

國立臺灣大學工學院工程科學及海洋工程學系

碩士論文

Department of Engineering Science and Ocean Engineering

College of Engineering

National Taiwan University

Master Thesis

基於服務導向架構之人工智慧決策支援機器人程序自

動化

Service-Oriented Architecture for Robotic Process

Automation of Artificial Intelligence Decision-Support

林宣佑

Hsuan-Yu Lin

指導教授：張瑞益 博士

Advisor: Ray-I Chang, Ph.D.

中華民國 109 年 7 月

July 2020



摘要

近年來人工智慧 (Artificial Intelligence, AI) 浪潮席捲全球，全球各行各業皆積極地引入 AI 技術以提升產業價值及推動產業轉型。隨著 AI 產業快速地發展，對相關人才的需求也更加強烈，出現以下問題：(一) 目前 AI 開發的入門門檻高，人才稀缺，無法滿足日益增長的商業需求。(二) AI 的知識發掘 (Knowledge discovery in databases, KDD) 與參數調整乃至執行過程都很冗長，需要背景執行的批次處理。(三) AI 模型的開發及訓練流程難以複製，不易將成功的經驗應用於其他領域。針對以上問題，我們研究將機器人程序自動化 (Robotic Process Automation, RPA) 引入 AI 模型的開發及訓練流程。(一) 新增圖型化使用者介面 (Graphic User Interface, GUI)，將模型開發過程中的各個步驟元件化，讓 AI 決策支援的流程得以通過拖曳及點擊方式完成，降低開發的入門門檻。(二) 提供背景執行的批次處理功能，新增度量閘道元件，可依所設定的邏輯閘道完成決策流程分流。(三) 新增決策規則管理系統，使用者可儲存決策流程，未來在面對新問題時可依其資訊 (如名稱、描述、資料型態、任務類別等)，以 AI 算法搜尋相似應用的決策流程，直接匯入調整對應參數，即可達到解決相關流程知識的再利用。本系統採用服務導向架構 (Service-Oriented Architecture, SOA)，未來若有其他應用需求，可優先重新組合所提供的服務即可，無須重新開發。

關鍵字：人工智慧、流程機器人、批次處理、服務導向架構

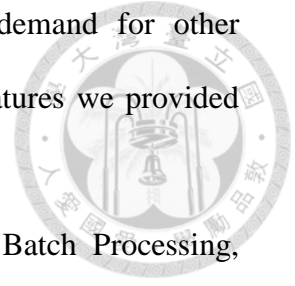


Abstract

Artificial intelligence (AI) has been going viral during the last years. People from all walks of life are introducing AI to enhance industrial value and promote industrial transformation. As this industry develops rapidly, there is a dramatic increase in the demand for talents and the following problems occurred. First, the high entry barrier of developing the AI and the existing talents are unable to meet the growing needs. Second, the processes of knowledge discovery in databases (KDD) and parameter adjustment and even the execution may turn out to be a long period of time. Therefore, we need to provide the feature of batching processing in the background. Lastly, duplicating the processes of model developing and model training is too hard to apply successfully in other fields. Measures to solve the above-mentioned problems, this study proposes a method of introducing Robotic Process Automation (RPA) into the field of AI's process of model developing and training. By componentizing each step in the process of model development, the decision-making process can be completed by dragging and clicking which avoid repeated steps of data pre-processing and model selection on the Graphic User Interface (GUI) we provided. Next, we provide the function of batch processing in the background and add the component of metrics gateway. Users can complete the route of decision-making process that based on the logical gateway that had set up. Finally, a new decision-making rule management system is added that users can store decision-making processes. When facing new problems, they can search for decision-making processes of similar applications with AI's algorithms based on their information (such as name, description, data type, task category, etc.). At the same time, users can import the decision-making processes and adjust the corresponding parameters directly to achieve the reuse of the knowledge process. This system chooses

to use Service-Oriented Architecture (SOA). If there is any demand for other applications in the future, all we need to do is reassemble the features we provided instead of redevelopment.

Keywords: Artificial Intelligence, Robotic Process Automation, Batch Processing, Service-oriented Architecture

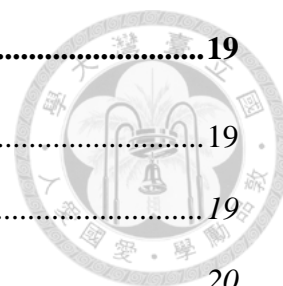


論文目錄



摘要.....	I
ABSTRACT.....	II
論文目錄.....	IV
圖目錄.....	VI
表目錄.....	VIII
第一章 緒論	1
1.1 研究動機與目的	1
1.2 論文架構	3
第二章 相關研究	4
2.1 知識發掘與人工智慧	4
2.2 相關人工智慧平台系統	6
2.3 機器人程序自動化	9
2.4 服務導向架構	11
2.5 長短期記憶模型	12
2.6 BERT.....	14
第三章 系統架構	15
3.1 系統設計	15
3.2 系統架構	15
3.2.1 資料蒐集模組.....	16
3.2.2 資料視覺化模組.....	17
3.2.3 資料分析模組.....	17

第四章 系統實作與實驗結果	19
4.1 系統實作	19
4.1.1 批次處理功能	19
4.1.2 機器人程序自動化	20
4.1.3 行動通訊系統	29
4.1.4 決策規則資訊系統	31
4.2 實驗流程	35
4.2.1 實驗設計	35
4.2.2 實驗結果與分析	36
第五章 結論與未來展望	39
5.1 結論	39
5.2 未來展望	39
參考文獻	41

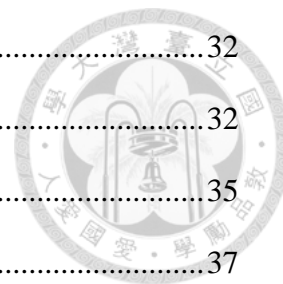


圖目錄



圖 1 AI 行動計畫整體推動架構[1]	1
圖 2 KDD 流程圖[4].....	4
圖 3 Azure Machine Learning designer[10].....	6
圖 4 Azure Machine Learning designer pipeline[10].....	7
圖 5 DataRobot Model Blueprints[11]	8
圖 6 流程智能化的四個階段[14]	10
圖 7 服務導向架構	11
圖 8 遞迴神經網路架構圖	12
圖 9 長短期記憶模型架構圖	13
圖 10 雙向長短期記憶模型架構圖	13
圖 11 BERT 架構圖	14
圖 12 系統架構圖	16
圖 13 資料分析模組	18
圖 14 批次處理功能架構圖	20
圖 15 傳統機器學習畫布	21
圖 16 單頁式機器學習畫布	21
圖 17 元件操作示意圖	24
圖 18 度量閘道元件示意圖	25
圖 19 資料轉化元件示意圖	26
圖 20 機器學習元件示意圖	27
圖 21 機器學習元件多輸入示意圖	28
圖 22 資料視覺化模組示意圖	29
圖 23 行動通訊系統架構圖	30
圖 24 行動通訊系統示意圖	31

圖 25 決策規則資訊系統	32
圖 26 決策規則推薦系統	32
圖 27 標籤預測模型架構圖	35
圖 28 使用者操作時間長條圖	37



表目錄



表 1 本研究與其他系統比較表	9
表 2 共同參數之權重比較表[15]	10
表 3 Azure 部分功能元件表	22
表 4 機器人程序自動化元件表	23
表 5 特徵圖例	27
表 6 詞向量模型參數設定	33
表 7 BiLSTM 模型參數設定	34
表 8 實驗受測者個人資料表	36
表 9 受測者分群數據表	38



第一章 緒論

1.1 研究動機與目的

近年來人工智慧 (AI) 在各個領域皆蓬勃發展，不論是在醫學影像、語音辨識、影像辨識等領域皆取得長足進展。在工業領域中，自德國於 2011 年提出了工業 4.0 的構想以後，工業 4.0 的浪潮席捲全球，其中智慧化更是在工業 4.0 中扮演著重要的地位，促使各領域皆積極地引入 AI 之相關技術以將產業智慧化。為有效的掌握 AI 發展趨勢，我國政府也於 2018 年提出了「台灣 AI 行動計畫」[1]，其中提出如圖 1 所示的五大主軸，啟動了 AI 產業化的進程。在各行業引入 AI 的浪潮下，對於 AI 人才的需求量也大大提升，中研院院士孔祥重在受訪時[2]亦提到，以現今十倍的招生量都還不足以因應產業的需求，因此解決 AI 產業人才缺額是迫在眉睫的問題。

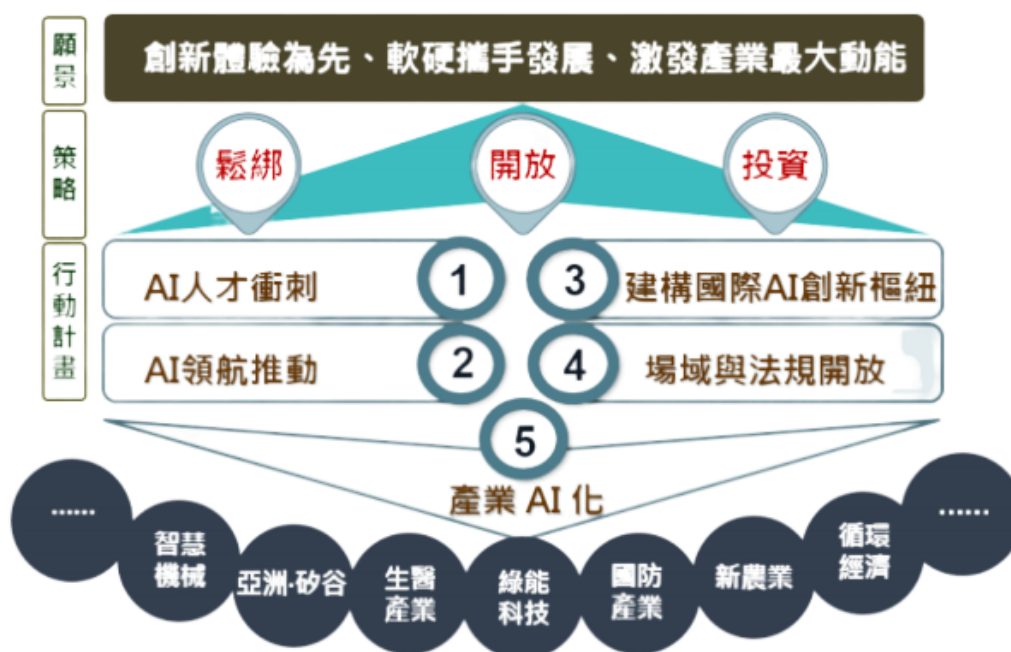



圖 1 AI 行動計畫整體推動架構[1]

在人才如此稀缺的情況下，不可避免的會有部分企業將現有的人才再培訓，以便應用於企業的產業 AI 化上，倘若能設計適當工具輔助進行 AI 開發與應用，可以大大降低企業引入 AI 的成本，提升台灣 AI 產業的競爭力。



在 AI 的開發過程中，資料轉化的方式、流程、演算法的選擇及如何優化模型等都會影響到結果的好壞，但在 AI 技術蓬勃發展的當下，資料轉化的方式與流程、演算法與參數的組合日漸增多，在開發過程中無法在有限的測試下找出最適當的方法。同時，隨著企業端資料量的累積，既有的模型準確率下降致使整個開發過程必須重新來過，曠日廢時且成效不彰；在企業成功開發了一套 AI 決策流程後亦難以將成功的經驗複製，因為資料的特性不同，就要採用不同的資料轉化方式與流程，所採用的演算法也會不同，因此即便擁有成功開發經驗的團隊，也難以保證可以將經驗成功複製。為了解決上述的問題，本研究基於服務導向架構[3] (SOA) 提出了不同於傳統撰寫程式碼開發 AI 決策支援演算法的方式，將複雜的資料轉化、演算法參數的選擇、特徵的選取等 AI 決策支援的步驟元件化，並以圖形化使用者介面 (GUI) 的方式呈現，讓現有的數據分析人才或是沒有產業實務經驗的演算法工程師，能夠更直覺的調整演算法；同時，服務導向架構能讓本系統更具有彈性與整合性，若有不同應用的需求也無須重新撰寫程式，只需要重新串接所提供的服務即可，大大的降低應用與開發上的成本。同時，本研究也利用新興的機器人程序自動化 (RPA) 的方式，將原始的知識發掘 (KDD) 以流程的方式呈現，讓演算法工程師可以直觀的瞭解資料的處理流程。此外，本系統也引入批次處理的功能，讓決策流程可以在系統背景執行，讓演算法工程師可以同時嘗試多種資料轉化方式，採用不同模型或參數進行訓練，省去等待與調整參數的時間，在決策流程完成後會主動通過即時通訊元件通知使用者，無須一直盯著螢幕查看變化，進而提升決策流程開發的效率。最後，本系統亦提供可將完成的決策流程匯出之功能，未來若有不同的應用想採用相同的決策流程，只需重新匯入並快速調整相應參數後即可嘗試應用，無須重新開發演算法，大大提升了決策流程重複利用的可能性，同時匯出的流程也會整併入決策規則資訊系統，提供其他使用者參考與調用，進一步加快演算法的開發並提升可用性。同時，在決策規則資訊系統中我們設計了決策規則查詢推薦系統，讓使用者更快速便捷取得相似決策流程。

1.2 論文架構

本論文第一章描述了研究動機與目的，為解決現有的演算法開發問題，及提升開發的效率引入機器人程序自動化及決策規則資訊系統。第二章為現有系統的探討，包括 Microsoft Azure 與 DataRobot，以及本研究所採用的相關技術。第三章則說明系統架構，闡述了系統的前後台設計。第四章為系統實作及相關實驗結果與討論。第五章針對系統總結並討論系統未來應用與展望。



