以助記詞為核心的虛擬通貨取證研究

Investigation into Cryptocurrency Forensics Focused on Mnemonic Phrases

高信雄
中央警察大學資訊管理學系

董正談
中央警察大學資訊管理學系

楊晟岑
中央警察大學資訊管理學系

*賀重愷
中央警察大學資訊管理學系
kyle891125@gmail.com

摘要

近年來，全球範圍內的犯罪組織及網路犯罪者廣泛使用比特幣等加密貨幣進行資金洗錢和跨國轉移，企圖逃避執法機關的偵查。面對此情況，各國政府紛紛制定相關法律與政策，以追回這些犯罪活動中所獲得的資產。在這一過程中，取得私鑰或助記詞成為關鍵，其中助記詞尤為重要，它是恢復和存取加密錢包的核心資訊。我們觀察到在許多案例中，執法機關若在行動時未能即時扣押加密貨幣，則可能導致資產被迅速轉移至其他錢包，大幅增加追蹤的複雜度。針對此問題，本研究開發了一種基於深度學習的創新鑑識方法。此方法專注於快速分析可能含有助記詞的可疑文件，例如犯罪分子電腦中的瀏覽器暫存檔或是用來隱藏助記詞的文本文件。為了增加方法的泛用性和靈活性，我們特別強調使用自然語言處理技術（NLP），支援包括英文和中文在內的多達 11 種語言的助記詞。此外，我們使用了 LSTM、BiLSTM、RNN、TextCNN 4 種 NLP 模型深度學習模型進行比較和評估，以確保鑑識方法的準確性和有效性，其準確率達到了 99%。這種方法不僅效率高（提升達到 80.27 倍），而且具有更高的實用性和泛用性，能夠適應各種不同的取證環境和需求。

關鍵字：加密貨幣、助記詞取證、深度學習、自然語言處理（NLP）

Abstract

In recent years, criminal organizations and cybercriminals globally have extensively used cryptocurrencies, such as Bitcoin, for money laundering and cross-border transfers in attempts to evade detection by law enforcement agencies. In response to this situation, governments around the world have been formulating relevant laws and policies to recover assets obtained through these criminal activities. In this process, acquiring private keys or mnemonic phrases has become crucial, with the mnemonic phrase being particularly important as it is the core information for restoring and accessing encrypted wallets. We have observed that if law enforcement cannot immediately seize the cryptocurrencies during actions, it may lead to the rapid transfer of assets to other wallets, significantly increasing the complexity of tracking. To address this issue, this study developed an innovative forensic method based on deep learning. This method focuses on the rapid analysis of suspicious documents that may contain mnemonic phrases, such as browser cache files on criminals' computers or text documents used to hide mnemonic phrases. To enhance the versatility and flexibility of the method, we particularly emphasize the use of natural language processing (NLP) technology, supporting mnemonic phrases in up to 11 languages, including English and Chinese. Additionally, we employed four types of NLP deep learning models—LSTM, BiLSTM, RNN, and TextCNN—for comparison and evaluation to ensure the accuracy and effectiveness of our identification method, achieving an accuracy rate of 99%. This method is not only efficient, with an improvement of 80.27 times, but also highly practical and versatile, capable of adapting to various forensic environments and requirements.

Keywords: Cryptocurrency, Mnemonic Phrase Forensics, Deep Learning, Natural Language Processing (NLP)

1. 前言

隨著網路科技的飛速發展，使用加密貨幣作為交易媒介的犯罪活動顯著增加。自 2008 年加密貨幣概念首次提出以來，比特幣、乙太幣和太達幣等已成為公眾關注的焦點。這些貨幣的分散式、點對點的交易特性以及高度匿名性，使得加密貨幣成為犯罪分子用於洗錢的首選方式。此外，洗錢和非法市場交易並非比特幣被利用於犯罪的唯一方式。比特幣已經成為各種形式的組織犯罪活動的流行資產，這包括勒索軟體攻擊、剝削兒童、販毒、勒索、資助恐怖主義和人口販運等。根據 chainalysis 在 2022 年的報告[1]，涉及加密貨幣的犯罪活動的總規模價值已超過 140 億美元。這一數據凸顯了加密貨幣在當代犯罪活動中所扮演的重要角色，同時也表明了對這種新型資產進行有效監管和犯罪打擊的迫切需求。

歐洲委員會在 1991 年的《洗錢和沒收犯罪所得公約》中提出的「從犯罪中獲取利潤」概念[2]，強調了通過沒收非法資金來削弱犯罪動機，進而遏制犯罪活動的重要性。然而，儘管區塊鏈和加密貨幣的研究日益豐富，關於犯罪現場的鑑識研究仍顯不足，特別是針對查扣加密貨幣時必要的助記詞（Mnemonic Phrase）證技術。在加密貨幣錢包的安全機制中，助記詞的角色至關重要。助記詞是一組特定的關鍵字或短語，用於加密錢包的恢復和存取。當犯罪分子刪除其加密錢包後，他們可透過輸入相應的助記詞來恢復對該錢包的存取，這一過程突顯了助記詞在加密貨幣交易中的核心地位。對於執法機構而言，若無法及時取得相關的助記詞，往往意味著執法者難以進行加密貨幣的查扣，這不僅嚴重阻礙了調查進展，還極大增加了將犯罪分子繩之以法的難度。

目前文獻中對於加密貨幣犯罪調查的研究相對有限，且主要集中在比特幣的取證方法上。[3]例如 Zollner 等人提出了一種專為 Windows 系統設計的自動化工具[4]，用於比特幣的即時取證和事後分析。該工具的目標是識別包括私鑰、比特幣地址、助記詞、錢包 ID 和電子郵件等在內的多種數據，並能夠對 RAM、hiberfil.sys 和 pagefile.sys 等文件進行分析。然而，這些工具存在一些顯著的不足之處。首先，在處理速度方面，這些工具常常無法滿足迅速取證的需求，特別是在高壓緊迫的犯罪現場調查中，即時性是至關重要的。其次，這些工具在助記詞的適用性上存在限制。助記詞不僅格式多樣，而且可能使用不同語言或包含特殊的組合模式，現有工具對於這些多樣性的支持不足，這種缺陷尤其會表現在識別隱藏或非標準格式的助記詞方面。本研究旨在深入探討犯罪現場對電子設備中助記詞的取證和鑑識方法，填補現有研究及工具的空白。考慮到目前市場上缺乏針對助記詞鑑識的完善工具，本研究的成果將對加密貨幣相關犯罪的有效打擊具有重要意義。

本研究為提高助記詞鑑識的效率和準確性，特別採用了自然語言處理技術（NLP）[5]來支援包括英文和中文在內的多達 11 種語言的助記詞識別。我們選用了長短期記憶網路(LSTM)、雙向長短期記憶網路 (BiLSTM)、 循環神經網路 (RNN)、文本卷積神經網路(TextCNN)等多種深度學習[6]模型，這些模型具有在捕捉助記詞的隱含特徵上的強大能力。考慮到助記詞的特性，這些神經網路模型能有效學習和識別這些特徵。經過大量數據的訓練，這些模型能夠顯著提高識別的準確率，達到約 99.99%的測試準確率。相比傳統的單字表比對方法，在識別速度上我們的方法更具優勢，能在有限時間內有效識別助記詞，提升速度達到 80.27 倍。本研究通過使用先進的神經網路模型，為在儲存設備中尋找隱藏助記詞提供了一種更有效的解決方案，從而適應多種取證環境和需求。

2. 背景知識

在本節中，我們介紹了儲存加密貨幣的虛擬錢包系統，包括目前市值最高的比特幣和虛擬錢包的助記詞密碼，接著我們展現了助記詞的蒐證方法和其使用的深度學習模型。

2.1. 比特幣

比特幣是加密貨幣的一種，其為中本聰 (Satoshi Nakamoto) 於 2008 年推出的加密貨幣和支付系統[7]。比特幣系統中的交易儲存在公共交易分類帳中，公共交易分類帳也就是區塊鏈，該系統是一個點對點的去中心化網路，付款直接在各方之間發送，而不是依賴中央機構來付款。因此比特幣有去中心化的貨幣發行和交易結算的特性。區塊鏈的安全性取決於比特幣挖礦的 SHA-256 演算法，這是一種計算密集型演算法(Computationally Intensive Algorithm)，該演算法可以防止比特幣的雙重支出和篡改已確認的交易。

