

國立臺北科技大學

資訊工程系碩士班

碩士學位論文

基於偏差-變異數分解攻擊與鬆弛聚合機制
於強化抵禦對抗樣本攻擊的聯邦學習系統

Federated Learning System for Enhancing
Defense against Adversarial Attacks based on Bias-
Variance Decomposition Attack and Slack
Aggregation Mechanism

研究生：洪綵緹

指導教授：陳彥霖 博士

中華民國一百一十二年七月

國立臺北科技大學
研究所碩士學位論文口試委員會審定書

本校 資訊工程系 研究所 洪綠綿 君

所提論文，經本委員會審定通過，合於碩士資格，特此證明。

學位考試委員會

委員：_____

李鳳榮

金凱儀

陳秀雲

林玉哲

指導教授：陳秀雲

所長：劉建宏

中華民國 一百一十二年七月二十四日

摘要

論文名稱 : 基於偏差-變異數分解攻擊與鬆弛聚合機制於強化抵禦對抗樣本攻擊的聯邦學習系統

頁數 : 105 頁

校所別 : 國立台北科技大學 資訊工程系碩士班

畢業時間: 一百一十一學年度 第二學期

學位 : 碩士

研究生 : 洪綵緹

指導老師 : 陳彥霖 博士

關鍵字: 聯邦學習 對抗樣本攻擊 聯邦對抗訓練 偏差-變異數分解攻擊 鬆弛聚合機制

聯邦學習是一種分散式的機器學習系統，該系統意旨利用去中心化的數據來訓練一個中心化的全域模型，目前被大量應用於需要數據隱私保護的場景。然而，聯邦學習的資訊安全問題仍不可忽視，因其奠基於機器學習之上，當集中式機器學習模型被證明在測試階段會受到對抗樣本攻擊的破壞，由於聯邦學習同時異質性問題影響，對此攻擊更加敏感，因此如何兼顧異質性與提升全域模型的對抗強健性為一大考驗。

首先，本論文假設聯邦學習系統只有在測試階段才會受到對抗樣本攻擊的影響，致力於提升聚合後的全域模型對於未知的對抗白盒攻擊的強健性，為了達成此目的，提出一種新的聯邦對抗訓練演算法: FedBVA_SAT_Slack。首先，客戶端只需訓練自身的原始資料集。然後，先放置一份極小的輔助資料集，利用偏差-變異數分解攻擊生成對抗範例後，再進行對抗訓練。最後，利用鬆弛加權聚合機制，將客戶端模型與對抗訓練模型組合，聚合成全域模型。依 MNIST 資料集為例，FedBVA_SAT_Slack 方法在測試階段受到 FGSM 與 PGD 攻擊的準確度分別達到 61.6% 與 67.1%，相較於 2022 年提出的 Fed_BVA 聯邦對抗演算法，提升了 18.6% 與 39.3% 的準確度。

ABSTRACT

Title: Federated Learning System for Enhancing Defense against Adversarial Attacks based on Bias-Variance Decomposition Attack and Slack Aggregation Mechanism

Number of Pages: 105

School: National Taipei University of Technology

Time: July ,2023

Degree: Master

Researcher: Tsai-Ti Hung

Advisor: Yen-Lin Chen Ph.D.

Keywords: Federated Learning, Adversarial Examples, Federated Adversarial Learning, Bias-Variance Decomposition Attack, Slack Aggregation Mechanism

Federated Learning is a decentralized machine learning system that aims to train a centralized global model using decentralized data, and is currently widely used in scenarios requiring data privacy protection. However, the information security issues of federated learning cannot be overlooked. Since it is built upon machine learning, the centralized machine learning model has been proven to be vulnerable to adversarial sample attacks during the testing phase. Due to the heterogeneity issues in federated learning, it is even more sensitive to such attacks, making it a significant challenge to balance heterogeneity and enhance the adversarial robustness of the global model.

Initially, this paper assumes that federated learning systems are only affected by adversarial sample attacks during the testing phase. We strive to improve the robustness of the aggregated global model against unknown adversarial white-box attacks. To achieve this, we propose a new federated adversarial training algorithm: FedBVA_SAT_Slack. Firstly, clients only need to train their original datasets. Then, a very small auxiliary dataset is placed first, and

after generating adversarial examples using bias-variance tradeoff attacks, adversarial training is performed. Finally, the client model and the adversarial training model are combined using a relaxed weighted aggregation mechanism, aggregating into a global model. Taking the MNIST dataset as an example, the accuracy of the FedBVA_SAT_Slack method under FGSM and PGD attacks during the testing phase reached 61.6% and 67.1%, respectively, which is an increase of 18.6% and 39.3% in accuracy compared to the Fed_BVA federated adversarial algorithm proposed in 2022.



致謝

感謝我的指導教授陳彥霖老師，給予我接觸人工智慧安全領域的機會。在研究過程中，老師的指導和鼓勵激發了我對人工智慧和聯邦學習領域的興趣。我非常感激老師的諄諄指教，這將永遠留在我的心中。感謝大學長，總能在我遇到研究低潮與困難，給予我最大的支持與建言。

同時，感謝協助收集線寬資料集的機械系學長奕沁，以及兩位學弟建維和鐵哥。還有實驗室的兩位學妹若安和劉萱，沒有你們的協助，我無法完成這篇論文。同樣地，感謝世昕教導我電晶體資料集的實驗細節，沒有你，我也無法完成論文。

感謝所有碩二的同學，很珍惜與你們一起修課學習的時光，謝謝你們總是能包容程度較差的我，不會忘記一起求知、一起寫作業，然後被大家電爆的時光，從你們身上學習到太多，也謝謝你們能包容我的身體狀況，替我處理許多事情。

我要感謝我的家人，無條件尊重我北上就讀資工所的決定，對我的身體狀況給予極大的理解，並且給我許多治療的支持。最後，特別感謝同為資工系教授的父親，不僅常在做研究遇到困難給予我極大的幫助，這八個月還要每週載我到遠的要命醫院就診，很慶幸人生有段時光與您一起在同一個領域學習，永存於心。

洪綵緹 謹誌於

國立臺北科技大學 資訊工程碩士班

中華民國 一百一十二年七月二十日

目錄

摘要 -----	i
ABSTRACT -----	ii
致謝 -----	iv
目錄 -----	v
表目錄 -----	ix
圖目錄 -----	xi
第一章 緒論 -----	1
1.1 研究背景與動機 -----	1
1.2 研究目的 -----	3
1.3 論文架構 -----	4
第二章 文獻探討 -----	5
2.1 聯邦學習 -----	5
2.1.1 聯邦學習的運作機制與安全聚合原則 -----	5
2.1.2 FedAvg 演算法 -----	6
2.1.3 FedProx 演算法 -----	10
2.2 對抗樣本攻擊 -----	11
2.2.1 黑盒與白盒攻擊 -----	11
2.2.2 FGSM 攻擊 -----	12
2.2.3 PGD 攻擊 -----	12
2.3 對抗強健性 -----	13
2.3.1 對抗強健性 -----	13
2.4 對抗訓練 -----	14

2.4.1 對抗訓練-----	14
2.5 聯邦對抗訓練-----	15
2.6 基於偏差-變異數的聯邦對抗訓練(Fed_BVA)-----	16
2.7 鬆弛聯邦對抗訓練-----	19
第三章 研究方法 -----	21
3.1 定義問題-----	21
3.2 研究方法與設計-----	22
3.2.1 系統設計-----	22
3.2.2 演算法流程-----	24
3.2.3 演算法比較與分析-----	27
3.3 偏差-變異數分解攻擊的多樣性-----	29
3.4 中心伺服器進行對抗訓練-----	31
3.5 反向鬆弛聚合機制的設計-----	33
3.6 實驗資料集、深度學習模型與聯邦學習參數設定-----	35
3.6.1 MNIST 手寫數字資料集與 CNN 模型-----	36
3.6.2 MNIST 資料集的非獨立同分布設定與訓練過程-----	38
3.6.3 Fashion-MNIST 資料集與 VGG-11 模型-----	39
3.6.4 Fashion-MNIST 資料集的非獨立同分布設定與訓練過程-----	40
3.6.5 線寬預測資料集與 Auto-Encoder 模型-----	41
3.6.6 線寬資料集的非獨立同分布設定與訓練過程-----	43
3.6.7 電晶體資料集與 VAE 模型-----	45
3.6.8 電晶體資料集的非獨立同分布設定與訓練過程-----	46
第四章 實驗結果與分析-----	47
4.1 實驗設備與環境-----	47
4.2 實驗流程-----	48

4.3 實驗評估指標 -----	49
4.3.1 分類問題的評估指標-----	49
4.3.2 迴歸問題的評估指標-----	50
4.4 FedBVA_SAT_Slack 鬆弛聚合機制的實驗分析-----	51
4.4.1 鬆弛聚合機制的運作分析-----	51
4.4.2 是否啟用鬆弛聚合機制的實驗分析-----	53
4.4.3 Fed_BVA 與 FedBVA_SAT_Slack 輔助資料集的數量比較-----	55
4.5 MNIST 資料集的實驗結果與分析-----	56
4.5.1 FedAvg 聚合方法下的不同聯邦對抗訓練演算法的訓練過程 -----	57
4.5.2 FedAvg 聚合方法下的面對 FGSM 攻擊的對抗強健性比較-----	58
4.5.3 FedAvg 聚合方法下的面對 PGD 攻擊的對抗強健性比較-----	60
4.5.4 FedProx 聚合方法下的不同聯邦對抗訓練演算法的訓練過程-----	62
4.5.5 FedProx 聚合方法下的面對 FGSM 攻擊的對抗強健性比較-----	63
4.5.6 FedProx 聚合方法下的面對 PGD 攻擊的對抗強健性比較-----	64
4.6 Fashion-MNIST 資料集的實驗結果與分析 -----	66
4.6.1 FedAvg 聚合方法下的不同聯邦對抗訓練演算法的訓練過程 -----	66
4.6.2 FedAvg 聚合方法下的面對 FGSM 攻擊的對抗強健性比較-----	68
4.6.3 FedAvg 聚合方法下的面對 PGD 攻擊的對抗強健性比較-----	69
4.6.4 FedProx 聚合方法下的不同聯邦對抗訓練演算法的訓練過程-----	70
4.6.5 FedProx 聚合方法下的面對 FGSM 攻擊的對抗強健性比較-----	72
4.6.6 FedProx 聚合方法下的面對 PGD 攻擊的對抗強健性比較-----	73
4.7 線寬資料集的實驗結果與分析 -----	75
4.7.1 FedAvg 聚合方法下的不同聯邦對抗訓練演算法的訓練過程 -----	75
4.7.2 FedAvg 聚合方法下的面對 FGSM 攻擊的對抗強健性比較-----	77
4.7.3 FedAvg 聚合方法下的面對 PGD 攻擊的對抗強健性比較-----	78

4.7.4 FedProx 聚合方法下的不同聯邦對抗訓練演算法的訓練過程-----	80
4.7.5 FedProx 聚合方法下的面對 FGSM 攻擊的對抗強健性比較 -----	81
4.7.6 FedProx 聚合方法下的面對 PGD 攻擊的對抗強健性比較 -----	83
4.8 電晶體資料集的實驗與分析 -----	84
4.8.1 FedAvg 聚合方法下的不同聯邦對抗訓練演算法的訓練過程 -----	84
4.8.2 FedAvg 聚合方法下的面對 FGSM 攻擊的對抗強健性比較 -----	87
4.8.3 FedProx 聚合方法下的不同聯邦對抗訓練演算法的訓練過程 -----	89
4.8.4 FedProx 聚合方法下的面對 FGSM 攻擊的對抗強健性比較 -----	90
4.9 四大資料集實驗結果的統整分析 -----	92
第五章 結論與未來工作-----	97
5.1 結論 -----	97
5.2 未來工作-----	98
參考文獻-----	99
附錄-----	104

表目錄

表 3.1 Proposed method 與現有的聯邦對抗訓練演算法的技術比較	27
表 3.2 利用 BV-FGSM 每五個溝通輪次所生成的全域對抗範例	30
表 3.3 利用 PGD 攻擊之每五個溝通輪次所生成的對抗範例	30
表 3.4 聯邦學習實驗參數於四種資料集之設定	36
表 4.1 實驗平台之規格	47
表 4.2 混淆矩陣與元素定義	49
表 4.3 MNIST:有無進行鬆弛聚合機制的對抗強健性比較.....	54
表 4.4 MNIST: Fed_BVA 與 FedBVA_SAT_Slack 對抗強健性比較.....	55
表 4.5 Fashion-MNIST: Fed_BVA 與 FedBVA_SAT_Slack 對抗強健性比較.....	56
表 4.6 線寬資料集: Fed_BVA 與 FedBVA_SAT_Slack 對抗強健性比較.....	56
表 4.7 MNIST:不同聯邦系統最終全域模型之準確度與 F1-Score 比較(FedAvg).....	58
表 4.8 MNIST 資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedAvg) ...	59
表 4.9 各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedAvg).....	61
表 4.10 MNIST:不同聯邦系統最終全域模型之準確度與 F1-Score 比較(FedProx).....	62
表 4.11 MNIST: 各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedProx).....	64
表 4.12 MNIST: 各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedProx)	65
表 4.13 Fashion-MNIST:各聯邦系統最終全域模型之準確度與 F1-Score 比較(FedAvg)..	67
表 4.14 Fashion-MNIST:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedAvg) .	69
表 4.15 Fashion-MNIST:各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedAvg) ...	70
表 4.16 Fashion-MNIST:各聯邦系統最終全域模型之準確度與 F1-Score 比較(FedProx).	71
表 4.17 Fashion-MNIST:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedProx))	73
表 4.18 Fashion-MNIST:各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedProx) ...	74
表 4.19 線寬資料集:各聯邦系統最終全域模型之 R-squared 比較(FedAvg)	76

表 4.20 線寬資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedAvg).....	78
表 4.21 線寬資料集:各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedAvg)	79
表 4.22 線寬資料集:各聯邦系統最終全域模型之準確度與 F1-Score 比較(FedProx).....	80
表 4.23 線寬資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedProx).....	82
表 4.24 線寬資料集:各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedProx).....	83
表 4.25 電晶體資料集:各聯邦系統最終全域模型之 R-squared 比較(FedAvg).....	85
表 4.26 電晶體資料集: Fed_BVA 與 FedBVA_SAT_Slack 對抗強健性比較(FedAvg).....	86
表 4.27 電晶體資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedAvg)....	88
表 4.28 電晶體資料集:各聯邦系統最終全域模型之準確度與 F1-Score 比較(FedProx)...	89
表 4.29 電晶體資料集:各聯邦對抗訓練系統對 FGSM 攻擊的 R-平方比較表(FedProx)..	91
表 4.30MNIST 資料集的實驗結果:以準確度為指標	93
表 4.31Fashion-MNIST 資料集的實驗結果:以準確度為指標	93
表 4.32 線寬資料集的實驗結果:以 R-平方為指標	94
表 4.33 電晶體資料集的實驗結果:以 R-平方為指標	95

圖目錄

圖 2.1 水平式聯邦學習系統的運作機制[26]	6
圖 2.2 FedSGD 聚合演算法[3]	7
圖 2.3 FedAvg 聚合演算法[3].....	8
圖 2.4 基於 MNIST-IID 的 FedSGD 與 FedAvg 演算法收斂比較圖[3].....	9
圖 2.5 FedProx 聚合演算法[31].....	11
圖 2.6 FGSM 的經典案例[33]	12
圖 2.7 聯邦對抗訓練(FAT)示意圖	16
圖 2.8 Fed_BVA 演算法[20]	18
圖 2.9 Slack Federated Adversarial Training 演算法[50].....	20
圖 3.1 聯邦學習基於對抗樣本攻擊威脅的階段	21
圖 3.2 FedBVA_SAT_Slack 系統時序圖	23
圖 3.3 FedBVA_SAT_Slack 演算法之參與客戶的訓練系統示意圖	25
圖 3.4 FedBVA_SAT_Slack 演算法之中心伺服器的對抗訓練防禦系統示意圖	25
圖 3.5 FedBVA_SAT_Slack 演算法之中心伺服器的聚合系統示意圖	25
圖 3.6 FedBVA_SAT_Slack 聯邦對抗訓練演算法	26
圖 3.7 FedBVA_SAT_Slack 聯邦對抗訓練框架:伺服器進行對抗訓練的流程圖	32
圖 3.8 FedBVA_SAT_Slack 聯邦對抗訓練框架: 伺服器進行對抗訓練的演算法	32
圖 3.9 FedBVA_SAT_Slack 聯邦對抗訓練框架：反向鬆弛聚合機制的流程圖	34
圖 3.10 FedBVA_SAT_Slack 聯邦對抗訓練框架：反向鬆弛聚合機制演算法	35
圖 3.11 MNIST 資料集	37
圖 3.12 MNIST 手寫數字辨識實驗: CNN 模型架構	38
圖 3.13 MNIST 資料集在聯邦學習之五個客戶端的非獨立同分布狀況	38
圖 3.14 Fashion MNIST 資料集	39

圖 3.15 Fashion MNIST 資料集分類實驗: VGG11 模型架構[54].....	40
圖 3.16 第一個客戶端在聯邦學習 Fashion-MINST 資料集實驗的各類別資料量	40
圖 3.17 Fashion-MNIST 資料集在聯邦學習之五個客戶端的非獨立同分布狀況	41
圖 3.18 線寬預測資料集的收集流程圖	41
圖 3.19 線寬資料集	42
圖 3.20 Auto-Encoder: 自編碼器模型[56].....	42
圖 3.21 線寬資料集的預測模型	43
圖 3.22 線寬資料集之十個客戶端的訓練資料量長條圖	44
圖 3.23 線寬資料集在聯邦學習之十個客戶端的非獨立同分布狀況.....	44
圖 3.24 a-IGZO TFT 的模擬橫切面視圖[25].....	45
圖 3.25 Variational Auto-Encoder: 變分自動編碼器模型[56]	45
圖 3.26 電晶體資料集的訓練過程[25]	46
圖 3.27 電晶體資料集在聯邦學習之三個客戶端的非獨立同分布狀況.....	46
圖 4.1 實驗流程圖	49
圖 4.2 MNIST 資料集: FedBVA_SAT_Slack 之所有模型的損失值折線圖(FedAvg)	52
圖 4.3 MNIST 資料集: FedBVA_SAT_Slack 之所有模型的損失值折線圖(FedProx)	52
圖 4.4 MNIST:是否執行鬆弛聚合機制的準確度比較(FedAvg)	53
圖 4.5 MNIST:是否執行鬆弛聚合機制的準確度比較(FedProx)	54
圖 4.6 MNIST:不同聯邦對抗訓練系統的聚合過程損失折線圖(FedAvg)	57
圖 4.7 MNIST 資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedAvg)	59
圖 4.8 MNIST:各聯邦對抗訓練系統對 PGD 攻擊強度的準確度折線圖(FedAvg)	61
圖 4.9 MNIST:不同聯邦對抗訓練系統的聚合過程損失折線圖(FedProx)	62
圖 4.10 MNIST: 各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedProx)	64
圖 4.11 MNIST:各聯邦對抗訓練系統對 PGD 攻擊的準確度折線圖(FedProx)	65
圖 4.12 Fashion-MNIST:各聯邦對抗訓練系統的聚合過程損失折線圖(FedAvg)	67

圖 4.13 Fashion-MNIST:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedAvg) .	68
圖 4.14 Fashion-MNIST:各聯邦對抗訓練系統對 PGD 擊的準確度折線圖(FedAvg)	69
圖 4.15 Fashion-MNIST:各聯邦對抗訓練系統的聚合過程損失折線圖(FedProx)	71
圖 4.16 Fashion-MNIST:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedProx)	72
圖 4.17 Fashion-MNIST:各聯邦對抗訓練系統對 PGD 擊的準確度折線圖(FedProx)	74
圖 4.18 線寬資料集:各聯邦對抗訓練系統的聚合過程損失折線圖(FedAvg)	76
圖 4.19 線寬資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedAvg).....	77
圖 4.20 線寬資料集:各聯邦對抗訓練系統對 PGD 攻擊強度的準確度折線圖(FedAvg) ..	79
圖 4.21 線寬資料集:各聯邦對抗訓練系統的聚合過程損失折線圖(FedProx).....	80
圖 4.22 線寬資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedProx).....	82
圖 4.23 線寬資料集:各聯邦對抗訓練系統對 PGD 攻擊的準確度折線圖(FedProx).....	83
圖 4.24 電晶體資料集:各聯邦對抗訓練系統的聚合過程損失折線圖(FedAvg)	85
圖 4.25 電晶體資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedAvg)....	88
圖 4.26 電晶體資料集:各聯邦對抗訓練系統的聚合過程損失折線圖(FedProx).....	89
圖 4.27 電晶體資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedProx)...	91

第一章 緒論

1.1 研究背景與動機

在資料共享的大數據時代下，個人的資料隱私問題也逐漸被世人重視。然而人工智能技術的一大挑戰，就是對資料的依賴性極強，若沒有完整且廣泛的資料，將大幅限縮機器學習與深度學習模型的泛用性[1]。因此，如何解決個人資料隱私的同時，也能保持模型泛用性，成為不容忽視的議題[2]。雪上加霜地，2016 年，歐盟通過 GDPR(General Data Protection Regulation)法案，該法條明確規定，所有與個人相關的數據都必須經過數據持有者的明確授權和同意才能夠使用。同樣地，台灣也於 2018 年通過了《資通安全管理法》，這讓資料共享變得更加困難。因此，聯邦學習(Federated Learning)[3]作為一種解決方案應運而生。

聯邦學習是一種分散式的機器學習策略，其主要由參與學習的客戶端（邊端裝置）以及負責聚合不同客戶端模型的中心伺服器所組成。在聯邦學習的框架中，各個客戶端負責訓練自身的模型，並且在訓練完成後，它們無需上傳任何訓練資料至中心伺服器，僅需傳送已訓練完成的模型參數。這種策略能在保護用戶敏感資料不被外洩的前提下，使系統從多個不同的客戶端學習，進而達到高度的泛用性，同時能保護客戶端的敏感資料。特別是在高度重視隱私性的醫學科技產業中，聯邦學習已經取得了不少的成功[4]。

不僅在醫療產業中，聯邦學習在工業領域受到重視[5-7]。隨著工業 4.0 和物聯網 (Internet of Things)的蓬勃發展，許多智慧工廠的邊端裝置多應用機器學習與深度學習技術。然而，由於這些邊緣裝置的敏感性，資料共享成為了一個重大的挑戰。此時，聯邦學習提供了一種解決方案[8, 9]:邊端裝置可以直接使用自身收集的資料來進行訓練，不僅可以防止機敏資料的洩露，同時也讓邊緣裝置保有訓練模型的主導權。即使有單一裝置決心脫離訓練團隊，聯邦學習的模型聚合事宜也不會中斷，因為聯邦學習不依賴特定客戶端的資料，新的邊端設備也可以隨時加入訓練。

聯邦學習旨在利用去中心化的數據來訓練一個中心化的全域模型，並且保證了客戶端原始資料的隱私性。然而，聯邦學習仍面臨三大挑戰[10]: 客戶端與中心伺服器的高昂溝通成本、客戶端的數據異質性與系統安全問題。針對聯邦學習的系統安全，可以分成三大階段進行分析。

首先，在客戶端的訓練階段，可能遭到後門攻擊[11]。由於聯邦學習的全域模型泛用性與客戶端之間的關係緊密，若有大量的客戶端訓練過程遭到破壞，將會對聯邦學習系統帶來嚴重的損壞。例如，Bagdasaryan 在研究中發現，即使只有少數的客戶端被攻擊，也可以在全域模型中引入後門攻擊，會造成嚴重的影響[12]。值得注意的是，有一些專門防禦後門攻擊的防禦策略已經被提出，如 Xie 等人所提出的 Byzantine SGD 防禦策略[13]、Cao 等人提出的 FLTrust 防禦策略[14]與 Pan 等人所提出的 Justinian's GAAvernor 防禦機制[15]等。

接著，客戶端與中心伺服器進行雙向的模型傳輸過程中，可能會被竊聽，從而窺探到客戶端的模型參數[16]。雖然模型參數相對於原始資料更難破解，但仍存在被駭客破解，從而恢復原始數據集和訓練超參數的風險。為了因應這一問題，學者們提出了許多防禦策略。例如，Kang 等人提出了一種基於差分隱私的方法，在聯邦學習進行模型聚合前，先添加高斯噪聲來保護模型參數[17]，從而防止數據洩露。另外，還有 Hardt 等人將同態加密技術應用於聯邦學習中[18]，可以有效保護聯邦學習的伺服器與客戶端之間的溝通安全性。

最後，在聯邦學習的全域模型測試階段(testing phase)，暴露著遭受對抗樣本攻擊的破壞。當集中式機器學習模型被證明在測試階段有被攻擊的風險[19, 20]，由於聯邦學習同時深受資料異質性的影響，使得對抗樣本攻擊對其的威脅更為劇烈，嚴重影響系統穩定性。

現今大多數的聯邦學習研究多著重於前二階段的攻防策略[21]，卻忽略了對抗樣本攻擊於測試階段也有長遠的影響[22, 23]。因此，本論文致力於提升對抗樣本攻擊之聯邦學習的最終全域模型的對抗強健性。

1.2 研究目的

本研究開發了一種全新的聯邦對抗訓練演算法:FedBVA_SAT_Slack，致力於提升聯邦學習系統全局收斂後，在測試階段抵禦未知的對抗樣本攻擊威脅的能力，亦即提升聯邦學習系統的全局強健性。

本論文的主要貢獻如下：

1. 著重在非獨立同分布(non-IID)[24]的狀況下進行研究：在許多現實應用中，各客戶端的數據分佈往往並不完全相同，也就是說，數據並不遵循獨立同分布(IID)的假設。因此，本研究特別著重在非 IID 的情況下，探討新提出的 FedBVA_SAT_Slack 演算法如何有效地對抗樣本攻擊，以增強聯邦學習系統的全局強健性。在實際的應用場景中，非獨立同分布(non-IID)的狀況更為常見，對於理解並改進聯邦學習系統在真實世界中的效能發揮著關鍵作用。
2. 提出在聯邦學習的中心伺服器進行對抗訓練的策略：在中心伺服器裡新增一個對抗訓練模組，專門負責對全域對抗範例進行對抗訓練；如此一來，可以減輕參與客戶端實施對抗訓練的成本。事實上，對抗訓練需要相當的運算資源[24]，然而在現實環境中，部分客戶端設備可能無法負擔進行對抗訓練的運算需求。
3. 運用反向鬆弛聚合機制，以協調客戶端模型與對抗訓練模型的平衡：由於乾淨數據的準確度與對抗樣本攻擊的強健性之間往往存在權衡關係，如何找到二者間的最佳平衡點便成為一項重大挑戰。透過改良後的鬆弛聚合機制，來協調代表乾淨數據集的客戶端模型與象徵對抗樣本攻擊強健性的對抗訓練模型之間的平衡，讓全域模型在乾淨準確度與對抗強健性盡可能達成平衡的狀態。
4. 驗證演算法的效能：使用四個資料集進行實驗，共有兩個公開的分類資料集：MNIST[25]與 Fashion-MNIST 資料集[26]，以及兩個實際工業資料集：線寬資料集[27]與電晶體資料集[28]，來驗證 FedBVA_SAT_Slack 的有效性，並與現有的其他聯邦對抗訓練演算法進行比較。

總而言之，本研究試圖解決聯邦學習系統面對未知的對抗樣本攻擊時的安全性與穩

定性問題，並提出新的聯邦對抗訓練演算法，希望能為人工智慧資訊安全領域注入全新的視角。

1.3 論文架構

本論文分成五個章節：第一章為緒論；第二章為文獻回顧，介紹對抗樣本攻擊、對抗訓練、聯邦學習聚合方法與聯邦對抗訓練的重要主題；第三章為研究方法，詳細介紹新的演算法 FedBVA_SAT_Slack 的運作機制，以及介紹用於驗證該演算法有效性的四種資料集和深度學習模型；第四章為實驗結果與分析，介紹實驗所使用的實驗平台軟硬體環境、並展示改良後的鬆弛聚合機制的實驗結果，同時將其與其他現有的聯邦對抗訓練演算法做出比較；第五章為結論與未來工作，將總結本論文的主要發現並討論未來可能的研究方向。



第二章 文獻探討

本章節著重於介紹本研究中的相關技術與文獻，將從聯邦學習的安全問題、對抗樣本攻擊、對抗樣本攻擊的防禦，直到聯邦學習之對抗樣本攻擊的防禦技術，進行全面地探討。

2.1 聯邦學習

聯邦學習(Federated Learning)，又稱為聯合學習、聯邦式學習，是一種分散式機器學習的框架，利用了去中心化資料來訓練聚合出中心化的全域模型，從而保證了客戶端原始資料的隱私性，使聯邦學習成為當今在異質性問題中，最熱門的分散式機器學習技術。接下來，將介紹聯邦學習的運作機制與安全聚合原則，以及兩大聯邦學習系統的聚合方法: FedAvg 與 FedProx，其中 FedProx 是一種專門為解決聯邦學習中異質性問題而設計的聚合方法。

2.1.1 聯邦學習的運作機制與安全聚合原則

聯邦學習的運作機制包含以下四個主要步驟，如圖 2.1 所示：

1. 客戶端更新(Client Update)：參與客戶端利用各自持有的資料集，來獨立訓練模型。
2. 前向溝通(Forward Communication)：參與客戶端將訓練完成的模型參數，上傳至中心伺服器，無須將參與客戶端所持有的原始資料集上傳。
3. 伺服器聚合(Server Aggregate)：當中心伺服器接收到所有客戶端的模型權重後，會運用聚合演算法，將所有參與客戶端的模型們，聚合成一個全域模型。
4. 反向溝通(Backward Communication)：當全域模型聚完成後，將全域模型回傳給所有參與客戶端。客戶端利用該全域模型進行下一輪的訓練，這個過程會不斷重複，直到全域模型達到收斂且穩定的狀態為止。

在聯邦學習的實踐中，以上述四個步驟完成一輪，稱為一個溝通輪次(communication round)。透過多輪的溝通輪次，讓全域模型將逐步優化和調整，提升預測準確性。

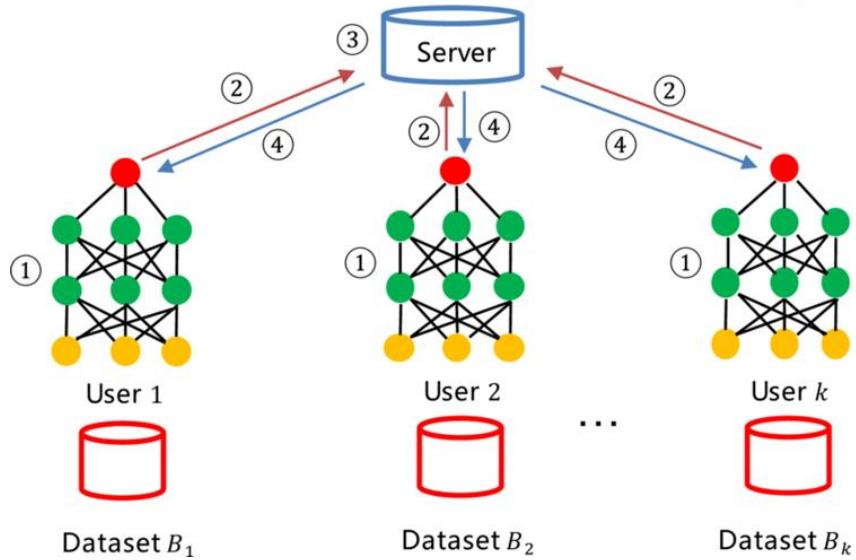


圖 2.1 水平式聯邦學習系統的運作機制[29]

在聯邦學習系統中，為了達成保護參與客戶端資料的隱私，必須嚴格遵守安全聚合原則(Secure Aggregation)[30-32]：

1. 客戶端資料不外流[3]：在這個原則下，每個客戶端的資料都只會在其裝置上進行處理，絕不會被傳送到其他地方，包括中心伺服器。換句話說，模型的訓練過程會在每個客戶端本地進行，並只有模型的權重（訓練完成的模型參數）會被傳送到中心伺服器以進行聚合，這就是第二個步驟：前向溝通(Forward Communication)的核心價值。
2. 客戶端與中心伺服器間，不直接傳輸訓練資料[31]：在聯邦學習的運作中，客戶端與中心伺服器之間絕不可涉及任何實際的訓練資料傳輸，而僅僅是模型權重和更新的交換。這種方式可以大大降低數據洩漏的風險。
3. 客戶端資料不共享[3]：每個客戶端的數據都只供自身使用，並不會與其他客戶端共享。這不僅保護了數據的隱私，也避免了不必要的資訊交換。

因此，無論在何種情況下，聯邦學習系統都必須嚴格遵守這些安全聚合的原則，以確保資料的隱私與安全。

2.1.2 FedAvg 演算法

Federated Averaging(FedAvg)[3]，由 Google 研究團隊於 2017 年提出，是一種在分散

式數據下訓練深度學習模型的演算法。意旨解決移動設備上進行機器學習的隱私與效率問題，可以在不將數據傳輸到中心伺服器的狀況下，聚合出全域模型。FedAvg 透過將模型訓練推送到客戶端，並使用加權平均來整合所有客戶端上傳至中心伺服器的模型參數，實現了分散式學習與客戶端資料的隱私保護。

介紹 FedAvg 演算法之前，必須先了解它的前身 FedSGD。FedSGD[3]是一種基於隨機梯度下降(Stochastic Gradient Descent)的分散式學習演算法，由客戶端負責執行模型訓練，且每次訓練時必須讓所有資料參與更新。因此，只要每更新一次梯度(gradient)，就必須將模型參數上傳至中心伺服器進行聚合。然而，此機制需要執行大量的溝通輪次(Communication rounds)，才能讓全域模型趨於穩定的狀態。圖 2.2 為 FedSGD 聚合演算法，可發現尚未有批次(batch)訓練的概念。

FedSGD Algorithm 1 FederatedSGD

```

1: procedure SERVER
2:   initialize  $\mathbf{w}_0$ 
3:   for  $t = 0; 1; 2; \dots$  do
4:     for all  $k$  in the  $K$  nodes in parallel do
5:        $\mathbf{w}_{t+1}^k \leftarrow \text{ClientUpdate}(k, \mathbf{w}_t)$ 
6:     end for
7:      $\mathbf{w}_{t+1} = \sum_{k=1}^K \frac{n_k}{n} \mathbf{w}_{t+1}^k$ 
8:   end for
9: end procedure
10: procedure CLIENTUPDATE( $k, w$ )
11:    $\mathcal{B} \leftarrow$  split  $\mathcal{P}_k$  to set of batches
12:   for all  $b \in \mathcal{B}$  do
13:      $\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla f(\mathbf{w}, b)$ 
14:   end for
15:   return  $\mathbf{w}$ 
16: end procedure

```

圖 2.2FedSGD 聚合演算法[3]

FedAvg 演算法為了解決溝通成本過高的問題，進行了相應的改良，引入了資料批次(batch size)的概念。該算法讓客戶端在完成一定次數的更新後，才將模型參數上傳至中心伺服器。這種方法能夠大幅降低客戶端與中心伺服器之間的溝通輪次。如圖 2.2 所示，FedAvg 聚合演算法已經加入了資料批次訓練的概念。

Algorithm 1 FederatedAveraging. The K clients are indexed by k ; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:

```

initialize  $w_0$ 
for each round  $t = 1, 2, \dots$  do
     $m \leftarrow \max(C \cdot K, 1)$ 
     $S_t \leftarrow$  (random set of  $m$  clients)
    for each client  $k \in S_t$  in parallel do
         $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$ 
     $w_{t+1} \leftarrow \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k$ 

```

```

ClientUpdate( $k, w$ ): // Run on client  $k$ 
     $\mathcal{B} \leftarrow$  (split  $\mathcal{P}_k$  into batches of size  $B$ )
    for each local epoch  $i$  from 1 to  $E$  do
        for batch  $b \in \mathcal{B}$  do
             $w \leftarrow w - \eta \nabla \ell(w; b)$ 
    return  $w$  to server

```

圖 2.3 FedAvg 聚合演算法[3]

介紹 FedAvg 演算法中的幾個重要的參數:

1. C: 代表每回合中參與訓練的客戶端比例，即每回合從所有客戶端中隨機選取一部分客戶端進行訓練。
2. K: 代表客戶端總數。
3. E: 表示客戶端在一個回合中訓練模型的次數。
4. B: 表示客戶端在一次本地更新中使用的樣本數量。
5. η: 表示客戶端的訓練學習率。

首先，在中心伺服器進行兩大步驟:

1. 初始化參數
2. 每一個溝通回合(communication round)隨機選取 m 位客戶端進行聚合，並且所有客戶端都必須進行本地訓練，更新本地的 $w_{t,k}$ 得到 $w_{t+1,k}$ 。所有客戶端更新結束後，將 $w_{t,k}$ 上傳至中心伺服器，伺服器將以客戶端資料量作為聚合依據，聚合出最新的全域模型參數 w_t 。

接著，探討客戶端的本地訓練機制:

