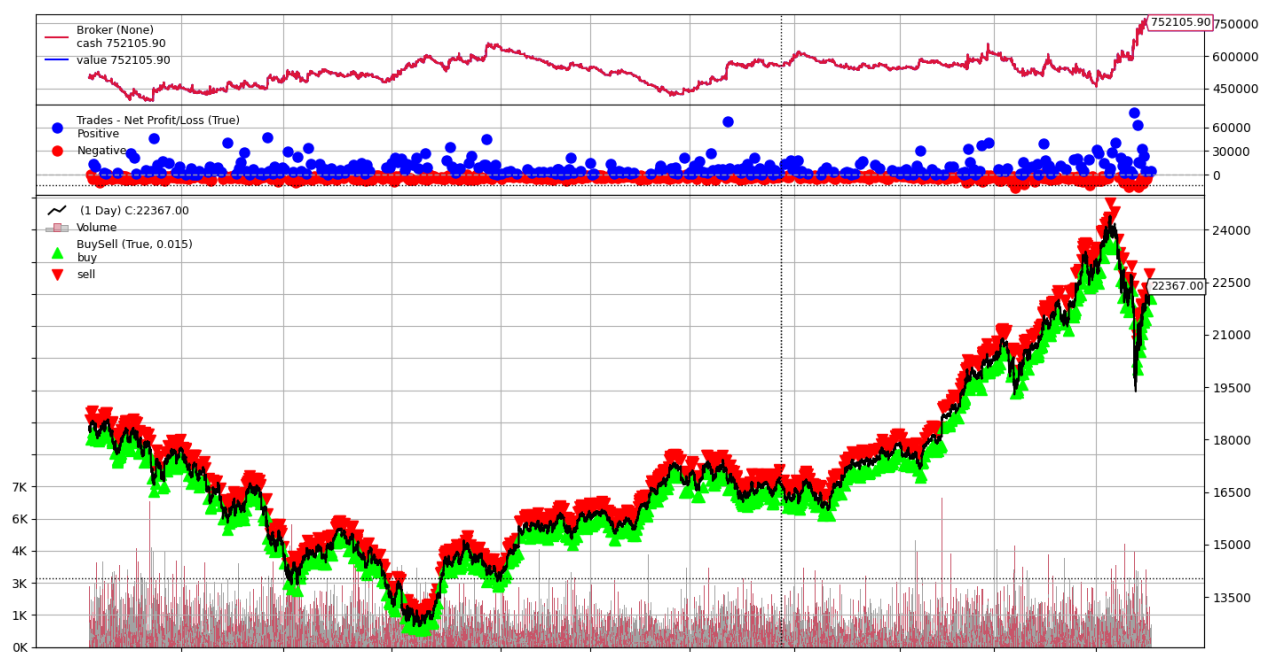


圖20 加入模型二號後的交易策略績效圖

#### 4. 加入模型三號

表12 加入模型三號的交易策略評估表

評估指標	數值
最終資金	935605.6
多單次數	總共3530／贏1641／輸1889
空單次數	總共4414／贏2001／輸2413
勝率	總共45.85％／多46.89％／空45.33％
夏普比率	1.218184
最大回撤	0.20959%
SQN	1.898358