2.2. 虛擬貨幣錢包的助記詞

由「比特幣區塊鏈第 39 號改進提案（BIP39）」(Haigh 等，2019)（https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki）所提出，其目的是為了幫助用戶記憶、記錄複雜的私鑰，避免抄寫錯誤。創建錢包的時候，通常系統為了安全和方便記憶、抄寫，並不會直接把冗長的私鑰顯示出來，而是轉換成一串由 12 到 24 個單字組成的序列，稱為助記詞（mnemonic phrase）。在區塊鏈中，助記詞扮演著關鍵角色，它是確定性錢包的生成種子和恢復機制，將複雜的隨機數字編碼轉化成人類可記憶的格式。這組單字序列更是用戶錢包資產的一個加密備份，能生成或恢復錢包及其所有金鑰，並成為存取錢包的密碼。當用戶遺失冷錢包或刪除了數字錢包應用程式時，可以透過助記詞恢復原本錢包，並回復其中的加密貨幣。

BIP39 提案提出助記詞的生成方式，首先生成一個隨機數（稱為熵），其長度通常為 128 位元，其次計算校驗碼（Checksum），對熵進行 SHA-256 雜湊，取雜湊值的前幾位作為校驗和，並添加校驗和，將校驗和添加到熵的末尾，然後劃分為 11 位的區塊，將結合了熵和校驗和的位序列劃分為 11 位長的多個區塊，接著轉換為單詞，使用一個預定義的 2048 個單詞的詞典（通常稱為“詞表”或“助記詞詞典”），將每個 11 位的區塊映射到一個特定的單詞，最後生成助記詞序列，最終的助記詞序列就是由這些單詞按順序排列而成的，如圖 1 所示。

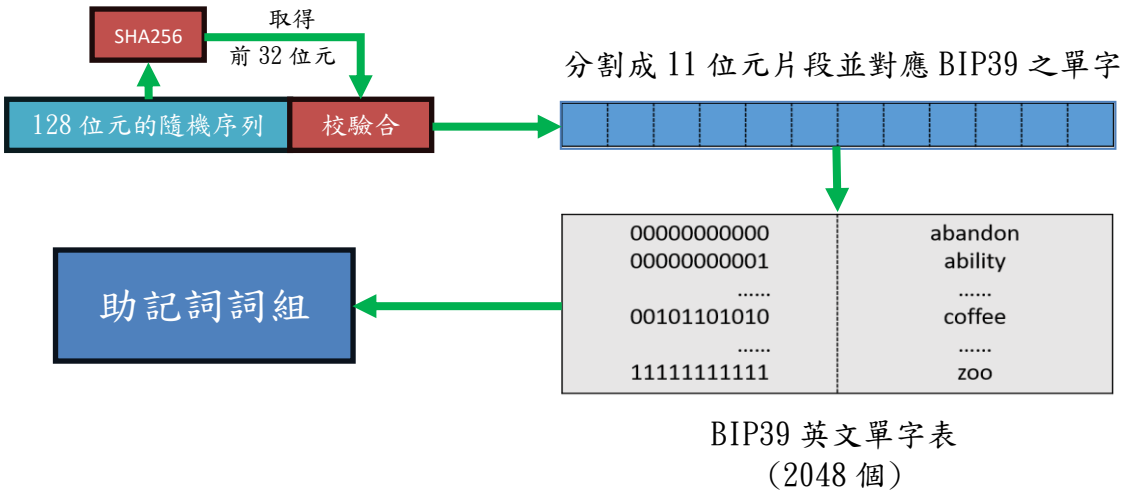

圖 1 助記詞生成方式圖

BIP39 提案中可以使用英文、日文、韓文、西班牙文、葡萄牙文、法文、義大利文、繁體中文、簡體中文、捷克文作為助記詞，而全球市占前 3 大的冷錢包產品 CoolWallet 使用了最新型態的含多國語言及數字作為助記詞，總共 11 種格式，皆為本實驗深度學習模型的學習範圍。

2.3. 助記詞蒐證方法

文獻中大多數關於加密貨幣的取證分析都集中在比特幣上，[8] [9] [10] [11] [12] [4] [13] ，而目前有助記詞的蒐證方法，僅有上述提到與私鑰蒐證合併為多合一方法的 WinBAS 程式，以及研究針對 Bitcoin Core 和 Electrum 中行程記憶體的數位證據[14]，其中包含了助記詞，但關於針對助記詞的鑑識方法的研究卻很少，因此為了填補助記詞蒐證在即時取證（Real-time Forensics）方面的缺乏，本研究採用一種新的高級神經網路來解決，研究了 NLP 自然語言學習模型辨識助記詞，分別實驗 LSTM、BiLSTM、RNN、TextCNN 4 種 NLP 模型，測試本研究的方法在各種深度學習模型中的泛用性，並最終作出綜合性的分析與比較。

2.4. 單字表比對方法

單字表比對方法以下稱為字庫法[15]，字庫法採用了 BIP39 規範[16]中的助記詞名單，並且比較測試檔案和助記詞名單中的字詞，以計算它們之間的相似度。字庫法首先會讀取一個文字檔並將其中的每一行 12 個詞彙儲存為列表。接著接受列表和助記詞名單作為輸入，計算它們的交集，並返回交集與第一個列表的比例，並將可能為助記詞之閾值設為 90%，最後，輸出相似度高於閾值之結果。

2.5. 文本卷積神經網路(TextCNN)

TextCNN 是一個 CNN (Convolutional Neural Networks)架構，利用卷積神經網路對文字進行分類的演算法。TextCNN 利用局部特徵進行操作，通過卷積操作提取句子的特徵。其基本運

作步驟的第一步是對本文進行斷詞，使用詞嵌入（Word Embedding）來為斷詞建立詞向量[17]。這樣做主要是為了將自然語言數值化，方便後續的處理。而第二步是對第一步得到的詞向量進行卷積。卷積是一種神經網路運算單元，主要用於對文本進行特徵提取。經過卷積操作後，會得到一個新的矩陣，卷積操作的特徵 c_i 由 h 長度的視窗為$x_{i:i+h-1}$所產生，如方程式 1 所示。

$$c_i = f(\omega \cdot x_{i:i+h-1} + b) \tag{1}$$

過濾器應用到句子上，如在$x_{1:h}$上卷積操作得到c_1、在$x_{2:h+1}$得到c_2，最終獲得特徵圖 c，如方程式 2 所示：

$$c_i = f(\omega \cdot x_{i:i+h-1} + b) \tag{2}$$

接著對上一步的結果進行最大池化（max-pooling），這個操作是從多個數值中選取一個最大值，這樣能保留主要特徵，然後使用 Softmax [18]：將最大池化的結果拼接起來，送入 Softmax 層中，得到各個類別的機率。TextCNN 詳細過程原理圖如圖 2 所示[19]：

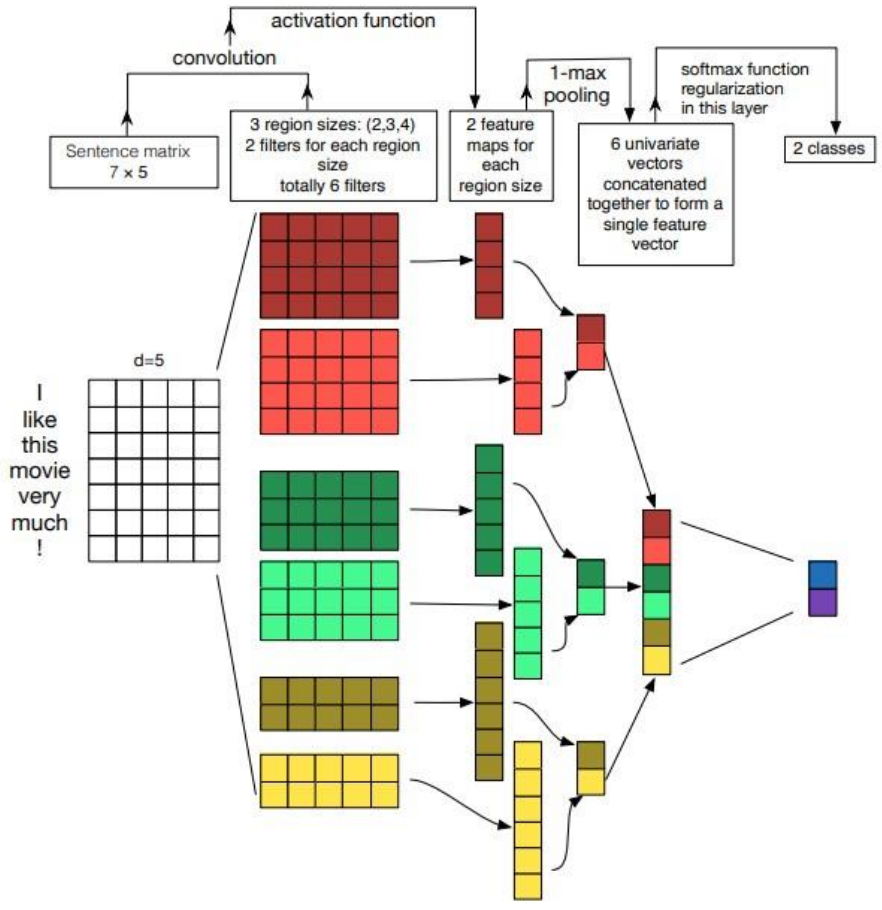

圖 2、TextCNN 模型原理圖

2.6. 循環神經網路 (RNN)