1. 所有客戶端將各自的訓練資料，將訓練資料以 B 資料大小分段。
2. 針對每段資料，進行 E 輪更新，透過本地客戶端模型訓練的損失函數，乘以固定的

學習率 η ，計算出本地更新後的模型權重 $w_{t+1,k}$ ， k 代表第 k 位客戶。

$$w_{t+1,k} = w_{t,k} - \eta \nabla F_k(w) \quad (2.1)$$

3. 將本地更新後的 w ，上傳至中心伺服器，伺服器將以客戶端資料量作為權重依據，聚合出最新的全域模型參數 w_t 。
4. 當中心伺服器聚合出最新的全域模型後，客戶端收到將收到更新後的全域模型參數，並進行新的本地訓練，直到全域模型收斂為止。

最後，探究中心伺服器的聚合機制，聚合公式如下，其中 n 為所有參與客戶的資料總量， n_k 代表第 k 位客戶的資料量。

$$w_t = \sum_{k=1}^k \frac{n_k}{n} w_{t,k} \quad (2.2)$$

最後，在 FedAvg 演算法的論文中[3]，比較 FedSGD 演算法與 FedAvg 演算法的全域模型收斂速度。結果顯示，FedAvg 演算法在溝通輪次小於 100 次時，便達到收斂狀態。相對地，FedSGD 演算法（表示為藍色實線），則需要超過 800 次的溝通才能達到收斂。因此，這項比較結果清楚地顯示，聯邦學習的 FedAvg 演算法能大幅降低溝通成本。

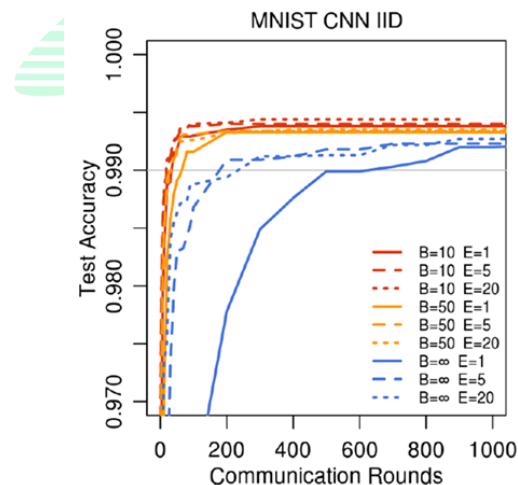


圖 2.4 基於 MNIST-IID 的 FedSGD 與 FedAvg 演算法收斂比較圖[3]

FedAvg 聚合演算法雖提升了收斂效率，但目前只能處理同質分佈(IID)數據，並尚未對非同質分佈(non-IID)的資料進行優化[33]。實際應用層面上，不同客戶端的數據分布必定存在差異，這會導致全域模型參數無法充分地學習所有客戶端的特徵。例如，在一個影像分類任務中，某些客戶端可能只包含某些特定類別的圖像，而其他客戶端可能

包含其他類別的圖像。在這種情況下，全域模型參數可能無法充分地反映所有類別的特徵，從而導致模型的性能(準確度)下降。目前，FedAvg 只能處理同質分佈(IID)數據，也讓針對非同質分佈(non-IID)數據優化演算法成為聯邦式學習的一大熱門研究方向。

2.1.3 FedProx 演算法

FedProx[34] 是一種聯邦學習演算法，旨在解決聯邦學習的異質性問題。這種異質性問題可以分為系統異質性(systems heterogeneity)和統計異質性(statistical heterogeneity)進行探討。系統異質性指的是現實中不同客戶端之間的設備差異，例如存儲、計算和通訊能力等方面的差異，設備的差異會對中心伺服器的模型聚合穩定度產生影響。而統計異質性意旨不同客戶端之間的資料分布和特徵可能存在差異，例如不同設備上的資料採集方式、資料特徵、資料量等方面上的差異，同義於非同質分佈(non-IID)。

FedProx 演算法引入了 proximal term(近端條件)，讓聯邦學習系統除了追求最小化損失函數 $F_k(\cdot)$ ，也考慮了本地與全域模型的距離，使得參數更新更加平滑，從而提高模型的準確性和泛化能力。以下加入 proximal term 的目標函數公式 h_k :

$$\min_w h_k(w; w^t) = F_k(w) + \frac{\mu}{2} \|w - w^t\|^2 \quad (2.3)$$

其中 w 為第 k 位參與客戶的本地模型， w^t 為第 t 輪溝通聚合的全域模型，利用 proximal term: $\frac{\mu}{2} \|w - w^t\|^2$ ， μ 為介於 0 到 1 之間的隨機變數；透過 proximal term 控制本地模型與全域模型的距離，也可以達到加速系統收斂。最後，FedProx 演算法建立在 FedAvg 之上，故中心伺服器進行模型聚合的運作機制與 FedAvg 演算法相同，中心伺服器的聚合演算法同樣以參與客戶端的資料量作為加權依據，FedProx 演算法之中心伺服器的聚合算法與公式 $wt = \sum_{k=1}^k \frac{n_k}{n} w_{t,k}$ (2.2)相同。

Algorithm 2 FedProx (Proposed Framework)

Input: $K, T, \mu, \gamma, w^0, N, p_k, k = 1, \dots, N$
for $t = 0, \dots, T - 1$ **do**
 Server selects a subset S_t of K devices at random (each device k is chosen with probability p_k)
 Server sends w^t to all chosen devices
 Each chosen device $k \in S_t$ finds a w_k^{t+1} which is a γ_k^t -inexact minimizer of: $w_k^{t+1} \approx \arg \min_w h_k(w; w^t) = F_k(w) + \frac{\mu}{2} \|w - w^t\|^2$
 Each device $k \in S_t$ sends w_k^{t+1} back to the server
 Server aggregates the w 's as $w^{t+1} = \frac{1}{K} \sum_{k \in S_t} w_k^{t+1}$
end for

圖 2.5 FedProx 聚合演算法[34]

2.2 對抗樣本攻擊

隨著人工智慧技術的蓬勃發展，雖然機器學習模型或深度網路模型具有良好的泛化能力，但也具有極高脆弱性，以分類為例：只要在輸入樣本內加入人眼難以觀察的細微擾動(perturbation)，模型會以高置信度給出一個錯誤的輸出。「由於數據在 Federated Learning 中的不可訪問性，使得保護數據免受對抗樣本攻擊變得更加困難。」[35]；這說明了本質亦為深度學習的聯邦學習系統，由於深受客戶端的異質性問題影響，對於對抗樣本攻擊更為敏感，也開啟了全新的研究領域大門；常見的對抗攻擊為 FGSM(Fast Gradient Sign Method)[24]與 PGD(Projected Gradient Descent)[36]，將針對這兩種攻擊進行探討。

2.2.1 黑盒與白盒攻擊

對抗樣本攻擊(Adversarial Examples)一般分成黑盒攻擊與白盒攻擊，黑盒攻擊意指攻擊者在未知模型真實樣貌的情況下，只破壞資料集內容，例如：單像素攻擊(One-Pixel Attack)[37]、邊界攻擊(Boundary Attack)；白盒攻擊則表示攻擊者完全得知模型的內部結構與參數，並透過這些資訊，生成出強大的對抗樣本攻擊，例如：Fast Gradient Sign Method(FGSM)[38]與 Projected Gradient Descent(PGD)[36]。因此，白盒攻擊的破壞性遠

大於黑盒攻擊，基於白盒攻擊的威脅性，本文針對白盒攻擊進行探討。

2.2.2 FGSM 攻擊

Fast Gradient Sign Method(FGSM)攻擊為最早被發明的對抗樣本攻擊之一，於 2015 年由伊恩·古德費洛(Ian J. Goodfellow)等人提出[36]，此攻擊透過竊取來的模型梯度來改變輸出目標函數(loss function)，FGSM 會在原始樣本的基礎上添加一個小的擾動，使得模型對該樣本進行錯誤的預測。

最有名的案例為，在熊貓(pandas)圖片加上 epsilon 僅 0.07 的細微擾動，卻造成深度模型將熊貓(pandas)辨識為長臂猿(gibbon)，很明顯是錯誤輸出，如圖 2.6 所示。

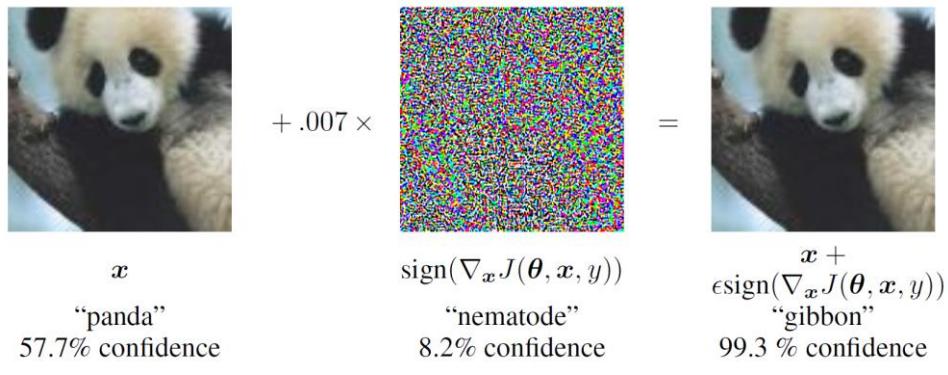


圖 2.6 FGSM 的經典案例[36]

FGSM 攻擊是透過竊取的模型梯度進行，首先利用 sign function 修改模型梯度的方向，並使用 ϵ (epsilon)來控制擾動向量的大小； θ 為模型參數、 x 為原始輸入樣本、 y 為輸出標籤，且 $J(\theta, x, y)$ 為損失函數，可以用公式 $\eta = \epsilon sign(\nabla_x J(\theta, x, y))$ (2.4)來表示 FGSM 攻擊所生成的擾動 η 。

$$\eta = \epsilon sign(\nabla_x J(\theta, x, y)) \quad (2.4)$$

最後，加上擾動後的輸入樣本(x^{adv})可以表示為公式 $x^{adv} = x + \epsilon \cdot sign(\nabla_x J(\theta, x, y))$ (2.5):

$$x^{adv} = x + \epsilon \cdot sign(\nabla_x J(\theta, x, y)) \quad (2.5)$$

2.2.3 PGD 攻擊

FGSM 攻擊為單次攻擊(One-time attack)，而 Project Gradient Descent(PGD)則是迭代攻擊(Iterative attack)，兩者皆是基於梯度變化的對抗樣本攻擊。可以將 PGD 視為多次迭代版本的 FGSM 攻擊[36]。PGD 攻擊於每次迭代時，是透過梯度計算出來的擾動，投影至 l_∞ -ball 範圍內[39]，若超過預設範圍，必須將其映射回規定範圍內。讓損失函數在一定範圍內，可以有效地讓擾動更接近原始樣本，更不易從人眼察覺攻擊的存在。以下兩個公式可以更理解 FGSM 與 PGD 攻擊之間的差異。

FGSM 為公式(2.6)， ε 為 epsilon:

$$x + \varepsilon \operatorname{sgn}(\nabla_x L(\theta, x, y)) \quad (2.6)$$

PGD 為公式(2.7)， t 為 t 時刻的輸入梯度， Π_{x+s} 代表必須落在限制範圍 S 內。

$$x^{t+1} = \Pi_{x+s} (x^t + \varepsilon \operatorname{sgn}(\nabla_x L(\theta, x, y))) \quad (2.7)$$

由於 PGD 攻擊是多次迭代攻擊，一般認為比 FGSM 攻擊更加強大[36]，對於非線性模型效果會更加顯著；但如果模型為線性狀態，FGSM 攻擊效果會較好，因為損失函數(loss function)的下降方向即使經過多次迭代，也不會有任何改變。雖然 PGD 攻擊非常簡單且有效，但該攻擊的計算效率卻是一大問題，因為 PGD 攻擊需要在每次迭代步驟中求解一個最佳化問題，所以需要消耗更大的計算資源。

2.3 對抗強健性

對抗強健性是指深度學習模型在面對對抗樣本攻擊時，是否仍能保持良好性能的指標，提升模型的對抗強健性(Adversarial Robustness)可以增強模型的安全性與可靠性，並有助於防止惡意攻擊與資料竄改。

2.3.1 對抗強健性

對抗強健性(Adversarial Robustness)[40]是指深度學習模型應付對抗樣本攻擊的能力，若某分類問題的深度學習模型在未受防禦的情況下，遭受某對抗樣本攻擊後的準確度為

m ；而該深度學習模型加入某種防禦機制的保護下，對於同一對抗樣本攻擊的準確度 n ，若 $n > m$ ，代表該防禦機制有效對抗了該種對抗樣本攻擊，也表示對抗樣本攻擊的強健性(Adversarial Robustness)有所提升，舉例來說，對抗訓練(Adversarial Training)是一種為了提升深度學習的強健性而誕生的防禦機制。

2.4 對抗訓練

對抗樣本攻擊是一種精心設計的攻擊，使原先正確分類的模型產生錯誤預測，這種攻擊對於安全關鍵領域會造成極其嚴重的影響，例如自動駕駛[41, 42]、金融詐欺[42, 43]。因此，如何建立對抗樣本攻擊的防禦系統變成非常重要的事情。常見的防禦對抗樣本攻擊的方法有防禦性蒸餾(Defensive Distillation)[44]、隨機平滑化(Randomized Smoothing)[42, 45]、與對抗訓練(Adversarial Training)[24]等；雖然不同的防禦方法都有其優缺點與防禦範圍限制，無法分出絕對優劣，但對抗訓練(Adversarial Training)仍被視為簡單且具有一定保護力的防禦方法[24, 46]。

2.4.1 PGD 對抗訓練

對抗訓練(Adversarial Training)的思路相當簡單，就是使用某種對抗樣本攻擊在原始樣本中添加細微擾動，並且將加入擾動後的樣本與原始樣本一起進行訓練[24, 46]，這種方法也被視為一種資料擴增(Data Augmentation)的方法[47]，藉此來提升深度神經網路的對抗樣本攻擊強健性(Adversarial Robustness)。為了更理解對抗訓練的運作機制，必須介紹最小最大化公式(min-max formulation)[36]，這開啟了對抗訓練機制全新的視野，公式(2.8)為最大最小化公式的定義：

$$\min_{\theta} \rho(\theta), \text{ where } \rho(\theta) = E_{(x,y) \sim D} [\max_{\delta \in S} L(\theta, x + \delta, y)] \quad (2.8)$$

其中， S 是對抗擾動的範圍， δ 是添加的對抗擾動，作者將對抗訓練分成兩個子問題，一為非凹內部最大化問題(non-concave inner maximization problem)，從攻擊者的角度出發，負責找出高攻擊性的對抗樣本，因此在本篇論文，也提出了比 FGSM 攻擊性

強的 PGD 攻擊；二為非凸外部最小化問題(non-convex outer minimization problem) ，以防禦者的視角切入，外層的 $\rho(\theta)$ 就是防禦者的目標函數，要盡可能在強力攻擊下找出最小損失，進而訓練出一定對抗強健性的模型。

由於 FGSM 攻擊在非線性的模型攻擊強度較低，換句話說，使用 FGSM 攻擊所生成對抗樣本可能與原始樣本相當接近，以影像資料為例，只有少數像素點的差異。當模型過度依賴 FGSM 對抗樣本時，會讓模型對於原始樣本的敏感度大幅下降，進而降低模型準確度。這個現象被稱為標籤洩漏(label leaking)；為了避免該現象的發生，提升對抗樣本的多樣性與複雜度成為首要目標。因此，進行對抗訓練時，選擇使用迭代次數更多且在非線性模型攻擊力更強的 PGD 攻擊生成對抗樣本，並且進行對抗訓練是更好的選擇。這種方法被稱為「PGD 對抗訓練」(PGD-based adversarial training)[48]。

對抗強健性(Adversarial Robustness)與乾淨資料集的準確度(Natural Accuracy)是一個經典的權衡問題(trade-off)，雖然對抗訓練(Adversarial Training)可以提升對抗強健性，但同時也會對模型在乾淨資料上的準確度微幅下滑。因為對抗訓練會加入對抗樣本來增強模型強健性，卻與乾淨的原始樣本有差異，讓深度模型在處理乾淨資料時出現性能下降的問題。因此如何拿捏對抗強健性與原始樣本的準確度成為一大課題[39]。

最後，PGD 對抗訓練(PGD-based adversarial training)還有效率低落的問題，PGD 攻擊需要進行多次迭代，且每次迭代都需要計算模型梯度與更新參數，將 FGSM 攻擊與 PGD 攻擊進行時間複雜度的比較，由於 FGSM 攻擊只計算一次梯度與模型參數，時間複雜度為 $O(\theta)$ ，其中 θ 是模型參數的總數；假設 PGD 攻擊需要進行 k 次迭代，每次迭代需要 $O(\theta)$ 的時間，PGD 攻擊時間複雜度為 $O(k*\theta)$ 。

2.5 聯邦對抗訓練

聯邦學習作為分散式的機器學習框架，同時也需面對機器學習中對抗式樣本攻擊所帶來的威脅。為了因應這一挑戰，「聯邦對抗訓練」(Federated Adversarial Training, 縮寫 FAT) [49]或被稱作「對抗強健聯邦學習」(Adversarial Robust Federated Learning) [23]逐漸

浮現。這種訓練方法融合了聯邦學習和對抗訓練的特點，旨在強化訓練過程中的資料隱私，同時提升中心伺服器的全域模型在面對汙染的測試集 D_{test} 的對抗強健性[23]。

在聯邦對抗訓練中，各客戶端利用其專屬資料集來生成 PGD 對抗範例，並執行 PGD 對抗訓練(PGD-based adversarial training)。隨後，這些經過強化的模型更新會被傳送到中心伺服器。中心伺服器透過聚合演算法，整合這些更新，形成一個具有強健性的全域模型。該模型再被發送回客戶端以進行進一步的對抗訓練，直至達到所需的對抗強健性為止。

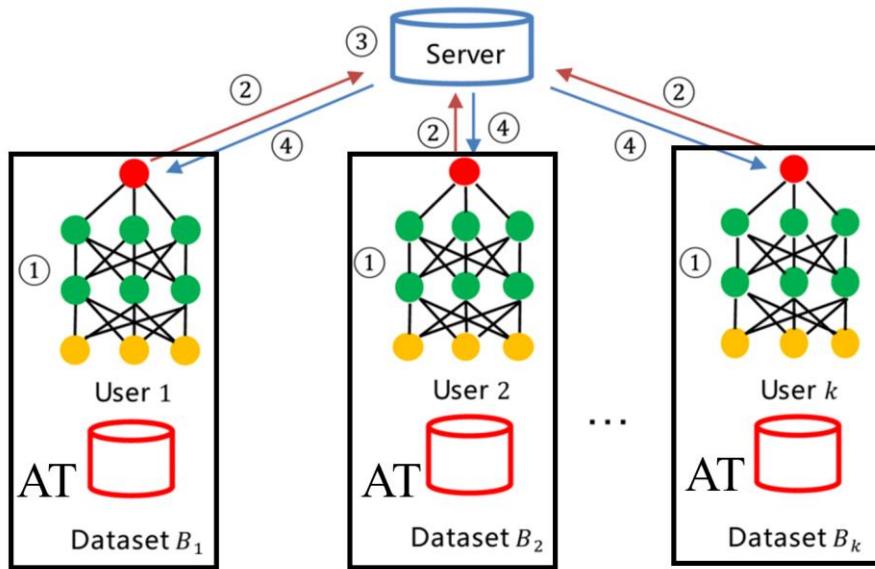


圖 2.7 聯邦對抗訓練(FAT)示意圖

儘管傳統的聯邦對抗訓練方法能有效地抵抗對抗樣本攻擊，並簡潔地呈現其機制，但它仍面臨著某些挑戰。在這種訓練方法中，每個客戶端設備都獨立進行 PGD 對抗訓練，這常常忽略了客戶端設備的計算能力限制。考慮到 PGD 對抗訓練本身已經需要相當的運算資源[50]，在真實應用場景中實施這種訓練方法可能不太現實[51]。為此，Zhou 及其團隊提議將對抗範例的生成過程交由中心伺服器來完成，並進一步提出了一種創新的聯邦對抗訓練框架——Fed_BVA[23]。

2.6 基於偏差-變異數的聯邦對抗訓練(Fed_BVA)

由於在現有的聯邦對抗學習研究領域中，缺乏系統性地分析聯邦對抗訓練與對抗強健性的研究方法，因此，在 Zhou 的論文[23]中，透過對本地客戶端生成的對抗樣本的偏差-變異數分解(Bias Variance Decomposition)[52]分析，發現直接在本地客戶端生成對抗樣本進行對抗訓練後，會使得本地客戶端的資料分布更加偏斜，不利於中心伺服器的聚合效果，因此提出了全新的聯邦對抗訓練框架——Fed_BVA。該框架除了能提升全域模型的對抗強健性，同時也能增加收斂效率。

該研究首先提出了對抗性偏差-變異數分解攻擊(Bias-Variance Decomposition Attack)，分別為 BV-FGSM 攻擊(Bias-variance based Fast Gradient Sign Method)與 BV-PGD 攻擊(Bias-variance based Projected Gradient Descent)，將 FGSM 攻擊與 PGD 攻擊以偏差與變異數的線性組合來表示，公式(2.9)與公式(2.10)分別為 BV-FGSM 與 BV-PGD 攻擊：

BV-FGSM 攻擊：

$$\hat{x}_{\text{BV-FGSM}} = x + \epsilon \cdot \text{sign}(\nabla_x B(x; w_1, \dots, w_K) + \lambda V(x; w_1, \dots, w_K)) \quad (2.9)$$

BV-PGD 攻擊：

$$\hat{x}_{\text{BV-PGD}} = \text{Proj}_{\Omega(x)}(\hat{x}^l + \epsilon \cdot \text{sign}(\nabla_{\hat{x}^l} B(\hat{x}^l; w_1, \dots, w_K) + \lambda V(\hat{x}^l; w_1, \dots, w_K))) \quad (2.10)$$

接著，將 BV-FGSM 攻擊或 BV-PGD 攻擊加入全新框架 Fed_BVA 中。Fed_BVA 演算法的目標是盡可能地減少聯邦學習系統中心伺服器全局收斂後的全域模型對於對抗樣本的威脅，換句話說，就是盡可能地增加收斂後全域模型在推理階段(testing phase)對於被惡意加入擾動的測試集 \hat{D}_{test} 的對抗強健性；該演算法的核心想法為在中心伺服器上放置一個小的輔助訓練集 D_s ， n_s 必須小於參與客戶端的資料集 D_k ，利用輔助訓練集 n_s 生成全域對抗範例(global adversarial examples)，並與客戶端共享全域對抗樣本。Fed_BVA 演算法分成以下步驟：

1. 在中心伺服器上使用輔助訓練集 D_s 生成全域對抗樣本：Fed_BVA 演算法使用偏差-變異數攻擊(BV-FGSM 或 BV-PGD)方法來生成全域對抗範例。本文中，以 \hat{D}_s 表示全域對抗範例。

2. 將全域對抗範例 \widehat{D}_s 分享給客戶端：中心伺服器將全域對抗範例 \widehat{D}_s 共享給所有參與客戶端。所有客戶端都必須使用 \widehat{D}_s 進行本地訓練。
3. 在客戶端進行對抗訓練：所有客戶端必須使用原始資料集與全域對抗範例 $(D_k \cup \widehat{D}_s)$ 進行對抗訓練；最後，將本地模型的參數傳到中心伺服器。
4. 中心服務器聚合所有客戶端的模型參數：中心伺服器以 FedAvg 聚合演算法，得到全域模型，然後再使用全域模型、偏差-變異數攻擊與輔助訓練集 D_s 生成新的全域對抗範例，並重複上述步驟，直到全局收斂。

以下為 Fed_BVA 演算法，該演算法是以 FedAvg 聚合演算法[53]為基礎，因此所有客戶端的使用的深度學習模型與訓練超參數必須相同，Fed_BVA 的起始條件為 $\widehat{D}_s = D_s$ ，意即客戶端只在第二個的溝通輪次(communication round)之後，才需加入全域對抗範例 \widehat{D}_s 一起進行訓練；圖 2.8 的第 6~15 行為客戶端本地訓練，第 16~19 行使用公式(2.9)或公式(2.10)為生成全域對抗樣本，第 20 行為 FedAvg 聚合演算法。

Fed_BVA 演算法在實驗中證明，在 MNIST 和 CIFAR-10 分類資料集上其效果優異。尤其在資料分佈為非同質分佈(non-IID)的情況下，Fed_BVA 演算法在相同的 FGSM 攻擊強度下，其準確率比 FedAvg 高出 20%。此外相較於傳統的聯邦對抗訓練，Fed_BVA 演算法也提高了 5%的準確度，代表對抗強健性具有一定的提升。

Algorithm 1 Fed_BVA

```
1: Input:  $K$  (number of clients, with local data sets  $\{\mathcal{D}_k\}_{k=1}^K$ );  $f$  (learning model);  $L$  (loss function);  $E$  (number of local epochs);  $F$  (fraction of clients selected on each round);  $B$  (batch size of local client);  $\eta$  (learning rate);  $\mathcal{D}_s$  (shared data set on server);  $\epsilon$  (perturbation magnitude).
2: Initialization: Initialize  $w_G^0$  and  $\hat{\mathcal{D}}_s = \mathcal{D}_s$ 
3: for each round  $r = 1, 2, \dots$  do
4:    $m = \max(F \cdot K, 1)$ 
5:    $S_r \leftarrow$  randomly sampled  $m$  clients
6:   for each client  $k \in S_r$  in parallel do
7:     Initialize  $k^{\text{th}}$  client's model with  $w_G^{r-1}$ 
8:      $\mathcal{B} \leftarrow$  split  $\mathcal{D}_k \cup \hat{\mathcal{D}}_s$  into batches of size  $B$ 
9:     for each local epoch  $i = 1, 2, \dots, E$  do
10:      for local batch  $(x, t) \in \mathcal{B}$  do
11:         $w_k^r \leftarrow w_k^r - \eta \nabla_{w_k} L(f_{\mathcal{D}_k}(x; w_k^r), t)$ 
12:      end for
13:    end for
14:    Calculate  $f_{\mathcal{D}_k}(x; w_k^r)$ ,  $\nabla_x f_{\mathcal{D}_k}(x; w_k^r)$  for  $\forall x \in \mathcal{D}_s$ 
15:  end for
16:  for  $(x, t) \in \mathcal{D}_s$  do
17:    Estimate the gradients  $\nabla_x B(x)$  and  $\nabla_x V(x)$ 
18:    Update  $\hat{x} \in \hat{\mathcal{D}}_s$  using BV-FGSM or BV-PGD
19:  end for
20:   $w_G^r \leftarrow \text{Aggregate}(w_k^r | k \in S_r)$ 
21: end for
22: return  $w_G$ 
```

圖 2.8 Fed_BVA 演算法[23]

雖然 Fed_BVA 演算法為聯邦對抗訓練開啟了新的視野，其方法卻涉及將由中心伺服器生成的全域對抗範例傳遞給各參與的客戶端以進行訓練。這一做法觸犯了聯邦學習的安全聚合原則：伺服器與客戶端之間應避免傳輸訓練資料[31]，而客戶端資料也應維持獨立性，不可任意共享資料[3]。針對這一問題，我們提出了一種全新的解決策略，旨在修正此問題，並打造出真正遵從聯邦學習安全聚合原則的系統。

2.7 鬆弛聯邦對抗訓練

鬆弛聯邦對抗訓練(Slack Federated Adversarial Training, SFAT)[54]是一種創新的框架，巧妙地融合了聯邦對抗訓練和 α -slack 聚合機制(α -slack mechanism)。在此架構中，SFAT 通過在聚合過程中為每位參與客戶端配置客製化的鬆弛度 (slack)，巧妙地解決了客戶端進行 PGD-對抗訓練(PGD-Adversarial Training)時，可能引起的資料異質性加劇的問題。這種鬆弛度分配機制被稱為 α -slack 機制，讓 SFAT 能維持對抗強健性的同時，有效地緩解聯邦學習中的資料異質性問題。

在中心伺服器上執行 α -鬆弛聚合機制，關鍵為修正了 Mardy's 對抗訓練的最大最小化問題，如公式(2.11)所示，而 α -鬆弛聚合機制是針對內部最大化問題提出優化策略，先使用 $\phi(\cdot)$ 操作，依據客戶端資料量與損失值的乘積 ($\frac{N_k}{N} * \mathcal{L}_k$) 進行升序排列，挑選出前 \hat{K} 個模型進行 $(1 + \alpha)$ 的加權，其他客戶端模型則以 $(1 - \alpha)$ 加權，如公式(2.12)所示。對原始目標函數進行了微小的鬆弛，從而使優化問題變得更容易求解，以盡可能減緩資料異質性加劇的效應，而圖 2.9 為鬆弛聯邦對抗訓練的完整演算法。

$$\min_{\theta} \rho(\theta), \text{ where } \rho(\theta) = E_{(x,y) \sim D} [\max_{\delta \in S} L(\theta, x + \delta, y)] \quad (2.11)$$

$$\begin{aligned} \mathcal{L}_{AT} &= \frac{1}{N} \sum_{n=1}^N \max_{\tilde{x}_n \in \mathcal{B}_\epsilon[x_n]} \ell(f_\theta(\tilde{x}_n), y_n) = \frac{1}{N} \sum_{k=1}^K \frac{N_k}{N} \left(\underbrace{\frac{1}{N_k} \sum_{n=1}^{N_k} \max_{\tilde{x}_n \in \mathcal{B}_\epsilon[x_n]} \ell(f_\theta(\tilde{x}_n^k), y_n^k)}_{\mathcal{L}_k} \right) \\ &\geq (1 + \alpha) \sum_{k=1}^{\hat{K}} \frac{N_{\phi(k)}}{N} \mathcal{L}_{\phi(k)} + (1 - \alpha) \sum_{k=1}^K \frac{N_{\phi(k)}}{N} \mathcal{L}_{\phi(k)} \\ &\doteq \mathcal{L}^\alpha(\hat{K}), \quad \text{s.t. } \alpha \in [0,1], \hat{K} \leq \frac{K}{2} \end{aligned} \quad (2.12)$$

Algorithm 1 Slack Federated Adversarial Training

Input: client number: K , communication rounds: T , local training epochs per round: E , initial server's model parameter: θ^0 , hyper-parameter for aggregation: α , number of enhanced clients: \hat{K} ;
Output: globally robust model θ^T ;

```

1: for  $t = 1, \dots, T$  do
2:   Clients: [ perform adversarial training]
3:   for client  $k = 1, \dots, K$  do
4:      $\theta_k^t, \mathcal{L}_k = AT(\theta_k^t, E)$  (Madry et al., 2018)
5:   end for
6:   Server: [ performs slacked aggregation]
7:    $\mathcal{L}_{all} \leftarrow [\frac{N_1}{N} \mathcal{L}_1, \frac{N_2}{N} \mathcal{L}_2, \dots, \frac{N_K}{N} \mathcal{L}_K]$ ,  $\mathcal{L}_{sorted} \leftarrow \text{Ascending\_Sort}(\mathcal{L}_{all})$ ;
8:    $\forall k, P_k = (\frac{1+\alpha}{1-\alpha} \cdot \mathbb{1}(\frac{N_k}{N} \mathcal{L}_k \leq \mathcal{L}_{sorted}[\hat{K}]) + 1 \cdot \mathbb{1}(\frac{N_k}{N} \mathcal{L}_k > \mathcal{L}_{sorted}[\hat{K}]))) / ((\sum_{k=1}^K P_k) + \frac{2\alpha}{1-\alpha})$ ;
9:    $\theta^{t+1} = \frac{1}{\sum_{k=1}^K N_k} \sum_{k=1}^K P_k N_k \theta_k^t$ ;
10:  end for

```

圖 2.9 Slack Federated Adversarial Training 演算法[54]

第三章 研究方法

介紹本研究提出的全新聯邦對抗訓練演算法——FedBVA_SAT_Slack，並且融入了偏差-變異數分解的全域對抗範例、在中心伺服器進行對抗訓練與執行反向鬆弛聚合機制，其目的提升聯邦學習系統的全域模型的對抗強健性。

3.1 定義問題

聯邦對抗訓練的目標是提高聯邦學習系統的全域模型在測試階段遭受對抗樣本攻擊的防禦能力，也就是提升全域模型的對抗強健性，如圖 3.1 所示。我們可以從中發現對抗樣本攻擊在測試階段對於聯邦學習系統具有相當大的威脅。即使將中心伺服器與客戶端之間的溝通進行了嚴密的加密和保護，仍會面臨對抗樣本攻擊的威脅[55]。因此，本研究聚焦於聯邦學習系統全局收斂後的全域模型(global model)防禦對抗樣本攻擊的能力。

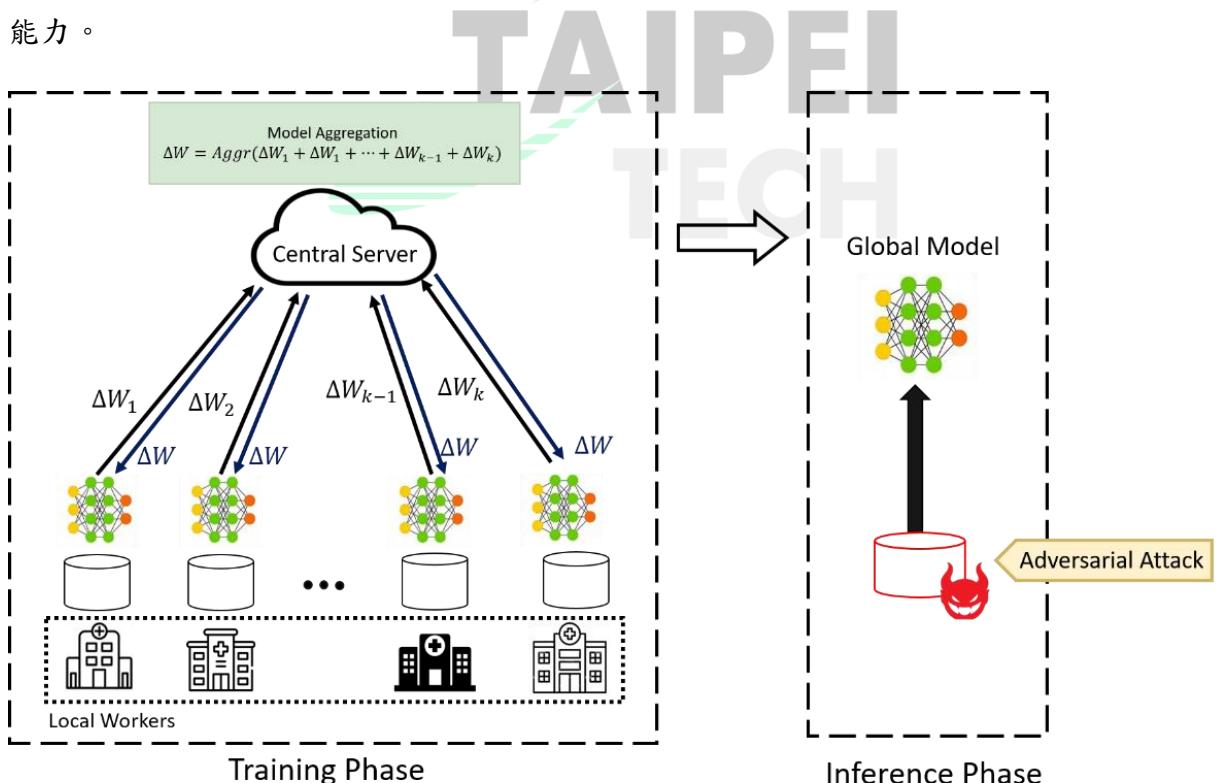


圖 3.1 聯邦學習基於對抗樣本攻擊威脅的階段

本研究提出的全新聯邦對抗訓練演算法，只在測試階段才會受到對抗樣本攻擊的影

響，無論是在中心伺服器加上對抗訓練模組，還是使用鬆弛聚合機制來生成全域模型，都是在未受到任何攻擊威脅的訓練階段(training phase)進行。當聯邦學習系統全局收斂且取得全域模型後，便可進入測試階段(testing phase)，將測試集 D_{test} 使用 FGSM 攻擊或 PGD 攻擊進行汙染，並將被汙染後的資料與全域模型進行測試，以得到對抗強健性的數據指標。

3.2 研究方法與設計

深入探討我們新提出的聯邦對抗訓練演算法——FedBVA_SAT_Slack。為了使讀者能更全面理解，本文分為 3.2.1 系統設計與 3.2.2 演算法流程兩個子節，幫助讀者對 FedBVA_SAT_Slack 的概念、設計原則和實際運作有深入的認識。另外，3.2.3 節將進行演算法比較，凸顯本研究所提出的演算法與現有文獻的差異性。

3.2.1 系統設計

FedBVA_SAT_Slack 演算法由三大系統組成，分別為參與客戶的訓練系統、中心伺服器的對抗訓練防禦系統，以及中心伺服器的聚合系統，說明如何相互協作並生成具有一定乾淨準確度與對抗強健性的全域模型，圖 3.2 為 FedBVA_SAT_Slack 系統時序圖。

1. 參與客戶端的訓練系統: FedBVA_SAT_Slack 系統依賴於一系列獨立的客戶端模型，每一個模型都在其自身的本地環境中訓練自身的原始資料集，訓練完成後，將客戶端模型與損失值上傳到中心伺服器。這種設計有效地減輕了客戶端的負擔，因為他們只需負責訓練自己的資料集，而無需涉及生成對抗範例或進行對抗訓練的繁瑣工作。
2. 中心伺服器的對抗訓練防禦系統: 首先，在中心伺服器放置一份極小的輔助資料集，並且利用此資料集與所有參與客戶端模型，來生成 BV-FGSM 全域對抗範例[23]，如公式(2.9)所示。接著，為了符合聯邦學習的安全聚合守則，必須將這些 BV-FGSM 全域對抗範例留在中心伺服器進行對抗訓練，也能減少客戶端的訓練負擔。

