CCF Based System Framework In Federated Learning Against Data Poisoning Attacks

Ibrahim M. Ahmed[1*] and Manar Younis Kashmoola[2]

[1] *College of Science, University of Mosul, Mosul, Iraq*
[2] *College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq*
[*] *Corresponding author. E-mail: ibrahim$_a lhlima@uomosul.edu.iq$*

Received: Aug. 17, 2022; Accepted: Sept. 04, 2022

Nowadays, smart systems attract a lot of attention as several smart applications are growing. Distributed machine learning such as federated learning has an essential role in smart systems including 6G applications. The main issues that face federated learning (F.L.) are security and performance, which are could be affected by the poisoning attack models. One of the most common poisoning attacks is an impersonation attack, such Sybil attack. This paper proposes a new framework that increases the security of federated learning against Sybil poisoning attacks. The proposed framework which is called FED_CCF, creates a hybrid environment using federate learning with Microsoft CCF (Confidential Consortium Framework). It provides a secure and reliable environment that misleads attackers targeting federated learning. The MNIST dataset is used to investigate the performance of F.L. model with FED_CCF in terms of accuracy. The F.L. model is evaluated by exploiting the MNIST dataset and 30% of malicious devices that use the Sybil attack. The experimental results show that F.L. system implementing FED_CCF outperforms Vanilla F.L. in terms of accuracy, where the former acquired approximately 95.2 % compared to the latter, which only obtains 2.55% employing Sybil poisoning attack.

Keywords: Data poisoning; Federated learning; CCF; Sybil poisoning attacks.

http://dx.doi.org/10.6180/jase.202307_26(7).0008

1. Introduction

6G networks and application scenarios aim to improve connectivity from a social and human perspective and build communication technologies to improve the overall quality of life [1]. IoT in cloud and fog technology, IoT with applications and IoT architecture for design and development using 6G sensors has become a feature of the next era of information technology [2]. Smart technology, including Artificial Intelligence technology (AI) is used in different fields. Distributed intelligent technologies such as Federated learning are emerging technology that grabbed the research community's attention [3, 4]. Federated Learning (F.L.) is a distributed machine training paradigm that trains a large-scale deep neural network. Each participant trains only on their local data with multiple epochs and sends your parameter to a global central aggregator [5, 6]. The main global aggregates collect the values of the updated parameters for the training round and then choose the best of them according to one of the aggregation methods. After determining the best local parameter updates, the global aggregator publishes them to the members of the F.L. models as an initial value for the next round [7, 8]. The F.L. system, by its nature, exposes to additional risks due to potentially malicious participants [9]. A malicious group of participants is trying to poison the F.L. system with fake parameters by sending model updates derived from the mislabeled data to the global model, which leads to a decrease in accuracy [10].

Moreover, the targeted data poisoning attacks against F.L. were studied extensively in our early works [11] with details, and it is suggested that it is necessary to provide security solutions that allow the F.L. model to avoid the risks of poisoning the model [12]. These solutions focus on building a reliable execution environment to avoid tampering with a malicious participant. Usually, the attackers use methods that target the aggregators in the F.L. system, such as Sybil attacks [13–15].

Attack on The F.L. model is the most complex process in the smart model attack chain because the attack occurs during the training process, making it very difficult to discover because the system does not have enough information about the participants [16]. A Sybil attack in which malicious participants exploit the creation of many copies to impersonate honest participants and propagate into the system. And performing similar activities by poisoning the Global model through label flipping [17, 18]. Sybil attack is a type of malicious computer network service attack. The attacker creates many identities with aliases and uses them to significantly affect the system's reputation through false information [19, 20]. In Sybil assaults, a single attacker leads many infected computers to launch a coordinated attack [21, 22]. In this paper, Federated learning systems face many security threats, including data poisoning attacks like Sybil poisoning attacks. To avoid this attack, we used the CCF framework in developing the F.L. system framework by getting a containment zone that filters out malicious participants and diagnoses its action. Then, this technology will prevent the attacker from joining the federated learning model as a participant in the training process. The main contributions of this paper include the following:

1. Proposed a new decentralized, federated learning based on CCF called FED_CCF.
2. Preventing and detecting poisoning attack model (Sybil) in the F.L. system.
3. The performance of the proposed framework (FED_CCF) is evaluated via many experiments using MNIST datasets and against Vanilla Federated learning for solving a classification problem with many device participants with 30% malicious devices depending on the Sybil attack. The rest of this paper includes the following: In section II, the recent related work is analyzed. The description of the CCF is provided in section III. Section IV consists of the methodology. Proposed Trusted F.L. based on CCF(FED_CCF). Finally, the analysis of the proposed framework is explained in section V.

2. Related work

The author's in [22] Investigation of targeted data poisoning attacks against F.L. systems. A malicious participant attempts to corrupt the global model by providing model updates based on misleading data. Which leads to serious consequences and reduces the accuracy of classification and retrieval. The researchers presented an innovative defence mechanism that detects and repels but does not prevent the attack. And in [23], the authors studied the problem of poisoning the federated learning model by Sybil's attack. They proposed a Fools-Gold defence model depending on the similarity of the customer's contribution. Fools-Gold can thwart a range of attacks.