循環神經網路(Recurrent Neural Network, RNN)，是一種深度學習模型，RNN 的設計靈感來

自於人類的思考和資訊處理方式，特別適用於具有時間依賴性或序列性質的任務。RNN 的關鍵特點在於其循環結構，允許資訊從前一個時間步傳遞到下一個時間步。每個時間步，RNN 接受一個輸入（通常是當前時間步的數據點）和一個隱藏狀態（包含了之前時間步的資訊），然後生成一個輸出和一個新的隱藏狀態。這一循環的過程使得 RNN 能夠捕捉數據中的順序、上下文和依賴關係。[20], [21] RNN 的概念如圖 3 所示。

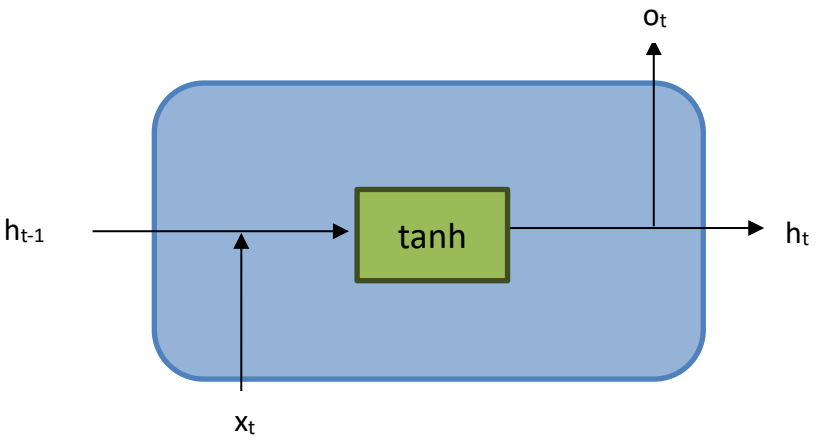

圖 3 RNN 概念圖

2.7. LSTM(長短期記憶網路)

LSTM[22], [23]於 1997 年由 Sepp Hochreiter、Jürgen Schmidhuber 引入，LSTM 處理資料在向前運作時傳遞訊息。LSTM 可以使其忘記或保留資訊。LSTM 的架構為一個記憶單元和各種閘，其使用單元狀態、輸入閘、輸出閘、遺忘閘門來儲存長期依賴性，以克服原始的 RNN 模型中梯度消失問題（Vanishing Gradient Problem）。

這種水平的單元狀態在整個 LSTM 結構中延伸，並通過不同的閘門機制來調節資訊的增加或刪除，從而實現資訊流的有效管理。訊息能藉由運作過程攜帶檔案訊息，一開始時間的特徵訊息可以攜帶到最後的時間，使結果能受長期記憶影響。資訊會隨著流程改變，從單元狀態新增到閘或是從閘刪除。其中包含 Sigmoid 函數，如方程式 3，Sigmoid 函數會產生 0 到 1 中的數字，並決定每個組件應通過多少。遺忘閘負責學習那些資訊要保留還是遺忘，如方程式 4。

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{4}$$

接著，輸入閘控制哪些新的資訊應該被存儲，如方程式 5，最後輸出閘決定了當前的記憶單元的哪些資訊應該被輸出，如方程式 6。

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{5}$$

輸入閘（Input gate）：用於決定哪些新的資訊應該被存儲在記憶單元中。

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

輸出閘（Output gate）：輸出閘決定了當前的記憶單元的哪些資訊應該被輸出。

LSTM 的概念如圖 4 所示。

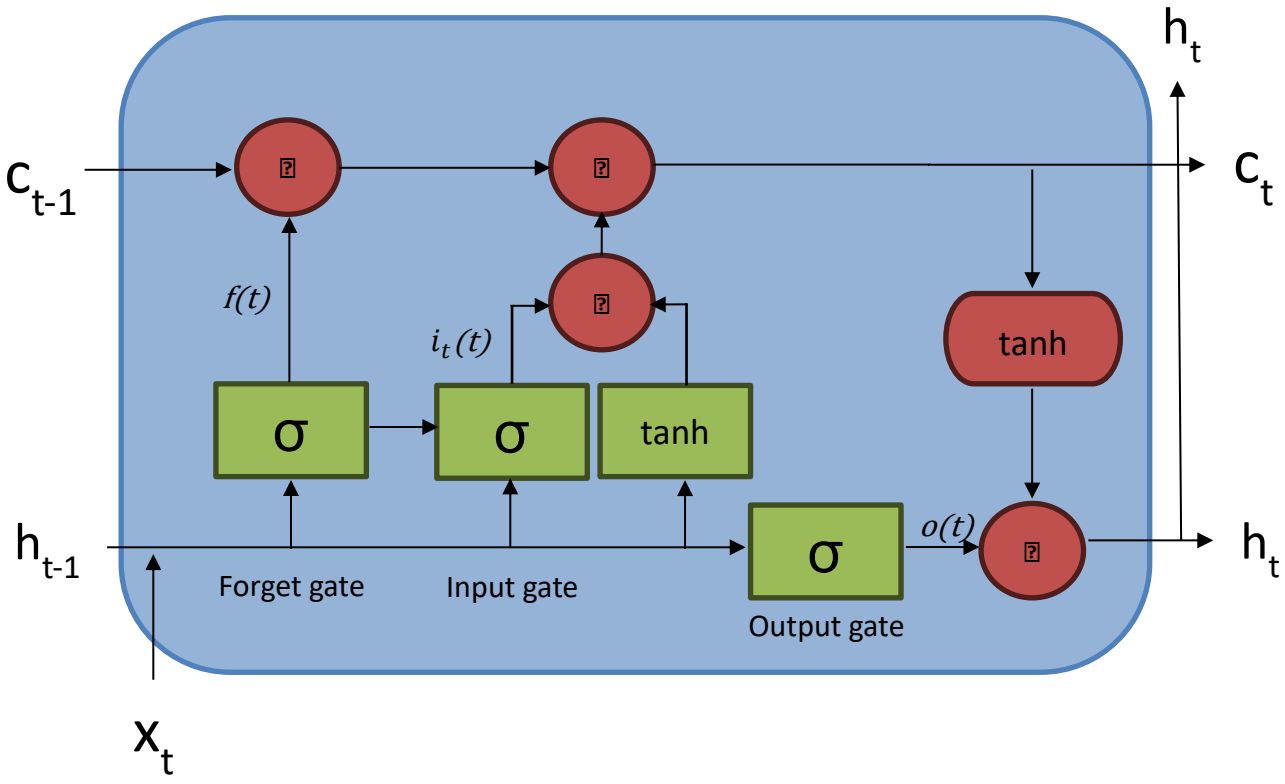

圖 4　LSTM 的概念示意圖

2.8. BiLSTM(雙向長短期記憶網路)

Bidirectional Long Short-Term Memory[24]，簡稱 BiLSTM，是一種深度學習模型，通常用於處理序列數據，特別是自然語言處理（NLP）和時間序列領域。BiLSTM 是對 LSTM（長短時記憶網路）的擴展，它的一個關鍵特點是能夠同時考慮前向和後向的上下文資訊。

BiLSTM 首先會接收一個序列作為輸入，這可以是自然語言句子或時間序列數據。接下來，模型將輸入序列分別送入前向 LSTM 和後向 LSTM 網路中進行計算。前向 LSTM 從序列的開頭開始處理，而後向 LSTM 則從序列的末尾開始處理。每個 LSTM 單元在處理輸入時會維護一個內部狀態，並生成一個輸出，其中包含了輸入序列的特徵資訊，隨後 BiLSTM 將前向和後向 LSTM 的輸出結果進行合併，以獲得完整的上下文資訊。這種結構使得 BiLSTM 在處理序列數據時能夠更全面地考慮上下文資訊，從而提高了模型的性能。BiLSTM 的概念如圖 5 所示。