3. 中心伺服器的聚合系統：由於客戶端模型與對抗訓練的模型差異過大，必須另外設計特殊的聚合機制，我們將其聚合機制稱為反向鬆弛聚合機制。首先，反向鬆弛聚合機制考量了生成訓練各模型的訓練資料大小，同時挑選出損失值最大的模型進行特殊加權。讓全域模型可以在保持乾淨資料準確性的同時，達到對抗強健性的平衡。

整個流程會重複進行，直到系統收斂為止。第一個優點為，所有參與的客戶端只需在本地環境中訓練自己的資料集，而不需要進行任何形式的對抗訓練，可以大幅度降低參與客戶端的運算成本。第二個優點為，與 Fed_BVA 演算法相比，FedBVA_SAT_Slack 演算法大幅度地降低了放置於中心伺服器的輔助資料集數量，以 MNIST 資料集為例，Fed_BVA 須放置 1280 筆資料，而 FedBVA_SAT_Slack 僅需 200 筆。兩者這種設計不僅可以有效地減輕客戶端的負擔，同時還能在保證乾淨資料準確性的前提下，實現全域模型的對抗強健性。

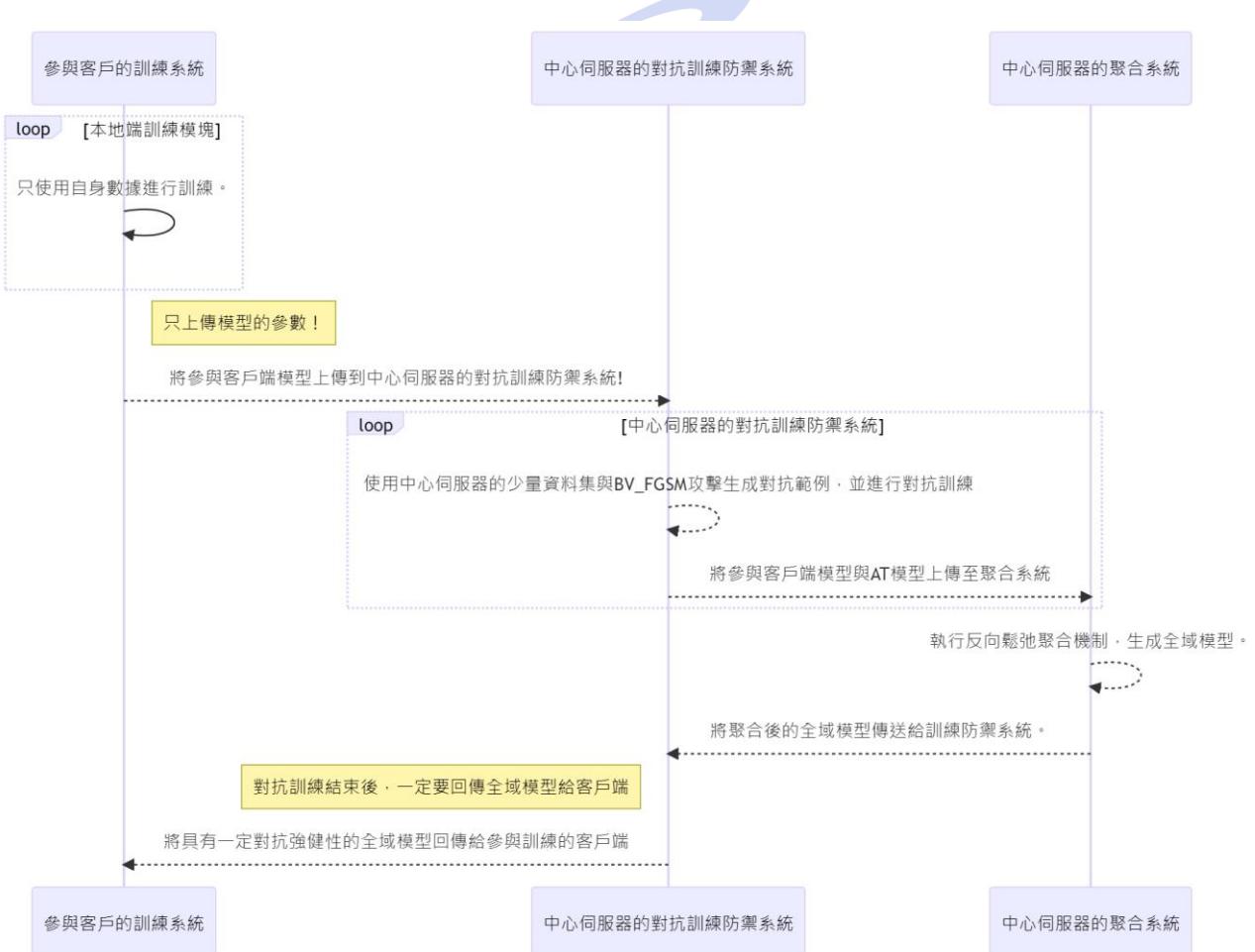


圖 3.2 FedBVA_SAT_Slack 系統時序圖

3.2.2 演算法流程

FedBVA_SAT_Slack 的演算法可以拆分成以下八個步驟，完整演算法請參考圖 3.6：

1. **客戶端訓練:**每個客戶端獨立訓練其自有的乾淨資料集。訓練過程中，對抗樣本的攻擊並不會影響其運作。
2. **將客戶端模型和損失值，上傳到用於生成全域對抗範例的位置:**客戶端訓練完成後，將所有參與客戶端的模型，上傳至欲生成全域對抗範例的部分。
3. **將客戶端模型與損失值，上傳到用於執行反向鬆弛聚合機制的位置:**客戶端訓練完成後，將所有參與客戶端的模型，上傳至欲進行反向鬆弛聚合機制的部分。

步驟 1~3 由參與客戶的訓練系統負責，如圖 3.3 所示；可對應圖 3.6 之第 2~13 行。

4. **中心伺服器生成全域對抗範例:**當中心伺服器收集到所有參與客戶端模型後，將所有客戶端模型進行線性組合，以生成 BV-FGSM 全域對抗範例。
5. **中心伺服器進行對抗訓練:**為了符合聯邦學習系統的安全聚合守則，將在中心伺服器生成的全域對抗範例，保留與此，並直接進行對抗訓練。
6. **將專門進行對抗訓練的模型與其損失值，上傳至進行反向鬆弛聚合機制之處:**將專門訓練全域對抗範例的模型，與客戶端模型一起進行聚合，以提升全域模型的對抗強健性。

步驟 4~6 由中心伺服器的對抗訓練防禦系統負責，如圖 3.4 所示；可對應圖 3.6 之第 15~25 行。

7. **中心伺服器對所有模型進行降序排列:**當聚合系統收集到所有參與客戶的模型與專門進行對抗訓練的模型後，為了選出最大損失值的模型所需的操作。
8. **中心伺服器進行反向鬆弛聚合機制:**當聚合系統收集到所有參與客戶的模型與專門進行對抗訓練的模型後，為了平衡兩者模型的差異，將最大損失值模型進行 P 倍加權，以保持乾淨準確度的前提下，提升全域模型的對抗強健性。

步驟 7~8 由中心伺服器的聚合系統負責，如圖 3.5 所示；可對應圖 3.6 第 27~29 行。

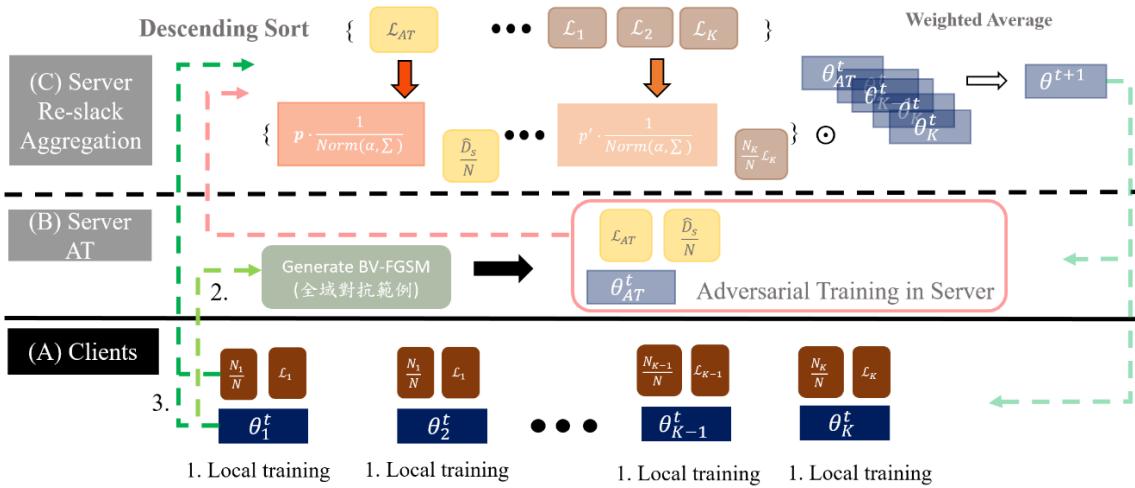


圖 3.3 FedBVA_SAT_Slack 演算法之參與客戶的訓練系統示意圖

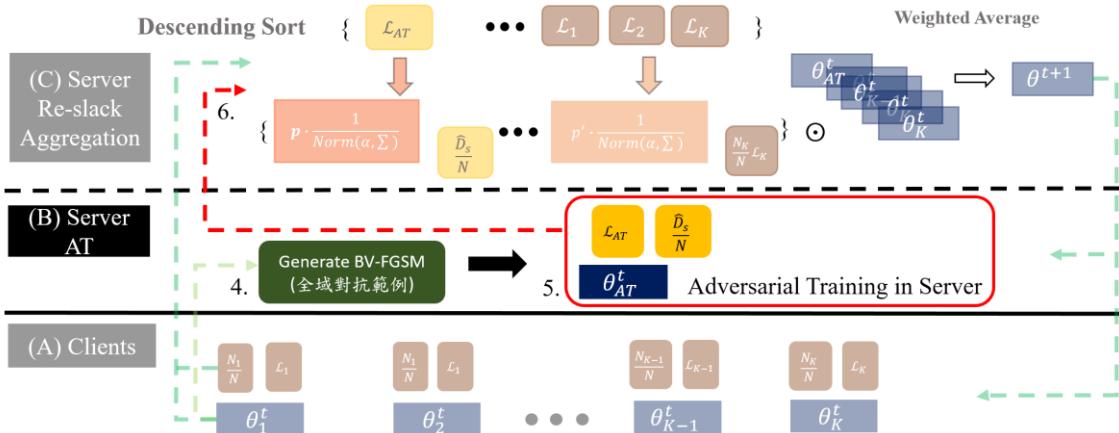


圖 3.4 FedBVA_SAT_Slack 演算法之中心伺服器的對抗訓練防禦系統示意圖

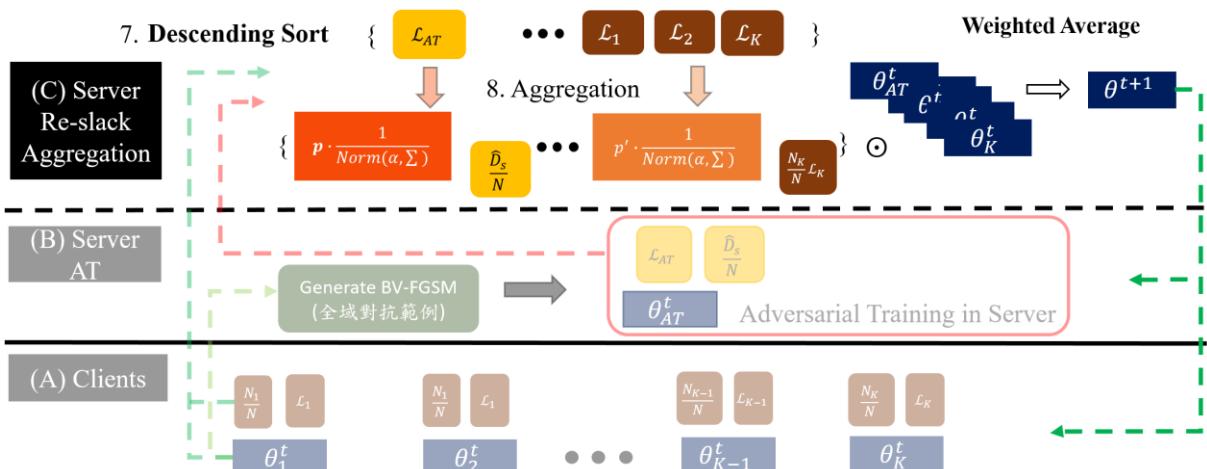


圖 3.5 FedBVA_SAT_Slack 演算法之中心伺服器的聚合系統示意圖

Algorithm FedBVA_Server_Slack

```

■ 1: Input:  $K$ (number of clients, with local data sets  $\{D_k\}_{k=1}^K$ );  $w_k$ (client k's weight),  $E$ (number of local epochs);  $\hat{E}$  (number of adversarial training);  $F$ (fraction of clients selected on each round);  $B$  (batch size of local client);  $\hat{B}$ (batch size of adversarial training in server)  $\eta$ (learning rate);  $D_s$ (the shared dataset on server);  $\epsilon$ (perturbation magnitude).
Output: globally robust model

    Clients: [ perform natural training]

2: Initialization: Initialize  $w_G^0$  and  $\hat{D}_s = D_s$ 

3: for each round  $r = 1, 2, \dots$  do

4:    $m = \max(F \cdot K, 1)$ ;  $S_r \leftarrow$  randomly sampled  $m$  clients

5:   for each client  $k \in S_r$  in parallel do

6:     Initialize  $k^{\text{th}}$  client's model with  $w_G^{r-1}$ 

7:      $B \leftarrow$  split  $D_k$  into batches of size  $B$ 

8:     for each local epoch  $e = 1, 2, \dots, E$  do

9:       for local batch  $(x, t) \in B$  do

10:         $w_k^r \leftarrow w_k^r - \eta \nabla_w L(f_{D_k}(x; w_k^r), t)$ 

11:      end for

12:    end for

13:    Calculate  $f_{D_k}(x; w_k^r)$ ,  $\nabla_w f_{D_k}(x; w_k^r)$ ,  $L_k$  for  $\forall x \in D_s$ 

14:  end for

15:  Server: [ performs adversarial training]

16:  for  $(x, t) \in D_s$  do

17:    Estimate the gradients  $\nabla_x B(x)$  and  $\nabla_x V(x)$ 

18:    Update  $\hat{x} \in \hat{D}_s$  using BV-FGSM

19:  end for

20:   $\hat{B} \leftarrow$  split  $\hat{D}_s$  into batches of size  $\hat{B}$ 

21:  for each local epoch  $e = 1, 2, \dots, E$  do

22:    for local batch  $(\hat{x}, t) \in \hat{B}$  do

23:       $w_{AT} \leftarrow w_{AT} - \eta \nabla_w L(f_{\hat{D}_s}(\hat{x}; w_{AT}), t)$ 

24:    end for

25:  end for

26:  Calculate  $w_{AT}$ ,  $L_{AT}$  for  $\forall x \in \hat{D}_s$ 

27:  Server: [ performs re-slacked aggregation]

28:   $w_G^r \leftarrow$  Slack Aggregate( $w_k^r | k \in S_r \cup w_{AT}$ ) in descending sort

29: return  $w_G$ 

```

圖 3.6 FedBVA_SAT_Slack 聯邦對抗訓練演算法

3.2.3 演算法比較與分析

本文提出的 FedBVA_SAT_Slack 演算法，參考了 Fed_AT、Fed_BVA 與 SFAT 三個不同的聯邦對抗訓練框架的技術而發展：

1. Fed_AT：一種基本的聯邦對抗訓練框架。直接在客戶端進行 PGD 對抗訓練，並且中心伺服器以模型的訓練資料量作為加權依據的方法。
2. Fed_BVA：中心伺服器生成全域對抗範例後，將其留給客戶端進行對抗訓練，最後伺服器也是直接使用訓練資料量作為加權依據。
3. SFAT：在客戶端進行對抗訓練的基礎上，伺服器使用 J. Zhu 所提出的鬆弛聚合機制，該機制每次選取損失值較小的前幾名模型進行特殊加權，並且同時考慮了模型的訓練資料量。
4. FedBVA_SAT_Slack：本文所提出的全新框架。在此設計下，客戶端僅需專注於訓練自身的資料集，避免了進行繁瑣的對抗訓練。為了強化系統的對抗強健性，我們採用了 Fed_BVA 所提出的全域對抗範例創建方法。隨後，為了符合聯邦學習的安全聚合原則，將全域對抗範例保留在中心伺服器，並在此進行對抗訓練。最後，在聚合模型的過程中，我們採用了一種創新的反向鬆弛聚合機制(Re-Slack Aggregation)。我們的方法是受到 J. Zhu 所提出的鬆弛聚合機制啟發，並進行適當的調整與優化，以平衡客戶端模型與伺服器的對抗訓練模型之間的差異。

接著，表 3.1 呈現了上述的四種不同的聯邦對抗訓練框架，於客戶端與中心伺服器所使用的技術差異。表格中，Our method 代表 FedBVA_SAT_Slack 演算法。可以透過表 3.1，清楚地瞭解到 FedBVA_SAT_Slack 如何借鑑並改進前人的方法，並提出一種更有效、更經濟，且在保證資料準確性的同時，能提高對抗攻擊強健性的新方法。

表 3.1 Proposed method 與現有的聯邦對抗訓練演算法的技術比較

	Fed_AT	SFAT	Fed_BVA	Our method
L:PGD 對抗訓練	V	V	X	X
S:生成全域對抗範例	X	X	V	V

L:訓練全域對抗範例	X	X	V	X
S:訓練全域對抗範例 (SAT)	X	X	X	V
S:鬆弛聚合機制	X	V	X	X
S:反向鬆弛聚合機制	X	X	X	V

(L = Local , 表客戶端。S=Server , 表伺服器端)

從上述的介紹與分析中，可以看出 FedBVA_SAT_Slack 演算法具有獨特的優勢與創
新之處：

1. 客戶端訓練負擔輕：與 Fed_AT 相比，FedBVA_SAT_Slack 的客戶端不需要進行任何形式的對抗訓練。這意味著客戶端只需在本地環境中訓練自己的資料集，大大降低了運算與網路通信成本，使得聯邦學習在多樣化的使用環境下有更好的實用性。
2. 減少輔助資料集數量：與 Fed_BVA 比較，我們可以看到 FedBVA_SAT_Slack 在伺服器端所需的輔助資料集大幅度減少，如以 MNIST 資料集為例，Fed_BVA 需要 2560 筆資料，而 FedBVA_SAT_Slack 只需 200 筆。這種設計節省了中心伺服器的資源需求，也同時顯示了 FedBVA_SAT_Slack 在利用現有資源上的高效率。
3. 中心伺服器的對抗訓練強化安全性：FedBVA_SAT_Slack 特別在中心伺服器上實施對抗訓練，這與 Fed_AT 和 Fed_BVA 有所區別，因為後兩者是在客戶端上執行對抗訓練。由於對抗訓練的模型被配置在中心伺服器上，這不僅滿足了聯邦學習的安全聚合準則，還減少了客戶端的訓練壓力。
4. 創新的反向鬆弛聚合機制，平衡準確度與強健性：FedBVA_SAT_Slack 使用了反向鬆弛加權聚合機制，這個機制是基於 SFAT 演算法的創新改進。不同於 SFAT 只選取損失前幾小的模型進行特殊加權，FedBVA_SAT_Slack 挑選出損失值最大的模型進行特殊加權，並且同時考慮模型的訓練資料量。這種獨特的設計可以在保持乾淨資料準確性的同時，達到對抗強健性的平衡。

3.3 偏差-變異數分解攻擊的多樣性

偏差-變異數分解是一種常用於機器學習分析的方法[56, 57]，主要目的為將模型的泛化誤差分解為偏差和變異數兩大部分。在非獨立同分佈(non-IID)的聯邦學習環境中，每位參與客戶端的資料集都具有其特殊性和差異。這樣的差異使得各客戶端訓練出的模型在不同數據集上可能產生偏差和變異。因此，在聯邦對抗訓練框架中，應用偏差-變異數分解來創造對抗全域範例是一個可行的方法。

簡單來說，偏差-變異數分解能將模型的泛化誤差拆分為兩大部分：一是偏差，評估所有參與客戶端模型與最佳模型之間的偏離程度；二是變異數，衡量所有參與客戶端模型之間的差異。首先，透過降低模型之間的偏差，可以提高模型的靈活性和適應性，使其能更好地適應不同的數據分布和模式，進而提高模型的泛化能力。接著，透過權衡不同客戶端模型的變異數，可以了解模型在不同訓練數據集上的穩定性與一致性。較低的變異數意味著模型在不同的數據集上的預測結果相對一致，具有較好的泛化能力。較高的變異數可能會導致模型對於未知攻擊的預測結果不穩定。

在本研究中，我們選擇使用 BV-FGSM 攻擊[23]生成全域對抗範例，與一般的 PGD 對抗範例相比，這兩種方法在生成對抗範例方面存在著差異。

首先，BV-FGSM 攻擊是基於每個溝通輪次中所有參與客戶端模型的反饋來生成對抗範例。因此，每回合生成的全域對抗範例都會有微小的差異，如表 3.2 所示。儘管這些變化非常微小，但人眼仍然能夠辨識出各溝通輪次所生成對抗範例之間的變化。

相反，PGD 對抗範例在每個溝通輪次中並不考慮所有參與客戶端模型的反饋，而是根據單個客戶端模型的反饋生成對抗範例。因此，PGD 對抗範例的差異較難以被肉眼察覺，並且相對較一致，如表 3.3 所示。

這種差異使得 BV-FGSM 攻擊生成的全域對抗範例具有更多的多樣與豐富性，而 PGD 對抗範例則較為一致。利用 BV-FGSM 攻擊生成的對抗範例策略，能夠讓聯邦對抗訓練系統學習到更多不同的全域對抗範例，進一步提升聯邦學習系統對未知攻擊的適應能力，並增強全域模型的對抗強健性。

因此，在本研究中選擇使用 BV-FGSM 攻擊生成全域對抗範例，以利用其多樣性特徵，提高聯邦學習系統應對未知攻擊威脅的能力。

表 3.2 利用 BV-FGSM 每五個溝通輪次所生成的全域對抗範例

溝通輪次=1	溝通輪次=5	溝通輪次=10	溝通輪次=15
Class: 6 	Class: 6 	Class: 6 	Class: 6
Class: 7 	Class: 7 	Class: 7 	Class: 7

表 3.3 利用 PGD 攻擊之每五個溝通輪次所生成的對抗範例

溝通輪次=1	溝通輪次=5	溝通輪次=10	溝通輪次=15
Class: 5 	Class: 5 	Class: 5 	Class: 5
Class: 0 	Class: 0 	Class: 0 	Class: 0

完整的 BV-FGSM 全域對抗範例與 PGD 對抗範例，請參考附圖 6.1 與附圖 6.2。

3.4 中心伺服器進行對抗訓練

儘管 Fed_BVA 演算法[23]在聯邦對抗訓練領域做出了重要貢獻，卻違反了聯邦學習中的安全聚合原則。簡而言之，Fed_BVA 演算法在中央伺服器上生成全域對抗範後，將全域對抗範例回傳給所有參與客戶端，以進行對抗訓練。然而，這種資料傳輸機制似乎違反了聯邦學習的兩項核心安全聚合原則，如章節 2.1.1 所述。第二個原則強調，客戶端與中央伺服器之間不應直接傳輸訓練資料，而僅交換模型權重和更新。第三個原則明確指出，每個客戶端的數據應僅供自身使用，不應與其他客戶端共享。因此，必須找到一種有效的解決方案來遵從這些原則，同時能提升聯邦學習系統的對抗強健性。

為了符合聯邦學習的安全聚合原則，本研究提出了在中心伺服器生成全域對抗範例後，直接在伺服器進行對抗訓練。這個方法可以視為奠基於許多先前研究的策略。例如，FLTrust[14]在中心伺服器端加入一個小且未受汙染的根資料集，利用根資料集訓練一個基準模型，用來對惡意客戶端模型的篩選。而另一個方法，Justinian's GAAvernor[15]則是利用強化學習技術，在中心伺服器端訓練出一個具有強健性的輔助模型，來協助伺服器在聚合階段作出決策。這些案例證明了為了提升聯邦學習系統的安全性，在中心伺服器進行訓練是一種可行的策略。

本研究所提出的 FedBVA_SAT_Slack 演算法，為了符合聯邦學習的安全聚合原則，當全域對抗範例生成後，必須將其留在伺服器進行對抗訓練，流程圖可參考圖 3.7；而完整的對抗訓練演算法，如圖 3.8 所示。當對抗訓練結束後，所有客戶端模型和對抗訓練模型，必須執行反向鬆弛聚合機制，以生成出全域模型，才能確保於乾淨準確度和對抗強健性之間取得平衡。

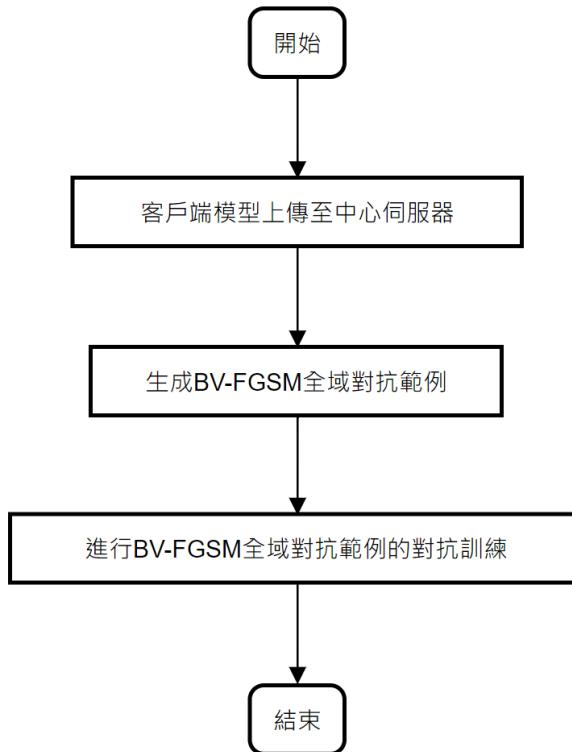


圖 3.7 FedBVA_SAT_Slack 聯邦對抗訓練框架:伺服器進行對抗訓練的流程圖

Algorithm 中心伺服器執行對抗訓練。

Input: K (number of clients, with local data sets $\{D_k\}_{k=1}^K$); \mathcal{L} (the loss); P_k (weighting ratio) ; $\widehat{\mathcal{D}}_s$ (number of global adversarial examples); T (communication rounds); θ^0 (initial server's model parameters); $\widehat{\mathcal{K}}$ (number of enhanced clients) .
Output: globally robust model: θ^T .

Server: [performs adversarial training]

- 1: **for** $(x, t) \in \mathcal{D}_s$ **do**
- 2: Estimate the gradients $\nabla_x B(x)$ and $\nabla_x V(x)$.
- 3: Update $\hat{x} \in \widehat{\mathcal{D}}_s$ using BV-FGSM .
- 4: **end for** .
- 5: $\mathcal{B} \leftarrow$ split $\widehat{\mathcal{D}}_s$ into batches of size \hat{B} .
- 6: **for** each server epoch $e = 1, 2, \dots, E$ **do**
- 7: **for** server batch $(\hat{x}, t) \in \mathcal{B}$ **do**
- 8: $w_{AT} \leftarrow w_{AT} - \eta \nabla_w L(f_{\widehat{\mathcal{D}}_s}(\hat{x}; w_{AT}), t)$.
- 9: **end for** .
- 10: **end for** .

圖 3.8 FedBVA_SAT_Slack 聯邦對抗訓練框架: 伺服器進行對抗訓練的演算法

總結來說，FedBVA_SAT_Slack 演算法透過在中心伺服器上直接進行對抗訓練，避免違反安全聚合原則並且在客戶端無須負擔進行對抗訓練的運算成本下，盡可能地提升聯邦學習系統的對抗強健性，並且透過適當地運行反向鬆弛機制，讓其能夠平衡乾淨準確度與對抗強健性。

3.5 反向鬆弛聚合機制的設計

當中心伺服器完成對抗訓練後，將所有參與客戶端模型們與對抗訓練的模型，必須經過反向鬆弛聚合機制，來生成全域模型，如圖 3.5 所示；其中， \hat{D}_s 代表 BV-FGSM 全域對抗範例的資料量。

在 FedBVA_SAT_Slack 架構中，鬆弛聚合加權機制扮演了極為重要的角色。當對抗訓練完成後，伺服器內會有兩種完全不同類型的深度學習模型：一種是僅訓練自身乾淨資料的客戶端模型；另一種則是僅使用 BV-FGSM 全域對抗範例來進行訓練的模型。由於這兩種模型之間存在著極大的差異，如果僅以訓練模型的資料量作為加權依據，而未應用本研究所提出的反向鬆弛聚合加權機制，難以在乾淨數據的準確度與對抗強健性之間取得適當的平衡。下一章節將提供完整的實驗數據，以利佐證此觀點。

在反向鬆弛聚合機制機制中，首先根據模型損失值大小進行降序排列($\mathcal{L}_1 \dots \mathcal{L}_k, \mathcal{L}_{AT}$)；然後，選出損失值最大的模型進行 p 倍加權，其他模型則以 p' 倍加權。因改成選擇較大損失值的模型，而稱為反向鬆弛聚合機制(re-slacked aggregation)。

反向鬆弛加權聚合機制的計算公式為(3.1)；其中， N_k 代表第 k 個參與客戶端的訓練資料量，而 N 則代表所有客戶端與對抗訓練模型的資料量總合。

$$\hat{\mathcal{K}} = 1, \mathcal{L}_k^{slack} = p \cdot \mathbb{1}\left(\mathcal{L}_{sorted}[\hat{\mathcal{K}}] \geq \frac{N_k}{N} * \mathcal{L}_k \vee \frac{\hat{D}_s}{N} \mathcal{L}_{AT}\right) + p' \cdot \mathbb{1}\left(\frac{N_k}{N} * \mathcal{L}_k \leq \mathcal{L}_{sorted}[\hat{\mathcal{K}}]\right) \quad (3.1)$$

公式(3.1)中，除了考慮 p 倍或 p' 倍加權，也會考慮每個模型的訓練資料比例。

圖 3.9 為反向鬆弛聚合機制的流程圖，選取最大損失值模型的設計，是希望每次進行鬆弛聚合機制時，都能選取到專門訓練全域對抗範例的模型，進行 p 倍加權，進而提升聯邦學習系統全域模型抵禦對抗樣本攻擊的能力。

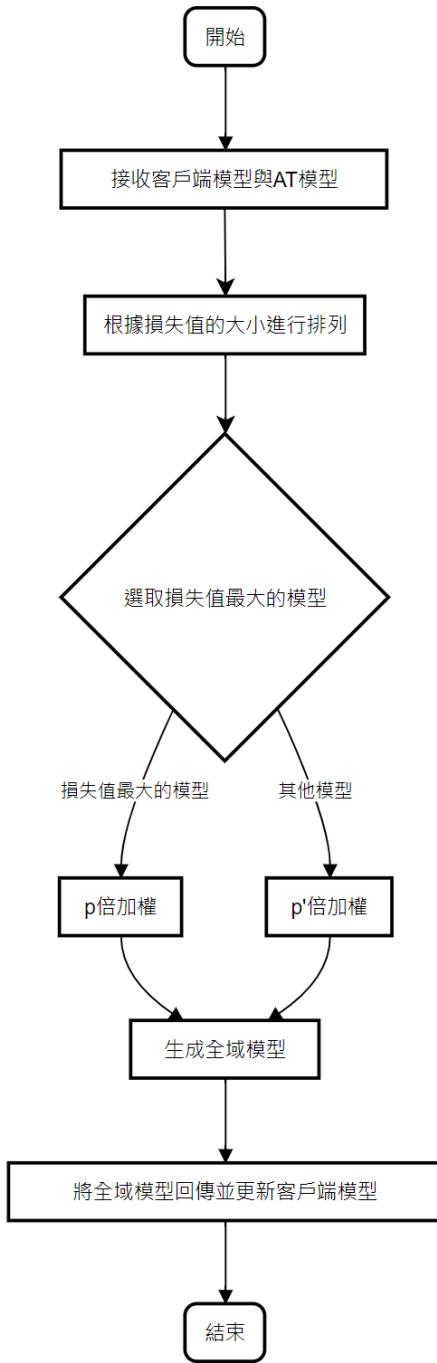


圖 3.9 FedBVA_SAT_Slack 聯邦對抗訓練框架：反向鬆弛聚合機制的流程圖

使用反向鬆弛聚合加權機制，具有以下兩大優勢：

1. 減少中心伺服器的輔助資料集數量：透過增加專門進行對抗訓練的模型加權比例，能夠減少儲存於中心伺服器的 BV-FGSM 全域對抗範例(\widehat{D}_s)的數量。因為聯生成全域對抗範例時需要迭代所有參與的客戶端模型，這過程消耗了相對大量的計算資源與成本。若能減少全域對抗範例的數量，就可以提高整體系統的運行效率。以 MNIST

資料集為例，Fed_BVA 聯邦對抗訓練演算法的輔助資料集數量為 1280，而本研究提出的 FedBVA_SAT_Slack 的輔助資料集數量僅為 200。不僅能提高運行效率，同時也增加了安全性，並降低資料外流的風險。

2. 增加全域模型的對抗強健性：反向鬆弛聚合機制有助於增強模型的對抗強健性。通過適當地調整各模型的加權比例，我們可以優先選擇專門訓練以抵禦對抗樣本攻擊的模型進行加權。不僅保持全域模型的性能，也提高了模型對於未知對抗樣本攻擊的防禦能力，讓我們在對抗未知的對抗樣本攻擊時有更多的策略選擇，開闢了全新的視角。

最後，圖 3.10 為 FedBVA_SAT_Slack 中的反向鬆弛聚合機制演算法。

Algorithm 中心伺服器的反向鬆弛聚合機制
Input: K (number of clients, with local data sets $\{D_k\}_{k=1}^K$); \mathcal{L} (the loss); P_k (weighting ratio); \hat{D}_s (number of global adversarial examples); T (communication rounds); θ^0 (initial server's model parameters); $\hat{\mathcal{R}}$ (number of enhanced clients)
Output: globally robust model: θ^T
Server: [performs re-slacked aggregation]
1: For $t = 1, 2 \dots T$ do
2: $\mathcal{L}_{all} \leftarrow [\mathcal{L}_1, \mathcal{L}_2, \dots, \mathcal{L}_k, \mathcal{L}_{AT}]$, $\mathcal{L}_{sorted} \leftarrow Descending\ Sort(\mathcal{L}_{all})$
3: $\hat{\mathcal{R}} = 1, \forall k, P_k, = (p \cdot \mathbb{1}(\mathcal{L}_{sorted}[\hat{\mathcal{R}}] \geq \frac{N_k}{N} * \mathcal{L}_k \text{ and } \frac{\hat{D}_s}{N} \mathcal{L}_{AT})) +$ $p' \left(\frac{N_k}{N} * \mathcal{L}_k \leq \mathcal{L}_{sorted}[\hat{\mathcal{R}}] \right) / ((\sum_{k=1}^K P_k))$
4: $\theta^{t+1} = \frac{1}{\sum_{k=1}^K N_k} \sum_{k=1}^K P_k N_k \theta_k^t$
5: end for

圖 3.10 FedBVA_SAT_Slack 聯邦對抗訓練框架：反向鬆弛聚合機制演算法

3.6 實驗資料集、深度學習模型與聯邦學習參數設定

在聯邦學習系統中，所有的參與客戶端確實都需要使用相同的模型進行訓練。因此，本章節將詳細闡述我們在實驗中所用到的資料集，以及相對應的深度學習模型。我們採用了四種資料集和模型：分別是利用卷積神經網路 (CNN) 模型訓練的 MNIST 資料集、

運用 VGG-11 模型訓練的 Fashion MNIST 資料集、以自編碼器（Auto-Encoder）模型訓練的線寬預測資料集，以及透過變分自編碼器（VAE）模型訓練的電晶體資料集。此外，本研究主要專注於處理非獨立同分佈（non-IID）的聯邦學習資料分布，而這四個資料集的分布型態有所不同，我們也將在此章節進行詳細說明。

接下來，表 3.4 將展示這四個資料集在聯邦學習實驗中的參數設定，我們將針對每個資料集進行具體的說明與分析。

表 3.4 聯邦學習實驗參數於四種資料集之設定

	MNIST	Fashion-MNIST	線寬資料集	電晶體資料集
Non-IID setting	n-shards	skew parameter	rate	Numbers
Communication rounds	15	20	20	50
Clients (K)	5	5	10	3
Epochs (E)	5	5	5	500
Local batch size	256	256	1024	128
Model	CNN	VGG11	Auto-Encoder	VAE
Optimizer	Adam	Adam	Adam	Adam
Learning rate	0.01	0.01	0.01	0.001

3.6.1 MNIST 手寫數字資料集與 CNN 模型

MNIST(Modified National Institute of Standards and Technology database ,MNIST)手寫數字資料集[25]，是一個在機器學習和影像處理領域被廣泛使用的資料集。這個資料集包含了 70000 張 28x28 的灰階圖片，訓練集共有 60000 張影像，測試集則為 10000 張影像，如圖 3.11 所示。MNIST 資料集由十個類別組成，分別由數字 0 至 9 組成。MNIST 資料集是聯邦學習領域實驗的必要資料集。