It is effective when users are infected and detected by diagnosing how Sybil's attack combines poisoned data with real data by adding intelligent noise to their updates. This team confirmed that Fools-Gold had provided better results than previous strategies. Still, it did not prevent the attack, which indicated that the attacker had breached the system, enabling him to sabotage later. In [24], the authors assumed that an attacker could gain access to some client machines during the learning process. Then attacker modifies the parameters of the local form on the client machines. As a result, the global model will have a high test error rate. Four Byzantine F.L. As methods of defence showed that individual protection could effectively resist innovative attacks. However, in other cases, defences are inadequate, underscoring the need for new defences against poisoning the local model of federated learning. In [25], the authors suggested a technique for improving training problems by using parameter estimation to improve updates to benign factors by producing data using various interpretation strategies. Typical intoxication can occur in benign and malignant forms, thus emphasizing the need for good defensive tactics in an F.L. environment. In [20], the authors suggested a technique for improving training loss by estimating parameters to improve updates for benign factors. However, the researchers have found that poisoned patterns can be carried and attacked while remaining undetected, thus emphasizing vulnerability and the need for good defensive tactics in a standardized learning environment. In [26], the authors presented a new, improved method for intelligent learning technology by integrating a measure to meet the intelligent model poisoning attack. However, experience shows that a multilateral attack can successfully escape detection tactics. So, when attackers run from these techniques, the system will be vulnerable to counterfeiting and thus may collapse. In [27], The authors proposed a feature-relevant (Fed-Fi) detrimental federated learning model detection method. Examining key features using LRP-based feature-importance reasoning and analyzing their similarities using Hamming distance. But their system results showed an attack effect of up to 3%. In [28], the authors proposed a "sniper" scheme to remove poisoned domestic models from malicious participants during training. By solving the maximal nerve problem, Sniper identifies benign local models, ignoring suspicious (poisoning) local models while updating the generic model. In [29], the authors have proposed a Federated User Verification (FedUV) model, a framework in which users jointly learn a set of vectors and maximize the correlation of their inclusions with a secret linear set of those vectors. In [30], the authors propose Local Malicious Factor (Local Malicious Factor) as a two-stage protection method (LoMar). The LoMar rates model is updated from each remote client in the first stage by calculating the relative distribution across their neighbours using a kernel density estimation approach. In the second stage, an appropriate threshold is estimated to distinguish between malicious and clean updates statistically. In [31], the authors provide a theoretical framework for studying the resilience of poisoning defenses (SparseFed) and provide a robustness and convergence analysis of our method. We undertake an open-source assessment at scale across various benchmark datasets for computer vision and federated learning to confirm its empirical effectiveness. In [32], the authors proposed aggregating rule dubbed Distance-based Outlier Suppression (DOS) using Copula-based Outlier Detection. The proposed technique computes the distance between local parameter changes of various clients and generates an outlier score for each client (COPOD).

The resultant outlier scores are turned into normalized weights using a SoftMax function. Table 1 shows the most recent proposed techniques in the literature; the aim of these techniques is to solve federated learning issues. Each proposed method has common limitations. The key point in F.L. is to detect and avoid Sybil poisoning attacks. Therefore, this paper proposes a decentralized, federated learning system based on a safe and reliable environment called FED_CCF.

3. Confidential consortium framework (ccf)

The Confidential Consortium Framework (CCF) is an open-source framework for building a new type of application that focuses on multiparty compute and data and are safe, highly available, and fast. CCF makes enterprise-ready multiparty systems possible by using the power of trusted execution environments (TEE or enclave), decentralized systems ideas, and cryptography. CCF is built on web technologies. A CCF network A group of Members runs a CCF network consortium. The Constitution, the ledger, spells out members' rules. With consensus protocol, members can vote to let a new trusted user send requests to

Table 1. Comparison of time performance and performance with different superpixel methods

Ref.	Methodology adopted	Finding	Weakness
[18]	Defending Against Label Flipping	Label flipping Attack	Not Avoid Attack
[19]	Fools-Gold	Sybil-based label-flipping and backdoor poisoning attacks	Sybil poisoning attack rate 91.17%
[20]	defend against our local model poisoning attacks	Byzantine failures of some client devices.	Only detect a local attack
[21]	a sequence of model poisoning attacks	Local poisoning attack proposed	Only detect the local proposed attack
[24]	attacks that have to analyze individual clients' model updates	backdoor Attacks	Not Avoid Attack
[27]	Fed-Fi	Sybil attack	Sybil poisoning attack rate 3.43%
[27]	HMFL	Sybil attack	Sybil poisoning attack rate 90.56%