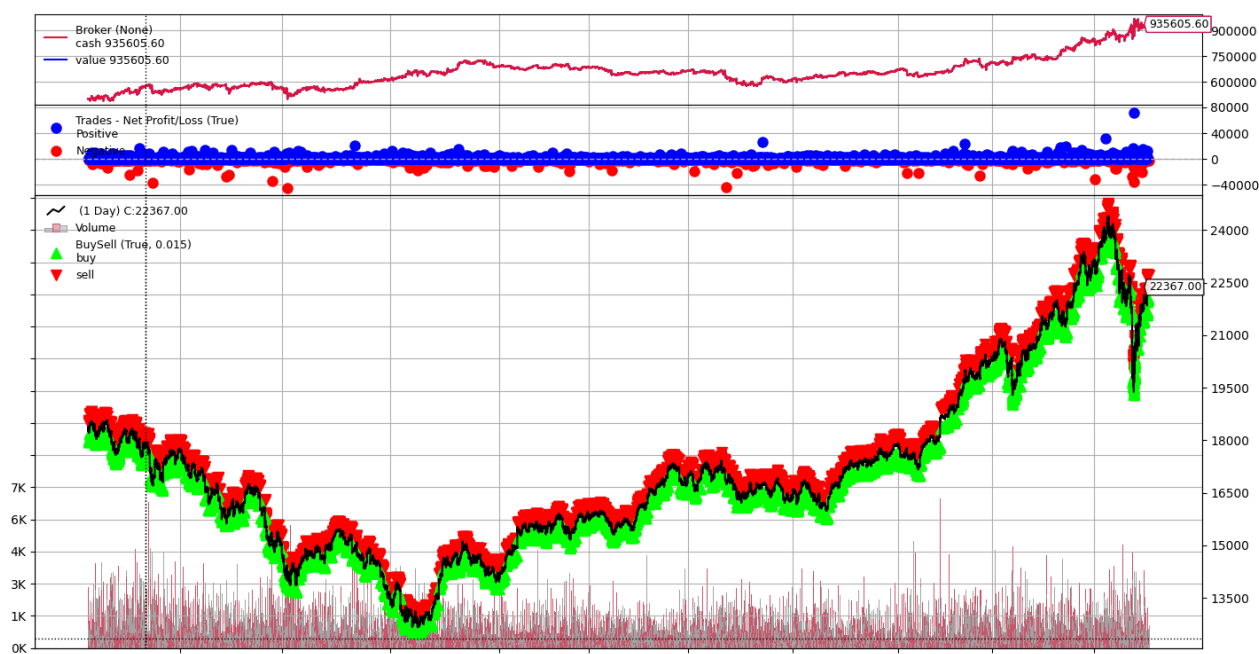


圖21 加入模型三號後的交易策略績效圖

## 五、結論

從實驗結果可以看到加入模型預測作為交易策略中進出場條件的一部份會犧牲一定的品質，而且其整體績效與品質會和模型的好壞掛勾。但是從另一方面來說，只要模型與交易策略搭配得當，兩者的組合是可以創造出一加一大於二的可能的。

## 六、未來展望

我們認為未來能改善的地方主要有兩點：模型的正確率以及交易策略的品質。

第一，我們認為模型的正確率可以再更高一點，因為就我們目前的模型來講只能做到正確率接近數據的分布，而我們認為除了利用不斷改良的其他 **Transformer-Based** 模型來進行時序資料的預測外，也許使用 **LSTM** 之父 **Sepp Hochreiter** 近期所提出的 **xLSTM**[8]來進行預測也可以訓練出更準確的模型。此外，我們認為於2024年5月所提出的神經模型架構 **KAN**[9]亦是能夠提高預測準確度的一大可能，因為我們可以利用 **KAN** 架構去結合金融、統計等相關的定理、公式，從而使新一代的 **Kansformer** 預測準確率大幅提升。

第二，我們認為目前的交易策略尚未能達到我們理想的品質。從 **SQN** 以及夏普比率可以知我們的策略品質在中下游的水準，從最終資金也可以知道我們的策略會輸給大盤。我們認為這原因除了透過調整以及利用其他技術指標去改良外，建立一個更好的倉位以及資金管理系統也可以幫助我們提升交易策略的品質。

## 參考文獻

- [1] 謝傑勝, “以 Transformer 模型預測台股期走勢之交易策略,” M.S. thesis, Dept. IIS., NTHU., Hsinchu, Taiwan, 2023. .
- [2] 夏鶴芸, “應用深度學習與自然語言處理新技術預測股票走勢 – 以台積電為例,” M.S. thesis, Dept. IIM., NTPU., New Taipei City, Taiwan, 2020. .
- [3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser and Illia Polosukhin, “Attention Is All You Need,” *NIPS*, June. 2017.
- [4] Zhaofeng Zhang, Banghao Chen, Shengxin Zhu and Nicolas Langrené, “Quantformer: from attention to profit with a quantitative transformer trading strategy,” *IEEE PAMI*, vol. 45, pp. 8813-8826, July. 2023.
- [5] Lorenzo D. Costa and Alexei M. C. Machado, “Prediction of Stock Price Time Series using Transformers,” *BRAZILIAN WORKSHOP ON ARTIFICIAL INTELLIGENCE IN FINANCE (BWAIF)*, Porto Alegre: Sociedade Brasileira de Computação, 2023, pp. 85-95.
- [6] Qianggang Ding, Sifan Wu, Hao Sun, Jiadong Guo and Jian Guo, “Hierarchical Multi-Scale Gaussian Transformer for Stock Movement Prediction,” in *29th IJCAI. Special Track on AI in FinTech*, Yokohama, 2020, pp. 4640-4646
- [7] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong and Wancai Zhang, “Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting,” in *35th AAAI Conf. on Artificial Intelligence*, Vancouver, 2021, pp. 11106-11115
- [8] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter and Sepp Hochreiter, “xLSTM: Extended Long Short-Term Memory,” *arXiv preprint*, arXiv:2405.04517, May. 2024, [Online]. Available: <https://arxiv.org/pdf/2405.04517>
- [9] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou and Max Tegmark, “KAN: Kolmogorov–Arnold Networks,” *arXiv preprint*, arXiv:2404.19756, Apr. 2024, [Online]. Available: <https://arxiv.org/pdf/2404.19756>