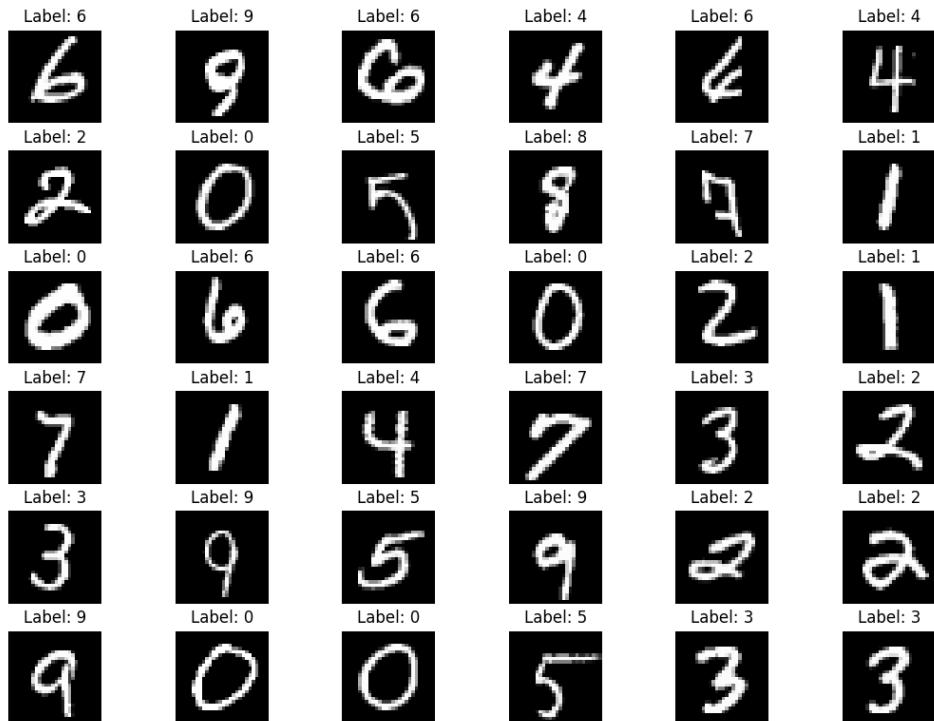


圖 3.11 MNIST 資料集

本研究針對 MNIST 手寫數字資料，使用卷積神經網路(Convolutional Neural Network, CNN)[58]作為主要的深度學習模型。此模型經過專門設計和優化，使其能夠有效地處理 MNIST 資料集的特點。圖 3.12 為 CNN 模型架構，包含兩個卷積層、兩個全連接層，以及兩個 Dropout 層。卷積層(Convolutional Layer)的主要任務在於從原始圖像中抽取有價值的特徵。每一個卷積層都使用 64 個 5×5 的卷積核(kernel)，並且採用 ReLU 作為激活函數，以增強模型的非線性表現力。經過兩層的卷積運算後，圖像的局部特徵將會被有效提取。

卷積層之後，模型將輸出結果經過 Dropout 層，隨機丟棄一部分的特徵。這樣的設計有助於防止模型過度依賴某些特定的特徵，並且能夠增進模型對新數據的泛化能力。之後，模型將 Dropout 層的輸出攤平並傳送到全連接層進行最終的分類。全連接層的節點數分別設定為 128 和 10，對應到特徵轉換和類別預測的功能。通過這樣的架構設計，模型能夠有效地學習並預測 MNIST 手寫數字資料的不同類別。

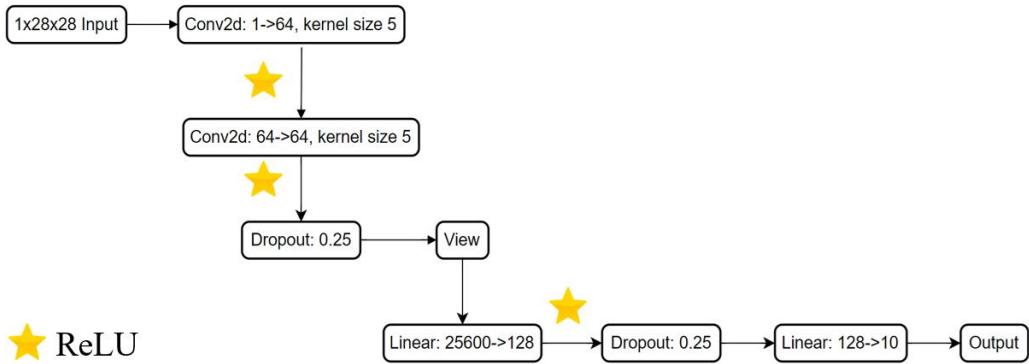


圖 3.12 MNIST 手寫數字辨識實驗: CNN 模型架構

3.6.2 MNIST 資料集的非獨立同分布設定與訓練過程

如表 3.4 所示，在 MNIST 資料集的實驗中，採用 n-shards 方法進行非獨立同分布的資料分配。n-shards 意味著每個客戶端最多擁有 n 個類別的資料[3]。

MNIST 資料集共含有六萬筆的訓練資料。在實驗設定中，共有五個客戶端參與。每個客戶端從這六萬筆訓練資料中，分別取得四個類別、總數 24000 筆的資料進行訓練，如圖 3.13 所示。這種分配方式模擬了現實情況下，每個客戶端通常只會有部分的資料類別。同時，為了能有效評估聯邦學習所產生的全域模型的效能，額外設置了一萬筆的測試資料集。透過這樣的設定，實驗不僅更能模擬真實環境，也使得聯邦學習的性能表現得以深入探討。

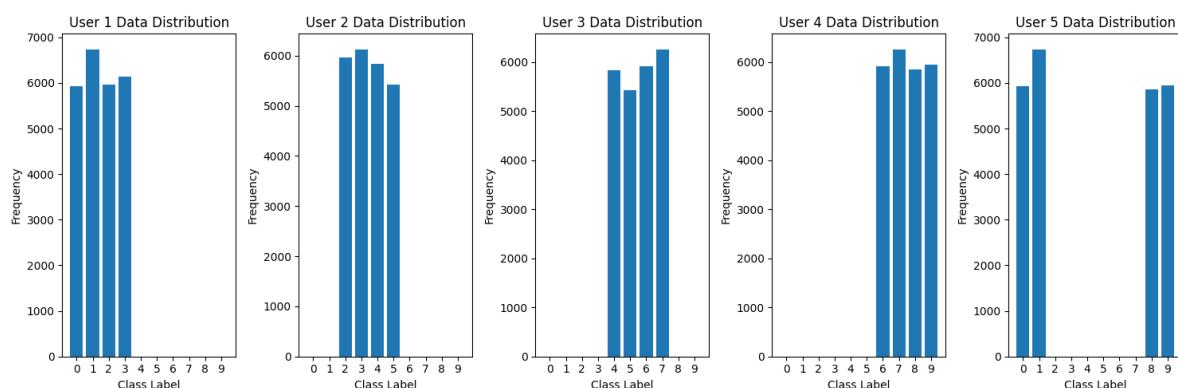


圖 3.13 MNIST 資料集在聯邦學習之五個客戶端的非獨立同分布狀況

3.6.3 Fashion-MNIST 資料集與 VGG-11 模型

Fashion-MNIST 是一個被廣泛使用於影像處理與聯邦學習領域的資料集。由 10 種不同的時尚商品類別所組成，分別為 T-shirt/top、Trouser、Pullover、Dress、Coat、Sandal、Shirt、Sneaker、Bag 和 Ankle boot。包含了 70000 張 28x28 的灰階圖片，每一類別都有 7000 張圖片，訓練集有 60000 張影像，而測試集為 10000 張影像，如圖 3.14 所示。



圖 3.14 Fashion MNIST 資料集

本研究中針對 Fashion-MNIST 資料集，使用 VGG[59]卷積神經網路模型 (Visual Geometry Group, VGG)來進行訓練；在 VGG 模型家族中，VGG11 表示該模型具有 11 個含權重的層，其中包括卷積層、全連接層和最大池化層；首先透過一系列卷積層來提取特徵，每一個卷積層都會經過卷積操作、批量標準化 (Batch Normalization) 和 ReLU 激活函數的處理。接著，模型將經過最大池化層，進行特徵的下採樣。隨著網絡深度的增加，卷積層的通道數也會從 64 增加至 512。在進行一系列的卷積和池化操作後，再經平均池化層進行全局特徵聚合，最後經全連接層進行分類，如圖 3.15 所示。

由於 Fashion-MNIST 為灰階影像資料集，因此 VGG11 的第一個積層層輸入通道數

必須為 1。而模型的全連接層輸出則應設定為 10，因為 Fashion-MNIST 數據集包含 10 個類別。

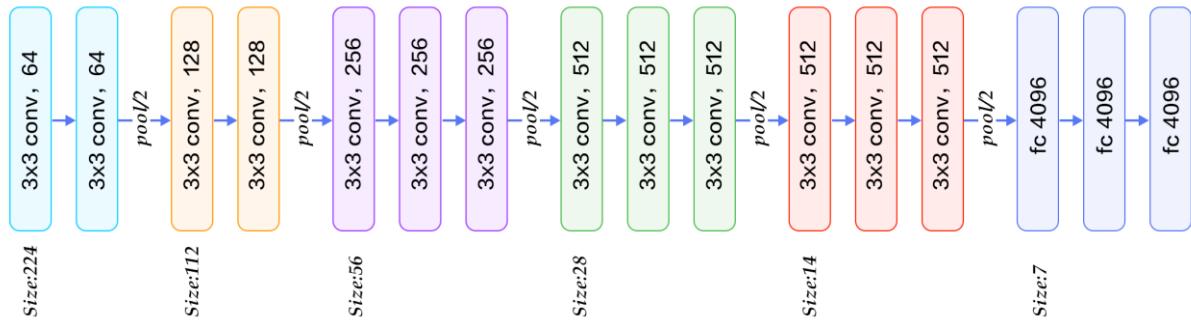


圖 3.15 Fashion MNIST 資料集分類實驗: VGG11 模型架構[59]

3.6.4 Fashion-MNIST 資料集的非獨立同分布設定與訓練過程

如表 3.4 所示，在 Fashion-MNIST 資料集的實驗中，採用了偏斜係數(skew parameter)方法，以進行非獨立同分布的資料分配。在這種方法中，客戶端的資料分布呈現偏斜。換句話說，每個客戶端只能在特定類別獲取大量資料，而只能從未指定的類別取得極少資料，如圖 3.16 所示。例如，圖中的第一個客戶端在類別一和類別二擁有超過 5500 張影像，但在其他類別中，最多只有不到 150 張影像。

Class/Label	Label 0	Label 2	Label 4	Label 6	Label 8
# number	5533	117	126	112	118
Class/Label	Label 1	Label 3	Label 5	Label 7	Label 9
# number	5507	123	114	128	122

圖 3.16 第一個客戶端在聯邦學習 Fashion-MINST 資料集實驗的各類別資料量

Fashion-MNIST 資料集中含有六萬筆訓練資料。在此實驗設定中，五個客戶端各自獲得了 12000 筆訓練資料。每個客戶端的資料主要來自指定的兩個類別，模擬出資料分布的偏斜性；而詳細的客戶端資料分布狀況，如圖 3.17 所示。此外，為了評估聯邦學習所產生的全域模型的對抗強健性，另外獨立了一萬筆的測試資料集。此設定不僅更接近現實狀況，也讓聯邦學習的性能表現得以深入探討與評估。

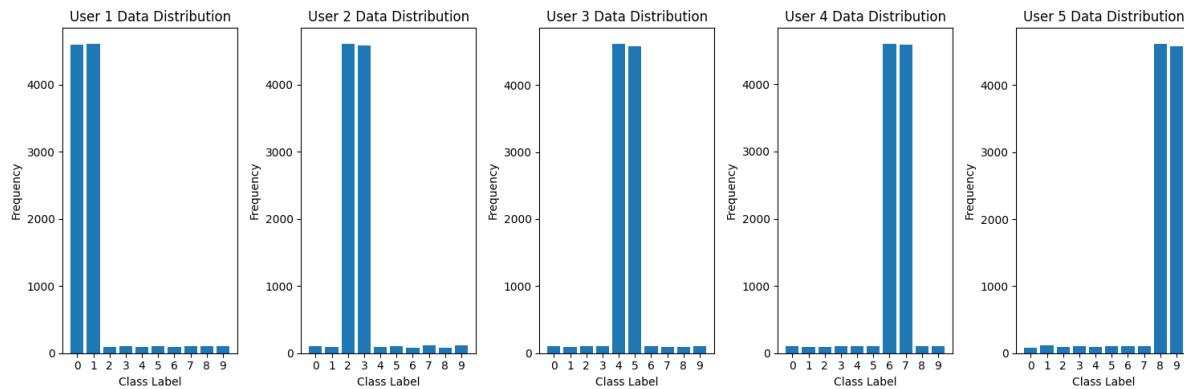


圖 3.17 Fashion-MNIST 資料集在聯邦學習之五個客戶端的非獨立同分布狀況

3.6.5 線寬資料集與 Auto-Encoder 模型

本研究中的線寬資料集為國立台北科技大學機械系教授的共同合作，該合作專案主要著重在 ITO 玻璃基板（Indium Tin Oxide）的剝蝕加工實驗[27, 60]。本實驗擁有極高的前瞻性，因為其潛力可以適用於如電極圖案化及手機面板製程等領域；在剝蝕加工實驗過程中，分別使用雷射機台與光纖機台進行剝蝕加工。隨後，透過共軛焦顯微鏡與專業量測軟體來評估量測結果，並將這些數據集整合進 Excel 電子表格。圖 3.18 顯示了實驗流程與蒐集線寬預測資料集的流程。

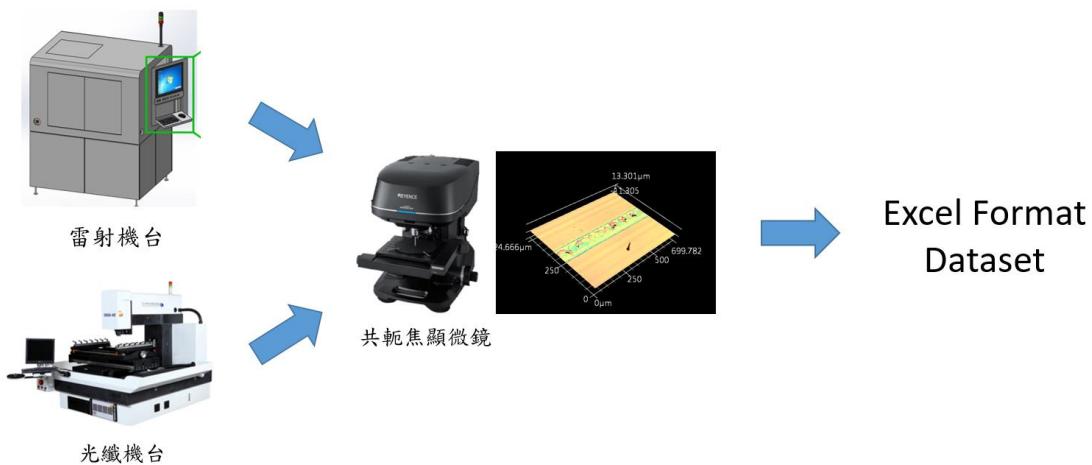


圖 3.18 線寬預測資料集的收集流程圖

線寬資料集涵蓋三個維度的資料，分別是雷射功率(power)、加工速度(speed)以及重複脈衝頻率(rate)。這些變數綜合考量用於預測線寬長度 (line width)，如圖 3.19 所示。

該實驗的主要目的是為了預測電極圖案化過程中的剝蝕加工線寬長度，藉此降低實驗成本。該問題為一典型的迴歸預測問題。資料集總共包含 16 萬筆資料，其中訓練集為 12 萬筆，測試集則包含 4 萬筆資料。

	A	B	C	D
1	power	speed	rate	line_width
2	2.89477	250	60	19555.7
3	6.68006	225	60	45189.6
4	6.53719	275	60	33726.6
5	4.64339	200	50	34317.1
6	2.60744	275	70	69176.8
7	2.80576	150	60	28666.3

圖 3.19 線寬資料集

本實驗使用的深度學習模型為自編碼器 (Auto-Encoder, AE) [61]。自編碼器是一種有效的非監督式學習模型，主要應用於提取數據的有效表現[66]，如圖 3.20 所示。該模型由編碼器和解碼器兩部分構成。編碼器的工作是將原始數據轉換為一種新的且通常是低維度的表現形式，而解碼器則將這種新的表現形式還原回原始數據。儘管自編碼器常被用於降維任務，但其實也能經過調整用於增維操作。

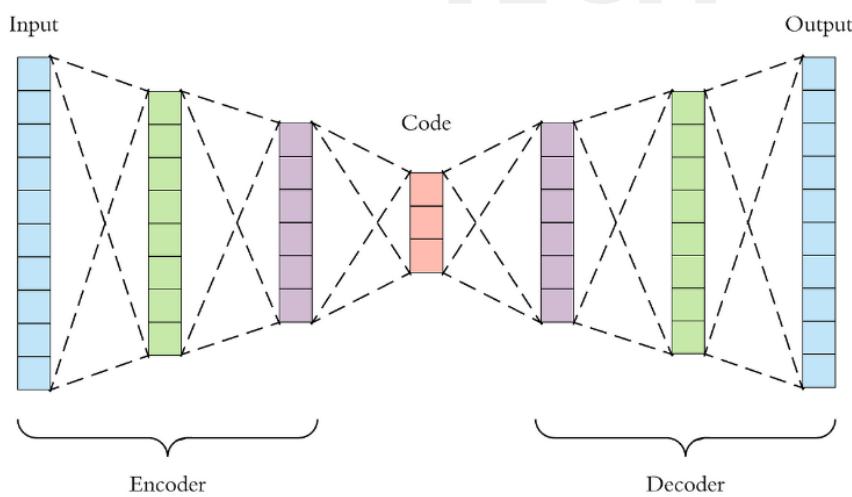


圖 3.20 Auto-Encoder: 自編碼器模型[61]

有鑑於線寬資料集的特徵維度相對較低，改以進行升維操作。然而，由於自編碼器的輸入與輸出維度必須一致，為了使模型具有預測線寬長度的功能，進一步增設了一個

由兩層組成的人工神經網路（2-ANN）迴歸模組來進行線寬預測，如圖 3.21 所示。這種結構不僅增加了模型的複雜度，也使預測能力能有所提升。

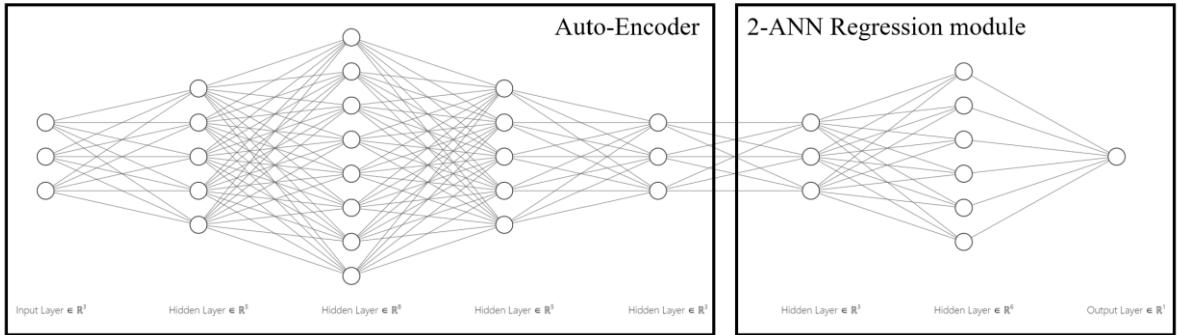


圖 3.21 線寬資料集的預測模型

3.6.6 線寬資料集的非獨立同分布設定與訓練過程

線寬資料集的「重複加工頻率」欄位包含 50 赫茲、60 赫茲和 70 赫茲。這三種數值。考慮到在不同的加工頻率下，雷射和光纖機台需要重新啟動並調整各項細微的機台數據，這是一項相當耗時且複雜的過程。因此，在實際的運作環境中，頻率的變動相對不易進行。為了使這一點在實驗設計中得到體現，如表 3.1 所示，以「重複加工頻率」作為依據，將線寬資料集進行非獨立同分布的分配。這種分配方式更貼近實際的操作情境，有助於更真實地模擬聯邦學習在實際環境下的運行表現。

線寬資料集中含有十二萬筆訓練資料。在此實驗設定中，共有十個客戶端參與。每個客戶端都根據不同的重複加工頻率取得相對應的資料子集，各自的數據量與資料分布均有所差異。圖 3.23 與圖 3.23 分別展示了各客戶端所獲取的資料數量及其資料分布情況。此外，為了評估聯邦學習所產生的全域模型的對抗強健性，我們獨立劃分四萬筆數據作為測試集。此設定不僅更接近現實情況，也為對聯邦學習的性能進行深入評估提供了便利。

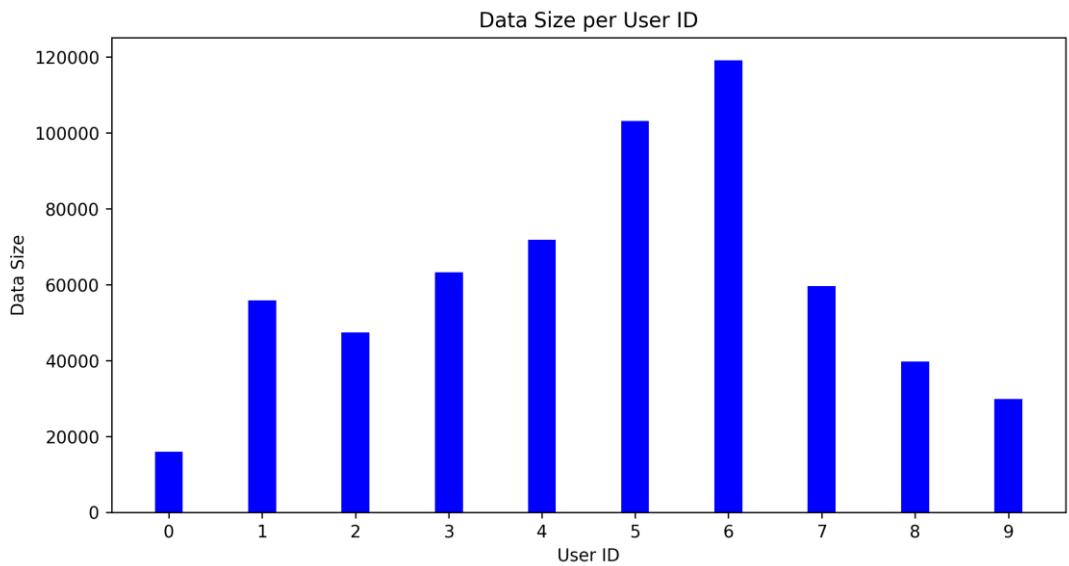


圖 3.22 線寬資料集之十個客戶端的訓練資料量長條圖

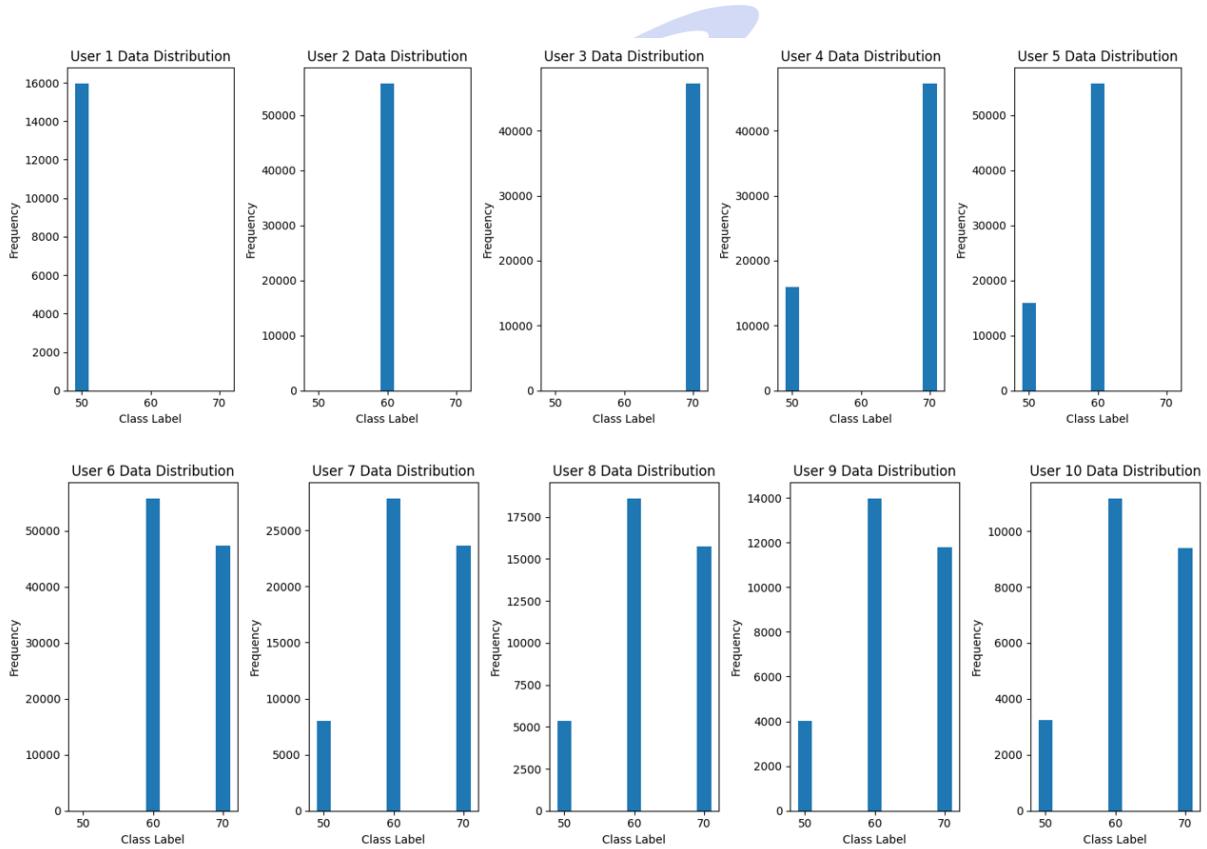


圖 3.23 線寬資料集在聯邦學習之十個客戶端的非獨立同分布狀況

3.6.7 電晶體資料集與 VAE 模型

電晶體數據集源自於與國立台北科技大學電子系的合作計劃。該計劃的焦點為應用變分自動編碼器（Variational Auto-Encoder，簡稱 VAE）來模擬非晶體銅鎵鋅氧化物 α -IGZO (Amorphous-Indium-Gallium-Zinc-Oxide) 薄膜電晶體（Thin Film Transistor，簡稱 TFT）的特性[28]。利用 VAE 模型，通過迴歸預測電性參數，來大幅度地減少元件開發的時間。本資料集共有 500 筆數據，訓練集為 375 筆，測試集 125 筆。

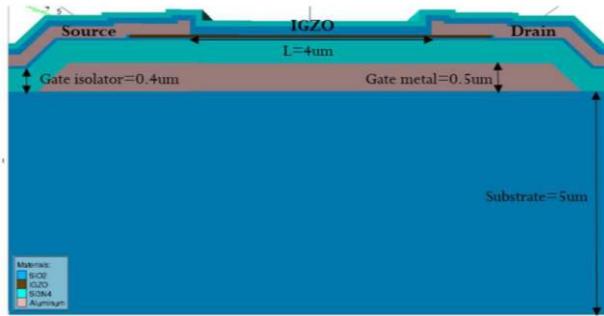


圖 3.24 a-IGZO TFT 的模擬橫切面視圖[28]

本實驗使用的深度學習模型為變分自動編碼器[62]。變分自動編碼器的主要分成兩大部分：編碼器與解碼器。在編碼器階段，VAE 會使用一個神經網路將輸入數據映射到隱含層(latent layer)的高維空間中。在解碼器階段，VAE 會從隱含變數的高維空間中抽樣出一個點，並且透過另一個神經網路將這個點轉換回原始數據的空間，如圖 3.25 所示。因此，在影像生成、異常檢測等領域被廣泛運用。然而，本實驗將 VAE 應用於預測 IGZO TFT 電晶體的電性參數，進一步拓寬了變分自動編碼器的應用範疇。

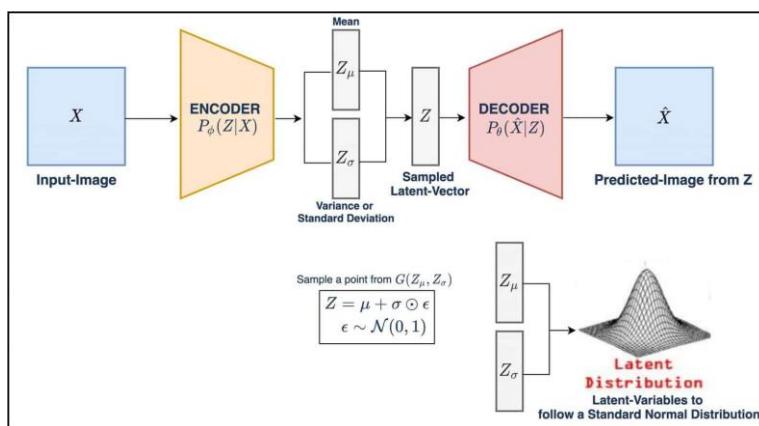


圖 3.25 Variational Auto-Encoder: 變分自動編碼器模型[61]

本實驗的預測模型，參考由 Chen 提出的電性預測方法[28]。將 α -IGZO 的能帶狀態密度(Subgap Density of State, DOS)做為模型的第一部分輸入資料，而第二部分的輸入資料為同質接面(HomoJunction)，第一部分資料與第二部分資料各自為 47 維。為了保證實驗的有效性，將兩大部分的資料一起匯入 VAE 模型進行預測[28, 63]，如所示。我們將這些數據萃取成一個 2 維的隱含層，並且在隱含層後加入 4-ANN 迴歸模型，以預測出特定的電性參數如: Ion、VTH、S.S.、DIBL、崩潰電壓等。

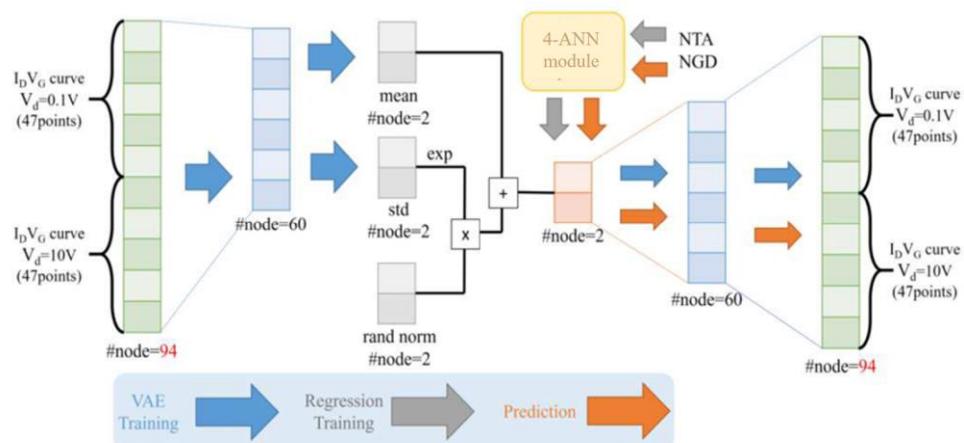


圖 3.26 電晶體資料集的訓練過程[28]

3.6.8 電晶體資料集的非獨立同分布設定與訓練過程

雖然電晶體資料集數量較少，但由於非獨立同分布在聯邦學習為最基礎的資料分布假設；本實驗中共有三個客戶端，每個客戶端各自從 375 筆的訓練資料集中取得資料，每個客戶端的資料量略為不同，如圖 3.27 所示。另外，為了評估聯邦學習所產生的全域模型的對抗強健性，獨立劃分 125 筆數據作為測試集。

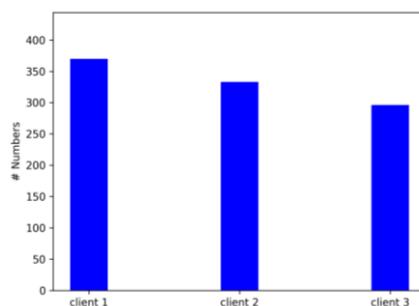


圖 3.27 電晶體資料集在聯邦學習之三個客戶端的非獨立同分布狀況

第四章 實驗結果與分析

4.1 實驗設備與環境

本章節將介紹實驗中使用的軟硬體環境及其規格。如表 4.1 所示；實驗平台的規格如下：作業系統為 Ubuntu 20.04 LTS，程式語言選用 Python。在深度學習框架的選擇上，選擇了 Torch 來實作聯邦學習系統，並利用 Pandas 來處理表格型態的資料，同時運用 PIL (Python Imaging Library) 來處理影像資訊。硬體上採用 Intel® Core™ i9-10900 CPU @ 2.80GHz 和 NVIDIA GeForce RTX™ 3070Ti 作為模型訓練平台。

表 4.1 實驗平台之規格

類型	項目	規格說明
硬體環境	CPU	Intel® Core™ i9-10900 CPU @ 2.80GHz
	GPU	NVIDIA GeForce RTX™ 3070Ti
	RAM	32GB
軟體環境	作業系統	Ubuntu 22.04.2 LTS(64-bit) / Window 11
	程式開發工具	Visual Studio Code
	程式語言	Python 3.10.11
	深度學習框架	Torch 1.12.1
	相依函式庫	PIL 9.5.0 Pandas 1.5.1 CUDA 11.3 cuDNN 8.3.2 Scikit-learn 1.1.3 numpy 1.23.4

4.2 實驗流程

本研究使用多種聯邦對抗訓練演算法進行實驗，包含 Fed_AT[49]、Fed_AT_Slack、Fed_BVA[23]、Fed_BVA_Slack、FedBVA_SAT 與 FedBVA_SAT_Slack。

首先，介紹 Fed_AT 與 Fed_AT_Slack。Fed_AT 為最基本的聯邦對抗訓練框架，直接在客戶端進行對抗訓練；Fed_AT_Slack 為在客戶端進行對抗訓練後，使用 J. Zhu 所提出的鬆弛聚合機制[54]進行模型聚合，因此，也被稱為 Fed_SFAT。

接著，探討 Fed_BVA 與 Fed_BVA_Slack。Fed_BVA 為在中心伺服器生成 BV-FGSM 全域對抗範例後，將全域對抗範例交給客戶端進行對抗訓練；Fed_BVA_Slack 為在 Fed_BVA 基礎之上，使用 J. Zhu 所提出的鬆弛聚合機制[54]進行模型聚合。

最後，談及 FedBVA_SAT 與 FedBVA_SAT_Slack。FedBVA_SAT 為在中心伺服器生成 BV-FGSM 全域對抗範例後，直接在中心伺服器執行對抗訓練的方法；FedBVA_SAT_Slack 為本研究提出的新方法，建立在 FedBVA_SAT 基礎之上，並使用反向鬆弛聚合機制(re-slack aggregation)進行模型聚合，期待實驗結果能證明本研究所提出的方法為最佳的解決方案。

無論採用哪種聯邦對抗訓練演算法，都是在未受到任何攻擊威脅的訓練階段(training phase)進行。當聯邦學習系統聚合出全域模型後，才進入測試階段。有別於訓練階段的安全環境，測試階段的測試集 D_{test} 會遭受未知的對抗樣本攻擊威脅，被攻擊污染後的測試集，稱為 \hat{D}_{test} 。在本實驗中，將 FGSM 攻擊與 PGD 攻擊視為未知的對抗樣本攻擊，並將汙染測試集 \hat{D}_{test} 與聯邦學習系統的全域模型一同進行對抗強健性數據指標的測試，如圖 4.1 所示。

對於分類問題，如 MNIST 與 Fashion-MNIST 資料集，選用準確度(Accuracy)作為衡量對抗強健性的數據指標；相反地，針對迴歸問題，如線寬資料集與電晶體資料集，則以決定係數(R-square score)作為評估對抗強健性的數據指標。