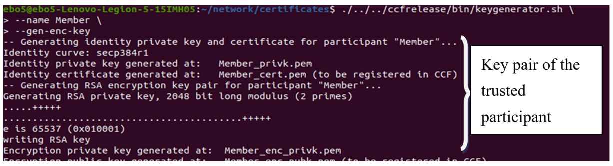
the application or to add a new member to the consortium. Governance rules for a CCF network are defined in a module called the Constitution. it contains four elements (set of executable actions, which members can propose and submit to a vote, validate () function, resolve () function and apply() function).in addition used Consensus Protocols. CCF provides Trusted Execution Environment (TEE) based on the blockchain with high throughput, high availability, and low latency while protecting application integrity and confidentiality [33]. CCF uses distributed ledger technology for saving data and transactions and also as an audit trail. CCF uses the Intel SGX to configure TEE by creating an Open Enclave as a protected section of memory. This Open Enclave just allows code execution for trusted host members of the CCF federation [34].

Trusted members of the consortium make voting on any transaction to allow the service they want to run depending on governance protocol. Members may be located in one or more cloud or corporate data center members, but all hosts support TEE regardless of location. The risk of untrusted hosts can be reduced by redundant service on multiple TEEs. CCF also provides an optional governance technology to recover from the replication protocol's catastrophic failures and ensure no other software code accesses private transactions within the TEE. Transactions are encrypted in the ledger and remain confidential even if hosts, networks, and most members are compromised [35]. In addition, the CCF maintains a Merkle tree for collecting transaction signatures. It systematically records signed evidence in the ledger to resist severe attacks, allowing service protocols to be restarted and malicious behaviors by its members detected and replicated [36].

4. Proposed framework

In secure F.L. systems, both authentication and provenance are critical to ensure the integrity of the final results, and this is done by validating the system's operation on both local data sources and models. In the methodology of the proposed system, each participant in F.L. system must obtain approval to join the Executive Environment after passing the checkpoint when attempting. Then, it will send a request to a system member backed by a node configuration certificate as per CCF specification. This point in the environment is the first containment zone that accepts only the honest participant and rejects any suspicious participants. The proposed method contains five phases as shown in Fig. 3. These phases are distributed between two containment areas. The first contentment area includes the first three phases which are responsible for allowing the trusted client to join FED_CCF framework. The second contentment areas includes phases 4 and 5 which are responsible for ensuring the aggregation process for FED_CCF framework. These contentment areas are used to prevent all attempts of model poisoning attack. The five phases are described in details as explained bellow:

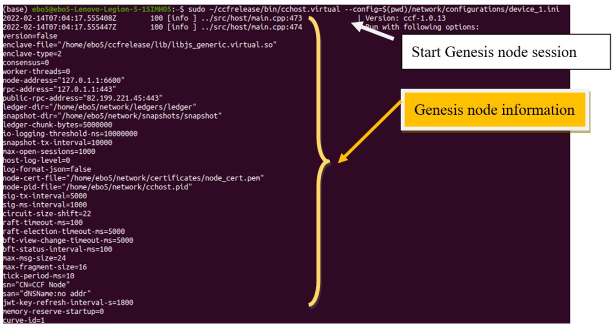
Phase 1: : when a trusted participant node is created, the arguments for that node must contain the unique identifier of the intended application. Then several steps are performed, such as key generation (public key signature; node secret key) and transition state creation. First, the public key is used to authenticate the TLS (Transport Layer Security) server and verify signed messages, as shown in Fig. 1. It then goes to the "trusted" or "pending" state.



```
ebo5@ebo5-Lenovo-Legion-5-15IMH05:~/network/certificates$ ./../../ccfrelease/bin/keygenerator.sh \
> --name Member \
> --gen-enc-key
-- Generating identity private key and certificate for participant "Member"...
Identity curve: secp384r1
Identity private key generated at:   Member_privk.pem
Identity certificate generated at:   Member_cert.pem (to be registered in CCF)
-- Generating RSA encryption key pair for participant "Member"...
Generating RSA private key, 2048 bit long modulus (2 primes)
.....+++++
..........................................+++++
e is 65537 (0x010001)
writing RSA key
Encryption private key generated at:  Member_enc_privk.pem
Encryption public key generated at:   Member_enc_pubk.pem (to be registered in CCF)
```

Fig. 1. Key generation of consortium participant.