第二章 相關研究

2.1 知識發掘與人工智慧

知識發掘[4, 5]是將低層數據轉換為高層知識的整個過程，目的是從資料中發掘出有效的、新穎的與潛在有用的資料，如圖 2 所示。其中，資料發掘的過程包含特徵選擇、資料前處理、資料轉化、資料探勘及資料評估等步驟。而其中，資料探勘是一項跨領域的技術，包括 AI、機器學習及資料庫等[6]。

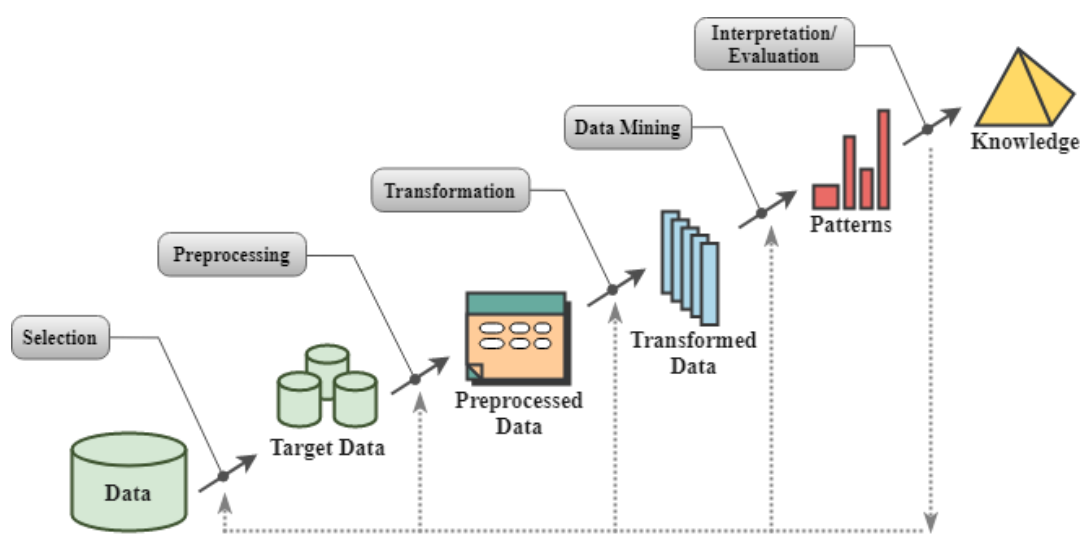
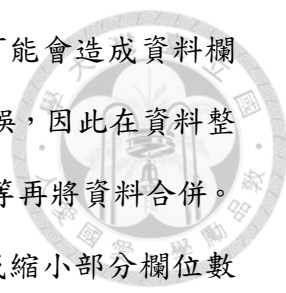


圖 2 KDD 流程圖[4]

資料特徵的好壞決定了知識發掘結果的優劣，特徵選擇的結果會直接決定模型訓練的效率與優劣，因此若能從原始資料中選取更適合的特徵即能有效的簡化模型取得有效參數的過程並提升開發效率，也能使模型的結構更簡單。在實務中所蒐集到的原始資料常會有資料缺失，雜訊與離群值等問題，因此在進行模型訓練前須先進行資料前處理與資料轉化工作，包括資料清理、資料整合與資料轉換[7]等步驟。

- 資料清理步驟是處理資料中的缺值，消除雜訊並修正資料中的不一致，對於資料缺失的處理有刪除缺失資料及填補眾數等方式。對於雜訊及離群值則會通過群集(Cluster)或迴歸(Regression)等方式將資料平滑化。

- 
- 資料整合步驟考慮到將不同來源的資料進行合併時可能會造成資料欄位不一致、資料蒐集時間不一致等問題，造成資料錯誤，因此在資料整合步驟時會先檢查資料的屬性、欄位名稱與資料匹配等再將資料合併。
 - 資料轉換步驟考慮到資料的不一致而造成過度放大或縮小部分欄位數值對模型的影響，常用平滑化、一般化及標準化等方式消除此類問題。平滑化常用在充滿大量雜訊的資料中，使雜訊產生的干擾降低；一般化可將資料的數值縮減到一個區間內，避免讓某個屬性的影響被放大或縮小；標準化則可幫助提升機器學習模型時的收斂速度，及提高模型的準確度。

資料探勘意指通過使用不同技術將龐大的資料建立模型，並從中找出隱藏的特殊關聯性及特徵的過程。包括六類常見的任務：異常檢測(Anomaly detection)、關聯分析(Association)、分類分析(Classification)、群集分析(Clustering)、迴歸分析(Regression)與匯總(Summarization)。

- 異常檢測是觀察訓練資料中與大部分資料不同的離群資料，常用於入侵檢測、故障檢測等。
- 關聯分析是分析資料庫中各資料彼此相依的機率，通常被用來分析公司各產品被同時購買的關係與頻率。
- 分類分析是透過研究資料庫中的特徵，將已知資料做出分類，並根據已知的特徵預測未經分類的新進數據。
- 群集分析和分類分析的概念相似，亦是將一資料庫的資料做出分類，並歸納出群集間的差異性及群集中的相似性。
- 迴歸分析是透過一系列的現有數據去預測未知數據的可能值。
- 匯總則提供了一個更緊湊的資料集表示，包括生成視覺化和報表。

資料評估步驟主要為通過驗證提供客觀證據對規定要求已得到滿足的認定，驗證可以由資料探勘者(也就是模型的建立者)自己完成，也可以通過其他人參與，以與資料探勘者毫無關聯的方式進行驗證。一般驗證過程中，資料探勘者是

不可能不參與的，但對於認定過程中的客觀證據的收集、認定的評估等過程如果通過與驗證提出者無關的人來實現，往往更具有客觀性。通過結果驗證，資料探勘者可以得到對自己所挖掘的資料價值高低之評估。[8]



2.2 相關人工智慧平台系統

目前市面上已經有人工智慧平台系統可為使用者提供分析使用者自己數據的服務，例如 Microsoft Azure、Google AI Cloud 與 DataRobot 等。但上述的平台所提供的服務與操作方式不盡相同，對應的程序與目標的客群也有所不同。Microsoft Azure [9]是 Microsoft 所提供的雲端服務組合，所提供的服務包括計算資源資料分析等，可協助使用者建置管理與部屬自己的資料分析應用程式，該平台於 2019 年發佈了輔助機器學習開發的 Azure Machine Learning designer 服務 [10]，如圖 3 所示，使用者可以在互動式畫布上直觀地連接資料集和模組，以創建機器學習模型。

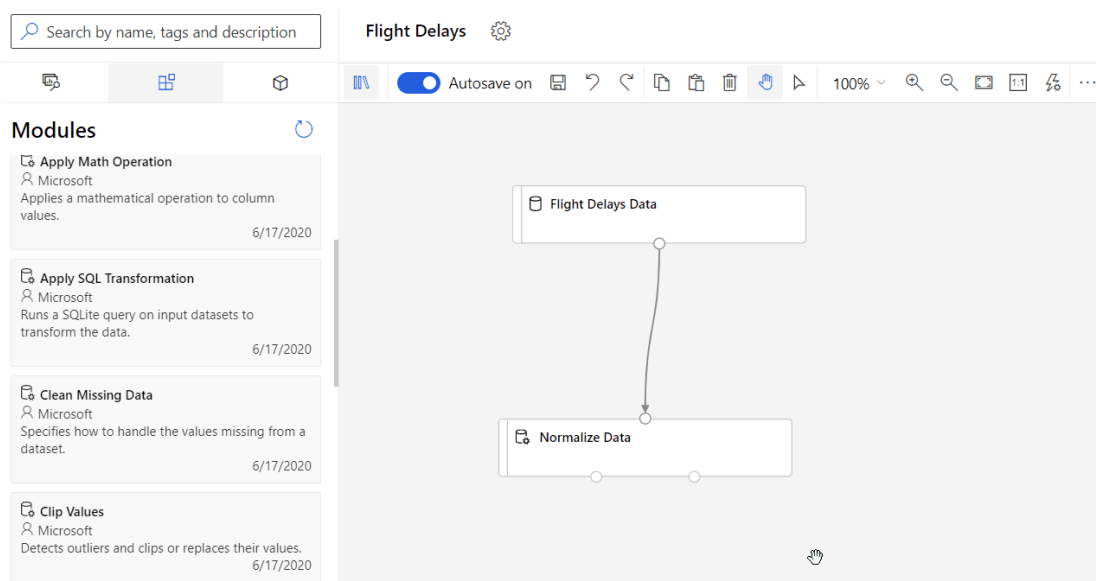


圖 3 Azure Machine Learning designer[10]

使用者可以將資料集與模組放置到互動式畫布上，並通過拖拉的形式連接資料與模組以創建一個或多個管道，如圖 4 所示互動畫布可以創建即時或批次進行預測的管道，也可以創建僅清理資料的管道，其中管道允許您重複使用工作並組織專案。

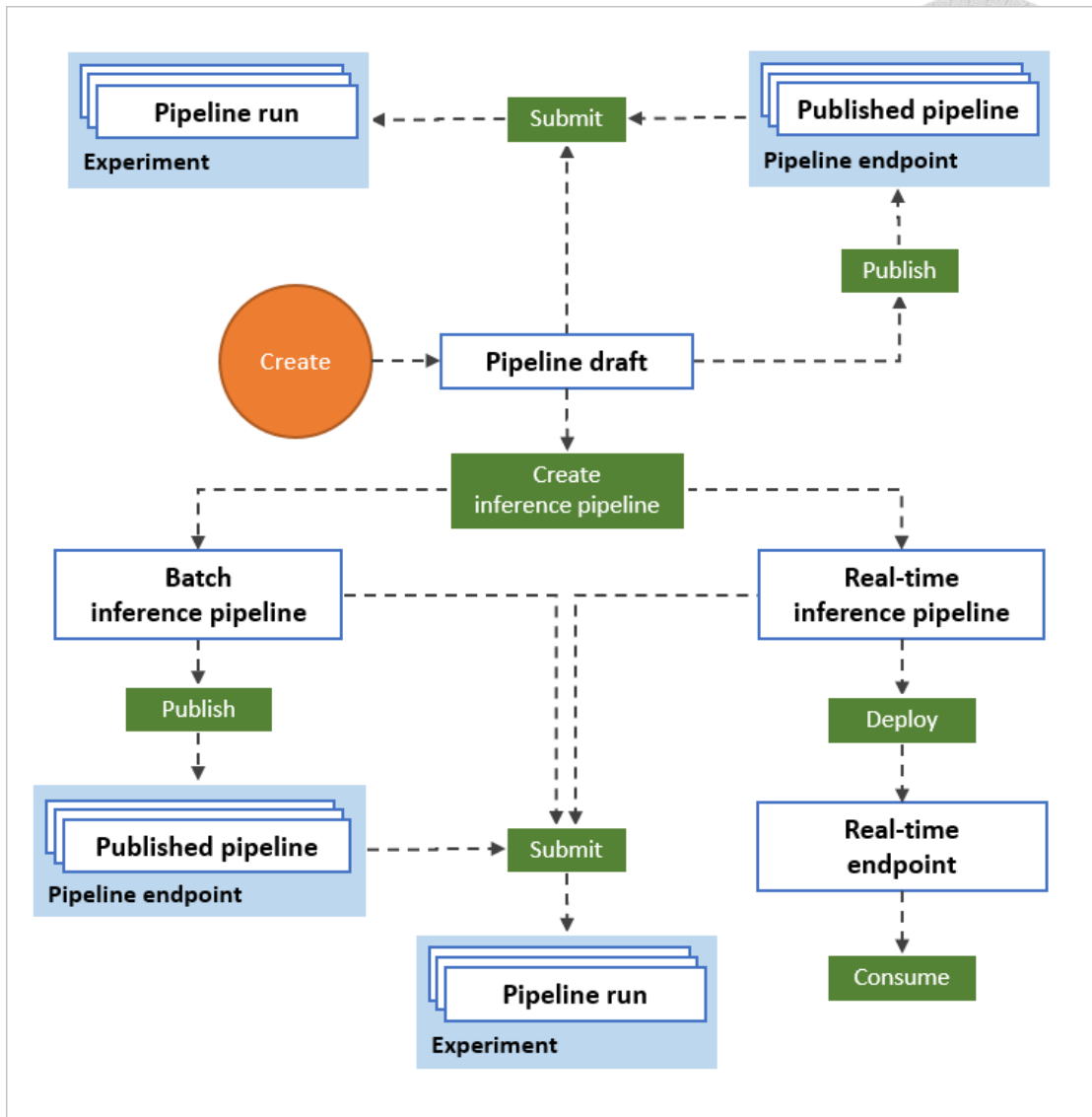


圖 4 Azure Machine Learning designer pipeline[10]

如圖 5 所示 DataRobot 也通過 Model Blueprints 服務[11]將 AI 的模型通過 GUI 直覺的以流程的方式展示給使用者。DataRobot 是基於專家系統(Expert System)提供開發團隊協助建置 AI 模型，使用者可以透過 DataRobot 的 Model Blueprints 了解到訓練過的模型中資料進行了哪些前處理步驟，做了哪些特徵工程以及使用哪些演算法與參數。然而 DataRobot 是一個 AutoML (Automated Machine Learning)的服務，此功能僅能讓使用者了解所使用的模型的流程，並不能對其中的參數進行修改。