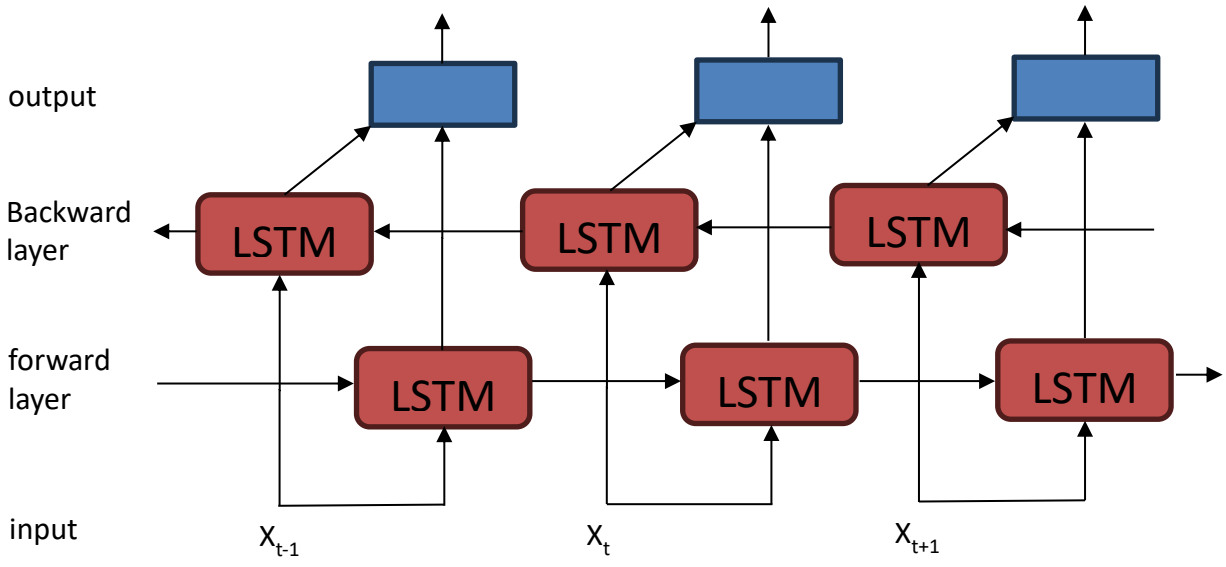

圖 5　BiLSTM 的概念示意圖

3. 實驗

本研究的實驗部分旨在驗證深度學習模型在助記詞辨識上的應用效果，並透過加入各種犯罪電腦蒐證中可能存在之文字檔案進行實驗，證實本研究的網路模型確實能夠從各種可能藏有助記詞之文字檔案中，從許多非助記詞數據中辨識出隱藏其中之助記詞。最後再比較各種深度學習模型應用於助記詞辨識的準確率，找出在效能和準確率最適合用來辨識助記詞的深度學習模型。

3.1. 規格設備

本實驗的實驗裝置使用 Python 3.9.1 及 Keras 2.6，配備 Intel Core i7-12700H 中央處理器、速度為 4.70 GHz、16GB RAM、顯示卡 GeForce RTX 3050 LAPTOP、作業系統 Windows11。

3.2. 實驗設計

在實驗設計上，我們安排了大量助記詞的數據集進行訓練，目的是讓深度學習模型能學習到每種助記詞的所有特徵，並安排了大量非助記詞和極端值的數據集進行訓練。目的是讓模型能學習到各種非助記詞和極端值數據存在的特徵。最後的測試階段則使用全新的助記詞和非助記詞數據，以評估網路的實際效能(包括正確率、型 I 及型 II 錯誤)。為了確保模型的可靠性，每一階段的數據都只在該階段中使用，並且在測試階段採用全新的數據，這些助記詞及非助記詞在網路的訓練及驗證階段都未曾出現，以最大程度反映真實現場助記詞蒐證的應用場景，接著訓練完的權重匯出到助記詞辨識程式工具，助記詞辨識程式工具會分析輸入之檔案，並輸出可能的助記詞和其機率。實驗設計的模型如圖 6 所示。

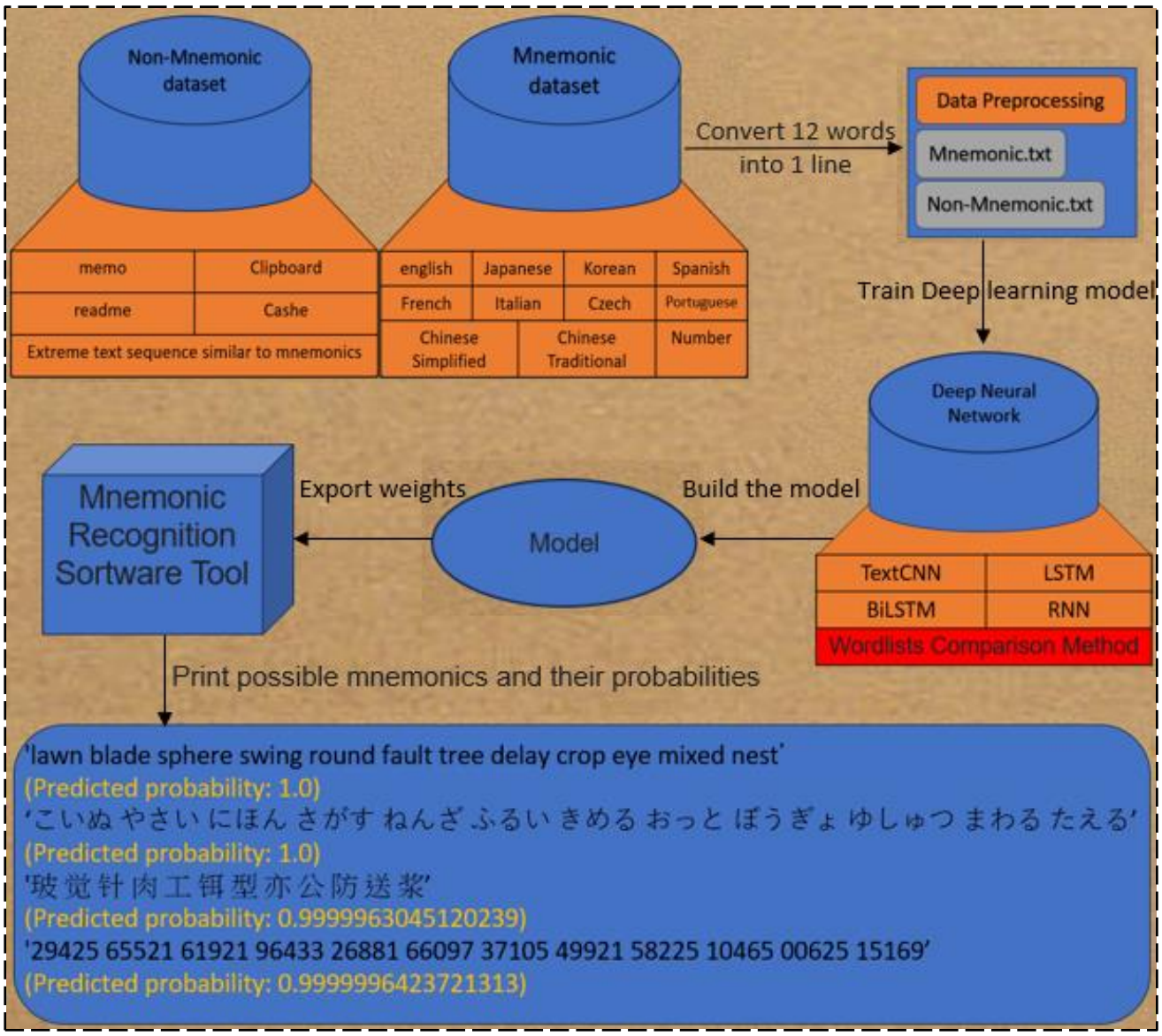

圖 6 所提方法的圖形簡介

3.3. 數據集

在進行深度學習模型訓練的過程中，我們針對助記詞和非助記詞兩大類別的數據集進行了廣泛且精細的挑選與整理。對於助記詞數據集，我們蒐集了遵循 BIP39 規範的 10 種語言的助記詞，包含了英文、日文等 10 國語言，以及加入市占率前三的冷錢包 Coolwallet 採用之新型數字助記詞。每種語言的助記詞均達到 10000 個，確保了數據的豐富性與多樣性。此外，為了確保數據的代表性和可靠性，我們確保了數據集中每個單詞至少出現 50 次，從而達到均衡和全面的數據分布。如表 1 所示。