Phase 2: when the service is opened, the service issues public key certificates to each trusted node once the starting node is configured with the signing key and the intermediate certificate. Still, before sending instructions, Consortium members can certify any of these nodes as part of the FED_CCF consortium through a members vote. In addition, the service takes governance directives when opening the application environment, and the first node is called the Genesis node, as shown in Fig. 2. Consortium members verify the member requesting Fig.2. Genesis to join the CCF Consortium through the certificates he submits. Therefore approval will be given to the client after verification, and an Ack will be issued. This process eventually activates the service and makes all its application commands available.



```
(base) ebo5@ebo5-Lenovo-Legion-5-15IMH05:~$ sudo ~/ccfrelease/bin/cchost.virtual --config=$(pwd)/network/configurations/device_1.ini
2022-02-14T07:04:17.555408Z        100 [info ] ../src/host/main.cpp:473                 | Version: ccf-1.0.13
2022-02-14T07:04:17.555447Z        100 [info ] ../src/host/main.cpp:474                 | Run with following options:
version=false
enclave-file="/home/ebo5/ccfrelease/lib/libjs_generic.virtual.so"
enclave-type=2
consensus=0
worker-threads=0
node-address="127.0.1.1:6600"
rpc-address="127.0.1.1:443"
public-rpc-address="82.199.221.45:443"
ledger-dir="/home/ebo5/network/ledgers/ledger"
snapshot-dir="/home/ebo5/network/snapshots/snapshot"
ledger-chunk-bytes=5000000
io-logging-threshold-ns=10000000
snapshot-tx-interval=10000
max-open-sessions=1000
host-log-level=0
log-format-json=false
node-cert-file="/home/ebo5/network/certificates/node_cert.pem"
node-pid-file="/home/ebo5/network/cchost.pid"
sig-tx-interval=5000
sig-ms-interval=1000
circuit-size-shift=22
raft-timeout-ms=100
raft-election-timeout-ms=5000
bft-view-change-timeout-ms=5000
bft-status-interval-ms=100
max-msg-size=24
max-fragment-size=16
tick-period-ms=10
sn="CN=CCF Node"
san="dNSName:no addr"
jwt-key-refresh-interval-s=1800
memory-reserve-startup=0
curve-id=1
```

Fig. 2. node working under CCF environment

Fig.3. shows the flow chart of the proposed method for two contentment areas.

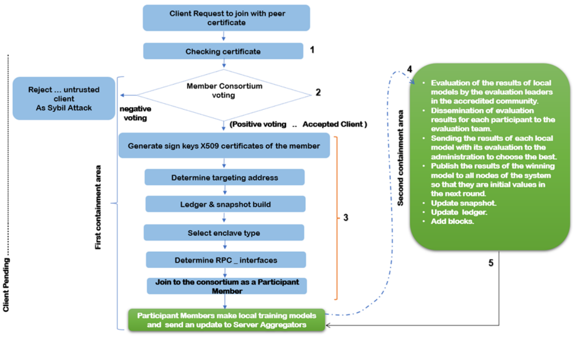

Client Request to join with peer certificate
Checking certificate
1
Member Consortium voting
2
Reject ... untrusted client As Sybil Attack
negative voting
(Positive voting .. Accepted Client)
Generate sign keys X509 certificates of the member
Determine targeting address
Ledger & snapshot build
Select enclave type
Determine RPC _ interfaces
Join to the consortium as a Participant Member
3
Participant Members make local training models and send an update to Server Aggregators
First containment area
Client Pending
4
Second containment area
Evaluation of the results of local models by the evaluation leaders in the accredited community.
Dissemination of evaluation results for each participant to the evaluation team.
Sending the results of each local model with its evaluation to the administration to choose the best.
Publish the results of the winning model to all nodes of the system so that they are initial values in the next round.
Update snapshot.
Update ledger.
Add blocks.
Global model Training
5

Fig. 3. first containment area by CCF checkpoint

Phase 3: after the service is configured and made available, CCF provides auditing and authentication capabilities to maintain high levels of service integrity. Even if some nodes or their keys are compromised, they will be immediately diagnosed by members of the CCF consortium. Blockchain technology [13] provides high reliability to CCF members through its effectiveness in documenting and tracking events in the environment in which it supports CCF members, distributed ledger replicas will be exchanged, and signatures will be stored across the network. The distributed ledger will provide additional security for the members. Also, to help ensure the integrity of the machine learning system by providing an audit trail for the training processes system or to evaluate its new data. It also saves the order of the different actions, as CCF delivers a receipt that can be used to ensure the reliability of the machine learning object. Moreover, a Merkle tree representing transaction signatures is used in the ledger's contents. Finally, the register verifies the sequence of actions stored in the snapshot file.

Phase 4: after the client joins the FED_CCF consortium as a participant, it will be received by the members of the consortium in F.L. They then gave a copy of the MNIST-CNN form with the latest update of the last Global model parameters. Finally, it is copied to its own distributed ledger, at which point it will resume the local training process as a participant worker and on its data, as shown in Fig. 4. Each participant worker makes several training epochs on the local data and sends to the evaluation by the global model members; then, the Members of the global model vote after testing the local parameter of the participant worker. And sending the voting result and the local participant worker parameters to the aggregator Members in FED_CCF.

Phase 5: after that, the global model members select the best local model parameter depending on equation 1 and sending back to all consortium members. All these operations are repeated in each round of training of the global model. The training process in the proposed FED_CCF is presented in Algorithm 1.