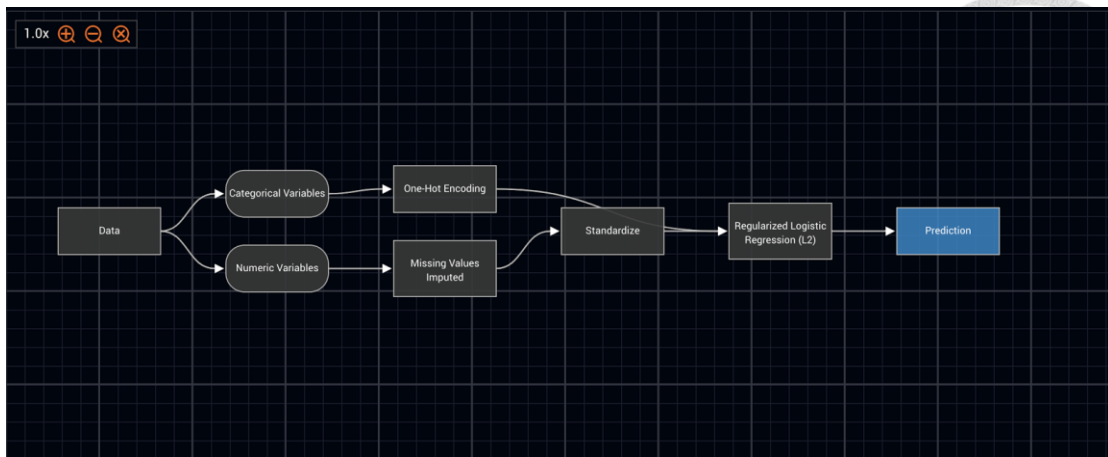


圖 5 DataRobot Model Blueprints[11]

Google AI Cloud[12]同樣提供分析使用者自己的數據的服務，同時它支持 Kubeflow，使用者可以建立可移植的機器學習管道(Pipeline)，這些管道可以在不同的環境中執行，不須大規模地改寫程式碼。同時，也通過 Notebooks 提供整合 JupyterLab 環境的代管服務，可以讓機器學習開發人員和數據資料學家建立 JupyterLab 的執行個體。此外，亦提供可直接使用其他用戶所公開發布的 AI Service 的功能。

InAnalysis [13] 為本實驗室所開發提供機器學習與深度學習服務的系統，讓使用者能通過網站點擊即可完成 KDD 資料蒐集、資料轉化、特徵工程與模型訓練等。本系統無須撰寫程式通過點擊即可完成上述功能，讓使用者能夠快速上手 AI 與機器學習流程，同時能夠快速建置 API 將模型佈署至任何應用，讓沒有 AI 開發經驗的開發者也能通過引導輕鬆上手。

隨著環境與技術的快速發展，數據分析的需求亦是日新月異，縱使目前已有許多商業平台與本實驗室的 InAnalysis 平台提供資料分析與模型訓練的服務，仍難以滿足使用者所有的需求。基於考量使用者在操作上的便利性與系統發展上的彈性，本研究之目標在於提供使用者更全面性的服務，整合所有使用者需求並提升操作上的便利性。如表 1 所示，針對本研究所提供的服務與市面上的商業平台系統進行比較，本研究集合了上述平台的優勢開發了新的 AI 與機器學習平台，讓使用者可以通過本系統的引導功能順利開發模型，同時有經驗的使用者也能開

發自己的演算法並上傳到系統，使用平台所提供的計算資源，縮短訓練時間。最後本系統也提供流程化的 GUI，使用者可以通過連接流程的方式，更直覺的瞭解演算法開發的步驟，並能輕鬆地調整流程。



表 1 本研究與其他系統比較表

系統 功能	Azure	DataRobot	Google AI	InAnalysis	本研究
資料分析	O	O	O	O	O
客製化演算法	O	X	O	O	O
流程系統	O	X	X	X	O
演算法視覺化	O	O	X	X	O
引用公開演算法	X	O	O	O	O
客製化 引用演算法	X	X	X	X	O

2.3 機器人程序自動化

近年來已有許多關於機器人程序自動化[14, 15](Robotic Process Automation, RPA)之相關研究，其中 Robotic Process Automation Through Advance Process Analysis Model[15]文獻指出 RPA 目前已廣泛的被使用於各種行業中，如金融業、人力資源及醫療保健等。在此研究中，透過所設計的分析模型將商用流程做詳細的分析，分別計算出傳統方法及使用 RPA 方法的流程中各項參數的權重，並以兩者間共同的參數做數據比較，如變化頻率、複雜度、使用時間、成交量及螢幕使用時間等做量化比較，如表 2 所示。其數據顯示兩者的共同參數項之權重和

於使用 RPA 方法下明顯降低約 25.48%，證實了使用 RPA 相對於傳統的方法更有效率。



表 2 共同參數之權重比較表[15]

Comparison Parameters	Traditional Model (%)	Proposed Model (%)
Frequency of Change	19.00	4.88
Degree of Complexity	9.50	9.75
Volume of Transactions	9.50	4.33
Time Taken	9.50	8.67
Screen Usage	9.50	3.90
Total	57.00	31.53

在上述的研究中已證實使用 RPA 方法可有效地提升系統的使用效率，但目前許多企業的系統中仍有許多工作需依賴人工處理，而這樣的工作大多是重複且枯燥的，而 RPA 能將這類重複的工作流程化、自動化，能有效的取代人力的投入，進而降低人力成本取得經濟效益[16, 17]。同時，在人工智慧的流程中亦存在許多重複的工作，目前仍沒有系統能完整地將 RPA 引入人工智慧的領域中，因此本研究嘗試將 RPA 引入人工智慧領域，降低使間成本以提升經濟效益。



圖 6 流程智能化的四個階段[14]

2.4 服務導向架構

服務導向架構[3]如圖 7 所示是一種架構模型，由網站服務技術等標準化元件組成，它允許使用者從現有的服務通過不同的組合建構應用程式[18]。服務提供了一個抽象的介面（Interface），移除了底層的複雜性，網站開發人員無須了解內部如何實作服務，只需根據應用的需求訪問服務即可。同時，抽象的介面亦有利於不同系統進行客製化。

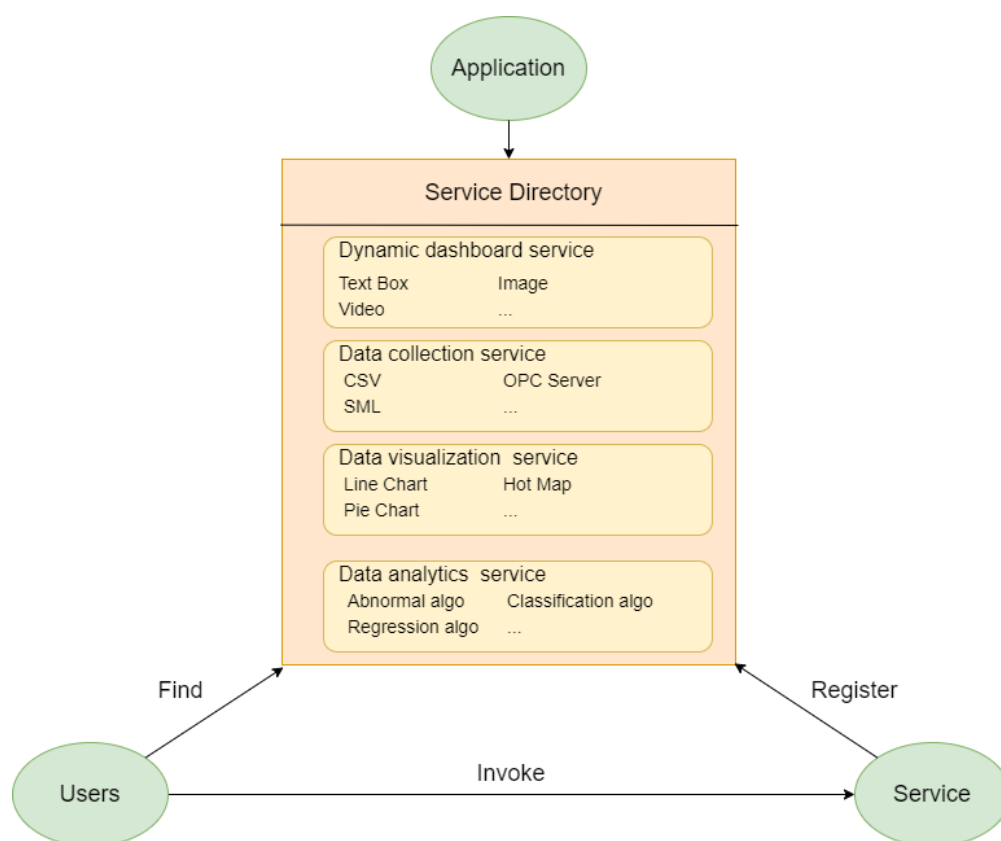


圖 7 服務導向架構

採用 SOA 能為使用者提供一個具彈性、可重複使用的整合性介面，促進內外部（如內部應用程式、用戶、與部門等相關單位）完美的溝通，盡速達到提升網路服務的目標[19, 20]。

SOA 具有以下技術特性：

- 分散式架構 (Distributed) — SOA 的組成元件是由許多分散在網路上的元件組合。

- 關係鬆散的界面 (Loosely coupled)—一般的系統主要是將應用系統功能切割成相互關聯的小零組件：模組、物件或元件。SOA 的作法是以界面 (Interface) 標準來組合系統，只要符合界面要求，零組件可以任意替換，大幅提高系統的彈性。
- 依據開放的標準 (Open standard)—使用開放標準是 SOA 的核心特色，許多軟體元件平台會採用專屬協定作為元件連結的規範，使得不同平台的元件無法相通。SOA 則著重於標準與互動性，將可避免不同平台開發程式間相互整合的困擾。
- 以流程角度出發—在建構系統時，首先了解特定工作的流程要求，並將其切割成服務界面，如此其他的發展者就可以依據服務界面開發合適的元件來完成工作。

2.5 長短期記憶模型

近年來關於自然語言處理 (Natural Language Processing, NLP) 的研究有了長足進展，目前相關研究大多採用遞歸神經網路[21](Recurrent Neural Networks, RNN) 進行，如圖 8 所示可以以遞歸神經網路中記憶的概念對文章進行理解，幫助模型理解語意。但當輸入為較長的文章時 RNN 無法處理隨著遞歸而產生權重指數爆炸或梯度消失 (Vanishing Gradient) 等問題，因此難以捕捉長時間關聯。

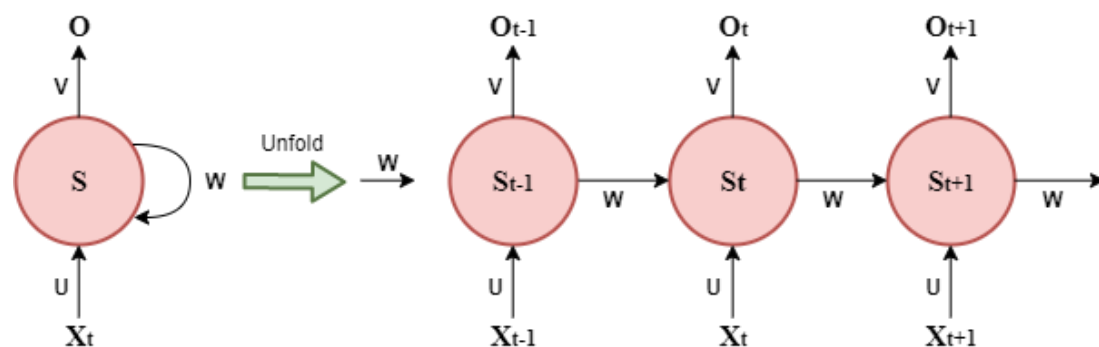


圖 8 遞迴神經網路架構圖

長短期記憶模型[20] (Long Short-Term Memory, LSTM) 能有效的解決梯度消失的問題，如圖 9 所示 LSTM 與 RNN 之不同在於 LSTM 實作了閘門 (Gate)

的機制以消除梯度消失的問題，其中 LSTM 的閘門包括輸入門 (Input Gate)、遺忘門 (Forget Gate)、輸出門 (Output Gate)，能讓運算結果選擇性的通過閘門進而達到避免梯度消失的問題。

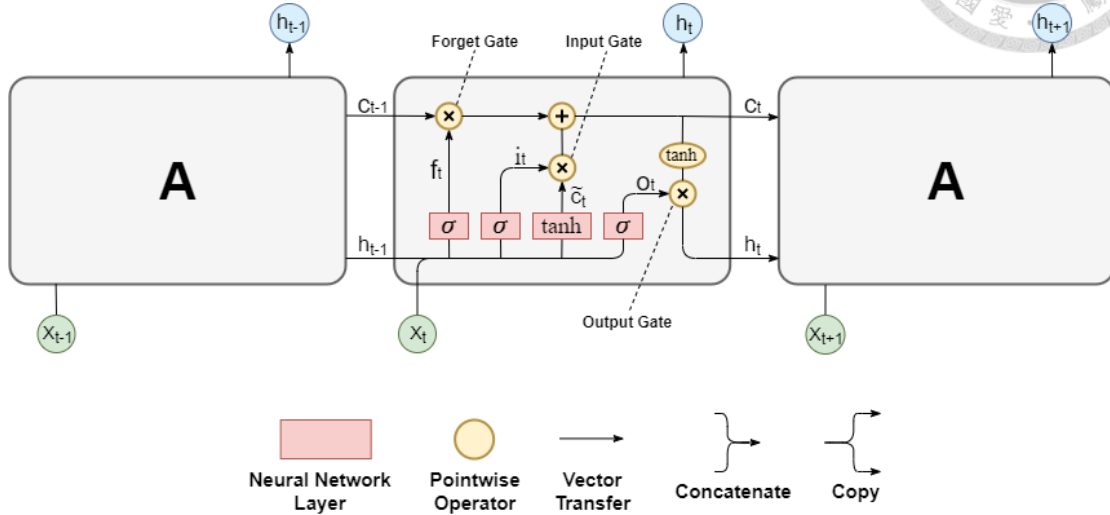


圖 9 長短期記憶模型架構圖

但上述長短期記憶模型僅能循序將文章輸入並進行訓練，因而無法利用後文的關係取得額外訊息，而雙向長短期記憶模型[22, 23] (Bidirectional LSTM, BiLSTM) 能進一步解決這個問題，如圖 10 所示 BiLSTM 能進行資料的雙向訓練，因本研究欲進行論文摘要的標籤標註須了解整篇文章的語意，故選擇使用了雙向長短期記憶模型。