表 1 助記詞數據集範例

	助記詞數據範例
英文	**napkin mandate tide sphere pass reduce similar misery tide limit reduce claw**
日文	**こくこ よっか ぬすむ このみ けとはす のりもの りくつ くらす すこい とあい ちしき きかい**
西班牙文	**amigo eterno fila captar donar enano bahía siesta público maceta alambre laguna**
簡體中文	**使未友八言输基辞寻份得弱**
繁體中文	**挖郭豐舒奴沉視八優阻柱丟**
法文	**correct intrigue froid frémir ardeur sénateur louve arsenic offrir abolir déjeuner loterie**
義大利文	**regresso pilota insano torrone parcella intonaco epocale cespo fortezza governo pettine uncinetto**
韓文	**운반 농민 우선 선장 전용 하순 정상 속옷 표현 마음 채널 저축**
捷克文	**pohnutka export podoba meloun sejmout vynikat skica mozol vstup honitba tenista samec**
葡萄牙文	**mesada brilho membro fiapo perplexo tomada plumagem fuligem tensor chefe revelar pegada**
數字	**92833 60721 74689 87073 71425 40033 25633 78337 78433 01297 77185 85681**

對於非助記詞數據集，我們特別強調了其在多樣性和實際應用兼容性上的重要性。這一點體現在數據源的廣泛性和數據類型的多樣性上。本實驗所採用的非助記詞數據集，來自於犯罪現場電腦可能存在的多種檔案類型，如備忘錄（memo）、剪貼簿（clipboard）、軟體使用說明（readme）、瀏覽器暫存檔（Cache）等，同時融合了大量的新聞文章和書籍句子。這些數據來源豐富多樣，不僅涵蓋了日常生活和專業場景中常見的文本類型，還包含了與真實世界文本使用習慣相關的內容，從而增強了數據集在實際應用中的代表性和有效性。

此外，非助記詞數據集還注重於包含一些極端值，即那些看似普通但包含大量與助記詞相似單詞的句子。這種做法旨在提升模型在處理邊緣案例時的鑑別能力。例如，句子“Meat Pie Recipe Homemade Australian Meat Pie Recipe Page Agendas can change”中，除了少數幾個單詞外，其餘都是助記詞。這樣的句子設計有助於模型更好地學習和理解在複雜和多變的真實世界情況下的文本特徵，從而在實際應用中發揮更好的性能，如表 2 所示。

表 2 非助記詞數據集範例

	非助記詞數據
memo	**this is a reminder that due the shortened holiday week employee**
clipboard	**no one gives a darn about losing a limb when you are**
readme	**Readme file is a markdown document that serves as the introductory**
article	**Party after party noise after noise All the sounds of misery At**
極端值	**Meat Pie Recipe Homemade Australian Meat Pie Recipe Page Agendas can change**

在驗證過程中，我們展現了極高的嚴謹性。透過仔細分析與對比助記詞和非助記詞數據，我們確保了模型在學習過程中能夠有效辨識和區分這兩類數據。此外，我們對數據進行了多層次的驗證，以確保模型的學習成果具有高度的準確性和適用性。通過這些細微的準備工作，我們的深度學習模型能夠有效地學習助記詞的規律和特徵，同時也能夠準確識別和處理非助記詞的數據，從而在實際應用中發揮其最大的效能。

3.4. 網路架構

3.4.1 LSTM/BiLSTM/RNN 網路架構

長短期記憶網路(LSTM)、雙向長短期記憶網路 (BiLSTM) 及 循環神經網路 (RNN) 均屬於循環神經網路的系列網路架構。這些網路架構尤其擅長處理時間序列數據，特別適合於文字序列識別任務。

在這一過程中，我們細致地探討了各種參數如何影響模型的表現，並展示了這些參數對於增強模型在特定任務上的適用性的關鍵作用。我們嘗試不同的參數設定來設計 LSTM，藉由改變各個參數，如最大詞彙量（MAX_NB_WORDS）、序列最大長度（MAX_SEQUENCE_LENGTH）、嵌入維度（EMBEDDING_DIM）、LSTM 單元數（LSTM_UNITS）、全連階層單元數（DENSE_UNITS，DU）、Dropout 參數（DROPOUT_RATE）、Epochs、批次大小（Batch Size），來精確地調整模型以達到最優性能。

在我們的研究中，最大詞彙量（MAX_NB_WORDS）設定為 95460，這正好對應於訓練數據集中的詞彙數量。這種設定避免了在模型中使用空白字彙填充，從而有助於提升模型的準確性。同時，我們將序列最大長度（MAX_SEQUENCE_LENGTH）設為 12，以最佳匹配助記詞的固有格式，因為標準的助記詞是由 12 個詞彙組成的。這些經過精心調整的設定使 LSTM

模型在訓練資料上表現優異，並在對未知測試數據的預測中達到了 99%的高精準度。

經過一系列的實驗和分析，我們確定了最優化的模型配置：嵌入維度設為 50，LSTM 單元數和全連階層單元數均設為 64，Dropout 層的參數設為 0.5。這樣的配置使得 LSTM 模型在驗證損失（val_loss）上表現最佳，有效減少了過度擬合問題，同時降低了模型的複雜性，從而提升了整體效能。此外，將批次大小設為 32 不僅能提供更精確的梯度估計，也有助於進一步降低過擬合的風險。為了公平比較不同深度學習模型的準確率，我們對所有模型采用了相同的參數設置。在構建模型時，我們根據助記詞的特性精心選擇了單元數、激活函數和初始權重，具體配置如表 3 所示。

表 3 LSTM 網路 使用參數

Parameters	Values
MAX_NB_WORDS	**95460**
MAX_SEQUENCE_LENGTH	**12**
EMBEDDING_DIM	**50**
LSTM_UNITS	**64**
DENSE_UNITS	**64**
DROPOUT_RATE	**0.5**
Epochs	**100**
Batch Size	**32**
Optimizer	**Adam**
Fully Connected Layer activation	**ReLU**
Output Layer	**Sigmoid**

下圖 3 的模型訓練的流程圖，描述了從開始到結束的整個訓練過程。從資料預處理（Data Preprocessing）開始，其包括數據清洗、特徵工程等，訓練-測試分割（Train-Test Split）將數據集分為訓練集和測試集，模型初始化（Model Initialization）選擇或構建一個初始模型，訓練迴圈（Training Loop）進行多個 epoch 的訓練，提前停止（Early Stopping）判斷是否達到提前停止的條件，模型評估（Model Evaluation）在測試集上評估模型性能，儲存模型（Save Model）將訓練好的模型儲存下來。