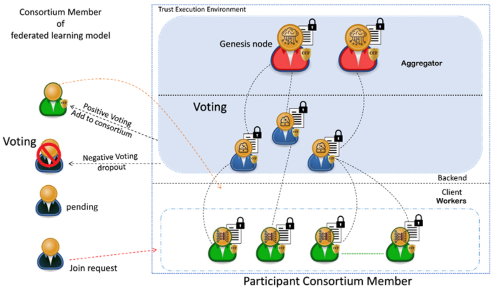

Consortium Member of federated learning model
Trust Execution Environment
Genesis node
Aggregator
Voting
Positive Voting Add to consortium
Voting
Negative Voting dropout
pending
Join request
Backend
Client Workers
Participant Consortium Member

Fig. 4. Proposed F.L. framework Methodology.

$$G_r^w = L_i^w \leftarrow \text{ACCi}\left(\text{eva}\left(L_i^w\right)\right) > \text{ACCr}\left(\text{eva}\left(G_r^w\right)\right) \tag{1}$$

The average accuracy of vanilla F.L. could be calculated using equation 2

$$\text{Accuracy FedAVG} = = \frac{ACCi}{\sum_i^k ACCi} \tag{2}$$

5. System implementation and evaluation

The implementation of the proposed FED_CCF is conducted on a virtual machine with a processor 2.6 core I7, 11th generation, 16 GB RAM, and 4 G.B. Nvidia 1650 TI GPU with the Ethereum blockchain platform.

Also, many programming languages are used in implementing the FED_CCF system, which are Python, Typescript, Jason and C++ in the Ubuntu 20.4 operating system. All experiments are executed using ten devices for both Vanilla F.L. and FED_CCF. Random distribution based on the IID method [29] has been used for allocating the MNIST training set for each participant.

The MNIST dataset is one of the benchmark datasets that is used in evaluating federated learning systems. It contains 70000 samples with 60000 samples as a training set and 10000 samples as a testing set. MNIST samples include handwritten grayscale digit images between 0 and 9. Each sample is represented as a feature Matrix consisting of 28 × 28 pixels. In addition, the neural network architecture which is used for implementing FED_CCF is the MNIST-CNN Convolutional Neural Network structure [37, 38]. With a learning rate of 0.01 and 10 batch size. All the experiments utilize the SGD optimizer.

The performance of FED_CCF is evaluated by running ten experiments with 20 communication rounds, ten epochs and 10 participants. 30% of the 10 participant devices were

Algorithm 1. FED_CCF

```
Server cite: For all CCF participants P
Initialize parameter of models $L^{pi}_{w0}$ MNIST Data Set D Shredding by D/P
for each round R = 1, 2, . . . .  do
  for for each participant, Pi ... do
    for for each epoch e in E  do
      $ACC_j$ =make cross-validation for local participant model on its Data set Dj
    end for
    return local_model update $L^{pi}_{w}$= mean (max($ACC_j$), min ($ACC_j$) to Global model aggregators in server site
  end for
  Global model_members make testing and validation for each participant's local_model update $L^{pi}_{w}$ by : Calculate the average loss value for each received local_model update $L^{pi}_{w}$   by $\Delta L^{pi}_{wr} = (O - O^0)$
  Determine the winning local_ model update $L^{pi}_{w0}$ and Send back the winning local_ model update $L^{pi}_{w0}$ to all participant
end for=0
```

Table 2. List of Notations.

Symbols	Meaning
R	no of training round (of the global model)
E,e	Total no. of Epochs, e is the number of local epochs.
P,kϵD	The participant's client's device is indexed in the Device Set
L^{pi}_{w}	Locally updated model by device d in round Rj
$\mathbf{G}^{W}_{r}$	Gj Global model constructed at the end of round Rj
ACCi	Accuracy of each local model
ACCr	Accuracy of the global model
IID	Independent and identically distributed

malicious. In the experiment, in each run, 30% of malicious devices were randomly selected out of the total number of devices; the number of malicious participants did not exceed 30% of the total number of participants who were run in various communication rounds.

The results of the proposed FED_CCF system that classified the MNIST dataset show that the malicious devices with Sybil attack could not penetrate the FED_CCF. Fig. 5. shows the classification accuracy of experiment 7 of the FED_CCF versus Vanilla F.L. with Sybil attack in communication round 3.

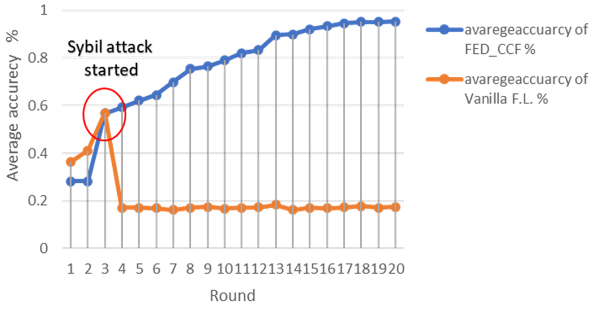


Fig. 5. The classification accuracy of experiment 7 for the FED_CCF and Vanilla F.L.