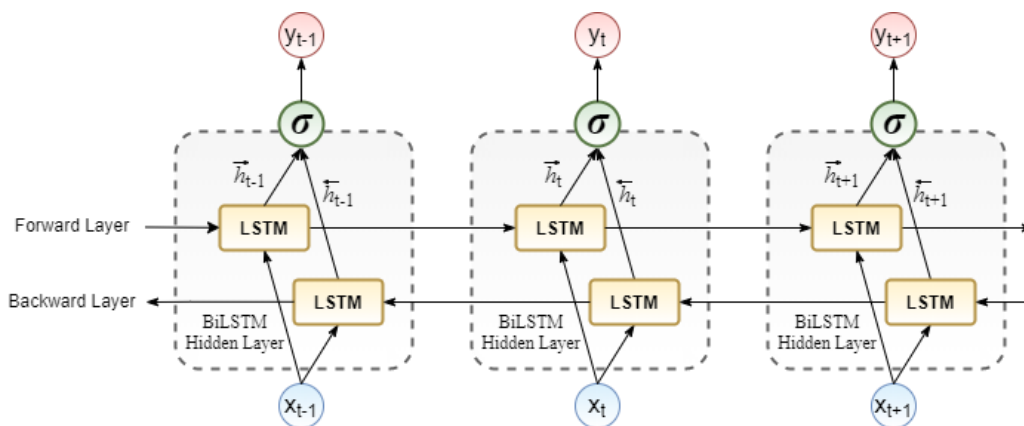


圖 10 雙向長短期記憶模型架構圖

2.6 BERT

BERT[24]是 2018 年 Google 所提供的語言模型，如圖 11 所示 BERT 採用雙向 Transformer 的 Encoder，BERT 會將輸入的文字先轉化為詞向量[25] (Word Embedding) 後輸入模型中，並進行預訓練 (Pre-training)，在預訓練時 BERT 使用維基百科資料與圖書語料庫共約 30 億個單字的未標記資料進行非監督式學習 (Unsupervised Learning) 進行遮罩語言模型 (Masked LM) 與下一句子預測 (Next Sentence Prediction) 的訓練，在訓練完成後 BERT 的輸出為詞向量，因而可以在 BERT 的基礎上進行微調 (Fine-Tuning) 即可滿足大部分的自然語言處理任務，同時採用 BERT 模型亦可以減少預訓練的時間與減少計算資源的浪費。

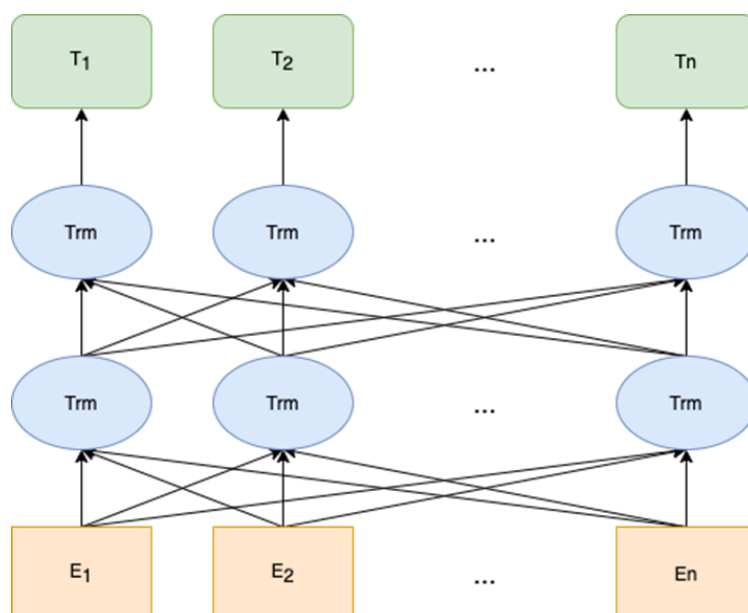


圖 11 BERT 架構圖

第三章 系統架構



3.1 系統設計

本系統除了沿用 InAnalysis 的資料前處理、特徵工程與訓練模型等一系列功能外，更進一步提供影像分析與自然語言處理功能。此外，本系統引入了決策流程功能，並實作機器人程序自動化。使用者可以使用單頁式機器學習畫布上的各元件，通過便利的 GUI 將元件以拖曳連線的方式達成資料分析、模型訓練與運維決策等操作流程，簡化了模型流程設計的難度，同時也能更直觀的瞭解資料處理流程。本系統也提供批次處理，讓工程師能夠一次嘗試多種資料前處理方式或是嘗試多種演算法與參數，本系統會在後台執行流程並回覆結果，使用者無須於螢幕前等候，在決策流程結束後，系統會通過行動通訊系統主動推送通知。最後，使用者還可以使用決策處理流程匯出與匯入的功能，將開發好的決策流程匯出與匯入。本研究亦開發了決策規則資訊系統，使用者可以將設計完成的決策流程存入並開放給其他使用者使用，若未來有類似的應用需求可以進入決策規則資訊系統尋找相關應用，匯入所需的決策流程並使用 GUI 調整流程中相應的參數即可快速嘗試，無須重新開發整個決策流程。同時，為了讓使用者在尋找流程時能更方便與迅速，本研究設計了決策規則查詢與推薦系統，協助使用者尋找相應的決策流程。

3.2 系統架構

目前主流的前端框架分別有 AngularJS[26]、React[27]與 Vue[28]，AngularJS 從原本的領先地位逐漸被 React 和 Vue 所超越，同時 Vue 在搜尋熱度呈現上升的趨勢，故考量了前端框架趨勢與未來維護系統的便利性，本系統採用 Vue 作為前端框架。考量前端框架的變化，本系統選擇採用 SOA，倘若未來前端架構趨勢有變化時，系統的核心功能仍無須重新開發，只需重新開發前台與連接服務的介面即可。同時也因為採用了 SOA，未來若有其他應用領域或其他介面的需求，皆

不需重新開發核心功能，使用者只需從現有的服務通過不同的組合建構應用程序即可；若有其他需求也可以通過應用程序擴充，極大化系統的彈性[29]。

如圖 12 所示，本系統的前台架構採用 Vue，後台框架採用 Python Flask[30]，並通過後台與核心服務溝通，建置本系統。其中，基於 SOA 本系統使用了三個子模組，分別為資料蒐集模組、資料視覺化模組與資料分析模組。

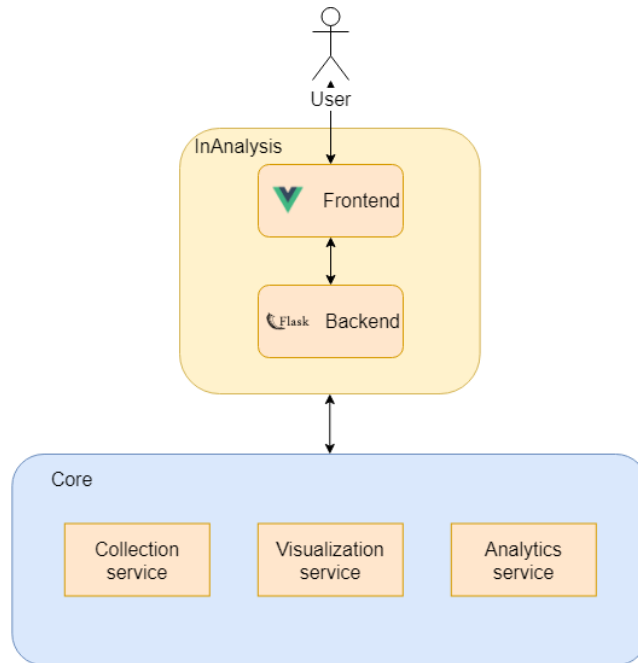


圖 12 系統架構圖

3.2.1 資料蒐集模組

資料蒐集模組提供了一個可供使用者上傳的服務，除原先的 CSV 與 XML 上傳以外本研究新增了 TSV 以及 ZIP 檔案，分別供數值專案、自然語言處理與圖像處理專案使用，通過本模組上傳的資料會直接上傳到核心服務供資料視覺化模組與資料分析模組使用。同時因原資料蒐集模組並不會對資料進行檢核，使用者須要等到在使用資料時才發生資料錯誤的問題，造成使用者操作流程中斷，因此本研究新增了資料蒐集模組在上傳檔案時會驗證使用者上傳的資料是否符合系統的要求，若資料為符合系統要求則會即時提示使用者，改善原系統亦造成使用者操作流程中斷之問題。

3.2.2 資料視覺化模組

在通過資料蒐集模組上傳資料後，使用者可以通過資料視覺化模組觀察所上傳的資料，其中資料視覺化模組是基於 Bokeh 所開發的。以往若系統有資料視覺化的需求，往往需要程式設計師自行實作視覺化的功能，倘若是多人開發的專案，則會因為每個人所使用的視覺化方法不同導致所呈現的視覺化方式不同進而影響到使用者體驗。採用資料視覺化模組後，只需要提供資料蒐集模組所回覆的 File Uid 與所要呈現的圖表類型，資料視覺化模組就會回覆圖表供使用者觀察，同時系統也會有一致的資料視覺化方式，給使用者一致性的使用者體驗。目前資料視覺化模組提供 PCA、折線圖、柱狀圖、圓餅圖等，能滿足多種不同的使用情境。

3.2.3 資料分析模組

資料分析模組提供了特徵選取、資料前處理、資料轉化、資料過濾等功能可以滿足 KDD[4, 5]的流程需求。資料分析模組會基於不同的資料分析需求提供不同的子模組，分別為迴歸模組、分類模組、異常偵測模組與分群模組[31]。使用者可以選取已通過資料蒐集模組上傳的檔案，並根據需求進行資料前處理流程；在特徵選取的過程中，使用者可以參考通過資料視覺化模組所提供的相關矩陣圖進行特徵選取；完成特徵選取後使用者可以使用相應的子模組進行模型訓練，模型訓練的過程是通過執行序控制，因此使用者可以同時訓練多個模型並隨時終止訓練。在模型訓練完成後使用者可以使用資料分析模組所提供的測試與預測子模組進行模型優劣的評估，其中模型測試功能採用資料視覺化模組，提供視覺化的圖表輔助使用者能更直覺地觀察及評估模型的優劣。

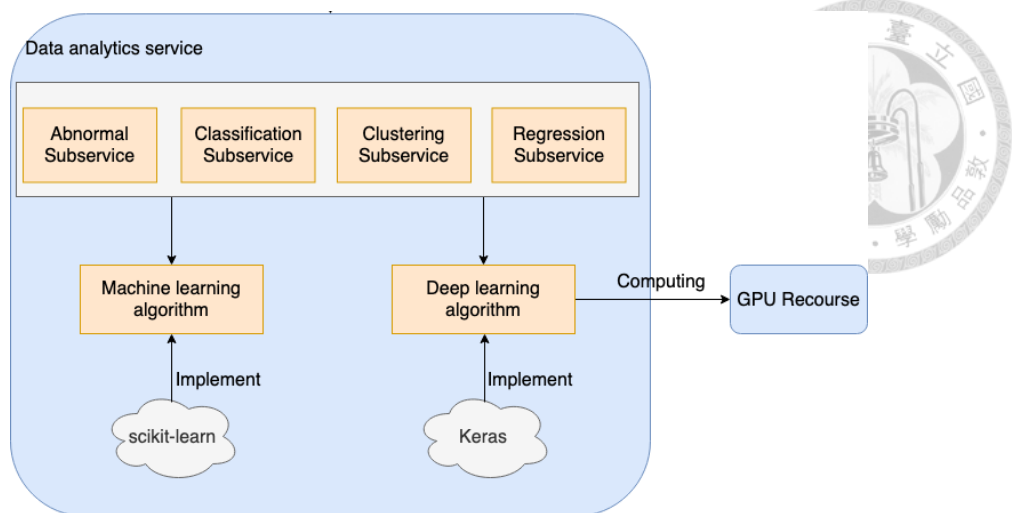


圖 13 資料分析模組

如圖 13 所示上述功能是基于 scikit-learn 和 Keras 開發的，這兩個機器學習模組是目前最為廣泛使用的套件，因此若有擴充相關性演算法或訓練模型的需求，資料分析模組擁有很大的相容性，能簡單地達到擴充的需求。當使用資料分析模組中的深度學習演算法時，資料分析模組會取用 GPU 資源，加速模型訓練，減少使用者等待的時間。