若某一聯邦對抗訓練演算法的數據指標較好，代表其對未知的對抗樣本攻擊具有較好的抵禦能力，也就是有較好的對抗強健性。

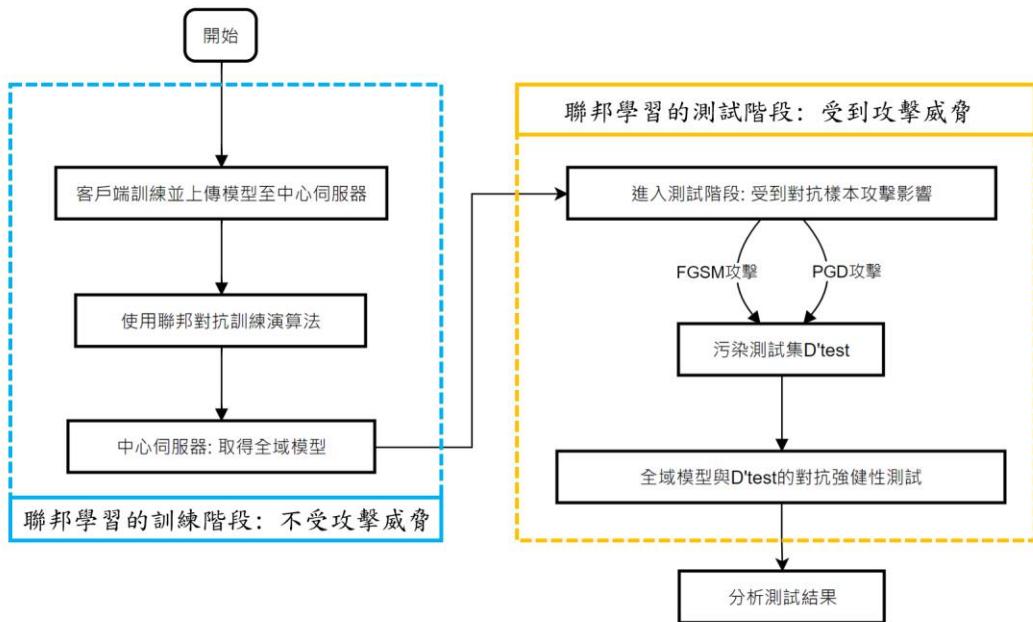


圖 4.1 實驗流程圖

4.3 實驗評估指標

介紹實驗評估指標，包含分類與迴歸問題的實驗參考指標。

4.3.1 分類問題的評估指標

本研究利用 MNIST 與 Fashion-MNIST 資料集進行分類實驗，並採用分類四大指標，分別為準確度 (Accuracy)、精確度 (Precision)、召回率 (Recall) 以及 F1-Score，以評估基於聯邦學習系統的分類問題的效能。四大指標皆由混淆矩陣(Confusion Matrix)的數值所構成，如表 4.2 所示，接下來將詳細介紹四大指標。

表 4.2 混淆矩陣與元素定義

	預測為陽性	預測為陰性
真實為陽性	真陽性(TP ,True Positive)	偽陰性(FN ,False Negative)
真實為陰性	偽陽性(FP ,False Positive)	真陰性(TN ,True Negative)

準確度 (Accuracy): 準確度是最直觀的評估指標，它是指模型預測正確的結果在所有預測中所佔的比例，也是聯邦學習領域最常使用的分類指標，如公式(4.1)所示。

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (4.1)$$

精確度 (Precision)：精確度是指模型在預測為正類別中實際為正類別的比例，簡單來說，就是判斷正面資料的準確率，如公式(4.2)所示。

$$\text{Precision} = \frac{(TP)}{(TP + FP)} \quad (4.2)$$

召回率 (Recall)：召回率是指在所有實際為正類別的樣本中，模型預測為正類別的比例，簡而言之就是在判斷成功的資料內，正面資料的準確率，如公式(4.3)所示。

$$\text{Recall} = \frac{(TP)}{(TP + FN)} \quad (4.3)$$

F1-Score：F1-Score 是一種同時考量了精確度和召回率的評估指標，它是精確度與召回率的調和平均數。F1-Score 的數值越接近 1，代表模型效能越好，如公式(4.4)所示。

$$\text{F1 - Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \quad (4.4)$$

在本研究的分類問題中，我們不僅僅注重準確度，同時也將精確度、召回率以及 F1-Score 納入考量，以全面評估聯邦學習演算法的表現。

4.3.2 迴歸問題的評估指標

迴歸問題的評估指標主要著重於預測值與真實數值之間的差異，與分類問題的評訓練時，皆選用了均方誤差(Mean Square Error ,MSE)作為損失函數，為了避免評估指標的重複性，本研究的所有迴歸指標主要採用決定係數 (R-square, R^2) 與均方根誤差 (Root Mean Square Error, RMSE) 來進行模型的評估。以下將詳細介紹這兩大指標。

決定係數 (R-squared, R^2)：決定係數，又稱為 R-平方，是用來描述模型對於數據的解釋能力。R 平方的範圍從 0 到 1，當數值越接近 1，代表越能解釋數據的變異性；反之，則為越無法解釋數據的變異性。當 R 平方小於零時，意味著該模型對數據的解釋能力非常差，無法提供任何有意義的解釋或預測。

R 平方值的計算公式為：

$$R^2 = 1 - (SSR/SST) \quad (4.5)$$

其中，SSR 是殘差平方和，也就是所有誤差的平方和；而 SST 為總變異平方和，反映了數據的總變異。

均方根誤差（Root Mean Square Error, RMSE）：RMSE 是一種常用來衡量模型預測誤差的指標。RMSE 簡單來說就是對誤差進行平均的運算，對於較大的誤差具有較高的懲罰。RMSE 的計算方法，如公式(4.6)所示：

$$RMSE = \sqrt{\sum_{i=1}^n (\hat{y}_i - y_i)^2 / n} \quad (4.6)$$

其中 n 為總數， \hat{y}_i 為預測值，而 y_i 為實際值。

R 平方提供了對模型解釋變異性的度量，而 RMSE 則對模型預測的誤差進行評估，使用不同的迴歸評估指標，可以讓我們從不同角度分析模型的預測能力。在實驗過程中，將使用這兩種指標來評估並比較不同模型的效能。

4.4 FedBVA_SAT_Slack 鬆弛聚合機制的實驗分析

4.4.1 鬆弛聚合機制的運作分析

在 FedBVA_SAT_Slack 聯邦對抗訓練演算法中，採取了一種將最大損失值的模型進行特殊加權的策略，並希望在每個溝通輪次都能選取到專門對抗訓練的模型（AT Module）。首先，先觀察此演算法在聚合所有客戶端模型及對抗訓練模型過程中的損失值變動情形，以 MNIST 資料集為例，如圖 4.2 與圖 4.3 所示。

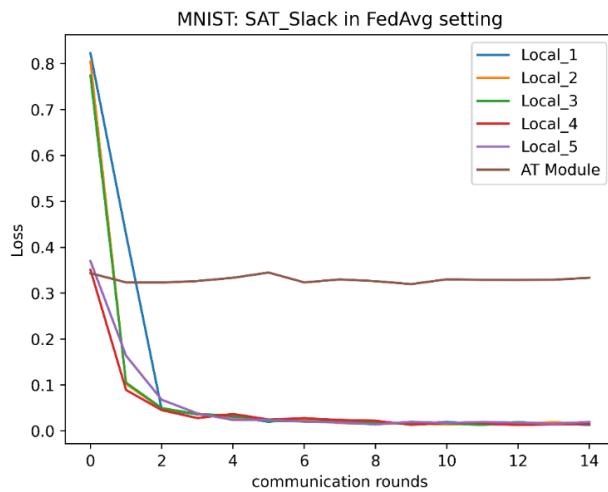


圖 4.2 MNIST 資料集: FedBVA_SAT_Slack 之所有模型的損失值折線圖(FedAvg)

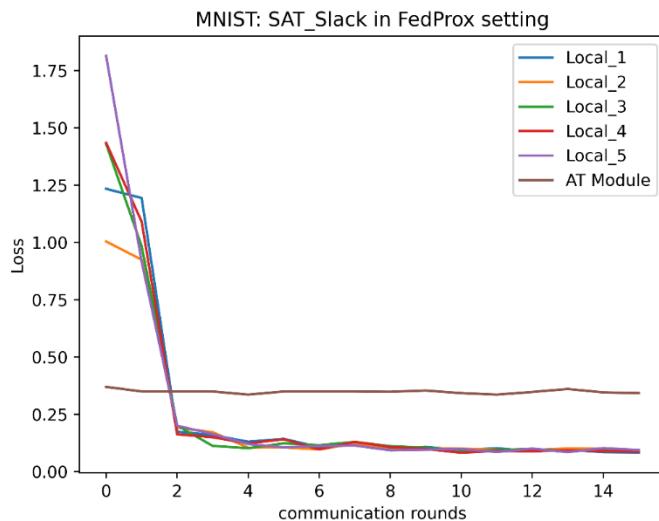


圖 4.3 MNIST 資料集: FedBVA_SAT_Slack 之所有模型的損失值折線圖(FedProx)

圖 4.2 呈現的是以 FedAvg 聚合方法為基礎的實驗結果，而圖 4.3 則描繪了基於 FedProx 聚合方法的相應結果。在這兩種情況下，我們可以觀察到一個共同的現象：當溝通輪次(communication rounds)超過兩次，所有客戶端的損失值均呈現大幅下降的趨勢。然而，只有代表對抗訓練模型損失值的棕色線段，無法呈現下滑趨勢而維持一定的水平，這是因為該模型是專門訓練受到高度攻擊的全域對抗範例。因此，對抗訓練模型的損失值較高且無法下降是合理的。

由此可知，FedBVA_SAT_Slack 演算法能在每次模型聚合時，有望穩定地選取出具有最大損失值的對抗訓練模型進行鬆弛加權，進而提升模型的對抗強健性。

在後續章節中，將針對四種資料集進行更多不同攻擊強度下的對抗強健性評估，進一步驗證 FedBVA_SAT_Slack 演算法在提升全域模型對抗強健性方面的優勢。

4.4.2 是否啟用鬆弛聚合機制的實驗分析

以下為 MNIST 資料集實驗中，基於 FedAvg 與 FedProx 聚合方法的 FedBVA_SAT 與 FedBVA_SAT_Slack 演算法之乾淨準確度(Natural Accuracy)的對比。

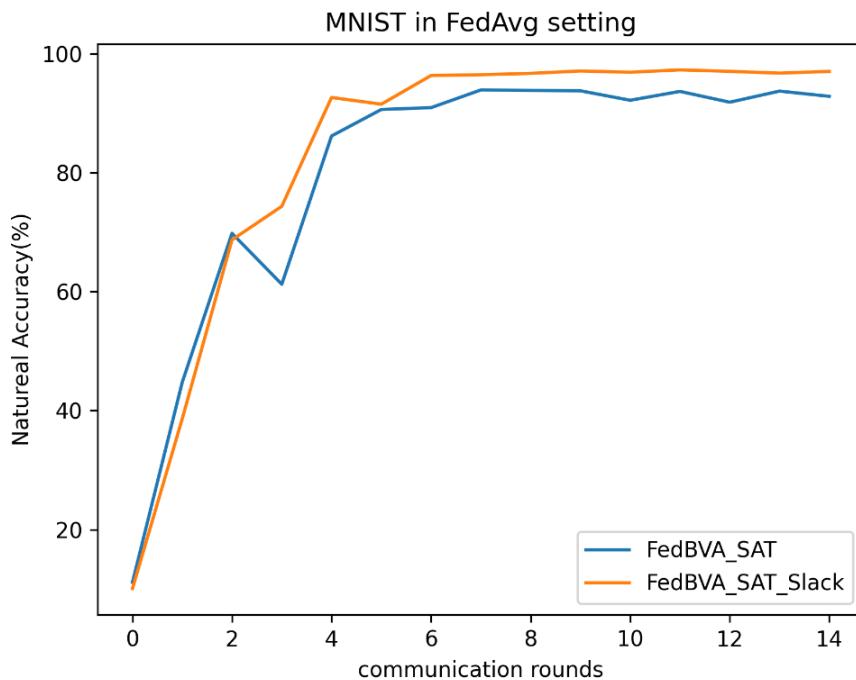


圖 4.4 MNIST:是否執行鬆弛聚合機制的準確度比較(FedAvg)

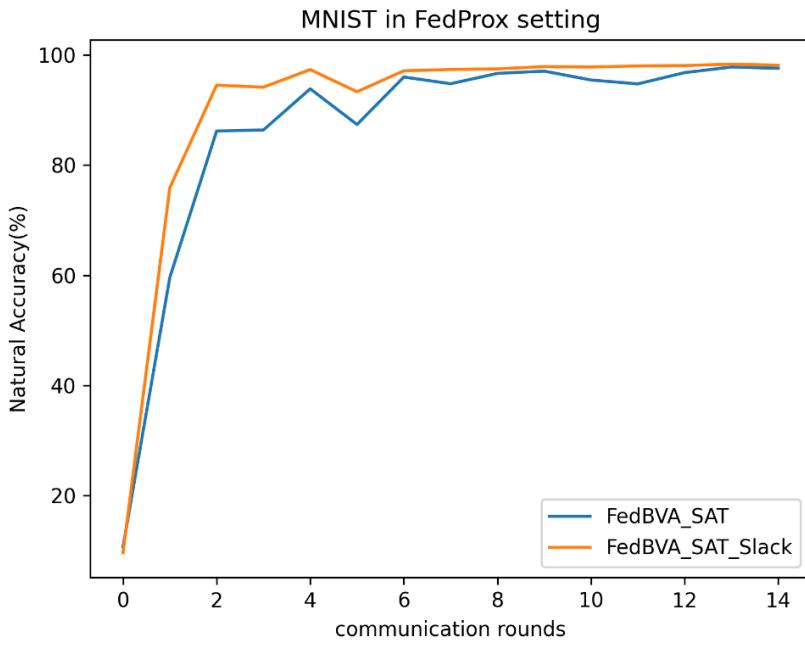


圖 4.5 MNIST:是否執行鬆弛聚合機制的準確度比較(FedProx)

圖 4.4 呈現的是以 FedAvg 聚合方法為基礎的實驗結果，而圖 4.5 則描繪了基於 FedProx 聚合方法的相應結果。在 FedAvg 與 FedProx 聚合方法下，有執行鬆弛聚合機制的 FedBVA_SAT_Slack 演算法，在聚合過程中的乾淨準確度較高。這代表著，FedBVA_SAT_Slack 在聯邦學習系統聚合全域模型的穩定性較高。

，P 代表對於損失函數最大的模型加權值，而 P' 代表了其他模型的加權值，但由於 FedBVA_SAT 並未實施鬆弛加權機制，P 與 P' 皆為 1。而 FGSM 與 PGD-20-10 欄位，則代表不同對抗樣本攻擊方法，這兩種方法的攻擊強度 ($\text{epsilon}, \epsilon$) 對分別設定為 0.3 和 0.2。

表 4.3 為 MNIST 資料集於進行 FedAvg 聚合方法的基礎上，FedBVA_SAT 和 FedBVA_SAT_Slack 兩種演算法經過 20 次聚合後產生的全域模型的準確度比較。比較這兩種演算法在面對乾淨數據以及未知的對抗樣本攻擊時的測試準確度。從結果來看，FedBVA_SAT_Slack 不只在處理乾淨數據時展現出更高的準確度，而且在面對各種不同的對抗樣本攻擊時也呈現出更佳的表現，顯示其具有更強的對抗強健性。

表 4.3 欄位 P/P' 中，P 代表對於損失函數最大的模型加權值，而 P' 代表了其他模型

的加權值，但由於 FedBVA_SAT 並未實施鬆弛加權機制， P 與 P' 皆為 1。而 FGSM 與 PGD-20-10 欄位，則代表不同對抗樣本攻擊方法，這兩種方法的攻擊強度 (epsilon , ϵ) 對分別設定為 0.3 和 0.2。

表 4.3 MNIST:有無進行鬆弛聚合機制的對抗強健性比較

MNIST Dataset	P/P'	Clean	FGSM	PGD-20-10
FedBVA_SAT	1.0/1.0	0.948	0.304	0.253
FedBVA_SAT_Slack	1.2/1.0	0.968	0.611	0.591

由以上分析，可以得出的結論是：在聯邦對抗訓練演算法中，先在中心伺服器對特定的資料的對抗訓練，再透過鬆弛聚合機制，能有效提升全域模型的準確度和穩定性。此外，該演算法在應對對抗樣本攻擊時，也表現出較高的強健性。

在後續章節中，將針對四種資料集進行更多不同攻擊強度下的對抗強健性評估，進一步驗證 FedBVA_SAT_Slack 演算法在提升全域模型對抗強健性方面的優勢。

4.4.3 Fed_BVA 與 FedBVA_SAT_Slack 輔助資料集的數量比較

Fed_BVA[23]和 FedBVA_SAT_Slack 演算法在生成全域對抗範例時，必須在中心伺服器上放置一份少量的輔助資料集作為輔助。新提出的 FedBVA_SAT_Slack 演算法，由於在中心伺服器上使用了鬆弛聚合機制，使得所需的輔助資料集數量可以進一步降低。

為了證實此論點，使用資料集數量超過萬筆的 MNIST 資料集、Fashion-MNIST 資料集以及線寬資料集進行實驗，並嘗試將 FedBVA_SAT_Slack 的輔助資料集數量降到最低。然而，由於電晶體資料集的數量相對較少，因此，在電晶體資料集實驗中，Fed_BVA 與 FedBVA_SAT_Slack 兩種演算法的輔助資料量皆設定為 50 筆。

在 Fed_BVA 演算法中，客戶端資料量與輔助資料量的比例設定為 64:600，而對於 FedBVA_SAT_Slack 演算法，將輔助資料集數量統一設定為 200。以下的三個表格分別展示了不同資料集在使用 Fed_BVA 和 FedBVA_SAT_Slack 兩種演算法時，其輔助資料集對對抗強健性表現的影響。

表 4.4 MNIST: Fed_BVA 與 FedBVA_SAT_Slack 對抗強健性比較

MNIST Dataset	輔助資料集數量	Clean	FGSM	PGD-20-10
Fed_BVA[23]	2560 筆	0.948	0.546	0.311
FedBVA_SAT_Slack	200 筆	0.968	0.611	0.458

在 MNIST 資料集中，每個客戶端的資料量約為 24000，因此 Fed_BVA 演算法的資料量為 2560，而表 4.4 中的 FGSM 與 PGD-20-10 欄位，分別代表 FGSM $\epsilon = 0.3$ 和 PGD-20-10 $\epsilon = 0.2$ ，而評估指標為準確度。

表 4.5 Fashion-MNIST: Fed_BVA 與 FedBVA_SAT_Slack 對抗強健性比較

Fashion-MNIST	輔助資料集數量	Clean	FGSM	PGD-20-10
Fed_BVA[23]	1280 筆	0.915	0.308	0.109
FedBVA_SAT_Slack	200 筆	0.918	0.368	0.213

在 Fashion-MNIST 資料集中，每個客戶端的資料量約為 12000，因此 Fed_BVA 演算法的資料量為 1280，表 4.5 中的 FGSM 與 PGD-20-10 欄位，分別代表 FGSM $\epsilon = 0.15$ 和 PGD-20-10 $\epsilon = 0.2$ ，而評估指標為準確度。

表 4.6 線寬資料集: Fed_BVA 與 FedBVA_SAT_Slack 對抗強健性比較

線寬資料集	輔助資料集數量	Clean	FGSM	PGD-20-10
Fed_BVA[23]	1280 筆	0.654	0.358	0.184
FedBVA_SAT_Slack	200 筆	0.968	0.842	0.74

在線寬資料集中，Fed_BVA 演算法的輔助資料量為 1280，表 4.6 中的 FGSM 與 PGD-20-10 欄位，分別代表 FGSM $\epsilon = 0.02$ 和 PGD-20-10 $\epsilon = 0.1$ ，評估指標為 R-平方。

結果顯示，與 Fed_BVA 相比，FedBVA_SAT_Slack 在輔助資料集數量更少的情況下，仍能在乾淨準確度和受攻擊準確度上呈現出更佳的表現。這項結果充分證明了 FedBVA_SAT_Slack 在使用較少輔助資料集的情況下，仍能維持優秀的對抗強健性。

最後，表 4.4、表 4.5 與表 4.6 均以 FedAvg 聚合方法進行分析，在後續章節中也將提供 FedProx 聚合方法的數據。

4.5 MNIST 資料集的實驗結果與分析

本實驗以 FedAvg 與 FedProx 聚合方法為基礎，進行了六種不同聯邦對抗訓練演算法的對抗強健性比較，比較的演算法包含 Fed_AT、Fed_AT_Slack、Fed_BVA、Fed_BVA_Slack、FedBVA_SAT 與 FedBVA_SAT_Slack，章節 4.2 已有完整的介紹。

在 MNIST 資料集實驗中，共有五個參與客戶端，並針對在中心伺服器進行了 15 次聚合後所生成的全域模型之對抗強健性進行深入分析，此分析涵蓋了全域模型對於乾淨資料的準確度，與對於不同強度 FGSM 與 PGD 攻擊的抵抗能力。

由於 MNIST 資料集為分類問題，選擇使用準確度與 F1-Score 作為評估指標；並且使用的模型為 CNN 模型。接下來的段落中，會以 FedAvg 與 FedProx 這兩個不同聚合方法進行實驗分析。

4.5.1 FedAvg 聚合方法下的不同聯邦對抗訓練演算法的訓練過程

FedAvg 表示在沒有採取任何防禦策略的情況下，僅使用標準的 FedAvg (Federated Averaging) 聚合方法進行實驗。而 Fed_AT、Fed_AT_Slack、Fed_BVA、FedBVA_SAT 以及 FedBVA_SAT_Slack 則是六種不同的聯邦對抗訓練演算法，共同目標為在聯邦學習系統收斂時，使全域模型的準確度，能夠盡可能接近未施加任何防禦策略下的 FedAvg 準確度。

MNIST in FedAvg and Non-IID setting.

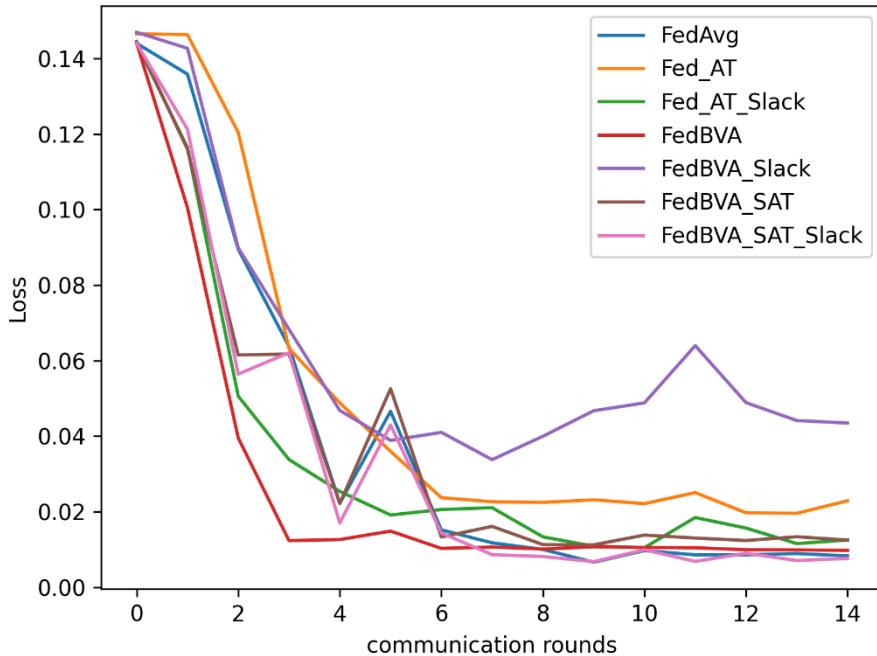


圖 4.6 MNIST:不同聯邦對抗訓練系統的聚合過程損失折線圖(FedAvg)

圖 4.6 展示了 FedAvg 與其他六種聯邦對抗訓練演算法在中心伺服器進行 15 次聚合後的損失變化。觀察損失值變化後，可以發現直接在客戶端執行對抗訓練的 Fed_BVA_Slack 演算法並未能有效地降低損失值。因此，如表 4.7 所示，這種演算法的準確度和 F1 分數明顯低於未經任何防禦手段的 FedAvg。這可能是因為在客戶端直接進行對抗訓練時，所有客戶端模型在中心伺服器進行聚合時，可能會遺失掉乾淨資料的特徵。這一觀察也間接反映出直接在客戶端進行對抗訓練的聯邦學習系統，仍存在提升的空間。

表 4.7 MNIST:不同聯邦系統最終全域模型之準確度與 F1-Score 比較(FedAvg)

	FedAvg	Fed_AT	Fed_AT_Slack	Fed_BVA	Fed_BVA_Slack	FedBVA_SAT	FedBVA_SAT_Slack
Accuracy	0.913	0.898	0.833	0.968	0.768	0.941	0.970
F1 Score	0.912	0.897	0.828	0.968	0.793	0.941	0.970

表 4.7 中可以清晰地顯示 Fed_BVA、FedBVA_SAT 與 FedBVA_SAT_Slack 在乾淨資

料的準確度與 F1-Score 指標上，都略優於未經對抗訓練防禦機制的 FedAvg，這正是希望看到的實驗結果。

4.5.2 FedAvg 聚合方法下的面對 FGSM 攻擊的對抗強健性比較

本章節中，比較了上述提到的五種主要聯邦對抗訓練演算法在面對不同的 FGSM 和 PGD 攻擊時的對抗強健性。評估指標為準確度(Accuracy)。以未進行任何防禦手段的 FedAvg 作為基準，如果其他的聯邦對抗訓練演算法在面對各種攻擊方法和攻擊強度時，其準確度仍然高於 FedAvg，便可認定該演算法提升了全域模型的對抗強健性。

首先，討論 FGSM 攻擊的對抗強健性比較。圖 4.7 與表 4.8 為不同聯邦對抗訓練系統經過 15 次聚合後的最終全域模型，對於 FGSM 攻擊強度在 $\epsilon = 0.05$ 至 0.40 範圍內的測試準確度。在該圖中，以 FedAvg(藍色線段)作為判斷標準，可以觀察到 Fed_BVA[23](紅色線段)與本論文提出的 FedBVA_SAT_Slack(粉色線段)方法在所有的攻擊強度下，其準確度皆比藍色線段還高，表示了這兩種演算法能夠提升全域模型的對抗強健性。換而言之，其他低於藍色線段的演算法，代表其性能較差。

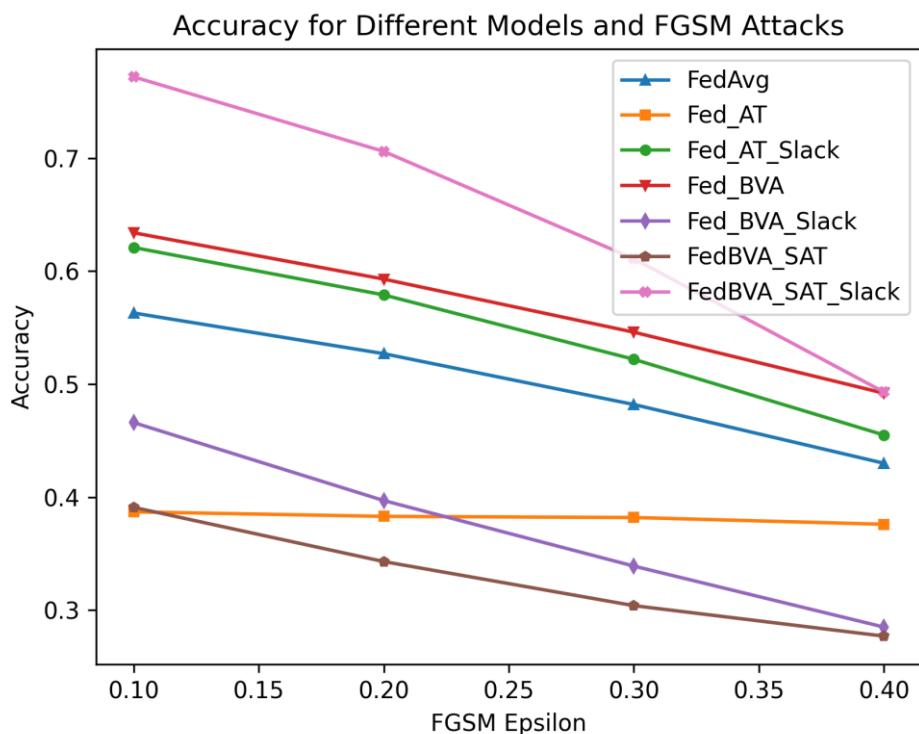


圖 4.7 MNIST 資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedAvg)

表 4.8 MNIST 資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedAvg)

FGSM eps 聯邦防禦方法	clean	0.1	0.2	0.3	0.4
FedAvg[3]	0.913	0.563	0.527	0.482	0.430
Fed_AT[49]	0.898	0.582	0.459	0.334	0.253
Fed_AT_Slack[54]	0.833	0.621	0.579	0.522	0.455
Fed_BVA[23]	0.968	0.634	0.593	0.546	0.492
Fed_BVA_Slack	0.768	0.492	0.423	0.365	0.312
FedBVA_SAT	0.941	0.391	0.343	0.304	0.277
FedBVA_SAT_Slack (proposed method)	0.970 (+0.020)	0.772 (+0.055)	0.706 (+0.051)	0.611 (+0.032)	0.493 (+0.001)

當攻擊強度小於 0.3 時，FedBVA_SAT_Slack 的準確度優於 Fed_BVA，然而，當攻擊強度達到 0.4 時，兩演算法的準確度極度接近，極有可能在攻擊強度更強的狀況下發生黃金交叉，這說明了 FedBVA_SAT_Slack 演算法對於高強度攻擊的防禦能力，還有進一步空間。

儘管如此，FedBVA_SAT_Slack 在乾淨數據的準確度以及輕度攻擊下的準確度均優於 Fed_BVA，我們認為我們提出的方法仍然具有其有效性。這證明了 FedBVA_SAT_Slack 在提升模型的對抗強健性方面能具有一定的潛力。

4.5.3 FedAvg 聚合方法下的面對 PGD 攻擊的對抗強健性比較

接著，來探討 PGD 攻擊的狀況。圖 4.8 與為不同聯邦對抗訓練系統經過 15 次聚合後的最終全域模型，對於 PGD 攻擊強度在 $\epsilon = 0.05$ 至 0.3 範圍內的測試準確度。在該圖中，以 FedAvg(藍色線段)作為判斷標準，會發現 Fed_AT_Slack(綠色線段)與

FedBVA_SAT_Slack (粉色線段)高於 FedAvg，代表這兩種方法的效果明顯比較好，但 $\epsilon > 0.15$ 後，Fed_AT_Slack 的準確度比本研究提出的 FedBVA_SAT_Slack 還高，但由於 FedBVA_SAT_Slack 的乾淨準確度為 0.970，明顯優於 Fed_AT_Slack 的 0.833；因此，儘管在面對高強度攻擊時 Fed_AT_Slack 的準確度較高，但考慮到 FedBVA_SAT_Slack 在乾淨環境下的優異表現，FedBVA_SAT_Slack 演算法的效能和效果是令人滿意的。

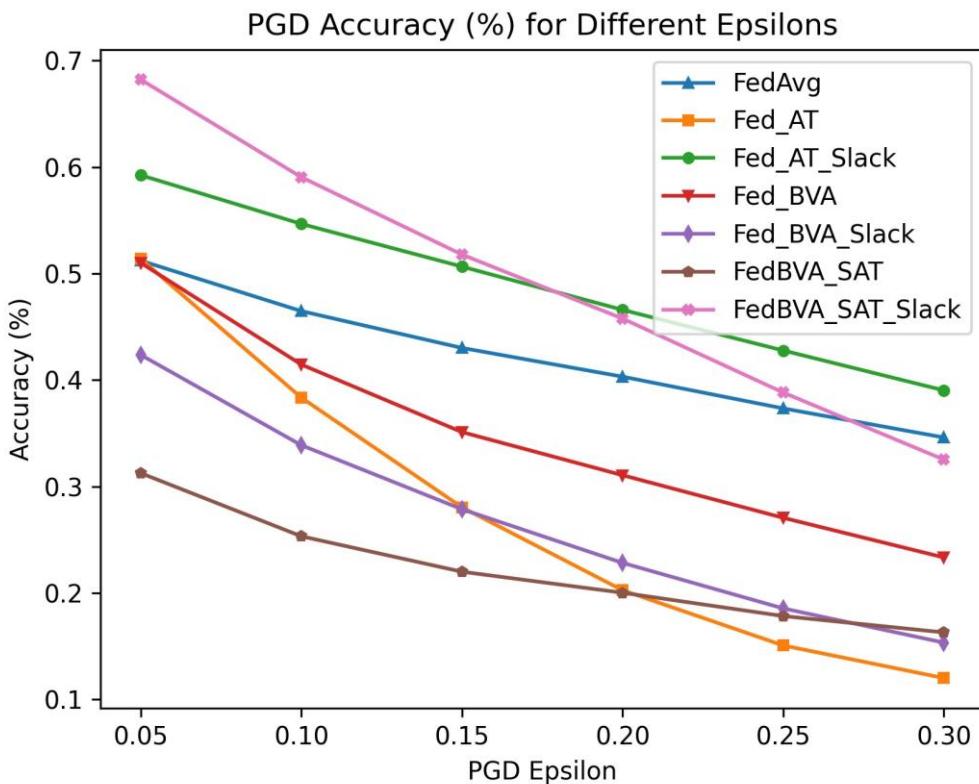


圖 4.8 MNIST:各聯邦對抗訓練系統對 PGD 攻擊強度的準確度折線圖(FedAvg)

表 4.9 各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedAvg)

聯邦防禦方法 \ PGD eps	clean	0.05	0.1	0.15	0.2	0.25	0.3
FedAvg[3]	0.913	0.512	0.465	0.430	0.403	0.373	0.346
Fed_AT[49]	0.898	0.514	0.384	0.280	0.203	0.151	0.120
Fed_AT_Slack[54]	0.833	0.593	0.547	0.507	0.466	0.428	0.390

Fed_BVA[23]	0.968	0.510	0.415	0.351	0.311	0.271	0.233
Fed_BVA_Slack	0.768	0.424	0.339	0.279	0.228	0.186	0.153
FedBVA_SAT	0.941	0.313	0.253	0.220	0.200	0.178	0.163
FedBVA_SAT_Slack (proposed method)	0.970 (+0.020)	0.682 (+0.089)	0.591 (+0.044)	0.518 (+0.011)	0.458 (-0.036)	0.388 (-0.04)	0.325 (-0.065)

4.5.4 FedProx 聚合方法下的不同聯邦對抗訓練演算法的訓練過程

誠如前述的 FedAvg，FedProx 表示在沒有採取任何防禦策略的情況下，僅使用標準的 FedProx 聚合方法進行實驗。而 Fed_AT、Fed_AT_Slack、Fed_BVA、Fed_BVA_Slack、FedBVA_SAT 以及 FedBVA_SAT_Slack 則是六種不同的聯邦對抗訓練演算法，共同目標為在聯邦學習系統收斂時，使全域模型的準確度，能夠盡可能接近未施加任何防禦策略下的 FedProx 準確度。

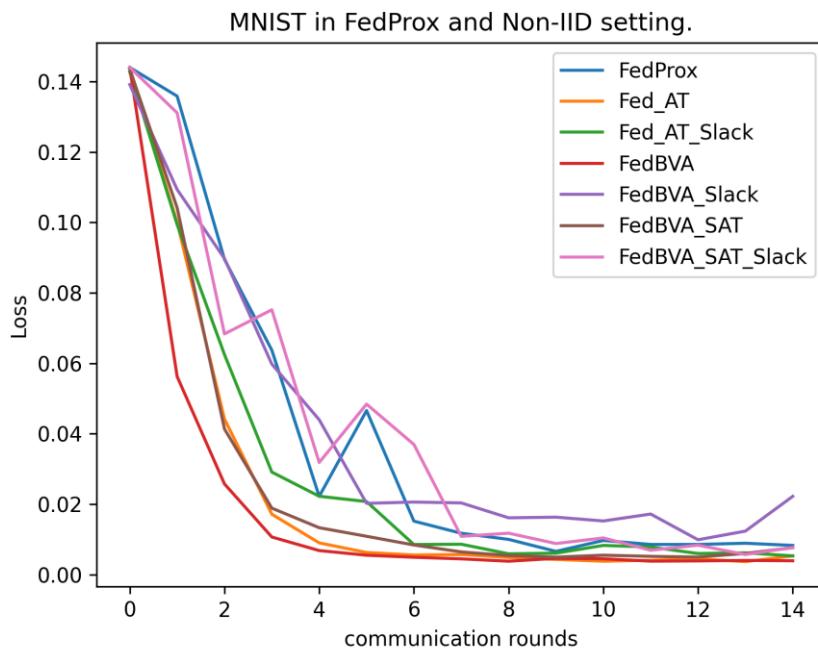


圖 4.9 MNIST:不同聯邦對抗訓練系統的聚合過程損失折線圖(FedProx)

表 4.10 MNIST:不同聯邦系統最終全域模型之準確度與 F1-Score 比較(FedProx)

	FedAvg	Fed_AT	Fed_AT_Slack	Fed_BVA	Fed_BVA_Slack	FedBVA_SAT	FedBVA_SAT_Slack
Accuracy	0.962	0.969	0.937	0.980	0.954	0.977	0.981
F1 Score	0.961	0.969	0.936	0.980	0.953	0.977	0.981

在表 4.10 中可以清晰地顯示所有演算法皆趨近或略優於未經對抗訓練防禦機制的 FedProx，這正是希望看到的實驗結果。

4.5.5 FedProx 聚合方法下的面對 FGSM 攻擊的對抗強健性比較

誠如先前所述，本實驗數據以未進行任何防禦手段的 FedProx 作為基準。即便在受到不同強度的 FGSM 攻擊下，只要準確度仍然高於 FedProx，就可以證明該演算法成功地提升了經過 15 次聚合後的聯邦對抗訓練全域模型的對抗強健性。

首先，討論 FGSM 攻擊的對抗強健性比較。圖 4.10 與表 4.11 為不同聯邦對抗訓練系統經過 15 次聚合後的最終全域模型，對於 FGSM 攻擊強度在 $\epsilon = 0.1$ 至 0.5 範圍內的測試準確度。