It is obvious, after applying the attack on the two systems in the 3rd communication round, the results demonstrate that the FED_CCF outperformed Vanilla F.L. with around 95% classification accuracy and was unaffected by the Sybil attack.

Fig. 6. illustrates the results of 10 experiments for both the FED_CCF and Vanilla F.L. with the Sybil attack. The result has shown the superiority of FED_CCF compared with the Vanilla F.L. in all the experiments. In experiment 1, the FED_CCF system achieved 93.44% classification accuracy rate, while the Vanilla F.L. achieved 15.21% classification accuracy rate only. Meanwhile, in experiment 10, the FED_CCF system achieved a 95.24% classification accuracy rate compared to 17.11% in Vanilla F.L. classification accuracy rate only. So, it is concluded that the proposed system FED-CCF has superior performance against the Sybil attach with very high classification accuracy. As shown in Fig. 6, which describes the Average accuracy of 10 experimental results. To do more investigation on the performance of the proposed FED_CCF through comparison with presented models in the literature. FED_CCF is able to prevent data poisoning attacks because of the possibility of diagnosing it during an attempt to attack. In contrast, the other models were limited to detecting attacks after the occurrence.

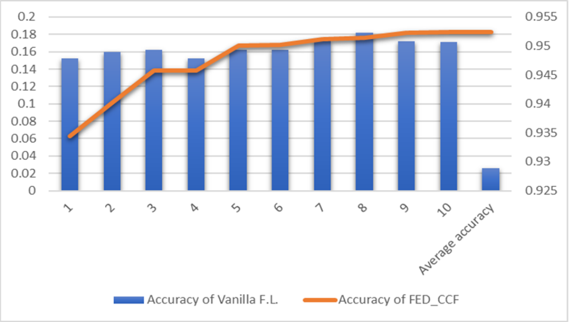


Fig. 6. Average accuracy of FED_CCF and Vanilla F.L.

Table 3 compares the accuracy and Sybil poisoning attack rate between FED_CCF and other algorithms. It can be seen that the accuracy of FED_CCF under collusive Sybil poisoning attack is high; at the same time, Fool Golds and HMFL algorithms can barely resist this attack; their attack rate is as high as 91.17% and 90.56%. However, in Fed-fi possibility of infection reaches 3.43%.

Table 3. Comparison with other algorithms

Technique	Accuracy	Sybil poisoning attack (%)
Fed-fi[27]	90.45	3.43
Fool golds[23]	79.27	91.17
Vanilla F.L.[33]	0.17	98.2
HMFL[27]	80.21	90.56
FED_CCF	95.24	0

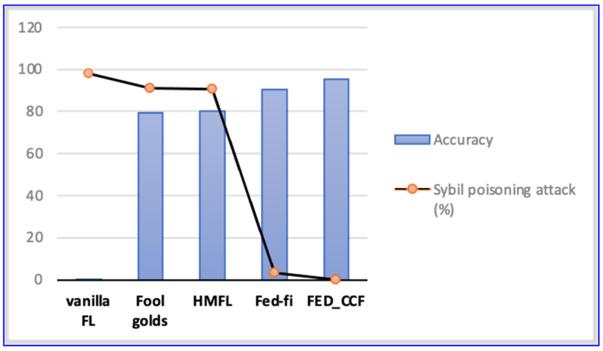


Fig. 7. Performance index of Sybil poisoning attacks detection algorithm.

The changing effect of attacks between these algorithms shows that the attack reached the system. And according to the effect ratios, it is evident that the attacker used different poisoning methods as a label flipping for a specific target in the MNIST dataset or backdoor attack. Experimental results showed that FED CCF is effective against Sybil attacks with zero attack rate, as shown in Fig. 7. because CCF avoids Sybil attacks by excluding malicious participants and adding there to the blocklist. And when the malicious participant attempts to affect other trusted members voting to reject its model, these two transactions provide a secure federated learning paradigm against Sybil's attack.

6. Conclusion

Federated learning is a new way of thinking about how to share data securely, and it can be used to do cross-domain collaborative analysis without gathering data from multiple parties. It is thought to be the best way to ensure that data is safe to exchange. Trying to create a situation like the Sybil attacks. One of the primary vulnerabilities of F.L. systems is targeting the aggregator in the F.L. system by inserting a poisoning model through malicious participants to put the global model under its control. In this study, a new decentralized, federated learning framework called FED_CCF is proposed based on the CCF environment to detect and avoid Sybil poisoning attacks. When the aggregator tasks are distributed, FED_CCF makes the attackers' task almost impossible through camouflage. FED_CCF will provide the mechanisms for honest participant identity and the ability to resume the work of the target nodes. In addition, it is provided more protection through using the Intel SGX microprocessor by building a protection zone for the honest participants.

Moreover, it will provide more integrity through the audit records which are represented by the distributed ledger. Furthermore, it provides high reliability due to its reliance on digital signatures and consensus protocols that the members of consortium FED_CCF adopt in accepting local participant models. The implementation of the proposed FED_CCF is executed based on MNIST-CNN architecture, and the MNIST dataset is used to evaluate the performance of FED_CCF in terms of accuracy. The experimental results have shown that the accuracy of FED_CCF under collusive Sybil poisoning attack is 95.24%, and the proposed FED_CCF is an effective malicious model for detection and prevention.