第四章 系統實作與實驗結果



4.1 系統實作

本研究基於 SOA 開發了包含有動態儀表板模組、資料蒐集模組、資料視覺化與資料分析模組的核心服務，並組合其中的資料蒐集模組、資料視覺化與資料分析模組重新建置了本實驗室的 AI 與機器學習平台 InAnalysis，新增了批次處理功能，讓使用者無須專注於螢幕前等待訓練結果，執行後會將各模組在背景執行，並在執行結束後通過行動通訊系統主動通知使用者，並使用上述模組建置了單頁式機器學習畫布，使用者可以在畫布上通過拖曳點擊等行為，使用新增的元件完成知識發掘流程，達到機器人程序自動化的目標。在完成上述決策流程的設計後，本研究設計了決策規則資訊系統，使用者可將決策流程存入，未來若有相似應用，使用者即可從決策規則資訊中尋找相似應用，調整相應參數後即可快速嘗試應用，無須重新開發決策流程。同時，考量使用者在尋找決策流程的過程中若增加推薦系統能更便利地尋找出適當的流程，因此本研究設計了決策規則推薦系統，採用關鍵字檢索、知識關係檢索與標籤檢索，協助使用者尋找決策流程。

4.1.1 批次處理功能

在原系統中核心服務各模組的操作行為建置於系統主程序中，因此單一使用者在同時僅能進行單一流程模組的操作，在模組未完成時會阻塞主程序，所以使用者須專注於系統的執行。因此本研究新增了批次處理功能，如圖 14 所示將流程中的各模組改由多執行序管控，使用者在完成各模組的對應參數設定後，執行模組時，批次處理功能會新建執行序以執行對應模組，並將執行序以唯一識別碼通過批次處理功能進行管理，使用者可以通過此唯一識別碼查詢模組的執行狀態，同時因為採用了多執行序執行各模組，因而在執行模組的過程中不會阻塞系統主程序，達成背景執行核心服務各模組的批次處理功能，讓使用者不再需要時刻專注於系統等待執行結果。

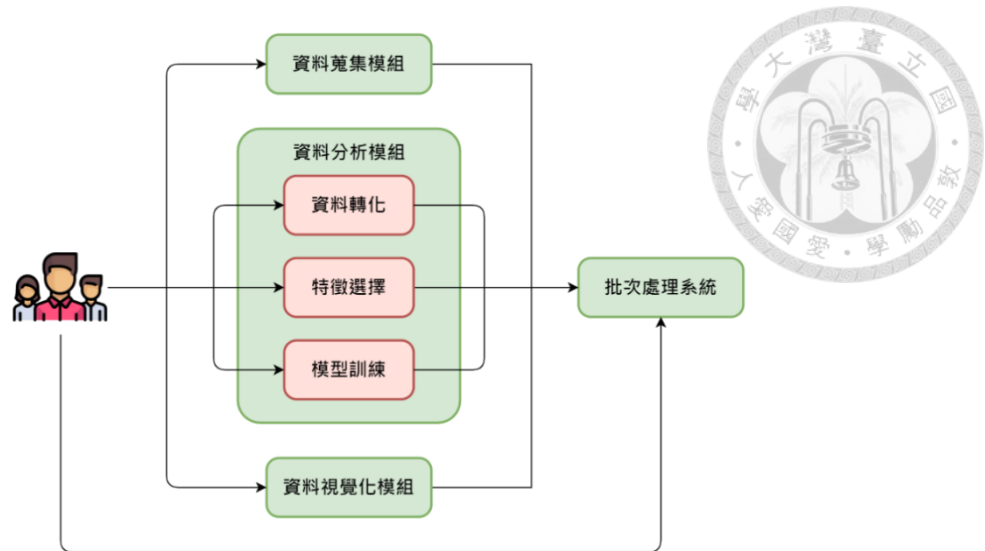


圖 14 批次處理功能架構圖

4.1.2 機器人程序自動化

為了加快模型開發時程、引導使用者開發模型及輔助使用者更直覺地了解演算法的開發步驟，本研究引入了機器人程序自動化的功能，開發了單頁式機器學系畫布。將核心服務的各模組元件化成便利的 GUI，使用者通過將元件拖曳連線等方式即可完成 KDD 中資料前處理、資料轉化與資料探勘等步驟。如圖 15 所示，在原系統的传统機器學習畫布中使用者須通過點擊按鈕，以網頁跳轉的方式完成 KDD 流程中的各步驟操作以完成模型訓練，而採用網頁跳轉的操作模式會導致使用者因網頁換頁或跳出訊息框等行為而反覆中斷使用者操作流程，容易導致使用者決策流程構想的中斷。為了解決上述問題本研究之機器人程序自動化功能改採用如圖 16 所示之單頁式設計，將新增元件與機器人程序自動化各功能置於左側邊欄中，將元件設定（圖中為度量閘道元件設定）置於右側邊欄中，使用者可以在同一頁面完成所有元件與功能操作，避免使用者決策流程構想中斷；同時如圖中所示本研究通過機器學習畫布將所有步驟以流程圖示展現，可以讓使用者更直觀的了解所設計的決策流程。

This is BETA version. Your account could be deleted after release.

InAnalysis Management Api Management RPA Document ICAN Sign Out

Project Management > Test

File List Total Files 3

File Name	File Type	Status	Actions
turbine_new	csv	Inuse	Preview Download Pre-processing Select to train Delete
pm2.5Train	csv	Inuse	Preview Download Pre-processing Select to train Delete
turbine_modify	csv	Inuse	Preview Download Pre-processing Select to train Delete

Model List Total Models 1

Model Name	Using File	Algorithm Name	Status	Actions
model	turbine_new	--	none	Preview Test Predict Management Delete

圖 15 傳統機器學習畫布

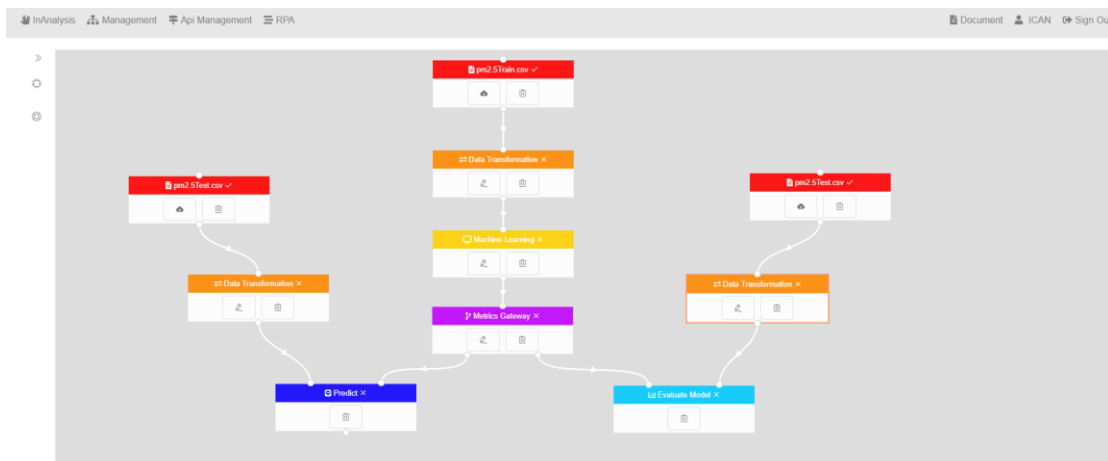








圖 16 單頁式機器學習畫布

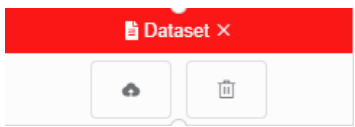
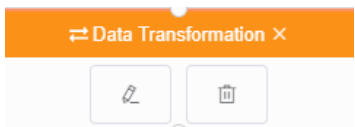
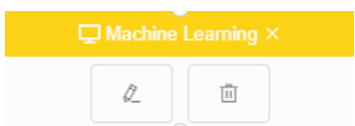
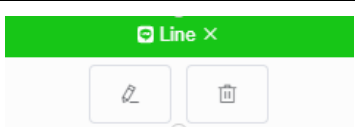
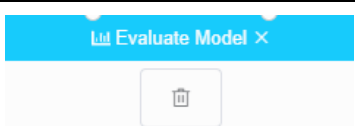
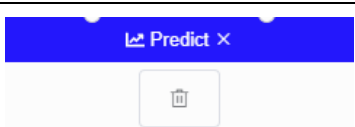
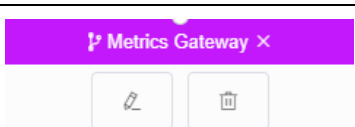
於文獻探討中得知目前僅有 Azure[10]提供流程系統的服務，並於其流程系統中提供功能元件供使用者設計流程，其部分功能元件如表 3 所示。因此本系統以此商業平台之設計為基礎，亦提供許多人工智慧相關功能元件於本研究之單頁式機器學習畫布中供使用者進行流程設計。

表 3 Azure 部分功能元件表

元件	圖示
資料集	 Automobile price data (Raw)
特徵選擇	 Select Columns in Dataset Exclude normalized losses which have many
分割資料	 Split Data Split the dataset into training set (0.7) and test
缺失值過濾	 Clean Missing Data Remove missing value rows
正規化	 Normalize Data
線性迴歸	 Linear Regression
模型訓練	 Train Model
模型評分	 Score Model
模型評估	 Evaluate Model

機器學習畫布上的各元件是通過將核心程式中的各模組元件化對應資料蒐集模組、資料視覺化模組與資料分析模組提出了如表 4 所示之資料取得、資料轉化、機器學習、Line 訊息、模型評估、模型預測與度量閘道 (metrics gateway) 等 GUI 元件，並同時採用元件圖示管理與元件色彩管理，讓使用者能清晰地選用本研究所提供的元件。由於本研究採用 SOA，在新開發之元件中可直接採用核心程式[29]的各模組功能，無須另外開發，達到提升開發時效與提升系統維護性的目的。

表 4 機器人程序自動化元件表

元件	圖示
資料取得	
資料轉化	
機器學習	
Line 訊息	
模型評估	
模型預測	
度量閘道	

機器學習畫布上的各元件之輸入輸出皆以拖曳或點擊方式供使用者進行設定，並可通過連線方式將元件進行連結，讓使用者能直覺且快速的完成決策流程

設計。如圖 17 所示，在使用者操作機器學習畫布時，元件上方的端口表示元件的輸入，元件下方端口即為此元件的輸出，使用者可以通過連線方式將一元件之輸出端與另一元件之輸入端進行連結，而當滑鼠懸浮於已連結元件時即自動出現刪除按鈕，使用者可以選擇是否將連結刪除。

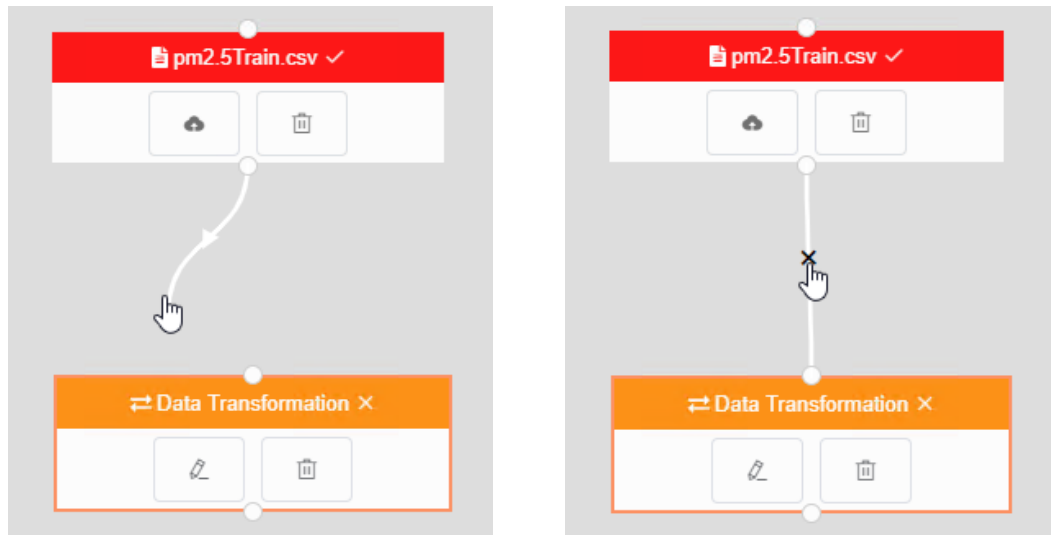
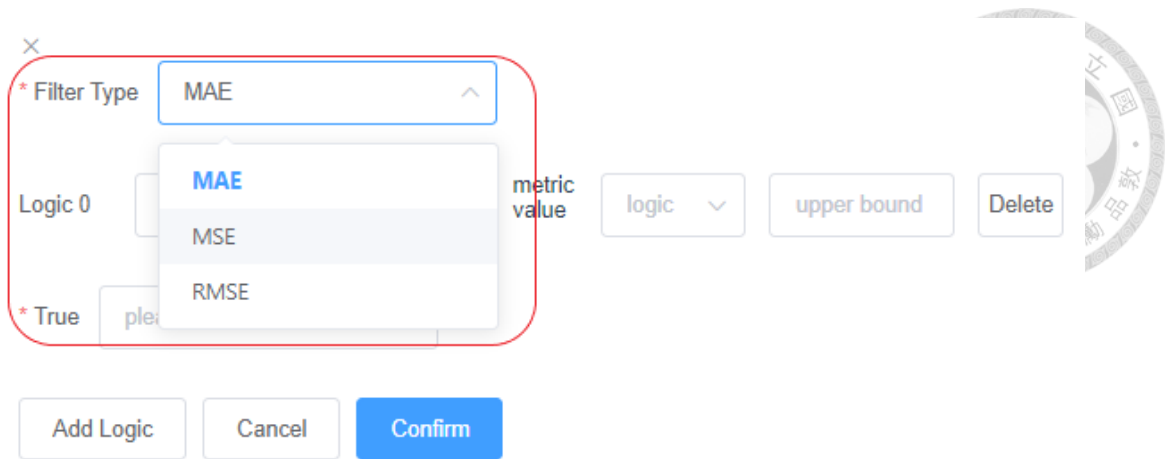
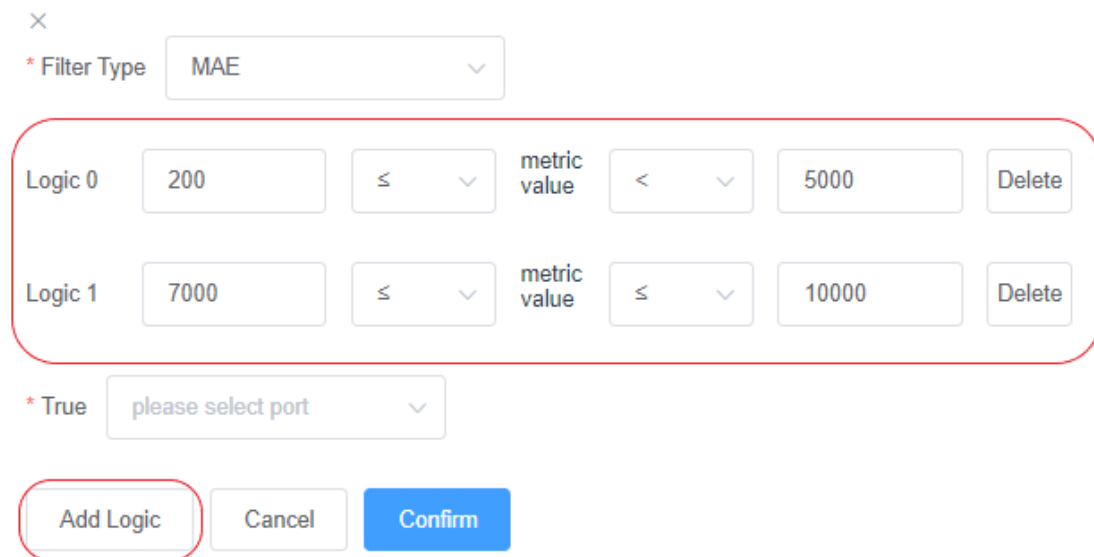