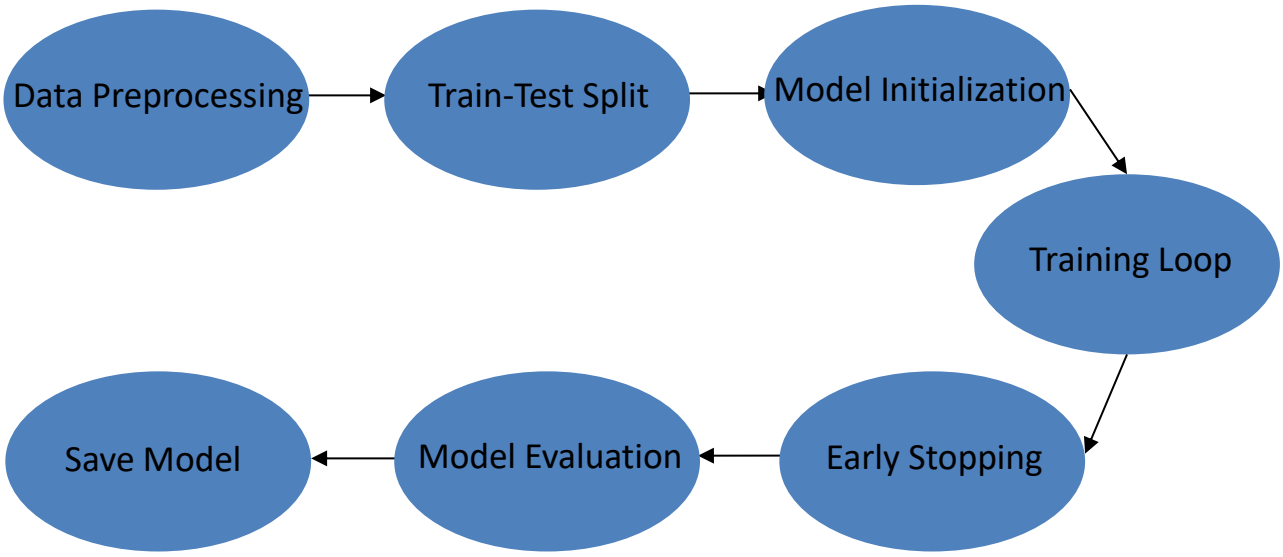

圖 7 網路訓練流程圖

3.4.2 TextCNN 網路架構

不同於前述的 LSTM/ BiLSTM/ RNN 網路架構等 RNN 架構模型，文本卷積神經網路(TextCNN)的原理基於卷積神經網絡（CNN），主要用於捕捉文本中的局部特徵。這種架構通過應用多個不同大小的卷積核（Convolutional Kernel），有效地識別出文本中的關鍵詞彙和語義模式，從而提取助記詞的關鍵資訊。其首先將文本轉換為向量形式，通過詞嵌入（Word Embedding）實現。詞嵌入過程中，每個單詞都被轉換為一個密集的向量，這些向量能夠捕捉詞彙之間的語義和語法關係。TextCNN 透過卷積操作（Convolution）捕捉這些向量中的模式，然後使用池化（Pooling）來降低特徵向量的維度並保留重要的特徵。最後生成的特徵向量經過全連接層（Fully Connected Layers）進行文本分類[25]，並通過 Sigmoid 函數將輸出轉換為 0 到 1 之間的概率值。TextCNN 的模型架構圖如圖 8 所示。經實驗證明，TextCNN 網路架構同樣能有效地處理助記詞的識別和分類工作，且 TextCNN 架構在處理具有高度變異性的助記詞時表現出了極高的準確性和效率，有效地支援了本研究的目標。

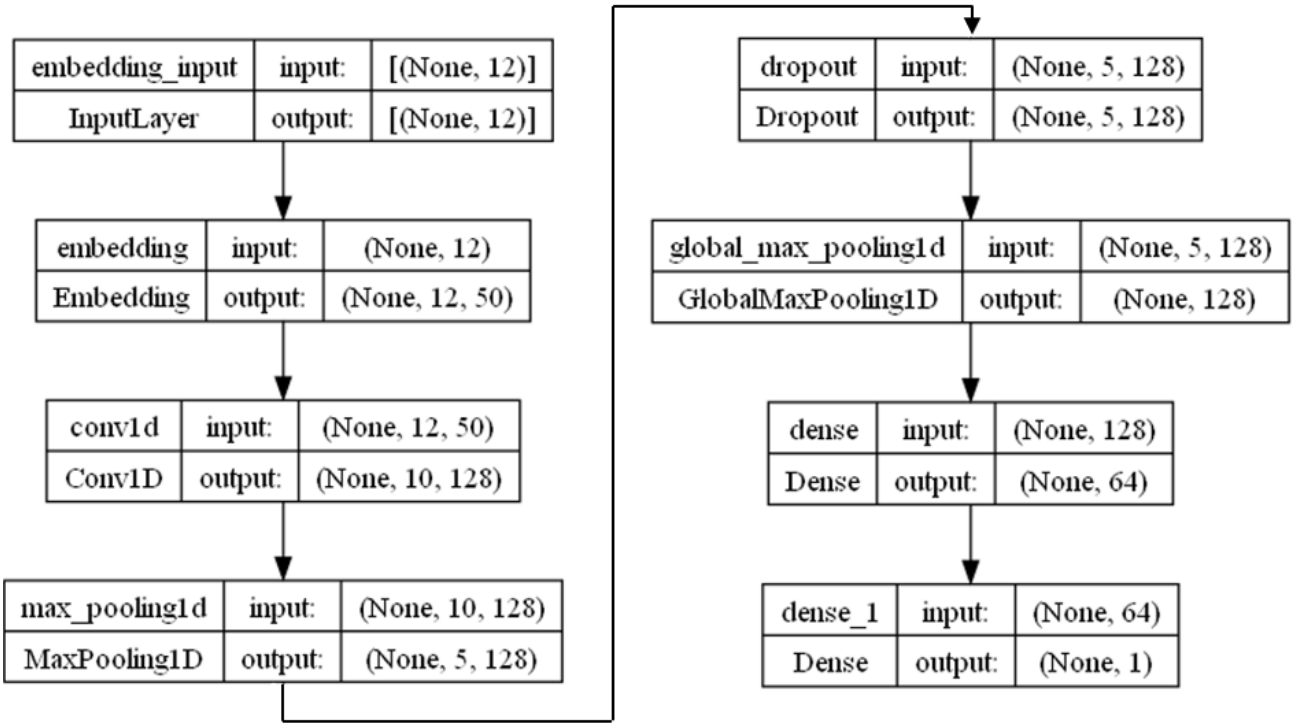

圖 8 TextCNN 模型架構圖

3.5. 助記詞辨識程式工具

助記詞辨識程式工具匯入 TextCNN 訓練完匯出的權重，其會分析輸入之可能藏有助記詞之文字檔案，並顯示其判斷可能為助記詞的文字序列和其是助記詞的概率，本實驗將可能為助記詞之閾值設為 0.99，當助記詞超過 0.99 時，會將助記詞記錄並輸出。

4. 實驗結果與分析

本研究展示了使用深度學習模型特別是 TextCNN 在助記詞辨識領域的有效性和高效性。實驗結果顯示，TextCNN 在測試樣本中的助記詞辨識方面取得了高達 99%準確率，超越了其他模型如 LSTM、BILSTM 和 RNN。這一成果得益於 TextCNN 優秀的數據適應性和處理速度，特別是在面對含有助記詞的大型文檔時，TextCNN 不僅保持了高準確率，還顯著提高了處理速度，達到了字庫法的 80 倍以上。

4.1. 機器學習模型比較

為了驗證 NLP 的各種模型對於助記詞辨識之優越性，本實驗採用了 LSTM、BILSTM、RNN、TextCNN 共 4 種模型測試準確率，並且針對各模型之準確率、型 I 錯誤率、型 II 錯誤率進行比較，目標是經過綜合比較找出最優越之深度學習模型。其中，型 I 錯誤率、型 II 錯誤率、準確率（Accuracy）是所有正確預測（真陽性和真陰性）佔所有預測（正確的和不正確的）的比例。計算公式如下：

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

型 I 錯誤率是模型錯誤預測為正類的負類實例（偽陽性）與所有實際負類實例的比率。計算公式如下：

$$\text{False Positive Rate (FPR)} = \frac{FP}{TN + FP} \tag{8}$$

型 II 錯誤率 (False Negative Rate, FNR)：型 II 錯誤率是模型錯誤預測為負類的正類實例（偽陰性）與所有實際正類實例的比率。計算公式如下：

$$\text{False Negative Rate (FNR)} = \frac{FN}{TP + FN} \tag{9}$$

實驗結果表明，雖然 LSTM、BiLSTM、RNN 和 TextCNN 這四種 NLP 模型均展現出了卓越的效能，並基本上都能勝任助記詞識別工作，但特別值得注意的是，TextCNN 在訓練、驗證和測試數據集上展現出了一致且卓越的性能。在訓練準確率方面，TextCNN 達到了 99.21%，而在驗證階段更是達到了高達 99.76%的準確率，顯示出其對於數據的強大適應能力。在測試數據集上，TextCNN 的準確率更是驚人地達到了 99.9993%，這一結果不僅展現了其卓越的預測能力，同時也證明了 TextCNN 在處理不同類型數據集時的穩定性和可靠性。這種穩定且一致的表現證明了 TextCNN 在處理不同數據集時沒有出現明顯的過度擬合現象。相比於其他模型如 LSTM、BiLSTM 和 RNN，TextCNN 在關鍵性能指標上顯示出更為穩定和可靠的表現，從而證明了其在助記詞識別任務上的優越性。

表 4 深度學習模型比較圖

	LSTM	BILSTM	RNN	TextCNN
訓練準確率	99.44%	98.88%	99.22%	99.21%
驗證準確率	95.35%	92.05%	93.67%	99.76%
測試準確率	99.998%	99.99%	99.995%	99.9993%
測試集 TP(真陽性)	60	59	60	60
測試集 TN(真陰性)	147528	147524	147524	147530
測試集 FP(偽陽性)	3	7	7	1
測試集 FN(偽陰性)	0	1	0	0
測試集型 I 錯誤率	0.002%	0.0047%	0.0047%	0.00068%
測試集型 II 錯誤率	0%	1.67%	0%	0%