In future work, the proposed system may evaluate against another type of poisoning model attack such as GAN.

References

[1] A. A. Laghari, A. K. Jumani, and R. A. Laghari, (2021) *"Review and state of art of fog computing"* **Archives of Computational Methods in Engineering** *28*(5): 3631–3643. DOI: doi.org/10.1007/s11831-020-09517-y.

[2] A. A. Laghari, K. Wu, R. A. Laghari, M. Ali, and A. A. Khan, (2021) *"A review and state of art of Internet of Things (IoT)"* **Archives of Computational Methods in Engineering**: 1–19. DOI: doi.org/10.1007/s11831-021-09622-6.

[3] M. B. Janjua, A. E. Duranay, and H. Arslan, (2020) *"Role of wireless communication in healthcare system to cater disaster situations under 6G vision"* **Frontiers in Communications and Networks *1***: 610879. DOI: doi.org/10.3389/frcmn.2020.610879.

[4] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, (2020) *"Federated optimization in heterogeneous networks"* **Proceedings of Machine Learning and Systems *2***: 429–450.

[5] X. Zhang and X. Luo, (2020) *"Exploiting defenses against GAN-based feature inference attacks in federated learning"* **arXiv preprint arXiv:2004.12571**: DOI: doi.org/10.48550/arXiv.2004.12571..

[6] L. Lyu, H. Yu, and Q. Yang, (2020) *"Threats to federated learning: A survey"* **arXiv preprint arXiv:2003.02133**: DOI: doi.org/10.48550/arXiv.2003.02133.

[7] J. Huang, R. Talbi, Z. Zhao, S. Boucchenak, L. Y. Chen, and S. Roos. "An exploratory analysis on users' contributions in federated learning". In: *2020 Second IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE. 2020, 20–29. DOI: doi.org/10.1109/TPS-ISA50397.2020.00014.

[8] Y. Liu, X. Yuan, R. Zhao, Y. Zheng, and Y. Zheng, (2020) *"Rc-ssfl: Towards robust and communication-efficient semi-supervised federated learning system"* **arXiv preprint arXiv:2012.04432**: DOI: doi.org/10.48550/arXiv.2012.04432.

[9] R. Nazir, K. Kumar, S. David, M. Ali, et al., (2021) *"Survey on wireless network security"* **Archives of Computational Methods in Engineering**: 1–20. DOI: doi.org/10.1007/s11831-021-09631-5.

[10] Z. A. Hamed, I. M. Ahmed, and S. Y. Ameen, (2020) *"Protecting windows OS against local threats without using antivirus"* **relation *29***(12s): 64–70. DOI: doi.org/10.3389/frcmn.2020.610879.

[11] I. M. Ahmed and M. Y. Kashmoola. "Threats on Machine Learning Technique by Data Poisoning Attack: A Survey". In: *International Conference on Advances in Cyber Security*. Springer. 2021, 586–600. DOI: doi.org/10.1007/978-981-16-8059-5_36.

[12] M. Waqas, K. Kumar, A. A. Laghari, U. Saeed, M. M. Rind, A. A. Shaikh, F. Hussain, A. Rai, and A. Q. Qazi, (2022) *"Botnet attack detection in Internet of Things devices over cloud environment via machine learning"* **Concurrency and Computation: Practice and Experience *34***(4): e6662. DOI: doi.org/10.1002/cpe.6662.

[13] M. Russinovich, E. Ashton, C. Avanessians, M. Castro, A. Chamayou, S. Clebsch, M. Costa, C. Fournet, M. Kerner, S. Krishna, et al., (2019) *"CCF: A framework for building confidential verifiable replicated services"* **Technical report, Microsoft Research and Microsoft Azure**:

[14] S. V. A. Amanuel and S. Y. A. Ameen, (2021) *"Device-to-device communication for 5G security: a review"* **Journal of Information Technology and Informatics *1***(1): 26–31.

[15] E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov. "How to backdoor federated learning". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2020, 2938–2948.

[16] Y. Zhao, J. Chen, J. Zhang, D. Wu, M. Blumenstein, and S. Yu, (2022) *"Detecting and mitigating poisoning attacks in federated learning using generative adversarial networks"* **Concurrency and Computation: Practice and Experience *34***(7): e5906. DOI: doi.org/10.1002/cpe.5906.

[17] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, and T. Goldstein, (2022) *"Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses"* **IEEE Transactions on Pattern Analysis and Machine Intelligence**: DOI: doi.org/10.1109/TPAMI.2022.3162397.

[18] Y. Jiang, Y. Li, Y. Zhou, and X. Zheng. "Sybil Attacks and Defense on Differential Privacy based Federated Learning". In: *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. IEEE. 2021, 355–362. DOI: doi.org/10.1109/TrustCom53373.2021.00062.