圖 17 元件操作示意圖

在原系統中，使用者須等待模型訓練完成後手動對模型進行評估，若不符合使用者預期，使用者須重新設計決策流程並等待。為了提供使用者更便捷的決策流程使用方式，本研究新增了度量閘道元件，如圖 18(a)所示度量閘道元件會自動排列輸入的度量選項，並以下拉選單的方式供使用者選擇（如圖中選擇 MAE），在選擇指定的度量後使用者可以指定邏輯閘道，設定當滿足所指定的邏輯閘道時決策流程應如何分流。如圖 18(b)所示若使用者欲指定多個邏輯閘道可點選左下角的 Add Logic 按鈕（如圖中所指定第一個邏輯閘道為 $if(200 \leq MAE < 5000)$ ，第二個邏輯閘道為 $if(7000 \leq MAE \leq 10000)$ ）。最後，在執行決策流程時決策流程元件即會自動通過上述使用者所指定的閘道對決策流程進行分流。



(a)



(b)

圖 18 度量閘道元件示意圖

此外，本研究亦針對基本的機器學習元件進行優化，在原系統中，使用者在使用資料前處理模組時每次只能針對單一欄位進行處理，若欲將多個欄位進行前處理或轉化時，使用者須反覆進行操作而造成使用上的繁瑣與不便。故本研究基於上述問題針對系統進行優化，如圖 19 所示資料轉化元件可自動的讀取輸入元件的檔案，且能同時對多個欄位進行處理（如圖中對 con_toptemperaturerup、grid_kvar 與 amb_windspeedpe 欄位進行一般化與平滑化），此外，資料轉化元件亦引入了 GUI 以採用下拉選單與開關方式供使用者操作，引導使用者進行資料轉化。

File Pre-Processing

Column Name	Filter Missing Value	Normalize Algorithm	Outliers Filter
Select All	<input checked="" type="checkbox"/>	Not to process	Not to process
mli_g01_nac_direction	<input checked="" type="checkbox"/>	Not to process	Not to process
mli_g01_hour_totalturbineok	<input checked="" type="checkbox"/>	Not to process	Not to process
mli_g01_phase3temperature_g	<input checked="" type="checkbox"/>	Not to process	Not to process
mli_g01_grid_inverterphase1temperature	<input checked="" type="checkbox"/>	Not to process	Not to process
mli_g01_con_toptemperature	<input checked="" type="checkbox"/>	Min-Max to 0~1	3rd standard d
mli_g01_grid_kvar	<input checked="" type="checkbox"/>	Min-Max to 0~1	2nd standard c
mli_g01_hour_totalwindok	<input checked="" type="checkbox"/>	Not to process	Not to process
mli_g01_amb_windspeedpe	<input checked="" type="checkbox"/>	Min-Max to 0~1	2nd standard c
mli_g01_hour_totalserviceon	<input checked="" type="checkbox"/>	Not to process	Not to process
mli_g01_grid_cosphi	<input checked="" type="checkbox"/>	Not to process	Not to process



圖 19 資料轉化元件示意圖

本研究亦對機器學習流程進行優化，在原資料分析模組中是以核取方塊 (Check box) 與輸入框提供使用者進行機器學習的特徵選擇與超參數 (Hyperparameter) 設定，這樣的介面雖能簡化操作但也可能會造成使用者的輸入錯誤。因此本研究針對機器學習元件進行優化，如圖 20 所示，機器學習元件會自動讀取輸入之資料集並分析特徵欄位屬性後將結果如表 5 所示以圖例呈現給使用者，讓使用者能直覺的了解所選取的特徵與模型輸入輸出兩者之間是否相符。同時，使用者能直接以拖曳的方式進行特徵選擇，倘若在圖例的引導下使用者仍選擇了與模型不符的特徵，本研究之機器學習元件會對使用者所選取之特徵進行檢核並提示使用者，降低使用者誤用特徵的可能。

表 5 特徵圖例






功能	圖標
可分類特徵	
整數特徵	
浮點數特徵	
圖像特徵	
文字特徵	



圖 20 機器學習元件示意圖

此外，如圖 21 所示機器學習元件亦優化了原資料分析模組模型只允許單一輸入或輸出的限制，若使用者所選取的演算法允許多個輸入或輸出，機器學習元件會於輸入框上方顯示 Multiple 的字樣提醒使用者。同時，針對超參數之設定本

元件亦進行優化，採以 GUI 方式引導使用者，當使用者選擇演算法後會讀取演算法可設定之超參數，如圖 21 上方區塊。不同的超參數型態會以不同的 GUI(如開關、滑塊等)引導使用者設定，相較於原系統之輸入框模式可有效地降低人為錯誤輸入的可能。

同時，如圖 22 所示本研究的各元件皆採用核心程式之資料視覺化模組，在原系統中使用者會通過不同視覺化套件實作視覺化，造成系統畫面呈現不一致，進而降低使用者體驗 (User Experience, UX)，故本系統引入了視覺化模組，在保留使用者客制化彈性的同時亦兼具了系統畫面的一致性，提升使用者體驗。

The image shows a user interface for configuring a machine learning model. At the top, there is a 'Select Algorithm' dropdown menu with 'in_linearRegression' selected. Below this, there are two hyperparameter settings: 'fit_intercept' with a green toggle switch turned on, and 'normalize' with a red toggle switch turned off. Underneath, there is a 'Select Feature' section with a table of features and their weights:

Feature Name	Weight
mli_g01_hour_totalturbineok	0.5
mli_g01_phase3temperature_g	0.5
mli_g01_grid_inverterphase1temperature	0.5
mli_g01_hour_totalwindok	0.5

Below the feature selection, there are two main sections: 'Input' and 'Output'. The 'Input' section is labeled 'X' and 'Multiple', and contains three features with weights of 0.5: 'mli_g01_amb_windspeedpe', 'mli_g01_con_toptemperature', and 'mli_g01_nac_direction'. The 'Output' section is labeled 'Y' and contains one feature with a weight of 0.5: 'mli_g01_grid_kvar'.

圖 21 機器學習元件多輸入示意圖

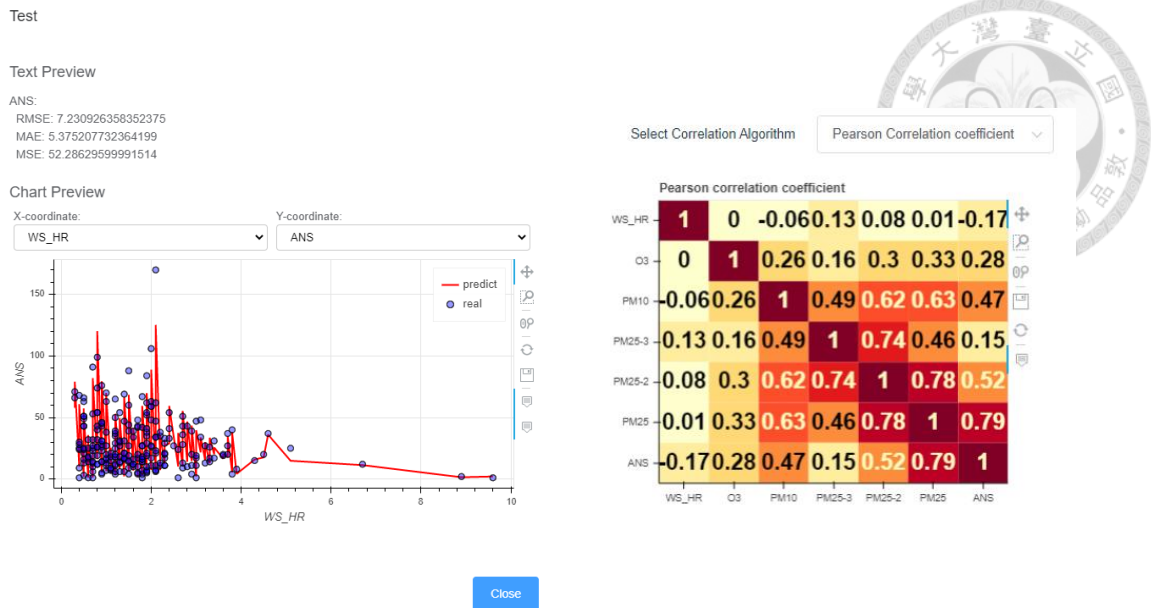


圖 22 資料視覺化模組示意圖

使用者通過上述各元件在單頁式機器學習平台完成決策流程設計後，搭配本研究之批次處理功能可在系統背景執行使用者所設計的決策流程，同時也因為搭配批次處理功能，使用者能在機器學習畫布中對同一檔案進行不同的資料轉化步驟並以此進行模型訓練，也可以採用不同的演算法或不同的參數，比較產出模型的優劣，省去反覆觀察結果及調整的時間，大大加速了決策流程的開發。

在使用者完成決策流程之設計後可使用匯出功能，將使用者所設計之決策流程以 json 檔案形式匯出，若未來有類似的應用即可匯入機器學習畫布，調整相應參數後即可嘗試應用，使複製成功的流程經驗成為可能。

4.1.3 行動通訊系統

在台灣網路報告中指出[32]有高達 94.8%的台灣網路使用者會使用行動通訊軟體，且行動通訊軟體已經實現跨裝置的使用，因此我們希望能與行動通訊軟體連接，改善原系統中使用者須在系統前等待資料轉化、模型訓練等相關步驟完成後才能進行後續操作之問題，達成使用者與系統通過行動通訊軟體雙向溝通的效果，並能更即時的觸及更多使用者，因而開發了行動通訊系統。

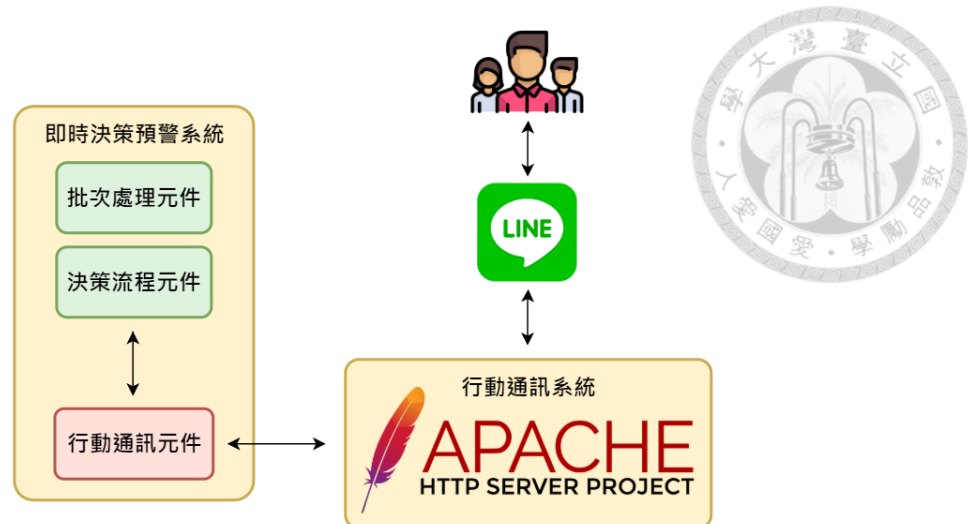


圖 23 行動通訊系統架構圖

如圖 23 所示，行動通訊軟體的使用者隱私是重要且必須被保護的資訊，若不能妥善的保護使用者於行動通訊軟體的唯一識別碼，可能會造成使用者的資訊遭到竊取進而造成使用者受到不明訊息騷擾的嚴重問題，因此本研究之行動通訊系統採用超文本傳輸安全協定[33] (HyperText Transfer Protocol Secure, HTTPS) 對荷載 (Payload) 進行加密以保護使用者資訊，基於此協定要求伺服器安裝合法且被信任的憑證頒發機構(CA)所頒發的憑證。故本研究之行動通訊系統使用 Apache[34]建立伺服器，並通過 Let's Encrypt 進行憑證申請與安裝，與各通訊軟體所提供的應用程式介面以 Webhook 即時監測，達成平台通過行動通訊軟體與使用者進行雙向溝通的目標，及確保使用者的網路通訊安全。