除此之外，我們還可以透過混淆矩陣中[26]來供了一個直觀的混淆矩陣比較，圖 9 展示了 LSTM、BiLSTM、RNN 以及 TextCNN 共 4 種模型在助記詞辨識任務上的表現。從中我們可以看出，4 種模型在助記詞辨識的任務上均展現了強大的識別能力。然而，TextCNN 在各項指標上的一致性和穩定性使其脫穎而出，並且在偽陽性（FP）和偽陰性（FN）的數量上都保持最低，特別是在偽陽性（FP）的數量上，TextCNN 只有 1 例，低於其他模型，這意味著它在判斷非助記詞資料時的誤判率極低。同樣，偽陰性（FN）的數量為 0，表明所有的真實助記詞都被正確識別，沒有被遺漏。這點對於取證工作尤為關鍵，因為在追蹤和取回非法資金流動過程中，遺漏任何一個助記詞都可能導致重大的資訊損失。因此，我們的數據進一步證實，在所有評估過的深度學習模型中，TextCNN 在確保鑑識過程中是更為可靠的選擇。

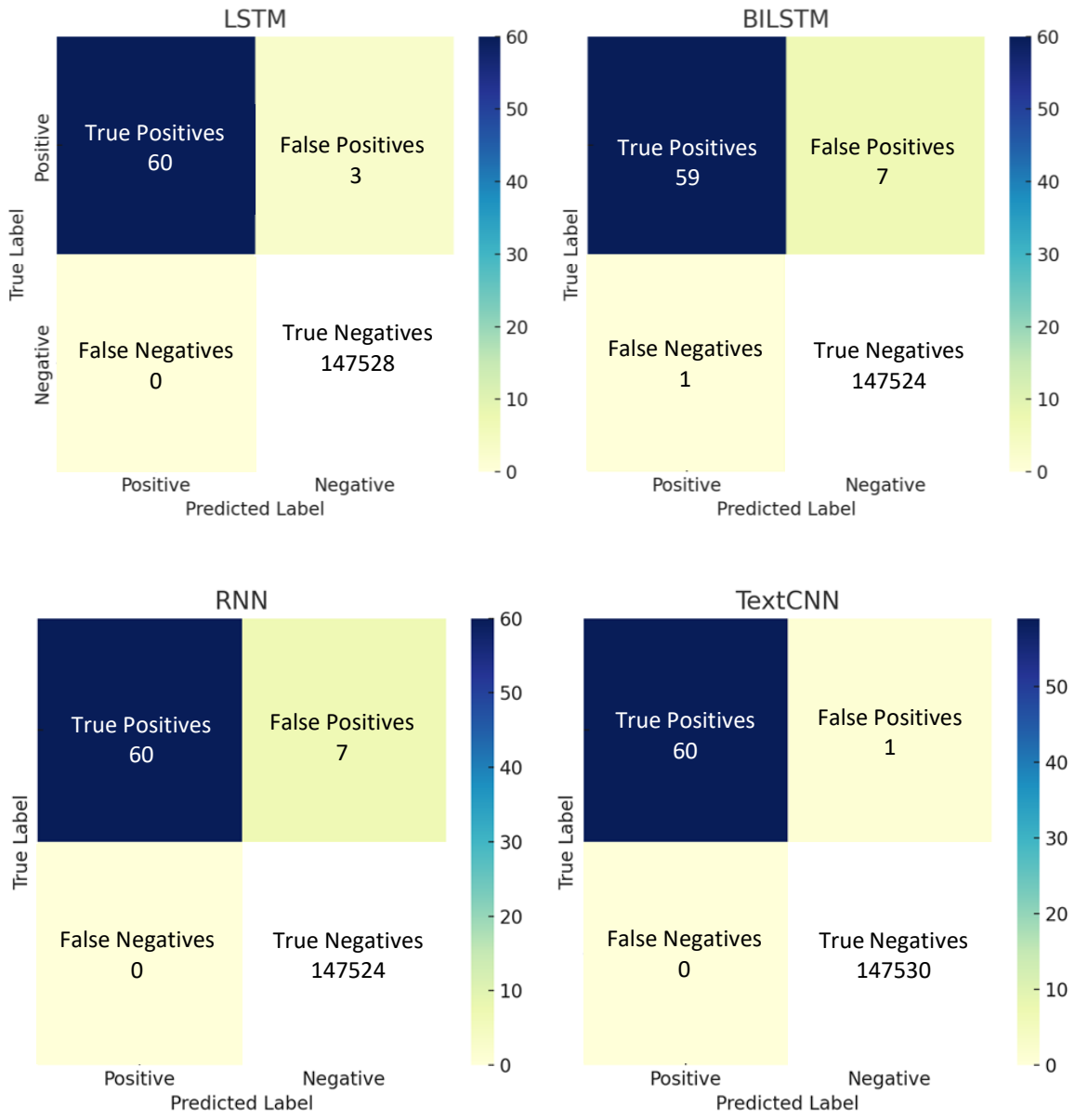

圖 9 各模型之混淆矩陣

註：在圖 9 的混淆矩陣中，真陰性（True Negatives, TN）的數量相比其他指標極為懸殊。因此，在視覺上不將 TN 的數值包含在色階對比中。

4.2. 實驗結果

總體來說，本研究經過比較之後，TextCNN 之測試結果最符合時記憶用場景的需求。因為英文是最常用之助記詞語言，因此英文的助記詞數據和非助記詞數據是最多的，而大量的非助記詞數據混雜屬於助記詞之單字，因此使英文助記詞之準確率無法達到 100%，但仍有超過 99.99%之表現，而其餘各國語言除了中文和數字，皆有達到機率為 100%的程度，此處之助記詞準確率是由確實是助記詞之判斷為助記詞之機率進行平均得出，而中文和數字由於文字之特殊性，在判別上機率略低，但仍有超過 99%之表現。助記詞辨識方法在藏有助記詞的文字檔案中的準確率為 99%以上，TextCNN 之實驗結果如表 5 所示。

總體來看，TextCNN 在處理不同語言的助記詞時表現皆極微出色。但由於英文是實際應用場景中最常使用的助記詞語言，因此在我們的數據集中，我們特別安排了大量的英文助記詞數據，以確保研究結果能夠更好地反映真實世界的應用情況。但這也意味著，在我們的數據集中，英文助記詞數據的豐富性帶來了特定的挑戰。為了更全面地測試模型的識別能力，我們甚至安排了一些極端案例（非常接近助記詞的測試資料，如本文表 2 中的非助記詞數據集範例），這些案例極易與真正的助記詞混淆。因此，雖然我們的模型在處理大量的英文非助記詞數據時面臨著將某些單詞誤判為助記詞的可能，但英文助記詞的準確率仍然超過了 99.99%。相比之下，其他語言，包括日文、西班牙文、簡體中文、繁體中文、法文、義大利文、韓文、捷克文、葡萄牙文以及數字，在我們的測試中助記詞的準確率均達到了 100%，如表 5 所展示。

表 5 TextCNN 實驗結果

	英文	日文	西班牙文	簡體中文	繁體中文	法文	義大利文	韓文	捷克文	葡萄牙文	數字
助記詞準確率	99.99	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%
測試數據集個數	42291	10000									
型 I 錯誤個數	1	0									
型 II 錯誤個數	0										

4.3. TextCNN 與字庫法的比較

在助記詞辨識方面，TextCNN 方法用於處理 8.6MB 的含助記詞文字檔案，耗時 7.36 秒，相比之下，字庫比對法則需 590.78 秒。這表示在處理相同檔案時，TextCNN 的速度是字庫法的 80.27 倍。需要注意的是，這一數值不是固定不變的，且在處理更大檔案時，速度差異可能會進一步擴大。關於準確率，兩者之間的差異不到 0.01%。表 6 展示了 TextCNN 與字庫法在優劣方面的具體比較。鑑於 TextCNN 在速度和準確率上的優勢，我們建議先使用 TextCNN 處理大量且龐大的檔案，以進行初步篩選。其後，可運用字庫法對 TextCNN 的結果進行進一步的細節檢查。

表 6. TextCNN 與字庫法比較圖

方法	優勢	劣勢
TextCNN	辨識速度相對字庫法效率提升 80.27 倍，且具多模態數位證據分析潛力	準確率高達 99.99%，但在大量數據下仍可能有型 I、型 II 錯誤的產生
字庫法	準確率 100%	辨識速度慢，日後發展潛力有限。

在辨識虛擬貨幣錢包助記詞的過程中，傳統的字庫法在辨識過程中存在一定的局限性。其主要劣勢包括辨識速度較慢，這可能影響到處理大量數據時的效率。此外，考慮到技術進步和未來發展的需求，字庫法的發展潛力相對有限。這主要是因為它依賴於預定義的字庫，使其在適應新的語言模式或多模態（Multimodal）擴展應用範圍方面可能不如其他更靈活的方法。