從上述的圖與表中，以 FedProx (藍色線段)作為判斷標準。可以觀察到當攻擊強度小於 0.3 時，除了 Fed_AT(橘色線段)與 Fed_AT_Slack(綠色線段)演算法的準確度比藍色線段低，其他聯邦對抗訓練演算法的準確度都比標準值還高。而當攻擊強度大於 0.3 後，所有聯邦對抗訓練演算法的準確度都比標準值高，代表這些方法都能對在 FedProx 聚合方法下有效提升全域模型的對抗強健性。其中，本研究提出的 FedBVA_SAT_Slack 在所有攻擊情境下的準確度表現均最為優秀，這明確地證明了該演算法在提升對抗強健性方面具有顯著的效能。

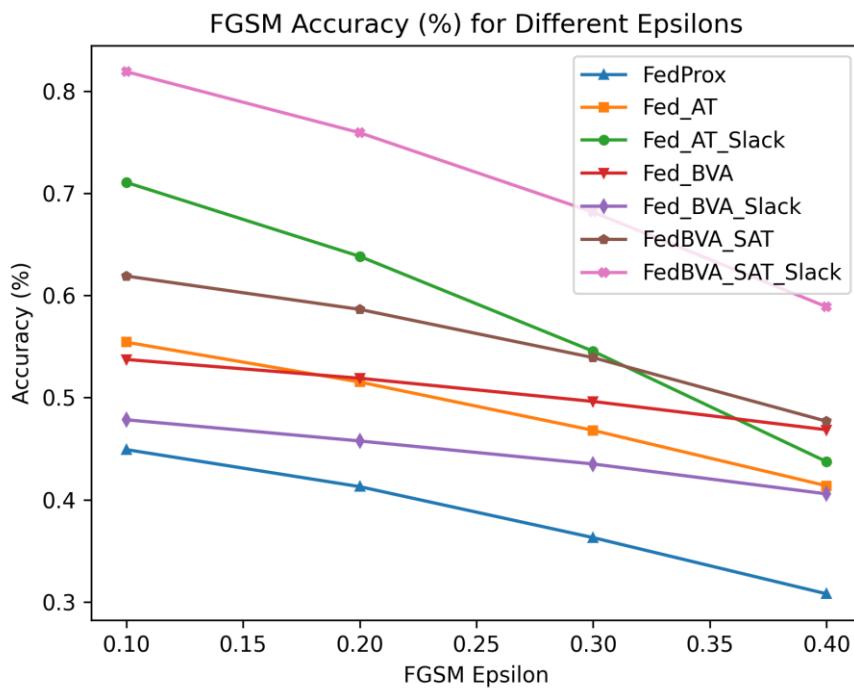


圖 4.10 MNIST: 各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedProx)

表 4.11 MNIST: 各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedProx)

聯邦防禦方法 \ FGSM eps	clean	0.1	0.2	0.3	0.4
聯邦防禦方法					
FedProx[31]	0.962	0.449	0.413	0.363	0.308
Fed_AT[49]	0.969	0.554	0.515	0.468	0.414
Fed_AT_Slack[54]	0.937	0.711	0.638	0.545	0.437
Fed_BVA[23]	0.980	0.537	0.519	0.496	0.469
Fed_BVA_Slack	0.954	0.478	0.458	0.435	0.406
FedBVA_SAT	0.977	0.619	0.586	0.539	0.477
FedBVA_SAT_Slack (proposed method)	0.981 (+0.001)	0.819 (+0.108)	0.76 (+0.122)	0.682 (+0.143)	0.589 (+0.112)

4.5.6 FedProx 聚合方法下的面對 PGD 攻擊的對抗強健性比較

接著，探討 PGD 攻擊的對抗強健性比較。圖 4.11 為不同聯邦對抗訓練系統經過 15 次聚合後的最終全域模型，對 PGD 攻擊強度在 $\epsilon = 0.05$ 至 0.25 範圍內的測試準確度。

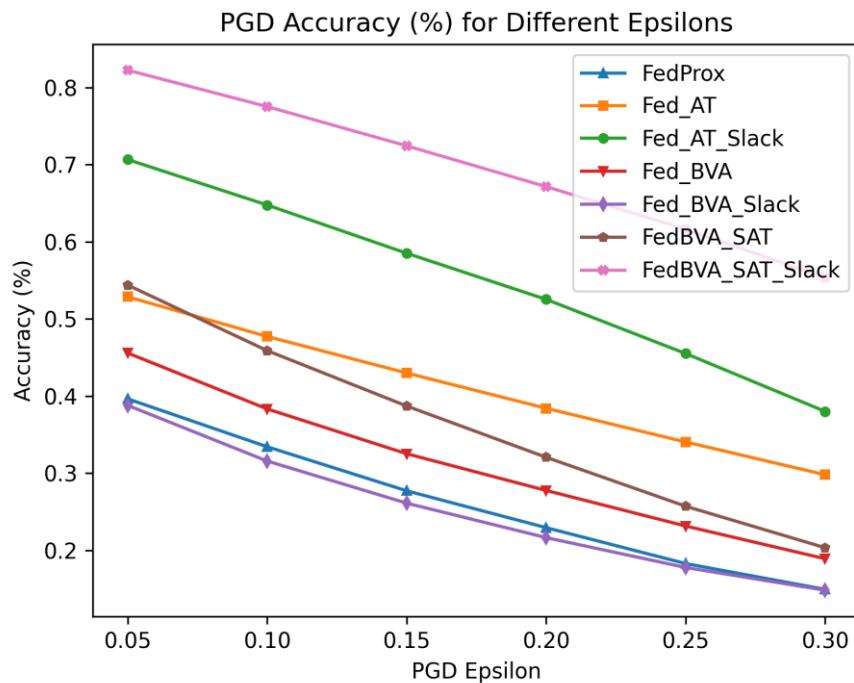


圖 4.11 MNIST: 各聯邦對抗訓練系統對 PGD 攻擊的準確度折線圖(FedProx)

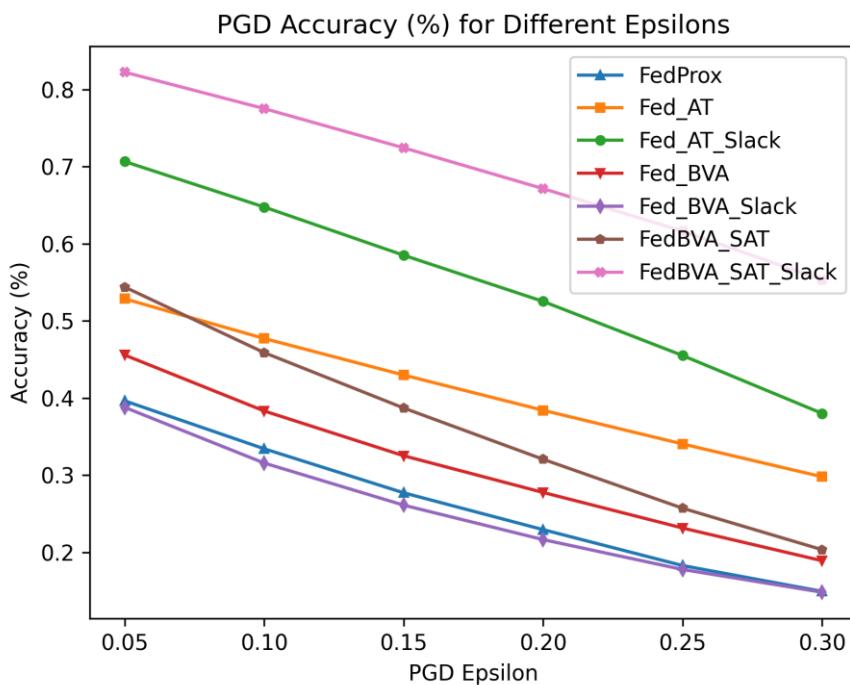


圖 4.11 與表 4.12 中，可經觀察得知，所有對抗訓練演算法在攻擊強度大於 0.1 後，準確度皆大於被視為比較基準的 FedProx(藍色線段)，並且本研究提出的 FedBVA_SAT_Slack(粉色線段)在所有攻擊情境下，準確度表現明顯優於其他方法，這明確地證明了該演算法在提升對抗強健性方面具有顯著的效能。

表 4.12 MNIST: 各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedProx)

PGD eps 聯邦防禦方法	clean	0.05	0.1	0.15	0.2	0.25
FedProx[3]	0.962	0.396	0.334	0.277	0.229	0.183
Fed_AT[49]	0.969	0.529	0.477	0.430	0.384	0.341
Fed_AT_Slack[54]	0.937	0.707	0.648	0.585	0.525	0.455
Fed_BVA[23]	0.980	0.456	0.383	0.325	0.278	0.232
Fed_BVA_Slack	0.954	0.388	0.316	0.261	0.217	0.178
FedBVA_SAT	0.977	0.544	0.459	0.387	0.321	0.257
FedBVA_SAT_Slack (proposed method)	0.981 (+0.001)	0.822 (+0.115)	0.775 (+0.127)	0.724 (+0.139)	0.671 (+0.146)	0.616 (+0.161)

4.6 Fashion-MNIST 資料集的實驗結果與分析

本實驗以 FedAvg 與 FedProx 聚合方法為基礎，進行了六種不同聯邦對抗訓練演算法的對抗強健性比較，比較的演算法包含 Fed_AT、Fed_AT_Slack、Fed_BVA、Fed_BVA_Slack、FedBVA_SAT 以及本研究提出的 FedBVA_SAT_Slack，在章節 4.2 有完整的介紹。

在 Fashion-MNIST 資料集實驗中，共有五個參與客戶端，所有客戶端使用的模型皆為 VGG11 模型，並針對在中心伺服器進行了 20 次聚合後所生成的全域模型之對抗

強健性進行深入分析，此分析涵蓋了全域模型對於乾淨資料的準確度，與對於不同強度 FGSM 與 PGD 攻擊的抵抗能力。

由於 Fashion-MNIST 資料集為分類問題，選擇使用準確度與 F1-Score 作為評估指標。接下來的段落中，會以 FedAvg 與 FedProx 這兩個不同聚合方法進行實驗分析。

4.6.1 FedAvg 聚合方法下的不同聯邦對抗訓練演算法的訓練過程

FedAvg 表示在沒有採取任何防禦策略的情況下，僅使用標準的 FedAvg (Federated Averaging) 聚合方法進行實驗。而 Fed_AT、Fed_AT_Slack、Fed_BVA、Fed_BVA_Slack、FedBVA_SAT 以及 FedBVA_SAT_Slack 則是六種不同的聯邦對抗訓練演算法，共同目標為在聯邦學習系統收斂時，使全域模型的準確度，能夠盡可能接近未施加任何防禦策略下的 FedAvg 準確度。

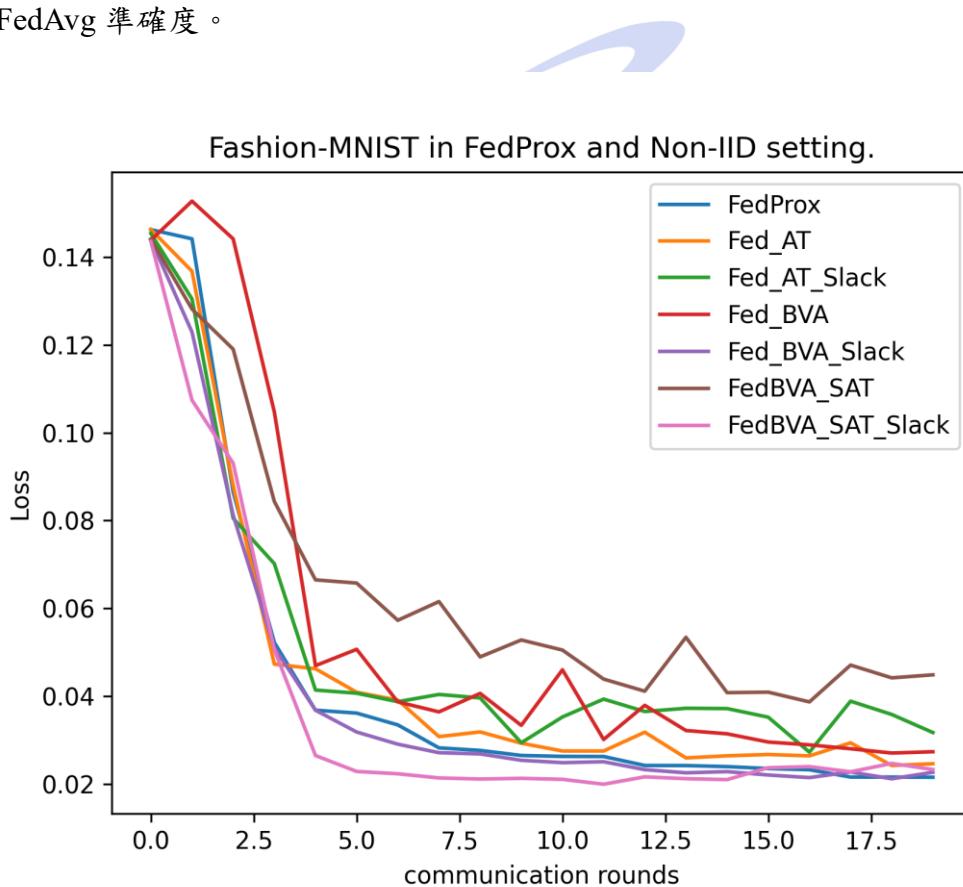


圖 4.12 Fashion-MNIST:各聯邦對抗訓練系統的聚合過程損失折線圖(FedAvg)

圖 4.12 展示了 FedAvg 與其他六種聯邦對抗訓練演算法在中心伺服器進行 20 次聚

合的損失變化。觀察損失值變化後，可以發現直接在客戶端執行對抗訓練的 Fed_AT_Slack(綠色線段)與 FedBVA_SAT(棕色線段)的聚合穩定性較差，可能代表聚合模型無法辨識乾淨資料與受到對抗樣本攻擊影響的特徵所致。

表 4.13 Fashion-MNIST:各聯邦系統最終全域模型之準確度與 F1-Score 比較(FedAvg)

	FedAvg	Fed_AT	Fed_AT_Slack	Fed_BVA	Fed_BVA_Slack	FedBVA_SAT	FedBVA_SAT_Slack
Accuracy	0.918	0.866	0.842	0.917	0.919	0.794	0.910
F1 Score	0.918	0.868	0.837	0.917	0.919	0.798	0.909

表 4.13 中可以清晰地顯示 Fed_BVA、Fed_BVA_Slack 以及 FedBVA_SAT_Slack 在乾淨資料的準確度與 F1-Score 指標上，相當接近 FedAvg 的準確度，代表這兩個聯邦對抗訓練演算法具有良好的泛用性。

4.6.2 FedAvg 聚合方法下的面對 FGSM 攻擊的對抗強健性比較

我們比較了六大聯邦對抗訓練演算法在面對不同強度的 FGSM 攻擊時之對抗強健性。評估指標為準確度(Accuracy)。以未進行任何防禦手段的 FedAvg 作為基準，如果其他的聯邦對抗訓練演算法在面對各種攻擊方法和攻擊強度時，其準確度仍然高於 FedAvg，便可認定該演算法提升了全域模型的對抗強健性。

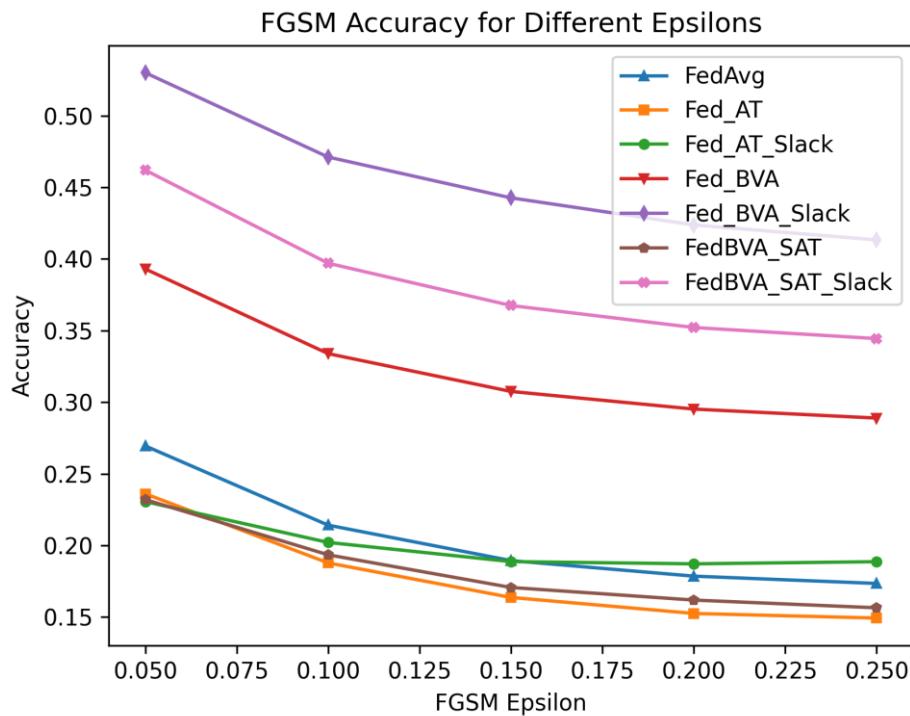


圖 4.13 Fashion-MNIST:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedAvg)

首先，討論 FGSM 攻擊的對抗強健性比較。圖 4.13 與表 4.14 為不同聯邦對抗訓練系統經過 20 次聚合後的最終全域模型，對於 FGSM 攻擊強度在 $\epsilon = 0.05$ 至 0.25 範圍內的測試準確度。圖 4.13 中，以 FedAvg(藍色線段)作為判斷標準，當攻擊強度 ϵ 大大於 0.05 後，除了 FedBVA_SAT(棕色線段)演算法之外，其他聯邦對抗訓練演算法的準確度皆高於藍色線段，以本論文提出的 FedBVA_SAT_Slack 演算法效果最為突出。

表 4.14 Fashion-MNIST:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedAvg)

聯邦防禦方法	FGSM eps	clean	0.05	0.1	0.15	0.2	0.25
FedAvg[3]		0.918	0.269	0.214	0.189	0.179	0.173

Fed_AT[49]	0.866	0.236	0.188	0.164	0.153	0.149
Fed_AT_Slack[54]	0.842	0.231	0.202	0.189	0.187	0.189
Fed_BVA[23]	0.917	0.393	0.334	0.308	0.295	0.289
Fed_BVA_Slack	0.919	0.462	0.397	0.368	0.352	0.344
FedBVA_SAT	0.794	0.232	0.193	0.171	0.162	0.156
FedBVA_SAT_Slack (proposed method)	0.910	0.560	0.502	0.471	0.448	0.438

4.6.3 FedAvg 聚合方法下的面對 PGD 攻擊的對抗強健性比較

接著，來探討 PGD 攻擊的狀況。

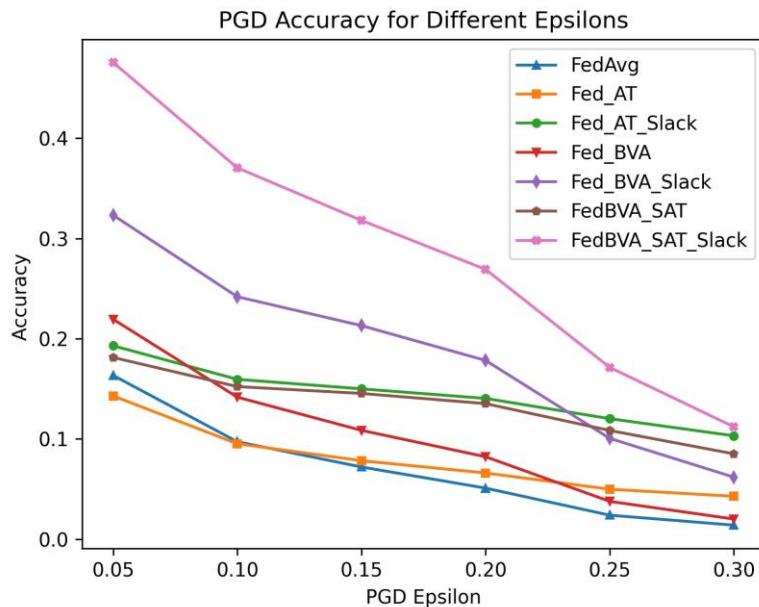


圖 4.14 Fashion-MNIST:各聯邦對抗訓練系統對 PGD 擊的準確度折線圖(FedAvg)

表 4.15 Fashion-MNIST:各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedAvg)

聯邦防禦方法	PGD eps	clean	0.05	0.1	0.15	0.2	0.25

FedAvg[3]	0.918	0.163	0.097	0.072	0.051	0.024
Fed_AT[49]	0.866	0.143	0.095	0.078	0.066	0.050
Fed_AT_Slack[54]	0.842	0.193	0.159	0.150	0.140	0.120
Fed_BVA[23]	0.917	0.219	0.142	0.109	0.082	0.038
Fed_BVA_Slack	0.919	0.323	0.242	0.213	0.178	0.101
FedBVA_SAT	0.794	0.181	0.152	0.145	0.135	0.108
FedBVA_SAT_Slack (proposed method)	0.910 (-0.007)	0.476 (+0.153)	0.371 (+0.129)	0.318 (+0.105)	0.269 (+0.091)	0.142 (+0.022)

圖 4.14 與表 4.15 為不同聯邦對抗訓練系統經過 20 次聚合後的最終全域模型，對於 PGD 攻擊強度在 $\epsilon = 0.05$ 至 0.25 範圍內的測試準確度。在該圖中，以 FedAvg(藍色線段)作為判斷標準，所有演算法在 PGD 攻擊下的準確度皆比藍色線段還高，表示了這些方法能夠提升全域模型的對抗強健性。而本研究所提出的 FedBVA_SAT_Slack 演算法在此攻擊範圍下的準確度皆高於其他算法，代表其具有一定的對抗強健性。

4.6.4 FedProx 聚合方法下的不同聯邦對抗訓練演算法的訓練過程

誠如前述的 FedAvg，FedProx 表示在沒有採取任何防禦策略的情況下，僅使用標準的 FedProx 聚合方法進行實驗。而 Fed_AT、Fed_AT_Slack、Fed_BVA、FedBVA_SAT 以及 FedBVA_SAT_Slack 則是六種不同的聯邦對抗訓練演算法，共同目標為在聯邦學習系統收斂時，使全域模型的準確度盡可能地趨近未施加任何防禦策略下的 FedProx 準確度。

圖 4.15 展示了 FedProx 與其他六種聯邦對抗訓練演算法在中心伺服器進行 20 次聚合後的損失變化。觀察損失值變化後，可以發現直接在客戶端執行對抗訓練的 Fed_AT_Slack(綠色線段)損失值無法進行有效地下降。

因此，如表 4.16 所示，Fed_AT_Slack 的準確度和 F1 分數明顯低於未經任何防禦手段的 FedProx，表示在客戶端進行對抗訓練後，將所有客戶端模型進行鬆弛聚合操作的

手段，可能會遺失對於乾淨資料的特徵，導致準確度不佳。

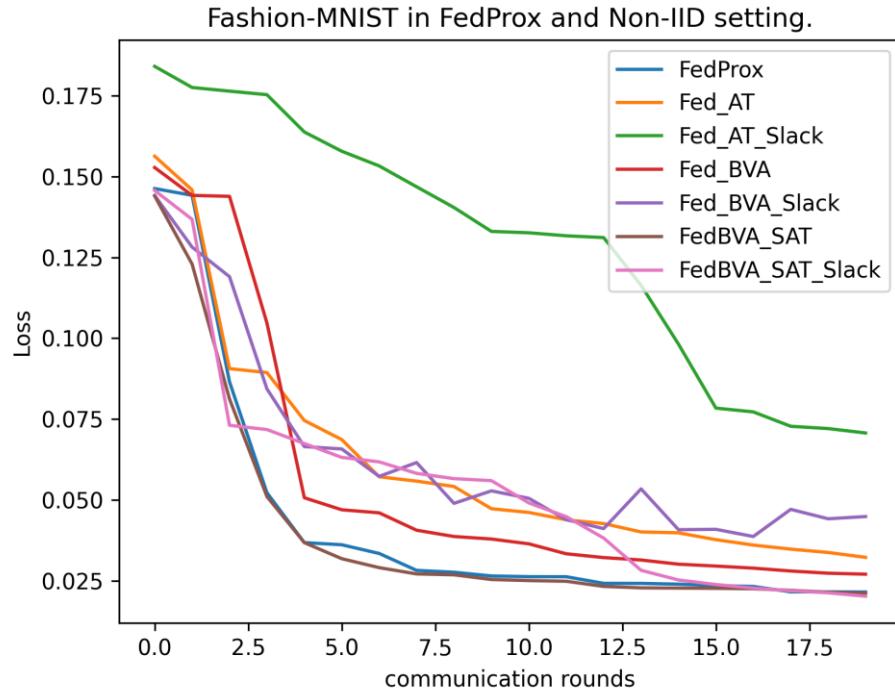


圖 4.15 Fashion-MNIST:各聯邦對抗訓練系統的聚合過程損失折線圖(FedProx)

表 4.16 Fashion-MNIST:各聯邦系統最終全域模型之準確度與 F1-Score 比較(FedProx)

	FedAvg	Fed_AT	Fed_AT_Slack	Fed_BVA	Fed_BVA_Slack	FedBVA_SAT	FedBVA_SAT_Slack
Accuracy	0.874	0.830	0.838	0.841	0.714	0.864	0.857
F1 Score	0.873	0.824	0.838	0.843	0.669	0.863	0.851

從表格中可以發現，Fed_BVA、FedBVA_SAT 以及 FedBVA_SAT_Slack 這三種演算法的乾淨準確度，最接近未經過任何防禦策略的 FedProx 的準確度，卻沒有超越。然而，從對抗樣本攻擊的防禦角度來分析，我們必須要理解，提升乾淨準確度與對抗強健性之間存在著一種固有的權衡關係（trade-off）。換句話說，要提升對抗強健性，往往可能需要付出一定程度的乾淨準確度下降的代價，這是一種不可避免的現象。因此，即使乾淨準確度有微幅的下降，但若能藉此顯著提升對抗強健性，這仍然是一個可以接受的結果。

4.6.5 FedProx 聚合方法下的面對 FGSM 攻擊的對抗強健性比較

誠如先前所述，本實驗數據以未進行任何防禦手段的 FedProx 作為基準。即便在受到不同強度的 FGSM 攻擊下，只要準確度仍然高於 FedProx，就可以證明該演算法成功地提升了經過 20 次聚合後的聯邦對抗訓練全域模型的對抗強健性。

首先，討論 FGSM 攻擊的對抗強健性比較。圖 4.16 與表 4.17 為不同聯邦對抗訓練系統經過 20 次聚合後的最終全域模型，對於 FGSM 攻擊強度在 $\epsilon = 0.05$ 至 0.25 範圍內的測試準確度。

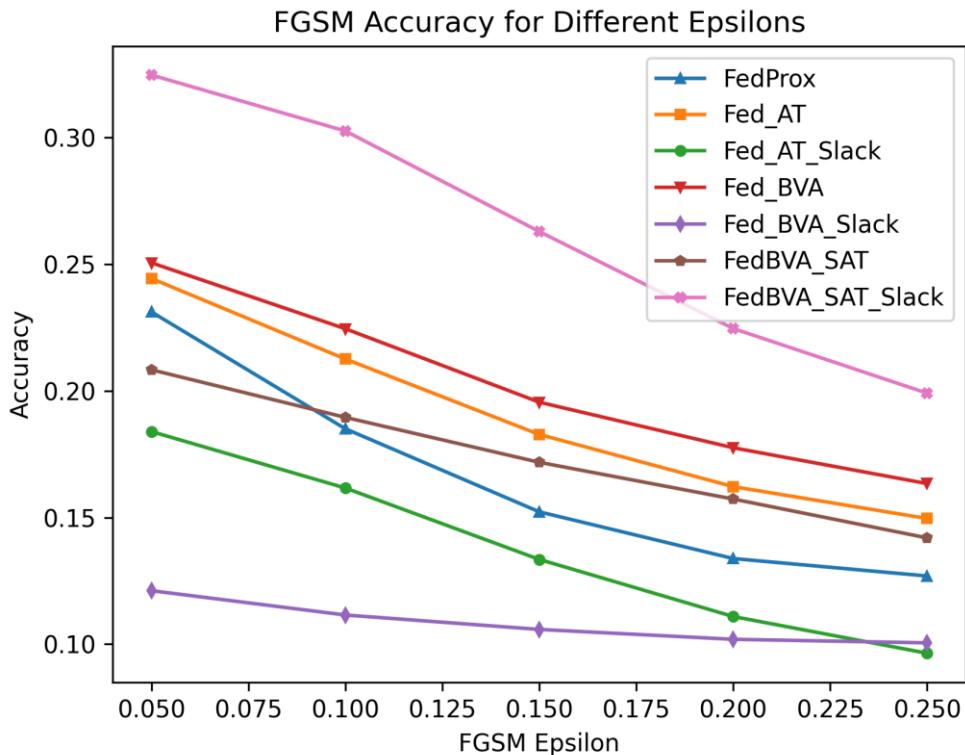


圖 4.16 Fashion-MNIST:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedProx)

表 4.17 Fashion-MNIST:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedProx)

聯邦防禦方法	clean	0.05	0.1	0.15	0.2	0.25
FGSM eps						

FedAvg[3]	0.874	0.231	0.185	0.152	0.134	0.127
Fed_AT[49]	0.830	0.244	0.213	0.183	0.162	0.150
Fed_AT_Slack[54]	0.838	0.184	0.162	0.133	0.111	0.096
Fed_BVA[23]	0.841	0.251	0.225	0.196	0.178	0.163
Fed_BVA_Slack	0.714	0.121	0.112	0.106	0.102	0.101
FedBVA_SAT	0.864	0.208	0.190	0.172	0.157	0.142
FedBVA_SAT_Slack (proposed method)	0.857	0.325	0.303	0.263	0.225	0.199

從上述的圖與表中，以 FedProx (藍色線段)作為判斷標準，可以觀察到當攻擊強度大於 0.05 後，所有聯邦對抗訓練演算法的準確度都比標準值高，代表這些方法都能對在 FedProx 聚合方法下有效提升全域模型的對抗強健性。其中，本研究提出的 FedBVA_SAT_Slack(粉色線段)在所有攻擊情境下的準確度表現均最為優秀，這明確地證明了該演算法在提升對抗強健性方面具有顯著的效能。

4.6.6 FedProx 聚合方法下的面對 PGD 攻擊的對抗強健性比較

接著，來探討 PGD 攻擊的狀況。圖 4.17 與表 4.18 為不同聯邦對抗訓練系統經過 20 次聚合後的最終全域模型，對於 PGD 攻擊強度在 $\epsilon = 0.05$ 至 0.25 範圍內的測試準確度。在該圖中，以 FedAvg(藍色線段)作為判斷標準，除了 Fed_AT 與 Fed_BVA_Slack 方法之外，其他演算法在不同強度的 PGD 攻擊下，準確度都比基準值 FedProx(藍色線段)還高，表示了這些演算法能夠提升全域模型的對抗強健性。

當攻擊強度小於 0.15($\epsilon < 0.15$)時，本研究提出的 FedBVA_SAT_Slack(粉色線段)準確度都比其他方法略高；而當攻擊強度等於 0.2 之後，Fed_AT_Slack(綠色線段)的準確度較高。儘管如此，由於 FedBVA_SAT_Slack 在乾淨準確度方面優於 Fed_AT_Slack，且除了 PGD $\epsilon = 0.2$ 時的攻擊強度稍遜於 Fed_AT_Slack 外，其他情況下的準確度保持較高水平。因此，我們可以認定本研究提出的方法仍然具有一定的效果。

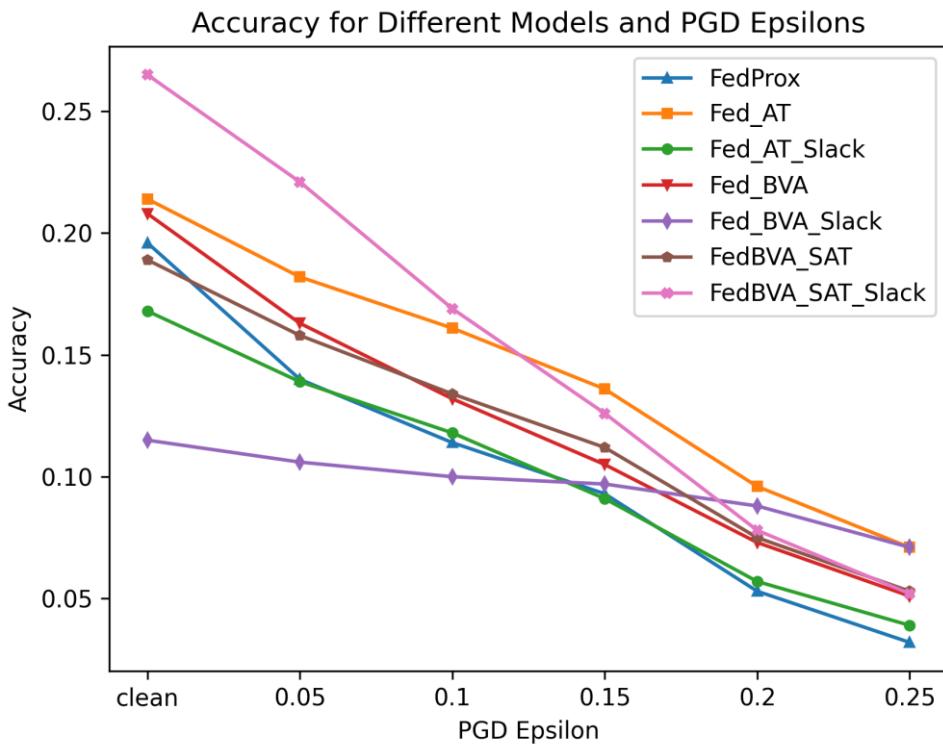


圖 4.17 Fashion-MNIST: 各聯邦對抗訓練系統對 PGD 擊的準確度折線圖(FedProx)

表 4.18 Fashion-MNIST: 各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedProx)

	PGD eps	clean	0.05	0.1	0.15	0.2	0.25
聯邦防禦方法							
FedAvg[3]	0.874	0.196	0.140	0.114	0.093	0.053	
Fed_AT[49]	0.830	0.214	0.182	0.161	0.136	0.096	
Fed_AT_Slack[54]	0.838	0.168	0.139	0.118	0.091	0.057	
Fed_BVA[23]	0.841	0.208	0.163	0.132	0.105	0.073	
Fed_BVA_Slack	0.714	0.115	0.106	0.100	0.097	0.088	
FedBVA_SAT	0.864	0.189	0.158	0.134	0.112	0.075	
FedBVA_SAT_Slack	0.857	0.265	0.221	0.169	0.126	0.078	
(proposed method)	(-0.017)	(+0.057)	(+0.028)	(+0.035)	(-0.015)	(-0.018)	

4.7 線寬資料集的實驗結果與分析

本實驗分別以 FedAvg 與 FedProx 聚合方法為基礎，並進行六個不同的聯邦對抗訓練演算法：Fed_AT、Fed_AT_Slack、Fed_BVA、Fed_BVA_Slack、FedBVA_SAT 與 FedBVA_SAT_Slack，在章節 4.2 有完整的介紹。

在線寬資料集實驗中，共有十個參與客戶端，所有客戶端使用的模型皆為 Auto-Encoder，並針對在中心伺服器進行了 20 次聚合後所生成的全域模型之對抗強健性進行深入分析，此分析涵蓋了全域模型對於乾淨資料的準確度，與對於不同強度 FGSM 與 PGD 攻擊的抵抗能力。

由於線寬預測資料集為迴歸問題，因此使用的評估指標為 R-平方(R-squared)。接下來的段落中，會以 FedAvg 與 FedProx 這兩個不同聚合方法進行實驗分析。

4.7.1 FedAvg 聚合方法下的不同聯邦對抗訓練演算法的訓練過程

FedAvg 表示在沒有採取任何防禦策略的情況下，僅使用標準的 FedAvg (Federated Averaging) 聚合方法進行實驗。而 Fed_AT、Fed_AT_Slack、Fed_BVA、FedBVA_SAT 以及 FedBVA_SAT_Slack 則是六種不同的聯邦對抗訓練演算法，共同目標為在聯邦學習系統收斂時，使全域模型的 R-平方，能夠盡可能接近未施加任何防禦策略下的 FedAvg R-平方。