在使用者通過上述機器人程序自動化完成決策流程設計後，系統會通過批次處理功能在系統背景執行，在決策流程執行完成後系統會主動通過行動通訊系統推播通知，如圖 24 所示，系統會通知使用者流程執行結果並通過資料視覺化模組產生圖表直覺的了解模型訓練結果。



圖 24 行動通訊系統示意圖

4.1.4 決策規則資訊系統

在使用者通過本研究的決策流程元件完成決策流程設計後，除使用匯出功能將流程以 json 檔案形式保存外，使用者亦可將決策流程存入決策規則資訊系統，為了讓其他使用者能夠快速地尋找相似應用，本研究選取了決策流程的關鍵資訊分別為：名稱、描述、資料型態、任務類別等資訊存入關聯式資料庫 MySQL 中。同時，本研究亦於資料庫中設計了 version 與 top_version 兩個欄位，若使用者調整後的決策結果不如預期，可回復至過往的流程中。如圖 25 所示存入決策流程資訊系統的流程會在畫面中顯示流程之敘述、類別等資訊，若未來其他使用者有相似應用可以於此尋找相似的決策流程匯出後，將相應參數調整至符合新應用後即可快速嘗試應用，無須重新開發決策流程，提升開發效率。

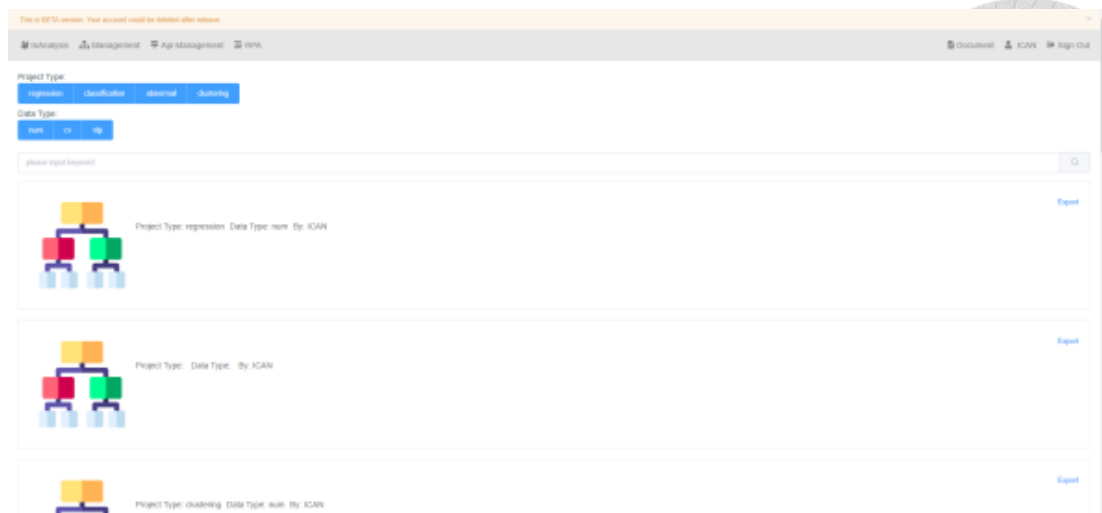


圖 25 決策規則資訊系統

同時考量若能增加搜尋與推薦之功能，系統能更方便使用者尋找適當的決策流程，故本研究參考了本實驗室之用例推薦研究[35]中的用例推薦模組，並針對本研究之決策流程進一步調整與優化，設計了如圖 26 所示之決策規則推薦系統，在使用者輸入關鍵字後，會將關鍵字分別輸入關鍵字檢索、知識關係檢索與標籤檢索，並將檢索結果整合後推薦給使用者。

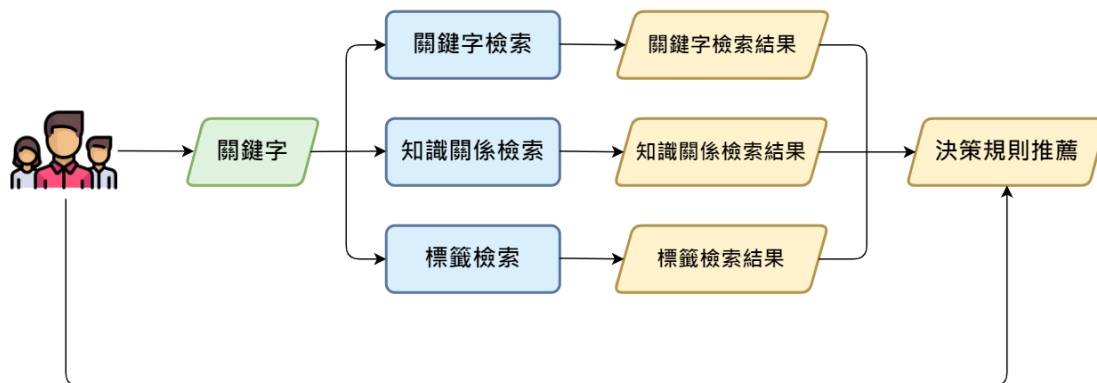


圖 26 決策規則推薦系統

首先在關鍵字檢索中，原研究採用資料庫中包含該關鍵字者即進行推薦，雖然這樣的設計可以涵蓋最多的推薦給使用者，但在大量資料與多使用者同時進行檢索的情況下，會造成資料庫搜尋的效能大幅下降，導致使用者需要長時間等待，因此在本研究的關鍵字檢索設計了一個建議的搜尋引擎，在使用者輸入關鍵字後，會將使用者所輸入的關鍵字與資料庫中所儲存之決策流程名稱與

描述進行評比，將評比分數關聯較高的推薦給使用者，避免將資料庫中的每一筆紀錄皆進行搜尋，解決了關鍵字搜尋的效能瓶頸。

在使用者輸入關鍵字時，可能會因為相關專有名詞翻譯差異的問題，因而採用關鍵字檢索時，無法推薦合適的決策流程給使用者，因此設計了知識關係檢索，使用維基百科中文文本與台灣碩博士論文網中與機器學習相關之論文摘要訓練詞向量模型，在使用者輸入關鍵字後會通過此詞向量模型取得與輸入之關鍵字相關的知識關係，並以此知識關係進一步進行檢索，達到關鍵字檢索功能的進一步輔助。

在訓練詞向量時，本研究使用上述蒐集之資料以結巴 (Jieba) 斷詞，將蒐集之資料以詞為基礎切分後，經過結巴之停用詞列表過濾後輸入詞向量模型中，並將模型如表 6 所示進行設定，完成訓練後將模型應用於知識關係檢索中。

表 6 詞向量模型參數設定

參數名稱	設定值
size	256
iter	5
windows	5
mini_count	5

最後本研究新增了標籤檢索，基於標籤標記[36]的研究提出了如圖 27 所示之標籤預測模型，進一步提升檢索效率及決策規則推薦的成效。我們使用臺灣碩博士論文網中與機器相關的論文摘要對其中約 200 篇摘要約八萬字針對使用演算法、關鍵字與專有名詞等三個類別進行 BIO 標記後輸入模型進行訓練，BIO 標記是指，將文章中的每一個元素標記為「B-X」、「I-X」、「O」，其中「B-X」的表示標記文字的開頭，「I-X」表示標記文字的中間，「O」表示非標記文字。

在完成資料標記後，首先會將已標記資料輸入 BERT[24]產出 token，因為摘要是有前後關聯的文本資料，因此我們選擇 BiLSTM[22]作為文章標籤預測模型，並如表 7 所示進行模型設定，同時文章標籤需要有序列化的輸出，因此我們加上一層條件隨機域(CRF, conditional random field)，使輸出序列化，對條件隨機域採用了新華社的標記後新聞資料進行預訓練。在使用者輸入決策流程描述時，系統會使用文章標籤預測系統，預測使用者所使用的演算法、關鍵字與專有名詞並存入資料庫中，在使用者使用決策流程推薦系統時，會將使用者所輸入的關鍵字與資料庫中的標籤進行比對，並將結果推薦給使用者，進一步增強系統的推薦效果。

表 7 BiLSTM 模型參數設定

參數名稱	設定值	描述
max_seq_length	202	標記文字的最大輸入序列長度
learning_rate	1e-5	模型學習率
num_train_epochs	400	訓練次數
dropout_rate	0.5	捨棄比率
lstm_size	128	LSTM 模型層數

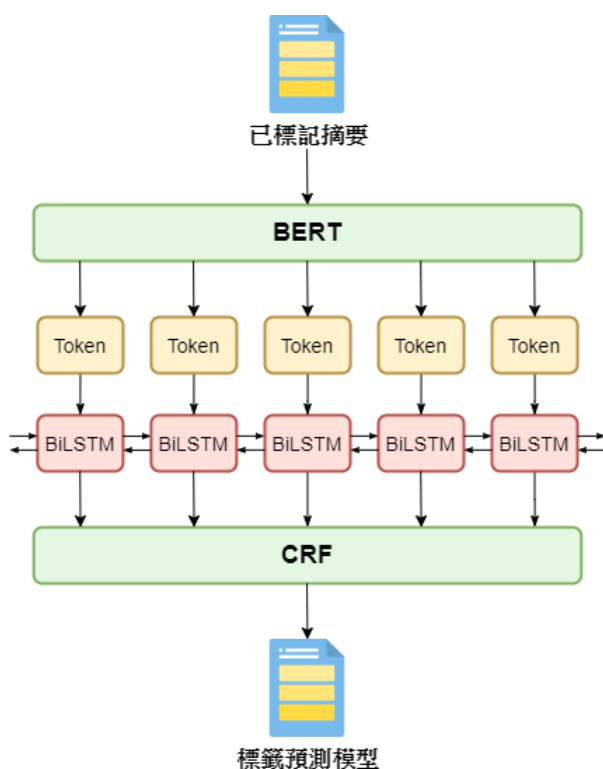


圖 27 標籤預測模型架構圖

4.2 實驗流程

本研究使用 UCI 上的資料集 Bike Sharing Dataset 進行實驗，首先將資料集隨機均分以模擬實際接收到新資料並重新訓練模型的情境，使用者針對資料中的所有欄位進行過濾缺值，並將其中 windspeed、hum 與 temp 進行正規化處理後使用系統中的 in_oneLayerNN 演算法設定超參數 epochs 為 100，而後選取 workingday、temp、hum 與 windspeed 作為輸入，cnt 作為輸出進行模型訓練。使用者分別使用傳統式機器學習畫布與本研究所新增的機器人程序自動化及決策規則資訊系統分別進行人工智慧流程設計。

4.2.1 實驗設計

首先因為傳統式機器學習畫布與本研究之單頁式機器學習畫布採用相同的使用者介面，為了消除對系統的熟悉程度造成使用者操作時間的影響，因此我將受測者分為兩群，其中一群先使用傳統式機器學習畫布，另一群先使用本研究之單頁式機器學習畫布。使用傳統式機器學習畫布的受測者，需使用另一批資料重新進行單頁式機器學習畫布操作，而使用本研究之單頁式機器學習畫布的使用者，

則使用本研究之決策規則資訊系統對第二批資料重新進行訓練，分別統計使用傳統式機器學習畫布與本研究之機器人程序自動化及決策規則資訊系統完成上述操作所花費時間。



4.2.2 實驗結果與分析

本實驗蒐集了 20 名不同的使用者，其中使用者的相關資訊如表 8 所示，本實驗之受測者皆就讀資訊相關系所，擁有一定的資訊背景。其中編號 1-5 的受測者為已使用過傳統式機器學習畫布的使用者，其餘皆為未使用過本系統之使用者，同時編號 1-10 的受測者有以自行撰寫程式進行機器學習流程的經驗，上述受測者分別如表 8 中之設定先後使用傳統式機器學習畫布與本研究之單頁式機器學習畫布進行上述流程操作並統計所花費時間（系統操作時間與螢幕使用時間總合），實驗結果如圖 28 所示，從圖中可以觀察出，除第 7 位受測者外使用本研究所設計之機器人程序自動化操作人工智慧流程的花費時間略低於使用傳統式機器學習畫布所需時間。分析第 7 位受測者使用本研究之單頁式機器學習畫布的使用時間較長的原因可能為：本實驗所設計的操作流程太簡單，可能會導致系統操作的熟悉程度影響程度無法完全消除，若進一步增加操作流程的複雜度與模型訓練所需時長，可能會使單頁式機器學習畫布之優勢更為顯著。為方便實驗數據蒐集，模型訓練時間也相對較短。

表 8 實驗受測者個人資料表

編號	性別	年齡	是否就讀資訊相關系所	是否使用過傳統式機器學習畫布	是否操作過機器學習流程	先使用何種流程
1	男	25	是	是	是	傳統式
2	男	24	是	是	是	單頁式
3	女	23	是	是	是	傳統式
4	男	24	是	是	是	單頁式
5	男	23	是	是	是	傳統式
6	男	24	是	否	是	單頁式
7	男	25	是	否	是	單頁式