而本研究使用的 TextCNN 顯示出與傳統字庫法相比的顯著優勢，尤其在彈性和未來可發展性方面。首先，TextCNN 在處理速度上具有明顯優勢，這使其能在犯罪現場迅速辨識出助記詞，進而查扣加密貨幣資產，這是傳統字庫法所難以匹敵的。其次，TextCNN 展現了卓越的數據適應性，能夠有效處理不同類型和規模的數據。此外，TextCNN 還能隨著時間的推移和數據的更新而持續進行自我優化。更重要的是，TextCNN 日後還具有應用於多模態數據（如文字、語音和影像）的潛力，使其有望成為一種全面的多模態數位證據分析工具，這是傳統方法永遠也不可能企及的。總結來看，TextCNN 不僅在短期內展現出高效的時間管理能力，也因其高度的彈性和可擴展性，展現出長遠的發展潛力，成為法律執行和數位鑑識領域中一個極具價值的工具。

4.4 TextCNN 與字庫法結合使用的策略

考慮到 TextCNN 在速度上顯著超越字庫法，提升了 80.27 倍，同時字庫法在準確率方面能

夠達到 100%的表現，我們提出一種創新的結合兩者優勢的助記詞辨識策略。這一策略的核心在於利用 TextCNN 在處理大規模數據集時的高速度和高準確率快速篩選檔案，以迅速縮小潛在的助記詞候選範圍。隨後，將 TextCNN 篩選出的結果提交給字庫法進行細節審查和確認，這一步驟利用了字庫法在精確度上的絕對優勢。通過這種雙重驗證機制，不僅大幅提升了搜尋的效率，也保證了最終結果的準確無誤，從而能夠迅速且確切地定位犯罪分子隱藏的助記詞。

5. 結論與未來展望

本研究證明了在分析加密錢包助記詞方面，TextCNN 這類深度學習模型在數位證據的犯罪現場採證工作中的可行性和有效性。雖然傳統字庫比對方法的準確率不容置疑地達到了百分之百，但在犯罪現場證據的迅速採集方面，時間壓力極為關鍵。在此背景下，深度學習模型不僅以接近完美的準確率（約 99%），同時還在處理速度上顯示出了巨大的優勢。這一點對於在有限時間內識別和保全數位證據來說至關重要。此外，隨著更多樣本的加入，這些模型有能力自我完善，進一步提高識別準確率。隨著本方法的廣泛應用，更豐富的現場檔案樣本將被輸入到深度學習模型中，這不僅將使方法變得更為精確，其即時更新和增量學習的特性也使其在使用上更為靈活和具有發展潛力。這對法律執行和數位鑑識領域將產生深遠的影響。在研究的最後，我們提出了結合 TextCNN 和字庫法的創新策略。利用 TextCNN 在速度上的顯著優勢以及字庫法的絕對準確率，這一結合策略有望開發出一種既快速又準確的助記詞識別方法。這將賦予執法人員迅速扣涉案虛擬資產的能力，有效防止犯罪者即時轉移非法資產，從而保障民眾的財產安全。

未來，如何提高深度學習模型在法律執行和數位鑑識領域的適法性和可解釋性[27], [28]，是個值得探索的問題，這部分我們將朝多個方向進行努力。首先，我們將整合解釋性技術，例如 LIME（Local Interpretable Model-agnostic Explanations）[29]和 SHAP（SHapley Additive exPlanations）[30]，以解釋模型的預測結果。透過這些方法，我們可以生成對模型預測的可解釋性解釋，有助於法庭理解模型的決策過程。此外，我們計劃展開更多的研究，以改進深度學習模型在法庭上的可接受性，包括對模型結構的調整，以使其更容易被理解，以及對模型內部特徵的分析並揭示其對最終預測的貢獻，以增加其透明度[31], [32]。透過這些改進，我們希望能夠使深度學習模型的運作過程更具可靠性及可解釋性，從而增強其運用於司法體系的效能和公正性。

誌謝：本研究感謝國科會研究計畫「區塊鏈技術在犯罪偵防中之整合應用」（編號：MOST 112-2222-E-015-001）之經費支持。

參考文獻

[1] G. Tziakouris, “Cryptocurrencies—a forensic challenge or opportunity for law enforcement? an interpol perspective,” *IEEE Security & Privacy*, vol. 16, no. 4, pp. 92–94, 2018.

[2] M. Levi, *Consent, dissent, and patriotism*. Cambridge University Press, 1997.

[3] A. Holmes and W. J. Buchanan, “A framework for live host-based Bitcoin wallet forensics and triage,” *Forensic Science International: Digital Investigation*, vol. 44, p. 301486, 2023.

[4] S. Zollner, K.-K. R. Choo, and N.-A. Le-Khac, “An automated live forensic and postmortem analysis tool for bitcoin on windows systems,” *IEEE Access*, vol. 7, pp. 158250–158263, 2019.

[5] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, “Natural language processing (almost) from scratch,” *Journal of machine learning research*, vol. 12, no. ARTICLE, pp. 2493–2537, 2011.

[6] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[7] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” *Decentralized business review*, 2008.

[8] A. Montanez, “Investigation of cryptocurrency wallets on iOS and Android mobile devices for potential forensic artifacts,” *Department Forensic Science, Marshall University: Huntington, WV, USA*, 2014.

[9] T. Haigh, F. Breitinger, and I. Baggili, “If I had a million cryptos: cryptowallet application analysis and a trojan proof-of-concept,” presented at the Digital Forensics and Cyber Crime: 10th International EAI Conference, ICDF2C 2018, New Orleans, LA, USA, September 10–12, 2018, Proceedings 10, Springer, 2019, pp. 45–65.

[10] A. G. Nabi, “Analytic Study on Android-based Crypto-Currency Wallets,” 2018.

[11] D. He *et al.*, “Security analysis of cryptocurrency wallets in android-based applications,” *IEEE Network*, vol. 34, no. 6, pp. 114–119, 2020.

[12] M. D. Doran, “A forensic look at bitcoin cryptocurrency,” 2014.

[13] E. Chang, P. Darcy, K.-K. R. Choo, and N.-A. Le-Khac, “Forensic Artefact Discovery and Attribution from Android Cryptocurrency Wallet Applications,” *arXiv preprint arXiv:2205.14611*, 2022.

[14] L. Van Der Horst, K.-K. R. Choo, and N.-A. Le-Khac, “Process memory investigation of the bitcoin clients electrum and bitcoin core,” *IEEE Access*, vol. 5, pp. 22385–22398, 2017.

[15] Y. Faqih, Y. Rahmanto, A. A. Aldino, and B. Waluyo, “Penerapan String Matching Menggunakan Algoritma Boyer-Moore Pada Pengembangan Sistem Pencarian Buku Online,” *Bulletin of Computer Science Research*, vol. 2, no. 3, pp. 100–106, 2022.

[16] X. N. Καρακάσης, “Mnemonic Maker 2D,” 2022.

[17] Y. Goldberg and O. Levy, “word2vec Explained: deriving Mikolov et al.’s negative-sampling word-embedding method,” *arXiv preprint arXiv:1402.3722*, 2014.

[18] M. Jiang *et al.*, “Text classification based on deep belief network and softmax regression,” *Neural Computing and Applications*, vol. 29, pp. 61–70, 2018.

[19] Y. Zhang and B. Wallace, “A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1510.03820*, 2015.

[20] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” *arXiv preprint arXiv:1409.2329*, 2014.

[21] P. Liu, X. Qiu, and X. Huang, “Recurrent neural network for text classification with multi-task learning,” *arXiv preprint arXiv:1605.05101*, 2016.

[22] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[23] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” *Advances in neural information processing systems*, vol. 28, 2015.
[24] Z. Huang, W. Xu, and K. Yu, “Bidirectional LSTM-CRF models for sequence tagging,” *arXiv preprint arXiv:1508.01991*, 2015.
[25] Q. Li *et al.*, “A survey on text classification: From traditional to deep learning,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 2, pp. 1–41, 2022.
[26] J. Davis and M. Goadrich, “The relationship between Precision-Recall and ROC curves,” presented at the Proceedings of the 23rd international conference on Machine learning, 2006, pp. 233–240.
[27] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, “Machine learning interpretability: A survey on methods and metrics,” *Electronics*, vol. 8, no. 8, p. 832, 2019.
[28] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, “A survey on neural network interpretability,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, 2021.
[29] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘ Why should i trust you?’ Explaining the predictions of any classifier,” presented at the Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.
[30] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” *Advances in neural information processing systems*, vol. 30, 2017.
[31] F. M. Granja and G. D. R. Rafael, “The preservation of digital evidence and its admissibility in the court,” *International Journal of Electronic Security and Digital Forensics*, vol. 9, no. 1, pp. 1–18, 2017.
[32] S. Goode, “The admissibility of electronic evidence,” *Rev. Litig.*, vol. 29, p. 1, 2009.