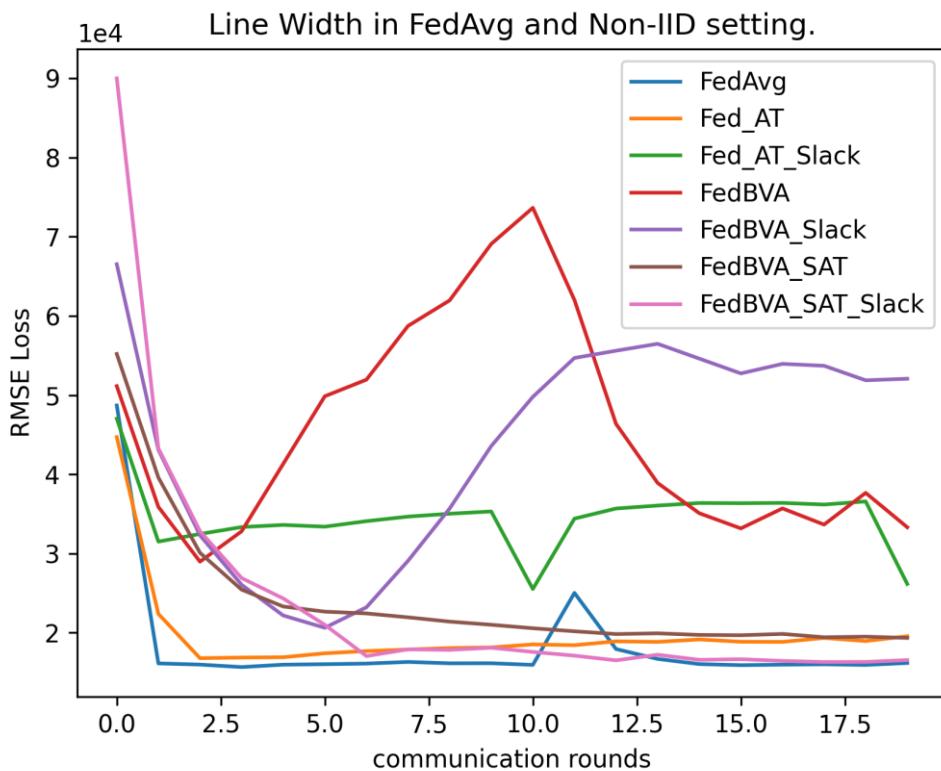


圖 4.18 線寬資料集:各聯邦對抗訓練系統的聚合過程損失折線圖(FedAvg)

表 4.19 線寬資料集:各聯邦系統最終全域模型之 R-squared 比較(FedAvg)

	FedAvg	Fed_AT	Fed_AT_Slack	Fed_BVA	Fed_BVA_Slack	FedBVA_SAT	FedBVA_SAT_Slack
R ²	0.928	0.905	0.439	0.651	0.770	0.868	0.931

圖 4.18 展示了 FedAvg 與其他六種聯邦對抗訓練演算法在中心伺服器進行 20 次聚合後的 RMSE 損失變化。觀察損失值變化後，可以發現直接在客戶端執行對抗訓練的 Fed_BVA 方法的損失值變化呈現倒 V 字型(紅色線段)，表示模型聚合的狀態極度不穩定，並不適用於線寬資料集；而使用鬆弛聚合機制方法的 Fed_BVA_Slack(紫色線段)，也出現損失值變化極不穩定的狀況。因此，如表 4.19 所示，這兩種演算法的 R-平方明顯低於未經任何防禦手段的 FedAvg，這是不樂見的情況；然而，也可以觀察到 Fed_AT 與 FedBVA_SAT_Slack 在乾淨資料的 R-平方上，接近或略勝於未經對抗訓練防禦機制的 FedAvg，代表實驗具有一定的穩定性。

4.7.2 FedAvg 聚合方法下的面對 FGSM 攻擊的對抗強健性比較

本章節中，比較了上述提到的六種主要聯邦對抗訓練演算法在面對不同強度的 FGSM 攻擊時之對抗強健性。評估指標為 R-平方(R-squared, R^2)。以未進行任何防禦手段的 FedAvg 作為基準，如果其他的聯邦對抗訓練演算法在面對各種攻擊方法和攻擊強度時，其 R-平方仍然高於 FedAvg，便可認定該方法提升了全域模型的對抗強健性。

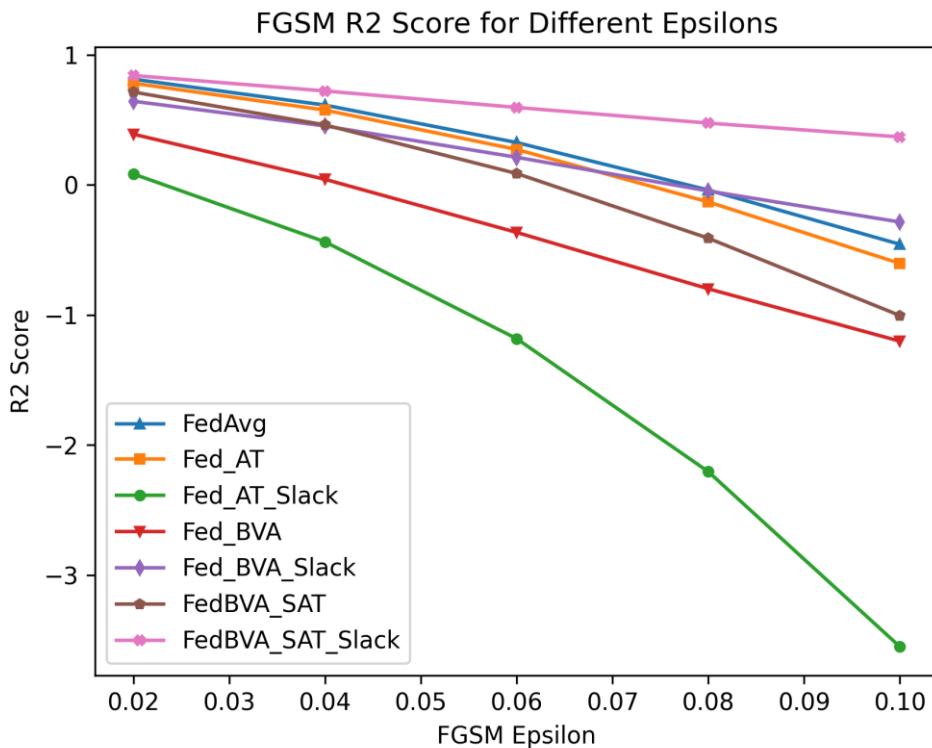


圖 4.19 線寬資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedAvg)

首先，討論 FGSM 攻擊的對抗強健性比較。圖 4.19 為不同聯邦對抗訓練系統經過 20 次聚合後的最終全域模型，對於 FGSM 攻擊強度在 $\epsilon=0.02$ 至 0.10 範圍內的測試 R- 平方。以 FedAvg(藍色線段)作為判斷標準，可以觀察到只有 FedBVA_SAT_Slack (粉色線段)在所有的攻擊強度下，其 R- 平方數值仍比藍色線段還高，表示了本研究提出的 FedBVA_SAT_Slack 演算法，不僅能提升分類問題，也能提高迴歸問題全域模型的對抗強健性。

在表 4.20 中，完整條列這六種聯邦對抗訓練演算法(聯邦防禦方法)在不同 FGSM 攻擊強度下的準確度。

表 4.20 線寬資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedAvg)

FGSM eps 聯邦防禦方法	clean	0.02	0.04	0.06	0.08	0.1
FedAvg[3]	0.928	0.813	0.614	0.326	-0.04	-0.45
Fed_AT[49]	0.905	0.782	0.576	0.273	-0.13	-0.60
Fed_AT_Slack[54]	0.439	0.084	-0.438	-1.18	-2.20	-3.55
Fed_BVA[23]	0.651	0.358	-0.067	-0.605	-1.2	-1.77
Fed_BVA_Slack	0.77	0.644	0.453	0.213	-0.047	-0.28
FedBVA_SAT	0.868	0.714	0.461	0.088	-0.41	-1.00
FedBVA_SAT_Slack (proposed method)	0.931 (+0.03)	0.842 (+0.03)	0.723 (+0.109)	0.596 (+0.27)	0.476 (+0.515)	0.369 (+0.7)

4.7.3 FedAvg 聚合方法下的面對 PGD 攻擊的對抗強健性比較

接著，來探討 PGD 攻擊的狀況。圖 4.20 與表 4.21 為不同聯邦對抗訓練系統經過 20 次聚合後的最終全域模型，對於 PGD 攻擊強度在 $\epsilon = 0.05$ 至 0.25 範圍內的測試 R- 平方。在該圖中，以 FedAvg(藍色線段)作為判斷標準，亦只有 FedBVA_SAT_Slack(粉色線段)在藍色線段之上，顯示了該方法無論在 FGSM 攻擊或 PGD 攻擊都能提升聯邦學習系統的全域模型對抗強健性。

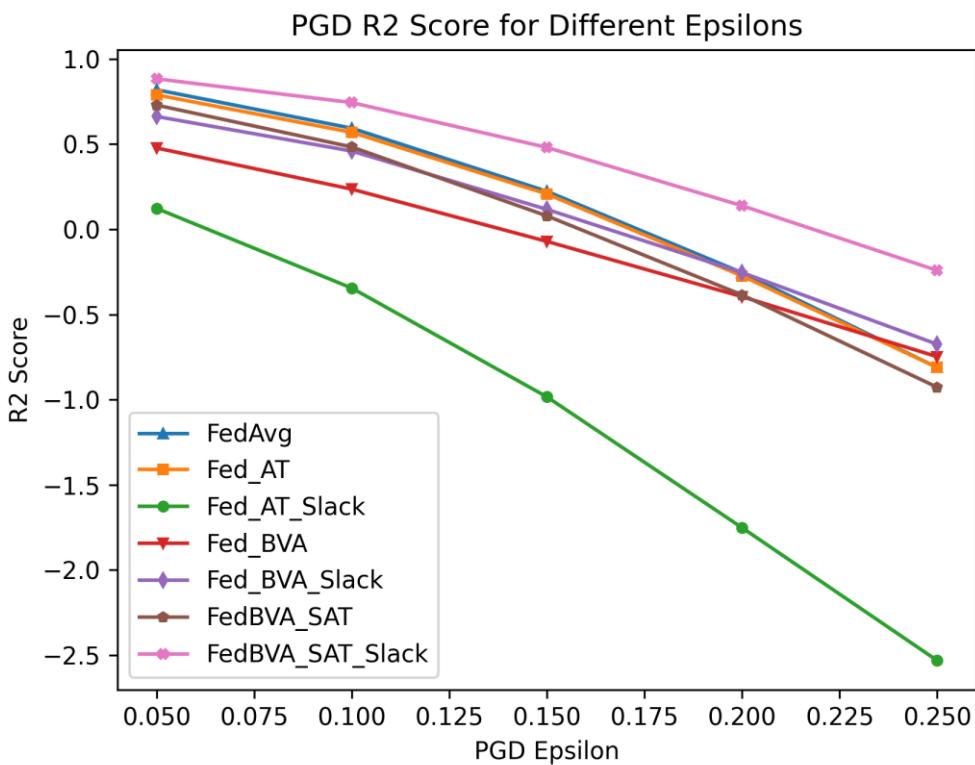


圖 4.20 線寬資料集:各聯邦對抗訓練系統對 PGD 攻擊強度的準確度折線圖(FedAvg)

表 4.21 線寬資料集:各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedAvg)

聯邦防禦方法 \ PGD eps	clean	0.05	0.10	0.15	0.20	0.25
PGD eps						
FedAvg[3]	0.928	0.819	0.593	0.223	-0.255	-0.811
Fed_AT[49]	0.905	0.789	0.569	0.207	-0.273	-0.811
Fed_AT_Slack[54]	0.439	0.123	-0.345	-0.983	-1.751	-2.531
Fed_BVA[23]	0.654	0.477	0.236	-0.071	-0.395	-0.748
Fed_BVA_Slack	0.769	0.662	0.459	0.117	-0.254	-0.675
FedBVA_SAT	0.868	0.729	0.483	0.079	-0.385	-0.927
FedBVA_SAT_Slack (proposed method)	0.931 (+0.03)	0.883 (+0.063)	0.739 (+0.148)	0.487 (+0.270)	0.142 (+0.406)	-0.214 (+0.436)

由表 4.21 可知，當 PGD 攻擊強度超過 0.25 後，本研究提出的 FedBVA_SAT

_Slack 演算法，R-平方數值已經小於 0。這表示，儘管 FedBVA_SAT_Slack 具有一定的抵禦能力，但仍有其一定的極限存在。

4.7.4 FedProx 聚合方法下的不同聯邦對抗訓練演算法的訓練過程

誠如前述的 FedAvg，FedProx 表示在沒有採取任何防禦策略的情況下，僅使用標準的 FedProx 聚合方法進行實驗。而 Fed_AT、Fed_AT_Slack、Fed_BVA、FedBVA_SAT 以及 FedBVA_SAT_Slack 則是六種不同的聯邦對抗訓練演算法，共同目標為在聯邦學習系統收斂時，使全域模型的 R-平方，能夠盡可能接近未施加任何防禦策略下的 FedProx R-平方。

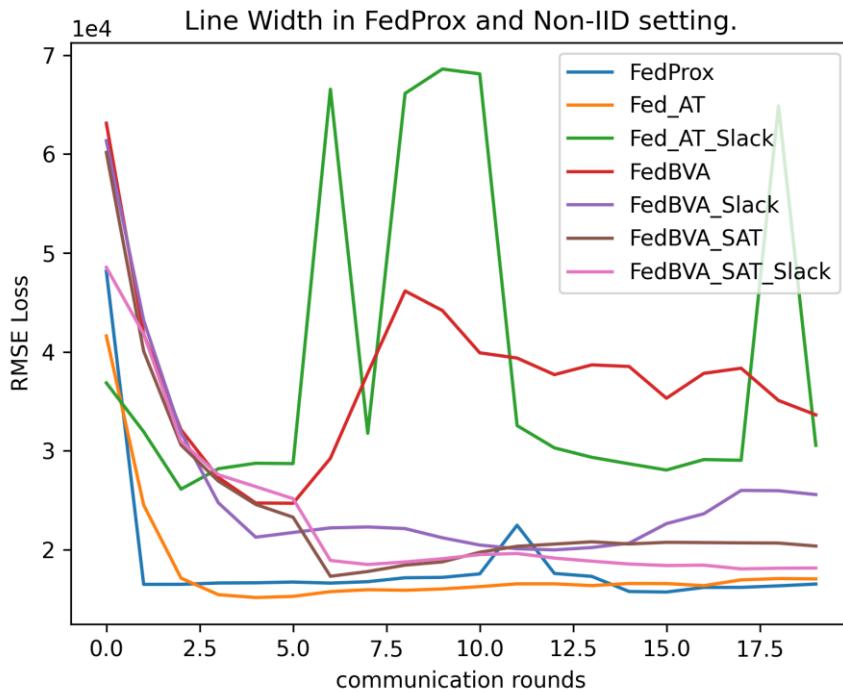


圖 4.21 展示了 FedProx 與其他六種聯邦對抗訓練演算法在中心伺服器進行 20 次聚合後的損失變化。觀察損失值變化後，可以發現直接在客戶端執行對抗訓練的 Fed_BVA 與 Fed_BVA_Slack 演算法的損失值曲線極度不穩定，暗示著這兩個演算法並不適合線寬資料集。因此，如表 4.22 所示，這兩種演算法的 R-平方數值偏低。

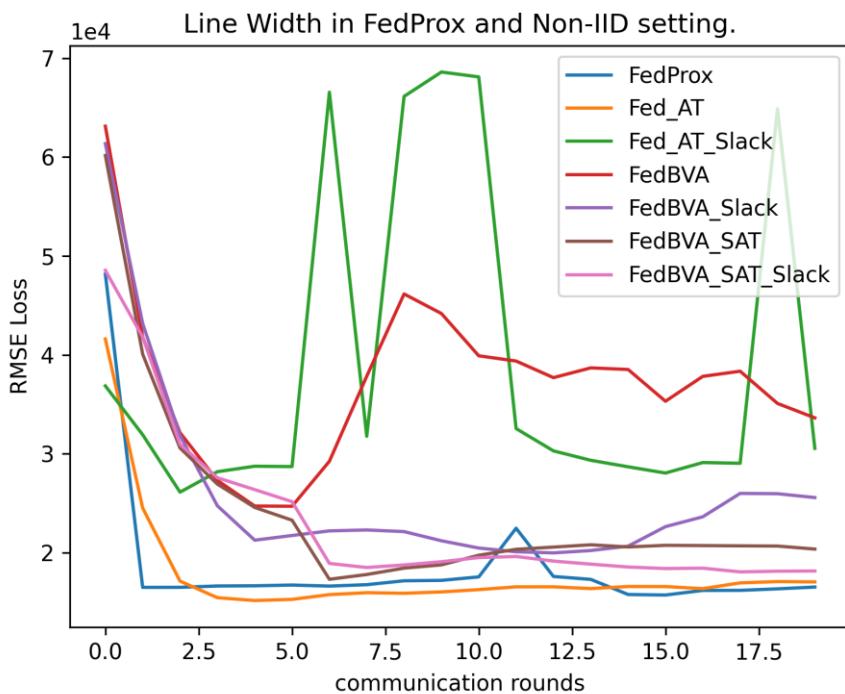


圖 4.21 線寬資料集:各聯邦對抗訓練系統的聚合過程損失折線圖(FedProx)

表 4.22 線寬資料集:各聯邦系統最終全域模型之準確度與 F1-Score 比較(FedProx)

	FedAvg	Fed_AT	Fed_AT_Slack	Fed_BVA	Fed_BVA_Slack	FedBVA_SAT	FedBVA_SAT_Slack
R-平方	0.916	0.923	0.678	0.567	0.76	0.871	0.929

在表格中可以清晰地顯示 Fed_AT 與 FedBVA_SAT_Slack 在乾淨資料的 R-平方指標上，略優於未經對抗訓練防禦機制的 FedAvg，這正是希望看到的實驗結果。

4.7.5 FedProx 聚合方法下的面對 FGSM 攻擊的對抗強健性比較

誠如先前所述，本實驗數據以未進行任何防禦手段的 FedProx 作為基準。即便在受到不同強度的 FGSM 攻擊下，只要 R-平方仍然高於 FedProx，就可以證明該演算法成功地提升了經過 20 次聚合後的聯邦對抗訓練全域模型的對抗強健性。

首先，討論 FGSM 攻擊的對抗強健性比較。圖 4.22 與表 4.23 為不同聯邦對抗訓練系統經過 20 次聚合後的最終全域模型，對於 FGSM 攻擊強度在 $\epsilon = 0.02$ 至 0.10 範圍

內的測試 R-平方。

從上述的圖與表中，以 FedProx (藍色線段)作為判斷標準，可以觀察到本研究提出的 FedBVA_SAT_Slack 在所有攻擊情境下的準確度表現均最為優秀，這明確地證明了該演算法在提升對抗強健性方面具有顯著的效能。

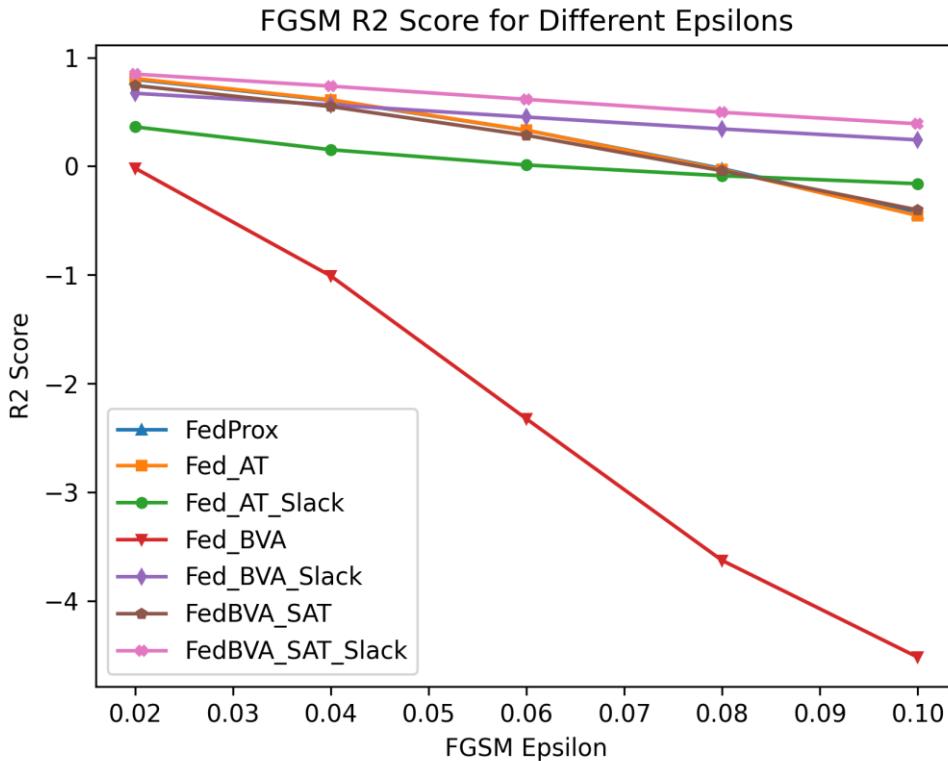


圖 4.22 線寬資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedProx)

表 4.23 線寬資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedProx)

聯邦防禦方法 \ FGSM eps	clean	0.02	0.04	0.06	0.08	0.1
FedProx[34]	0.916	0.800	0.607	0.331	-0.022	-0.433
Fed_AT[49]	0.923	0.808	0.612	0.332	-0.031	-0.454
Fed_AT_Slack[54]	0.678	0.362	0.152	0.012	-0.087	-0.161
Fed_BVA[23]	0.567	-0.017	-1.009	-2.325	-3.628	-4.519
Fed_BVA_Slack	0.76	0.672	0.566	0.453	0.343	0.243

FedBVA_SAT	0.871	0.743	0.550	0.285	-0.042	-0.402
FedBVA_SAT_Slack (proposed method)	0.929 (+0.13)	0.848 (+0.05)	0.738 (+0.128)	0.616 (+0.278)	0.497 (+0.508)	0.391 (+0.135)

4.7.6 FedProx 聚合方法下的面對 PGD 攻擊的對抗強健性比較

接著，來探討 PGD 攻擊的狀況。圖 4.23 與表 4.24 為不同聯邦對抗訓練系統經過 20 次聚合後的最終全域模型，對於 PGD 攻擊強度在 $\epsilon = 0.02$ 至 0.10 範圍內的測試 R- 平方。在該圖中，以 FedProx(藍色線段)作為判斷標準，亦只有 FedBVA_SAT_Slack (粉色線段)在藍色線段之上，顯示了該方法無論在 FGSM 攻擊或 PGD 攻擊都能提升聯邦學習系統的全域模型對抗強健性。

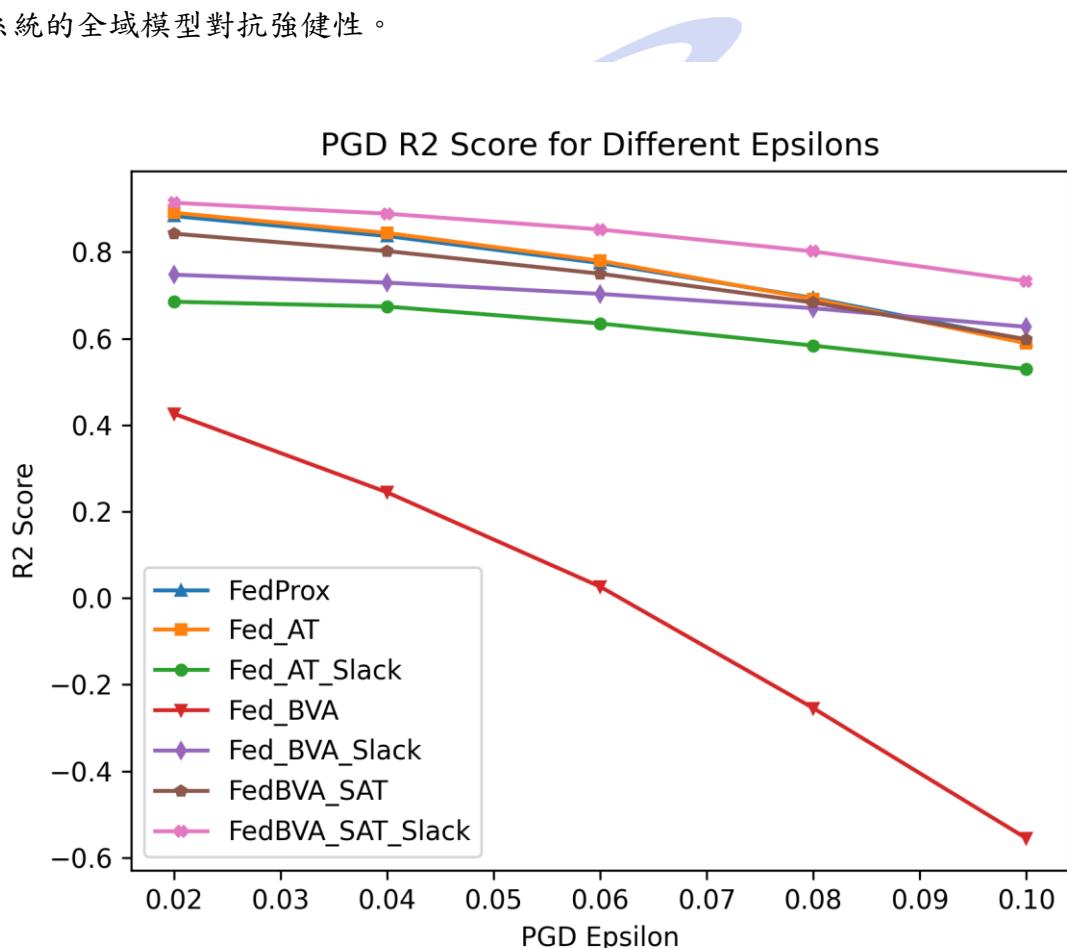


圖 4.23 線寬資料集:各聯邦對抗訓練系統對 PGD 攻擊的準確度折線圖(FedProx)

表 4.24 線寬資料集:各聯邦對抗訓練系統對 PGD 攻擊的準確度比較表(FedProx)

聯邦防禦方法 \ PGD eps	0	0.02	0.04	0.06	0.08	0.1
FedProx[34]	0.916	0.883	0.837	0.774	0.694	0.596
Fed_AT[49]	0.923	0.891	0.844	0.780	0.691	0.589
Fed_AT_Slack[54]	0.678	0.685	0.674	0.635	0.584	0.529
Fed_BVA[23]	0.567	0.426	0.245	0.027	-0.254	-0.555
Fed_BVA_Slack	0.76	0.748	0.729	0.703	0.670	0.627
FedBVA_SAT	0.871	0.842	0.802	0.750	0.683	0.598
FedBVA_SAT_Slack	0.929	0.914	0.889	0.852	0.801	0.732
(proposed method)	(+0.13)	(+0.024)	(+0.045)	(+0.073)	(+0.107)	(+0.106)

4.8 電晶體資料集的實驗與分析

本實驗分別以 FedAvg 與 FedProx 聚合方法為基礎，並進行六個不同的聯邦對抗訓練演算法：Fed_AT、Fed_AT_Slack、Fed_BVA、Fed_BVA_Slack、FedBVA_SAT 與 FedBVA_SAT_Slack，在章節 4.2 有完整的介紹。

在電晶體資料集實驗中，共有三個參與客戶端，並針對在中心伺服器進行了 50 次聚合後所生成的全域模型之對抗強健性進行深入分析，此分析涵蓋了全域模型對於乾淨資料的準確度，與對於不同強度 FGSM 攻擊的抵抗能力。

由於電晶體預測資料集為迴歸問題，因此使用的評估指標為 R-平方(R-squared)。接下來的段落中，會以 FedAvg 與 FedProx 這兩個不同聚合方法進行實驗分析。

4.8.1 FedAvg 聚合方法下的不同聯邦對抗訓練演算法的訓練過程

FedAvg 表示在沒有採取任何防禦策略的情況下，僅使用標準的 FedAvg (Federated Averaging) 聚合方法進行實驗。而 Fed_AT、Fed_AT_Slack、Fed_BVA、Fed_BVA_Slack、FedBVA_SAT 以及 FedBVA_SAT_Slack 則是六種不同的聯邦對抗訓練演算法，共同目標為在聯邦學習系統收斂時，使全域模型的 R-平方，能夠盡可能接近未施加任何防禦策略下的 FedAvg R-平方。而在電晶體資料集與 VAE 模型下，只有 FedAvg、Fed_BVA、FedBVA_SAT 以及 FedBVA_SAT_Slack 可以完整收斂，六種演算法的準確度如下表。

表 4.25 電晶體資料集:各聯邦系統最終全域模型之 R-squared 比較(FedAvg)

	FedAvg	Fed_AT	Fed_AT_Slack	Fed_BVA	Fed_BVA_Slack	FedBVA_SAT	FedBVA_SAT_Slack
R^2	0.716	-0.314	-0.259	0.797	-32	0.489	0.981

因為有三個系統無法收斂，因此只討論其他有收斂的訓練過程損失函數，如下圖：

圖 4.24 電晶體資料集:各聯邦對抗訓練系統的聚合過程損失折線圖(FedAvg)

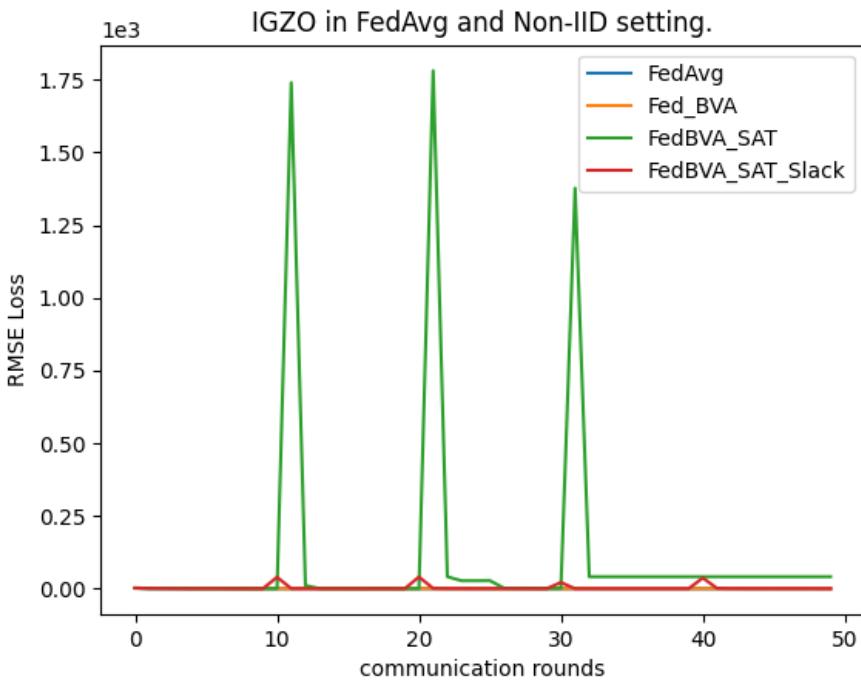


圖 4.24 展示了 FedAvg 與 Fed_BVA、FedBVA_SAT、FedBVA_SAT_Slack 三種可收斂的聯邦對抗訓練演算法在中心伺服器進行 50 次聚合後的 RMSE 損失變化，表 4.25 則為各聯邦系統最終全域模型之 R-squared 比較。觀察損失值變化後，會發現 FedAvg 與 Fed_BVA (橘色線段)損失值保持平穩。

由於 PGD 攻擊對於本資料集與所使用的 VAE 模型的攻擊強度太高，讓客戶端訓練與中心伺服器進行模型聚合時，無法有效提取出乾淨特徵。另外，因應本資料集的特殊性，進一步解釋損失值的狀況。

接著，仔細看圖 4.24 會發現 FedBVA_SAT(綠色線段)與 FedBVA_SAT_Slack (紅色線段)在本實驗的損失曲線圖中呈現一種鋸齒狀的趨勢。這是為了因應電晶體資料集及其使用的 VAE 模型對對抗樣本攻擊的過度敏感性。如果每次聚合都引入專門訓練的全局對抗範例的對抗訓練模型，可能導致模型無法有效學習來自乾淨資料的資訊，進而產生模型失真的問題。為了解決這種問題，我們對 FedBVA_SAT 與 FedBVA_SAT_Slack 演算法做出了調整，改為每十次聚合時才加入對抗訓練模型，使其在其他回合可以收斂至更為穩定的狀態。FedBVA_SAT 演算法在每十次加入對抗訓練模型一起聚合時，損失函數飆高的程度遠高於 FedBVA_SAT_Slack，這與 P 倍加權的比例有關係，下一段將仔細

說明，而當溝通輪次高於 30 後，會發現 FedBVA_SAT 演算法的損失函數無法下降，系統進入無法完整收斂的狀況，直接導致全域模型失去預測的能力，R-平方值直接變成 0.0。而經過特殊加權的 FedBVA_SAT_Slack 仍有損失值下降的趨勢，且準確度較為合理。

結果顯示，這種策略使 FedBVA_SAT_Slack 演算法能夠在全域模型對乾淨資料的 R-平方值達到了 0.905 的理想狀態。這種機制也提升了中心伺服器在經過 50 次聚合後，生成的全域模型對於未知攻擊的防禦能力，如表 4.26 所示，該表格的欄位 P/P' 中，P 代表對於損失函數最大的模型加權值，而 P' 代表了其他模型的加權值，但由於 FedBVA_SAT 並未實施鬆弛加權機制，P 與 P' 皆為 1。

表 4.26 電晶體資料集: Fed_BVA 與 FedBVA_SAT_Slack 對抗強健性比較(FedAvg)

電晶體資料集	加入 AT Model 進行聚合的次數	P/P'	Clean (R-Squared)	FGSM (R-Squared)
FedAvg	-	-	0.962	0.59
FedBVA_SAT	每 10 次加入聚合	1/1	0.489	0.0
FedBVA_SAT_Slack	每 10 次加入聚合	0.05/1	0.929	0.713

接下來，我從另一個角度來分析電晶體資料集以及其使用的 VAE 模型對對抗樣本攻擊的高敏感性。從上表我們可以看出，若不利用鬆弛聚合機制來減緩對抗訓練模型(AT Model)對於全局系統的收斂影響力，即便將 FedBVA_SAT 的策略從每一次聚合都加入對抗訓練模型，改為每五個溝通輪次只進行一次，對乾淨資料準確度也會有一定程度的降低。

相反地，使用了鬆弛聚合機制的 FedBVA_SAT_Slack，透過調整加權比例 P 至 0.05，能有效降低對抗訓練模型的對於聯邦系統全局收斂的衝擊力，如此一來，才能讓聯邦學習系統的最終模型於乾淨資料的 R-平方值保持在 0.972 的水準，也凸顯了鬆弛聚合機制的重要性。

4.8.2 FedAvg 聚合方法下的面對 FGSM 攻擊的對抗強健性比較

這裡比較了上述提到的六種主要聯邦對抗訓練演算法在面對不同強度的 FGSM 的對抗強健性。評估指標為 R-平方 (R-squared, R^2)。以未進行任何防禦手段的 FedAvg 作為基準，如果其他的聯邦對抗訓練演算法在面對各種攻擊方法和攻擊強度時，其 R-平方仍然高於 FedAvg，便可認定該演算法提升了全域模型的對抗強健性。

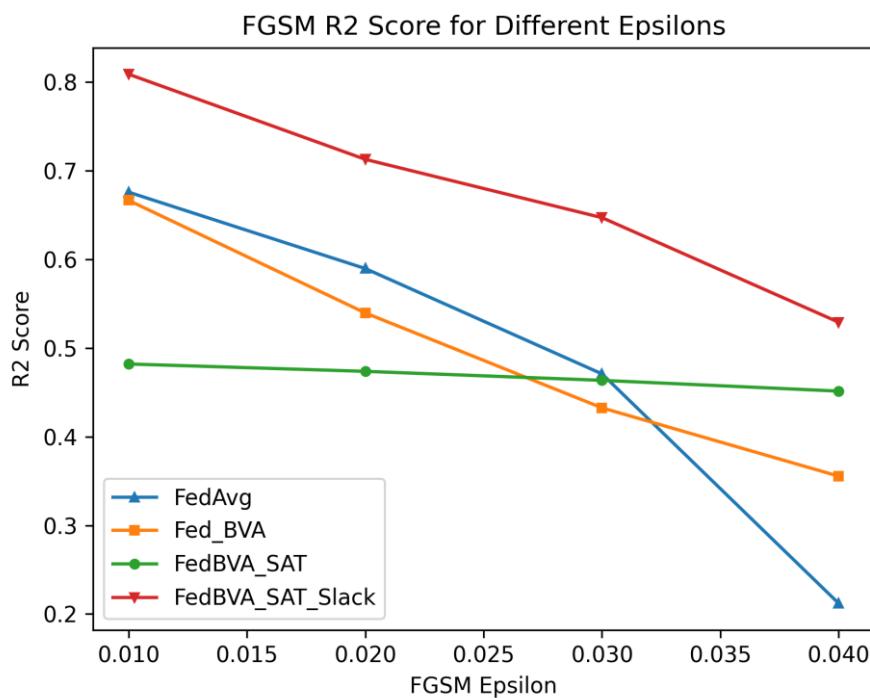


圖 4.25 電晶體資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedAvg)

表 4.27 電晶體資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度比較表(FedAvg)

聯邦防禦方法 \ FGSM eps	clean	0.01	0.02	0.03	0.04
聯邦防禦方法					
FedAvg[3]	0.962	0.676	0.59	0.471	0.211
Fed_BVA[23]	0.798	0.667	0.539	0.432	0.356
FedBVA_SAT	0.482	0.474	0.463	0.451	0.482

FedBVA_SAT_Slack (proposed method)	0.972 (+0.026)	0.809 (+0.133)	0.713 (+0.123)	0.646 (+0.175)	0.53 (+0.079)
---------------------------------------	--------------------------	-------------------	-------------------	-------------------	------------------

圖 4.25 與表 4.27 為不同聯邦對抗訓練系統經過 50 次聚合後的最終全域模型，對於 FGSM 攻擊強度在 $\epsilon = 0.01$ 至 0.04 範圍內的測試 R-平方。在該圖中，以 FedAvg(藍色線段)作為判斷標準，可以觀察到 Fed_BVA_Slack(紅色線段)與 FedBVA_SAT_Slack(粉色線段)趨近於基準值 FedAvg(藍色線段)。所有攻擊強度下，本研究所提出的 FedBVA_SAT_Slack 演算法效果較佳，證明該方法仍具有提升聯邦學習系統對抗強健性的效果。