8	男	22	是	否	是	單頁式
9	女	24	是	否	是	傳統式
10	男	24	是	否	是	傳統式
11	女	22	是	否	否	傳統式
12	女	22	是	否	否	傳統式
13	女	22	是	否	否	傳統式
14	女	22	是	否	否	傳統式
15	女	20	是	否	否	傳統式
16	男	22	是	否	否	單頁式
17	女	21	是	否	否	單頁式
18	女	21	是	否	否	單頁式
19	女	20	是	否	否	單頁式
20	女	20	是	否	否	單頁式

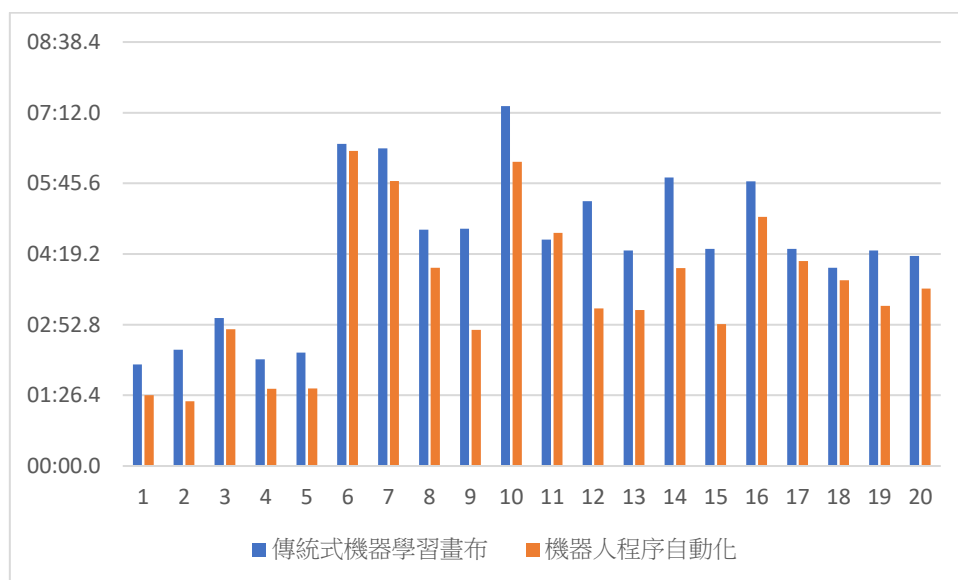


圖 28 使用者操作時間長條圖

本實驗所統計的數據顯示使用傳統機器學習畫布所花費平均時間為 4 分 28 秒，而使用本研究所開發的機器人程序自動化及決策規則資訊系統平均時間為 3 分 35 秒，相對於原系統所採用的傳統機器學習畫布所花費的時間約減少了 20%，證明採用機器人程序自動化能有效的降低使用者開發時間並提升開發效率。

若再將上述實驗結果分為已使用過傳統式機器學習畫布之受測者記為群體一（受測者編號 1-5），與未使用過傳統式機器學習畫布但有自行撰寫程式進行機器學習流程的經驗之受測者記為群體二（受測者編號 6-10），與皆無上述經驗

之受測者記為群體三（受測者編號 11-20）進行進一步分析，結果如表 9 所示。觀察所蒐集之實驗數據可知，三個群體中群體一，也就是已使用過傳統式機器學習畫布之受測者花費時間減少比率最高，分析此結果可能原因為：此群體受測者已熟悉元件操作方式，因此傳統式機器學習畫布跳頁與信息框等方式所增加的等待時間經由本研究之單頁式機器學習畫布可以得到最大程度的減少。群體二所減少之花費時間比率最少，可能之原因為：在觀察此群體使用者操作本系統時，此群體之受測者會在操作流程的同時觀察本系統之其他功能，相較於群體三僅直接按照設定之操作流程操作，一定程度上增加了操作時間，同時可能因此群體對機器學習流程操作有設計經驗，對傳統式機器學習畫布之操作也有一定程度上的理解，因此花費時間減少比率可能較低。

表 9 受測者分群數據表

群體	傳統式機器學習畫布平均使用時間	單頁式機器學習畫布平均使用時間	花費時間減少比率
群體一	2 分 23 秒	1 分 44 秒	27.3%
群體二	6 分 00 秒	5 分 02 秒	16.1%
群體三	4 分 45 秒	3 分 37 秒	20.4%

第五章 結論與未來展望




5.1 結論

本研究基於 SOA 建構了一個 AI 決策流程系統，通過引入機器人程序自動化，將原本複雜的 AI 與機器學習開發流程，透過流程的連結與 GUI 以直覺的參數設定方式簡化了開發步驟，降低使用者開發機器學習演算法的門檻。本系統也開發了批次處理功能，允許使用者在 RPA 中嘗試不同的資料轉化方式，並同時訓練多個演算法，能降低使用者反覆嘗試不同前處理流程、訓練參數與調整演算法的時間。同時，本研究亦實做了批次處理功能，使用者無需時刻專注於螢幕等待執行結果，待流程結束後本系統即會通過行動通訊系統以行動通訊軟體主動通知使用者，降低了使用者等待的時間，亦提升開發效率。此外，本系統也開發了決策規則資訊系統，使用者可以通過此系統引用其他使用者所開放的決策流程，並依照使用需求進行客制化調整，達到演算法開發流程再利用的效果，提高模型訓練的成功率，使複製成功的決策流程經驗成為可能。同時，為了讓使用者能更便利的尋找到合適的決策流程，本研究亦開發了決策規則推薦系統，提升使用者尋找決策流程的便利性。最後，本研究基於 SOA 所開發的核心服務，允許使用者通過組合不同的現有服務以建構應用程序，本核心服務已成功使用於建構智慧風場管理平台與數位學習平台[29]，證明了此核心服務應用於系統及演算法之開發的高彈性。

5.2 未來展望

目前本實驗之受測者皆擁有資訊相關背景且皆年齡分布多集中為 20-25 歲，在受測者資料蒐集的受測者背景層面上缺乏全面性，同時受測者年齡不夠廣泛可能會存在不同年齡層對新系統熟悉快慢之差異，此外本研究僅蒐集 20 筆測試資料，無法完全確認已消除受測結果之特異性。因此，未來若能針對不同背景及年齡層進一步蒐集更多測試資料，可能可以更全面地證明引入 RPA 及 SOA 能進一步提升系統使用效率。




同時，從相關文獻中可知，Azure 將所提供的元件分得更為細碎，這樣的設計可以提升流程設計之彈性，而本研究提供之功能元件則是以 KDD 流程為基礎進行元件切分，可引導缺乏實務經驗的使用者進行 AI 流程設計，但於一定程度上犧牲了細節調整的彈性，可能無法滿足專家調整流程的所有需求。因此，若能如 Azure 進一步將本系統之功能元件依據不同步驟進行切分，並同時保留本研究已提供之功能元件，即可滿足更多使用者的使用需求，觸及更多層面之使用者。

未來機器人程序自動化功能，可以進一步結合自然語言處理技術，通過對碩博士論文網的論文摘要進行爬蟲擷取並應用 NER、CRF、BERT 等模型對實驗流程進行分析，並自動匯入決策流程資訊系統，便可以在系統中引入成功的研究模型，增加可供選擇的演算法開發流程。此外，在資料轉化與演算法參數的選擇可以導入 AutoML 的方法，自動為資料選擇最合適的轉化方式與演算法，進一步減少使用者所需調整的參數，簡化開發流程。同時，決策流程資料還可以進一步結合圖資料庫，並對使用者所建立的流程進行分析，發掘其中的價值，並可以在使用者設計流程時提供參考建議，進一步達到引導使用者的效果。

參考文獻



1. 行政院, 臺灣AI 行動計畫核定本. 2018. Available from: <https://digi.ey.gov.tw/File/4C622B6A10053DAD>
2. 自由時報. 中研院士孔祥重: 台灣AI 人才遠不足需求. 2019 [cited 2020 4/2]; Available from: <https://news.ltn.com.tw/news/life/paper/1341254>.
3. Erl, T., *Service-oriented architecture: concepts, technology, and design*. Pearson Education India.
4. Fayyad, U., G. Piatetsky-Shapiro, and P.J.A.m. Smyth, *From data mining to knowledge discovery in databases*. 1996. **17**(3): p. 37-37.
5. Fayyad, U., G. Piatetsky-Shapiro, and P.J.C.o.t.A. Smyth, *The KDD process for extracting useful knowledge from volumes of data*. 1996. **39**(11): p. 27-34.
6. Frontender. *Data Mining*. 2020 [cited 2020 4/8]; Available from: <http://frontender.com/blog/enablers/data-mining/>.
7. García, S., J. Luengo, and F. Herrera, *Data preprocessing in data mining*. 2015: Springer.
8. 維基百科. 資料探勘. 2020 [cited 2020 4/8]; Available from: <https://zh.wikipedia.org/wiki/%E6%95%B0%E6%8D%AE%E6%8C%96%E6%8E%98>.
9. Microsoft. *What is Azure?* 2020 [cited 2020 5/15]; Available from: <https://azure.microsoft.com/en-us/overview/what-is-azure/>.
10. Microsoft. *What is Azure Machine Learning designer (preview)?* 2019 [cited 2020 5/15]; Available from: <https://docs.microsoft.com/en-us/azure/machine-learning/concept-designer>.
11. DataRobot. *How Do Model Blueprints Add Value to DataRobot?* 2018 2018/1/9 [cited 2020 5/15]; Available from: <https://blog.datarobot.com/data/model-blueprints-add-value-datarobot>.
12. Google. *AI and machine learning products*. [cited 2020 4/8]; Available from: <https://cloud.google.com/products/ai/>.
13. 林佳穎, 利用機器學習方法預測織布製程參數於虛實整合系統, in 工程科學及海洋工程學研究所. 2018, 國立臺灣大學. Available from: <https://hdl.handle.net/11296/8v6833>
14. van der Aalst, W.M., M. Bichler, and A. Heinzl, *Robotic process automation*. 2018, Springer.
15. Timbadia, D.H., et al. *Robotic Process Automation Through Advance Process Analysis Model*. in *2020 International Conference on Inventive Computation Technologies (ICICT)*. 2020. IEEE.

- 
16. Gao, J., et al. *Automated robotic process automation: A self-learning approach*. in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*. 2019. Springer.
 17. Lacity, M., L.P. Willcocks, and A. Craig, *Robotic process automation at Telefonica O2*. 2015.
 18. *Chapter 1: Service Oriented Architecture (SOA)*. 2016 [cited 2020 4/4]; Available from:
<https://web.archive.org/web/20160206132542/https://msdn.microsoft.com/en-us/library/bb833022.aspx>.
 19. Erl, T., *Service-oriented architecture*. 2005: Pearson Education Incorporated.
 20. 曾保彰. *服務導向架構(Service-Oriented Architecture , SOA) 簡介*. 2007 [cited 2020 4/4]; Available from:
http://www.cc.ntu.edu.tw/chinese/epaper/20070620_1008.htm.
 21. Elman, J.L.J.C.s., *Finding structure in time*. 1990. **14**(2): p. 179-211.
 22. Graves, A., N. Jaitly, and A.-r. Mohamed. *Hybrid speech recognition with deep bidirectional LSTM*. in *2013 IEEE workshop on automatic speech recognition and understanding*. 2013. IEEE.
 23. Schuster, M. and K.K.J.I.t.o.S.P. Paliwal, *Bidirectional recurrent neural networks*. 1997. **45**(11): p. 2673-2681.
 24. Devlin, J., et al., *Bert: Pre-training of deep bidirectional transformers for language understanding*. 2018.
 25. Mikolov, T., et al., *Efficient estimation of word representations in vector space*. 2013.
 26. Google. *Angular*. 2020 [cited 2020 5/20]; Available from: <https://angular.io/>.
 27. Facebook. *React*. 2020 [cited 2020 5/20]; Available from:
<https://reactjs.org/>.
 28. You, E. *Vue.js*. 2020 [cited 2020 5/20]; Available from: <https://vuejs.org/>.
 29. Lin, H.-Y., et al. *Service-Oriented Architecture for Intelligent Management with Data Analytics and Visualization*. in *2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)*. 2019. IEEE.
 30. *Flask*. [cited 2020 4/6]; Available from: <https://palletsprojects.com/p/flask/>.
 31. Lee, C.-Y., J.-Y. Lin, and R.-I. Chang, *Improve quality and efficiency of textile process using data-driven machine learning in industry 4.0*.
 32. 財團法人台灣網路資訊中心, *台灣網路報告*. 2019. Available from:
https://report.twnic.tw/2019/assets/download/TWNIC_TaiwanInternetReport_2019_CH.pdf
 33. wikipedia. *超文本傳輸安全協定*. 2020 [cited 2020 6/7]; Available from:
<https://zh.wikipedia.org/wiki/%E8%B6%85%E6%96%87%E6%9C%AC%E4>

[%BC%A0%E8%BE%93%E5%AE%89%E5%85%A8%E5%8D%8F%E8%A
E%AE.](#)

34. Apache. *APACHE HTTP SERVER PROJECT*. 2020 [cited 2020 6/7]; Available from: <https://httpd.apache.org/>.
35. 廖偉傑, 使用深度學習與虛實整合系統之工業數據分析用例推薦, in *工程科學及海洋工程學研究所*. 2020, 國立臺灣大學. p. 1-33.
36. Huang, Z., W. Xu, and K.J.a.p.a. Yu, *Bidirectional LSTM-CRF models for sequence tagging*. 2015.