[19] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, (2016) *"Infogan: Interpretable representation learning by information maximizing generative adversarial nets"* **Advances in neural information processing systems *29***:

[20] G. Sun, Y. Cong, J. Dong, Q. Wang, L. Lyu, and J. Liu, (2021) *"Data poisoning attacks on federated machine learning"* **IEEE Internet of Things Journal**: DOI: doi.org/10.1109/JIOT.2021.3128646..

[21] Y. Wang, P. Mianjy, and R. Arora. "Robust learning for data poisoning attacks". In: *International Conference on Machine Learning*. PMLR. 2021, 10859–10869.

[22] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu. "Data poisoning attacks against federated learning systems". In: *European Symposium on Research in Computer Security*. Springer. 2020, 480–501. DOI: doi.org/10.1007/978-3-030-58951-6_24.

[23] M. Fang, X. Cao, J. Jia, and N. Gong. "Local model poisoning attacks to {Byzantine-Robust} federated learning". In: *29th USENIX Security Symposium (USENIX Security 20)*. 2020, 1605–1622.

[24] C. Fung, C. J. Yoon, and I. Beschastnikh, (2018) *"Mitigating sybils in federated learning poisoning"* **arXiv preprint arXiv:1808.04866**: DOI: https://doi.org/10.48550/arXiv.1808.04866.

[25] A. N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo. "Model poisoning attacks in federated learning". In: *Proc. Workshop Secur. Mach. Learn.(SecML) 32nd Conf. Neural Inf. Process. Syst.(NeurIPS)*. 2018, 1–23.

[26] Z. Chen, P. Tian, W. Liao, and W. Yu, (2021) *"Towards multi-party targeted model poisoning attacks against federated learning systems"* **High-Confidence Computing *1***(1): 100002. DOI: doi.org/10.1016/j.hcc.2021.100002..

[27] C. Zhou, Y. Sun, D. Wang, and Q. Gao, (2022) *"Fed-Fi: Federated Learning Malicious Model Detection Method Based on Feature Importance"* **Security and Communication Networks *2022***: DOI: doi.org/10.1155/2022/7268347.

[28] D. Cao, S. Chang, Z. Lin, G. Liu, and D. Sun. "Understanding distributed poisoning attack in federated learning". In: *2019 IEEE 25th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE. 2019, 233–239.

[29] H. Hosseini, H. Park, S. Yun, C. Louizos, J. Soriaga, and M. Welling. "Federated Learning of User Verification Models Without Sharing Embeddings". In: *International Conference on Machine Learning*. PMLR. 2021, 4328–4336.

[30] X. Li, Z. Qu, S. Zhao, B. Tang, Z. Lu, and Y. Liu, (2021) *"Lomar: A local defense against poisoning attack on federated learning"* **IEEE Transactions on Dependable and Secure Computing**: DOI: doi.org/10.1109/TDSC.2021.3135422.

[31] A. Panda, S. Mahloujifar, A. N. Bhagoji, S. Chakraborty, and P. Mittal. "SparseFed: Mitigating Model Poisoning Attacks in Federated Learning with Sparsification". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, 7587–7624.

[32] X. Cao and N. Z. Gong. "MPAF: Model Poisoning Attacks to Federated Learning based on Fake Clients". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 3396–3404. DOI: doi.org/10.48550/arXiv.2203.08669..

[33] A. Shamis, P. Pietzuch, M. Castro, E. Ashton, A. Chamayou, S. Clebsch, A. Delignat-Lavaud, C. Fournet, M. Kerner, J. Maffre, et al., (2021) *"PAC: Practical accountability for CCF"* **arXiv preprint arXiv:2105.13116**: DOI: doi.org/10.48550/arXiv.2105.13116.

[34] A. Moghimi, G. Irazoqui, and T. Eisenbarth. "Cachezoom: How SGX amplifies the power of cache attacks". In: *International Conference on Cryptographic Hardware and Embedded Systems*. Springer. 2017, 69–90.

[35] H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, H. Möllering, T. D. Nguyen, P. Rieger, A.-R. Sadeghi, T. Schneider, H. Yalame, et al. "SAFELearn: secure aggregation for private federated learning". In: *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE. 2021, 56–62. DOI: doi.org/10.1109/SPW53761.2021.00017..

[36] W. Huang, T. Li, D. Wang, S. Du, J. Zhang, and T. Huang, (2022) *"Fairness and accuracy in horizontal federated learning"* **Information Sciences *589***: 170–185. DOI: doi.org/10.1016/j.ins.2021.12.102.

[37] 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*. PMLR. 2017, 1273–1282.

[38] S. M. S. A. Abdullah, S. Y. A. Ameen, M. A. Sadeeq, and S. Zeebaree, (2021) *"Multimodal emotion recognition using deep learning"* **Journal of Applied Science and Technology Trends 2**(02): 52–58. DOI: doi.org/10.38094/jastt20291.