4.8.3 FedProx 聚合方法下的不同聯邦對抗訓練演算法的訓練過程

FedProx 表示在沒有採取任何防禦策略的情況下，僅使用標準的 FedProx 聚合方法進行實驗。而 Fed_AT、Fed_AT_Slack、Fed_BVA、FedBVA_SAT 以及 FedBVA_SAT_Slack 則是六種不同的聯邦對抗訓練演算法，共同目標為系統收斂時，使全域模型的 R-平方，能夠盡可能接近未進行任何防禦策略下的 FedProx R-平方。而在電晶體資料集與 VAE 模型下，只有 FedAvg、Fed_BVA、FedBVA_SAT 以及 FedBVA_SAT_Slack 可以完整收斂，以下為這四個演算法的訓練過程的收斂損失圖，請見圖 4.26。

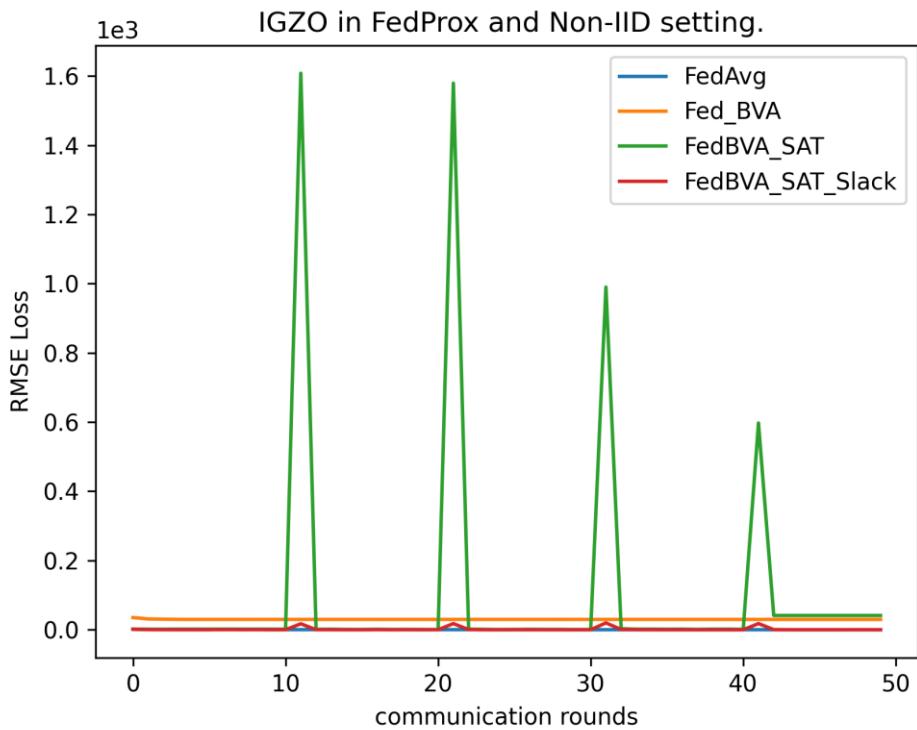


圖 4.26 電晶體資料集:各聯邦對抗訓練系統的聚合過程損失折線圖(FedProx)

圖 4.26 展示了 FedAvg 與 Fed_BVA、FedBVA_SAT、FedBVA_SAT_Slack 三種可收斂的聯邦對抗訓練演算法在中心伺服器進行 50 次聚合後的 RMSE 損失變化，表 4.25 則為各聯邦系統最終全域模型之 R-squared 比較。觀察損失值變化後，會發現 FedAvg 與 Fed_BVA (橘色線段)損失值保持平穩。

由於 PGD 攻擊對於該資料集與所使用的 VAE 模型的攻擊強度太高，讓客戶端訓練與中心伺服器進行模型聚合時，無法有效提取出乾淨特徵。另外，因應本資料集的特殊性，進一步解釋損失值的狀況。

接著，仔細看圖 4.26 會發現 FedBVA_SAT(綠色線段)與 FedBVA_SAT_Slack (紅色線段)在本實驗的損失曲線圖中呈現一種鋸齒狀的趨勢。這是為了因應電晶體資料集及其使用的 VAE 模型對對抗樣本攻擊的過度敏感性。如果每次聚合都引入專門訓練的全局對抗範例的對抗訓練模型，可能導致模型無法有效學習來自乾淨資料的資訊，進而產生模型失真的問題。為了解決這種問題，我們對 FedBVA_SAT 與 FedBVA_SAT_Slack 演算法做出了調整，改為每十次聚合時才加入對抗訓練模型，使其在其他回合可以收斂至

更為穩定的狀態。FedBVA_SAT 演算法在每十次加入對抗訓練模型一起聚合時，損失函數飆高的程度遠高於 FedBVA_SAT_Slack，這與 P 倍加權的比例有關係，下一段將仔細說明，而當溝通輪次高於 30 後，會發現 FedBVA_SAT 演算法的損失值高於其他演算法，經過特殊加權的 FedBVA_SAT_Slack 仍有損失值下降的趨勢。

表 4.28 電晶體資料集:各聯邦系統最終全域模型之準確度與 F1-Score 比較(FedProx)

	FedAvg	Fed_AT	Fed_AT_Slack	Fed_BVA	Fed_BVA_Slack	FedBVA_SAT	FedBVA_SAT_Slack
R^2	0.817	-1.149	-0.335	0.669	0.841	0.814	0.985

從圖 4.26 展示了 FedAvg 與 Fed_BVA、FedBVA_SAT、FedBVA_SAT_Slack 三種可收斂的聯邦對抗訓練演算法在中心伺服器進行 50 次聚合後的 RMSE 損失變化，表 4.25 則為各聯邦系統最終全域模型之 R-squared 比較。觀察損失值變化後，會發現 FedAvg 與 Fed_BVA (橘色線段)損失值保持平穩。

由於 PGD 攻擊對於該資料集與所使用的 VAE 模型的攻擊強度太高，讓客戶端訓練與中心伺服器進行模型聚合時，無法有效提取出乾淨特徵。另外，因應本資料集的特殊性，進一步解釋損失值的狀況。

接著，仔細看圖 4.26 會發現 FedBVA_SAT(綠色線段)與 FedBVA_SAT_Slack (紅色線段)在本實驗的損失曲線圖中呈現一種鋸齒狀的趨勢。這是為了因應電晶體資料集及其使用的 VAE 模型對對抗樣本攻擊的過度敏感性。如果每次聚合都引入專門訓練的全局對抗範例的對抗訓練模型，可能導致模型無法有效學習來自乾淨資料的資訊，進而產生模型失真的問題。為了解決這種問題，我們對 FedBVA_SAT 與 FedBVA_SAT_Slack 演算法做出了調整，改為每十次聚合時才加入對抗訓練模型，使其在其他回合可以收斂至更為穩定的狀態。FedBVA_SAT 演算法在每十次加入對抗訓練模型一起聚合時，損失函數飆高的程度遠高於 FedBVA_SAT_Slack，這與 P 倍加權的比例有關係，下一段將仔細說明，而當溝通輪次高於 30 後，會發現 FedBVA_SAT 演算法的損失值高於其他演算法，經過特殊加權的 FedBVA_SAT_Slack 仍有損失值下降的趨勢。

表 4.28 表 4.29 可以清晰地顯示 Fed_BVA_Slack 與 FedBVA_SAT_Slack 在乾淨資料的 R-平方指標上，皆略優於未經對抗訓練防禦機制的 FedProx，這正是希望看到的實驗結果。

4.8.4 FedProx 聚合方法下的面對 FGSM 攻擊的對抗強健性比較

誠如先前所述，本實驗數據以未進行任何防禦手段的 FedProx 作為基準。即便在受到不同強度的 FGSM 攻擊下，只要 R-平方仍然高於 FedProx，就可以證明該演算法成功地提升了經過 50 次聚合後的聯邦對抗訓練全域模型的對抗強健性。

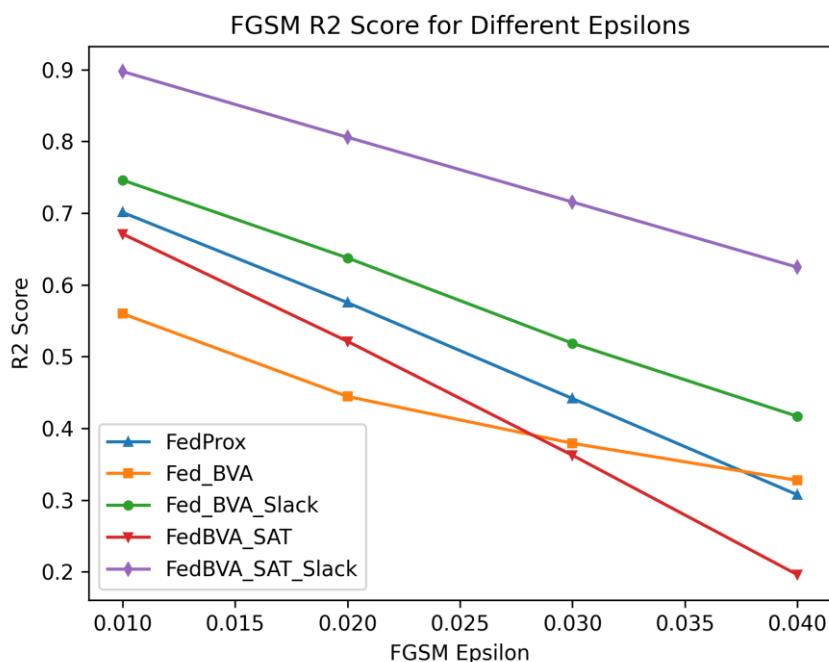


圖 4.27 電晶體資料集:各聯邦對抗訓練系統對 FGSM 攻擊的準確度折線圖(FedProx)

圖 4.27 為不同聯邦對抗訓練系統經過 50 次聚合後的最終全域模型，對於 FGSM 攻擊強度在 $\epsilon = 0.01$ 至 0.04 範圍內的 R-平方(決定係數)折線圖，詳細數據可參考表 4.29。我們以 FedProx (藍色線段)作為判斷標準，可以觀察到本研究提出的 FedBVA_SAT_Slack 在所有攻擊情境下的準確度表現均最為優秀，證明了該演算法在提升對抗強健性方面具有顯著的效能。

表 4.29 電晶體資料集:各聯邦對抗訓練系統對 FGSM 攻擊的 R-平方比較表(FedProx)

FGSM eps 聯邦防禦方法	clean	0.01	0.02	0.03	0.04
FedProx[34]	0.817	0.701	0.575	0.442	0.307
Fed_BVA[23]	0.669	0.560	0.444	0.379	0.327
Fed_BVA_Slack	0.841	0.746	0.638	0.519	0.417
FedBVA_SAT	0.814	0.671	0.521	0.362	0.196
FedBVA_SAT_Slack (proposed method)	0.985 (+0.144)	0.897 (+0.151)	0.806 (+0.168)	0.715 (+0.196)	0.625 (+0.208)

4.9 四大資料集實驗結果的統整分析

這裡將彙整第 4.5 章至第 4.8 章的實驗結果，這些章節內我們對六種不同的聯邦對抗訓練演算法進行了比較：Fed_AT、Fed_AT_Slack、Fed_BVA、Fed_BVA_Slack、FedBVA_SAT 與 FedBVA_SAT_Slack。在這六種演算法中，我們將選出表現最佳的三種方法來進行分析與研究。考量到本研究提出的方法在各項測試中都表現出色，因此 FedBVA_SAT_Slack 演算法必然會被選入。然而，考慮到不同的聯邦聚合方法和資料集特性，選出第二和第三優秀的聯邦對抗訓練演算法可能會因情況而異。

首先，針對影像資料集的 MNIST 與 Fashion-MNIST 進行探討。從第 4.5 章與第 4.6 章的實驗結果，我們可以理解本研究所提出的 FedBVA_SAT_Slack 方法在提高全域模型的對抗強健性上具有相當的效果，如表 4.29 所示。然而，值得注意的是，在 FedAvg 聚合方法下，當遭受高強度的 PGD 攻擊時，FedBVA_SAT_Slack 的性能似乎不及基本的聯邦對抗訓練演算法 Fed_AT，這一點可見於表 4.29 與表 4.30 的紅字部分。這是因為 PGD 攻擊是一種多步迭代的對抗攻擊方法，其特點是能夠針對多維度的影像資料進行深入的干擾，進而導致模型判斷出現誤差。隨著 PGD 攻擊的強度增加，它所生成的干擾變得

更難預測和控制，這也使得 FedBVA_SAT_Slack 應用於影像資料集時，在面對高強度 PGD 攻擊時的防禦能力較差。

儘管 Fed_AT 在抵抗高強度 PGD 攻擊方面表現較佳，但在處理未受攻擊的資料時，其準確度遠不及 FedBVA_SAT_Slack 的高達 96.8%。這顯示 Fed_AT 在抵抗攻擊與保持準確度之間的平衡上仍需改進。總結而言，FedBVA_SAT_Slack 在 FGSM 攻擊與較低強度的 PGD 攻擊的情境下，已展現出卓越的對抗強健性，具體的實驗數據請參見第 4.5 章與第 4.6 章。然而，當涉及高強度的 PGD 攻擊，FedBVA_SAT_Slack 的性能還有進一步優化的可能，值得後續研究與優化。

表 4.30MNIST 資料集的實驗結果:以準確度為指標

聚合方法	聯邦對抗訓練演算法	Clean(%)	FGSM(%)	PGD-20(%)
FedAvg	Epsilon(攻擊強度)	---	0.2	0.2
	Fed_AT[49]	77.4%	38.3%	49.4%
	Fed_BVA[23]	94.8%	59.3%	31.1%
	FedBVA_SAT_Slack (proposed method)	96.8% (+2%)	70.6% (+11.3%)	45.8% (-3.6%)
	Epsilon(攻擊強度)	---	0.2	0.2
FedProx	Fed_AT[49]	91.7%	37.7%	31.5%
	Fed_BVA[23]	98.1%	51.9%	27.8%
	FedBVA_SAT_Slack (proposed method)	98.1% (+0%)	76% (+24.1%)	67.1% (+35.6%)
	Epsilon(攻擊強度)	---	0.2	0.2

表 4.31Fashion-MNIST 資料集的實驗結果:以準確度為指標

聚合方法	聯邦對抗訓練演算法	Clean(%)	FGSM(%)	PGD-20(%)
FedAvg	Epsilon(攻擊強度)	---	0.1	0.25
	Fed_AT_Slack[54]	64.6%	27.9%	17.3%

	Fed_BVA_Slack	91.7%	39.7%	10.1%
	FedBVA_SAT_Slack (proposed method)	91.9% (+0.2%)	50.2% (+10.5%)	14.2% (-3.1%)
	Epsilon(攻擊強度)	---	0.1	0.25
FedProx	Fed_AT_Slack[54]	86.7%	27.5%	10%
	Fed_BVA_Slack	86.4%	11.2%	8.8%
	FedBVA_SAT_Slack (proposed method)	86.4% (-0.3%)	30.3% (+2.8%)	10.8% (+0.8%)

接下來，我們將注意力轉移到真實的工業資料集：線寬資料集。FedBVA_SAT_Slack 再次在提升全域模型的對抗強健性方面表現出色，詳情請參見第 4.7 章。值得強調的是，線寬資料集是以數據表格式（Excel）呈現的資料集。相較於高維度的影像資料集，線寬資料集的預測維度僅有三維，因此其數據結構相對簡單。這使得當這個資料集遭受高強度的 PGD 攻擊時，其所受的影響遠不及高維度影像資料。這一特性也使得 FedBVA_SAT_Slack 演算法在線寬資料集上具有更高的對抗性優勢。

表 4.32 是線寬資料集的實驗整合比較表。特別值得注意的是，當使用 FedAvg 聚合方法時，FedBVA_SAT_Slack 在面對 $\epsilon = 0.15$ 的 PGD 攻擊時，R-平方達到了 0.487。同樣地，在使用 FedProx 聚合方法時，FedBVA_SAT_Slack 方法的 R-平方也達到了 0.731。與其他方法相比，這一結果顯示在低維度的線寬資料集中，該方法具有極其卓越的強健性提升。

表 4.32 線寬資料集的實驗結果:以 R-平方為指標

聚合方法	eps/聯邦對抗訓練演算法	Clean	FGSM	PGD-20
FedAvg	Epsilon(攻擊強度)	---	0.04	0.15
	Fed_AT[49]	0.905	0.576	0.2
	FedBVA_SAT	0.868	0.461	0.081

	FedBVA_SAT_Slack (proposed method)	0.931 (+0.03)	0.723 (+0.109)	0.487 (+0.406)
FedProx	Epsilon(攻擊強度)	-	0.04	0.08
	Fed_AT[49]	0.923	0.612	0.592
	FedBVA_SAT	0.871	0.55	0.597
	FedBVA_SAT_Slack (proposed method)	0.929 (+0.06)	0.735 (+0.123)	0.731 (+0.134)

最後，我們深入探討另一個真實的工業資料集：電晶體資料集。雖然電晶體資料集也以數據表格式（Excel）呈現，但其輸入維度高達 94，相對於線寬資料集，其複雜性明顯提升。值得注意的是，對抗白盒攻擊的策略利用模型的梯度資訊來調整訓練的損失函數。因此，在考慮資料特性的同時，也必須考慮模型的架構。

本研究中，我們採用了 VAE 模型，其目的是將高維度的輸入資料轉化成具有代表性的潛在特徵碼。使用 VAE 模能夠精確地抓取資料的關鍵特徵。然而，該模型對於高維度且具有強烈攻擊性的 PGD 攻擊顯得特別脆弱。鑑於電晶體資料集本身的高維度特性，再結合 VAE 模型的敏感性，即使面臨極微小的 PGD 攻擊擾動，R-平方數值會出現劇烈的變動，甚至直接降至負值。因此，在這一階段的研究，我們主要著重於研究模型在 FGSM 攻擊情境下的對抗強健性，如表 4.33 所示。

最終，根據表 4.33 所示，FedBVA_SAT_Slack 在 FedAvg 或 FedProx 的聚合策略下均實現了顯著的對抗強健性提升，特別是在 FedProx 聚合策略下的效果尤為卓越。這再一次驗證了該方法能夠有效提升對抗強健性，更多詳情請參閱第 4.8 章。

表 4.33 電晶體資料集的實驗結果:以 R-平方為指標

聚合方法	eps/聯邦對抗訓練演算法	Clean	FGSM	FGSM
FedAvg	Epsilon(攻擊強度)	-	0.02	0.04
	FedAvg[3]	0.962	0.59	0.211

	Fed_BVA[23]	0.946	0.539	0.356
	FedBVA_SAT_Slack (proposed method)	0.972 (+0.026)	0.713 (+0.045)	0.53 (+0.174)
	Epsilon(攻擊強度)	-	0.02	0.04
FedProx	FedProx[34]	0.916	0.558	0.167
	Fed_BVA[23]	0.918	0.710	0.516
	FedBVA_SAT_Slack (proposed method)	0.995 (+0.079)	0.86 (+0.15)	0.75 (+0.234)



第五章 結論與未來工作

5.1 結論

本論文提出了一種創新的 FedBVA_SAT_Slack 聯邦對抗訓練演算法，目標是強化聯邦學習系統的全域模型，使其能更有效地抵禦未知對抗樣本的攻擊。

首先，為了提升聯邦學習系統的抵抗對抗攻擊的能力，選擇使用在中心伺服器上生成 BV-FGSM[23]全域對抗範例，目的為增加對抗樣本的多元性。接著，為了遵守聯邦學習的安全聚合原則，將生成出來的 BV-FGSM 全域對抗範例，將其留伺服器上進行專門的對抗訓練。與此同時，每個客戶端只使用自己的數據集進行訓練，並不直接與中央伺服器共享訓練資料。這樣的設計確保了客戶端與中央伺服器之間的資料流程完全分離，以符合聯邦學習中客戶端和中央伺服器不直接傳輸訓練資料的原則。最後，運用了反向鬆弛聚合機制，於每次聚合時從客戶端模型與對抗訓練模型中選取最大損失值並進行特殊加權，此策略能讓聯邦學習系統全局收斂後的最終模型在乾淨準確度與對抗強健性之間找到一個適當的平衡。

本研究涵蓋了四個不同的資料集，包括 MNIST、Fashion-MNIST、線寬資料集和電晶體資料集。值得一提的是，與 Zhou 的 Fed_BVA[23]研究相比，我們不僅探討了 FedAvg 聚合策略下的全域模型強健性，還擴展到了 FedProx 的聚合策略，進一步豐富了研究的深度和廣度。

總體而言，FedBVA_SAT_Slack 在不同的聯邦學習聚合方法和四個不同的資料集下均能有效地在保持乾淨準確度的前提下，提升聯邦學習最終全域模型的對抗強健性。無論是處理影像分類任務:MNIST 和 Fashion-MNIST，還是處理工業數據表的迴歸問題:線寬資料集和電晶體資料集，其都在高準確度的基礎上，展現了顯著的對抗性提升。儘管如此，面對高強度的 PGD 攻擊，本研究所提演算法仍有進一步的提升空間。

最後，關於 FedBVA_SAT_Slack 演算法在各大資料集上的實驗超參數，我們在附錄中已提供詳盡的資料，供讀者參考。

5.2 未來工作

雖然本研究在強化聯邦學習的對抗強健性上有所成效，但仍有許多可以深入探討與改善的地方。首先，對抗訓練不僅能提高模型對抗樣本攻擊的防禦能力，同時也可以在模型洩漏攻擊的防禦上提供貢獻。未來研究可以從這個角度切入，進一步優化該演算法以防止模型洩漏攻擊[16]。

此外，目前的演算法依然需要依賴對資料的先驗知識，以決定最佳的鬆弛參數與加權比例。未來研究可以尋找方法，讓演算法自我調整這些參數，使其能在不同的資料集與攻擊狀況下達到最佳的效能。

最後，嘗試將本演算法應用在其他模型上，例如自然語言處理(NLP)或是影像辨識等領域，驗證其在不同領域的應用性與效能。透過不斷的優化與擴展，我們期望能進一步提升 FedBVA_SAT_Slack 演算法在聯邦學習環境中的應用價值與對抗強健性。



參考文獻

- [1] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843-852.
- [2] R. Shokri and V. Shmatikov, "Privacy-preserving deep learning," in *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 2015, pp. 1310-1321.
- [3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, 2017: PMLR, pp. 1273-1282.
- [4] T. S. Brusimi, R. Chen, T. Mela, A. Olshevsky, I. C. Paschalidis, and W. Shi, "Federated learning of predictive models from federated electronic health records," *International journal of medical informatics*, vol. 112, pp. 59-67, 2018.
- [5] Z. Su *et al.*, "Secure and efficient federated learning for smart grid with edge-cloud collaboration," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, pp. 1333-1344, 2021.
- [6] B. Li, Y. Wu, J. Song, R. Lu, T. Li, and L. Zhao, "DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems," *IEEE Trans. Ind. Inform.*, vol. 17, no. 8, pp. 5615-5624, 2020.
- [7] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Blockchain and federated learning for privacy-preserved data sharing in industrial IoT," *IEEE Trans. Ind. Inform.*, vol. 16, no. 6, pp. 4177-4186, 2019.
- [8] D. C. Nguyen, M. Ding, P. N. Pathirana, A. Seneviratne, J. Li, and H. V. Poor, "Federated learning for internet of things: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1622-1658, 2021.
- [9] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1759-1799, 2021.
- [10] K. Zhang, X. Song, C. Zhang, and S. Yu, "Challenges and future directions of secure federated learning: a survey," *Frontiers of computer science*, vol. 16, pp. 1-8, 2022.
- [11] C. Xie, K. Huang, P.-Y. Chen, and B. Li, "Dba: Distributed backdoor attacks against federated learning," in *International conference on learning representations*, 2019.
- [12] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to backdoor federated learning," in *International conference on artificial intelligence and statistics*, 2020: PMLR, pp. 2938-2948.
- [13] L. Zhao, J. Jiang, B. Feng, Q. Wang, C. Shen, and Q. Li, "Sear: Secure and efficient aggregation for byzantine-robust federated learning," *IEEE Transactions on*

Dependable and Secure Computing, vol. 19, no. 5, pp. 3329-3342, 2021.

- [14] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping," in *ISOC Network and Distributed System Security Symposium (NDSS)*, 2021.
- [15] X. Pan, M. Zhang, D. Wu, Q. Xiao, S. Ji, and Z. Yang, "Justinian's {GAAvernor}: Robust Distributed Learning with Gradient Aggregation Agent," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 1641-1658.
- [16] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE symposium on security and privacy (SP)*, 2017: IEEE, pp. 3-18.
- [17] K. Wei *et al.*, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3454-3469, 2020.
- [18] M. Hardt and G. N. Rothblum, "A multiplicative weights mechanism for privacy-preserving data analysis," in *2010 IEEE 51st annual symposium on foundations of computer science*, 2010: IEEE, pp. 61-70.
- [19] N. Papernot, P. McDaniel, A. Sinha, and M. Wellman, "SoK: Towards the Science of Security and Privacy in Machine Learning."
- [20] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574-2582.
- [21] L. Lyu *et al.*, "Privacy and robustness in federated learning: Attacks and defenses," *IEEE transactions on neural networks and learning systems*, 2022.
- [22] L. Sun *et al.*, "Adversarial attack and defense on graph data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2022.
- [23] Y. Zhou, J. Wu, H. Wang, and J. He, "Adversarial robustness through bias variance decomposition: A new perspective for federated learning," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 2753-2762.
- [24] I. J. Goodfellow, J. Shlens, and C. Szegedy, "EXPLAINING AND HARNESSING ADVERSARIAL EXAMPLES," *stat*, vol. 1050, p. 20, 2015.
- [25] Y. LeCun, C. Cortes, and C. Burges, "MNIST handwritten digit database," ed: Florham Park, NJ, USA, 2010.
- [26] P. Dwivedi and A. Upadhyaya, "A Novel Deep Learning Model for Accurate Prediction of Image Captions in Fashion Industry," in *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2022: IEEE, pp. 207-212.
- [27] S.-F. Tseng, W.-T. Hsiao, K.-C. Huang, and D. Chiang, "The effect of laser patterning

- parameters on fluorine-doped tin oxide films deposited on glass substrates," *Applied surface science*, vol. 257, no. 21, pp. 8813-8819, 2011.
- [28] X.-Z. Chen, S.-S. Hu, H.-H. Hu, Y.-L. Chen, and K.-M. Chen, "Machine Learning-Based Approach to Analyze the Effect of Density of States on the Electrical Properties of a-IGZO TFT," in *2022 IEEE International Conference on Consumer Electronics-Taiwan*, 2022: IEEE, pp. 413-414.
- [29] X. Huang, Y. Ding, Z. L. Jiang, S. Qi, X. Wang, and Q. Liao, "DP-FL: a novel differentially private federated learning framework for the unbalanced data," *World Wide Web*, vol. 23, pp. 2529-2545, 2020.
- [30] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: challenges and applications," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 2, pp. 513-535, 2023.
- [31] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*, 2019: PMLR, pp. 634-643.
- [32] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning," in *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 1-15.
- [33] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the Convergence of FedAvg on Non-IID Data," in *International Conference on Learning Representations*, 2019.
- [34] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *Proceedings of Machine learning and systems*, vol. 2, pp. 429-450, 2020.
- [35] N. Rodríguez-Barroso, D. Jiménez-López, M. V. Luzón, F. Herrera, and E. Martínez-Cámara, "Survey on federated learning threats: Concepts, taxonomy on attacks and defences, experimental study and challenges," *Information Fusion*, vol. 90, pp. 148-173, 2023.
- [36] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," 2021.
- [37] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828-841, 2019.
- [38] Y. Liu, S. Mao, X. Mei, T. Yang, and X. Zhao, "Sensitivity of adversarial perturbation in fast gradient sign method," in *2019 IEEE symposium series on computational intelligence (SSCI)*, 2019: IEEE, pp. 433-436.
- [39] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness May Be at Odds with Accuracy," in *International Conference on Learning Representations*, 2019, no. 2019.

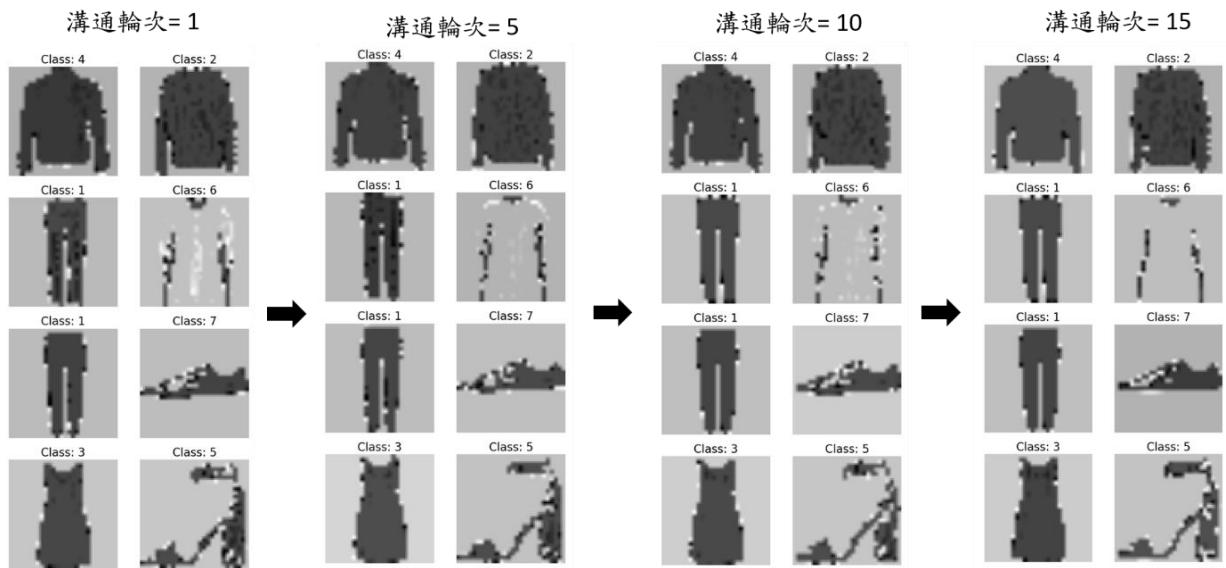
- [40] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," in *International Conference on Learning Representations*, 2016.
- [41] Y. Deng, X. Zheng, T. Zhang, C. Chen, G. Lou, and M. Kim, "An analysis of adversarial attacks and defenses on autonomous driving models," in *2020 IEEE international conference on pervasive computing and communications (PerCom)*, 2020: IEEE, pp. 1-10.
- [42] Y. Li, M. Cheng, C.-J. Hsieh, and T. C. Lee, "A review of adversarial attack and defense for classification methods," *The American Statistician*, vol. 76, no. 4, pp. 329-345, 2022.
- [43] I. Fursov *et al.*, "Adversarial attacks on deep models for financial transaction records," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 2868-2878.
- [44] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*, 2016: IEEE, pp. 582-597.
- [45] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *international conference on machine learning*, 2019: PMLR, pp. 1310-1320.
- [46] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*, 2019: PMLR, pp. 7472-7482.
- [47] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979-1993, 2018.
- [48] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*: Chapman and Hall/CRC, 2018, pp. 99-112.
- [49] G. Zizzo, A. Rawat, M. Sinn, and B. Buesser, "FAT: Federated Adversarial Training," in *Annual Conference on Neural Information Processing Systems*, 2020.
- [50] Y. Wang, X. Ma, J. Bailey, J. Yi, B. Zhou, and Q. Gu, "On the Convergence and Robustness of Adversarial Training," in *International Conference on Machine Learning*, 2019: PMLR, pp. 6586-6595.
- [51] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "A survey on federated learning for resource-constrained IoT devices," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 1-24, 2021.
- [52] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1-58, 1992.

- [53] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, vol. 2, 2016.
- [54] J. Zhu, J. Yao, T. Liu, J. Xu, and B. Han, "Combating Exacerbated Heterogeneity for Robust Models in Federated Learning," in *The Eleventh International Conference on Learning Representations*, 2022.
- [55] P. Liu, X. Xu, and W. Wang, "Threats, attacks and defenses to federated learning: issues, taxonomy and perspectives," *Cybersecurity*, vol. 5, no. 1, pp. 1-19, 2022.
- [56] Z. Yang, Y. Yu, C. You, J. Steinhardt, and Y. Ma, "Rethinking bias-variance trade-off for generalization of neural networks," in *International Conference on Machine Learning*, 2020: PMLR, pp. 10767-10777.
- [57] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15849-15854, 2019.
- [58] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.
- [59] C. Duan, P. Yin, Y. Zhi, and X. Li, "Image classification of fashion-MNIST data set based on VGG network," in *Proceedings of 2019 2nd International Conference on Information Science and Electronic Technology (ISET 2019). International Informatization and Engineering Associations: Computer Science and Electronic Technology International Society*, 2019, vol. 19.
- [60] S.-F. Tseng, W.-T. Hsiao, K.-C. Huang, and D. Chiang, "Electrode patterning on PEDOT: PSS thin films by pulsed ultraviolet laser for touch panel screens," *Applied Physics A*, vol. 112, pp. 41-47, 2013.
- [61] G. Hinton and R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," *IEEE Trans. Microw. Theory Tech*, vol. 47, p. 2075, 1999.
- [62] J. Sun, X. Wang, N. Xiong, and J. Shao, "Learning sparse representation with variational auto-encoder for anomaly detection," *IEEE Access*, vol. 6, pp. 33353-33361, 2018.
- [63] 王宣融, "基於邊緣運算之聯邦式學習系統惡意攻擊防護," 碩士, 資訊工程系, 國立臺北科技大學, 台北市, 2022. [Online]. Available: <https://hdl.handle.net/11296/q99cj9>

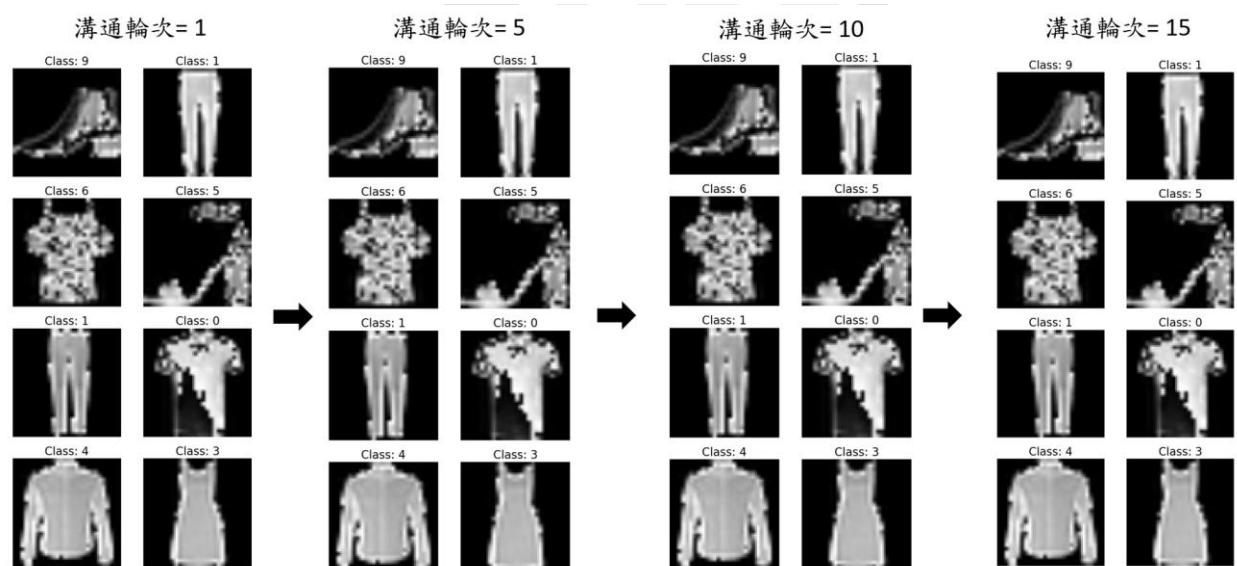
附錄

附表 6.1 FedBVA_SAT_Slack 演算法在四個資料集的實驗超參數

	MNIST	Fashion-MNIST	線寬資料集	電晶體資料集
Non-IID setting	n-shards	skew parameter	rate	Numbers
溝通輪次(T)	15	20	20	50
Clients (K)	5	5	10	3
Local epochs (E)	5	10	10	500
Local batch size	256	128	1024	64
Model	CNN	VGG11	Auto-Encoder	VAE
Optimizer	Adam	Adam	Adam	Adam
Local learning rate	0.01	0.01	0.01	0.001
輔助資料集數量	200	200	200	50
BV-FGSM epsilon	0.3	0.45	0.3	0.005
Server epochs	50	50	200	500
Server batch size	128	128	512	64
Server Optimizer	Adam	Adam	Adam	Adam
Server learning rate	0.01	0.01	0.01	0.1
p/p'(加權比例)	1.2/1	1.2/1	1.2/1	0.005/1
Slack Aggregation 的使用時機	溝通輪次>3 開始啟用	每三個輪次， 使用一次；其 他輪次只聚合 客戶。	溝通輪次>2 開始啟用	每十個輪次， 使用一次；其 他輪次只聚合 客戶。



附圖 6.1 BV-FGSM 全域對抗範例



附圖 6.2 PGD 對抗範例