electronics

MDPI

Article

Enhancing the Resilience of a Federated Learning Global Model Using Client Model Benchmark Validation

Algimantas Venčkauskas, Jevgenijus Toldinas *, Nerijus Morkevičius, Ernestas Serkovas and Modestas Krištaponis

Department of Computer Science, Kaunas University of Technology, 44249 Kaunas, Lithuania; algimantas.venckauskas@ktu.lt (A.V.); nerijus.morkevicius@ktu.lt (N.M.); ernestas.serkovas@ktu.lt (E.S.); modestas.kristaponis@ktu.edu (M.K.)

* Correspondence: eugenijus.toldinas@ktu.lt

Abstract: Federated learning (FL) makes it possible for users to share trained models with one another, thereby removing the necessity of publicly centralizing training data. One of the best and most cost-effective ways to connect users is through email. To steal sensitive information, spam emails might trick users into visiting malicious websites or performing other fraudulent actions. The developed semantic parser creates email metadata datasets from multiple email corpuses and populates the email domain ontology to facilitate the privacy of the information contained in email messages. There is a new idea to make FL global models more resistant to Byzantine attacks. It involves accepting updates only from strong participants whose local model shows higher validation scores using benchmark datasets. The proposed approach integrates FL, the email domain-specific ontology, the semantic parser, and a collection of benchmark datasets from heterogeneous email corpuses. By giving meaning to the metadata of an email message, the email's domain-specific ontology made it possible to create datasets for email benchmark corpuses and participant updates in a unified format with the same features. In order to avoid fraudulently modified client updates from being applied to the global model, the experimental results approved the proposed approach to strengthen the resiliency of an FL global model by utilizing client model benchmark validation.

Keywords: federated learning; cyber threat intelligence; email; domain ontology

check for updates

Academic Editors: Meng Han, Tong Qiao, Zhuojun Duan, Liyuan Liu, Seyedamin Pouriyeh and Subir Halder

Received: 28 January 2025
Revised: 10 March 2025
Accepted: 18 March 2025
Published: 19 March 2025

Citation: Venčkauskas, A.; Toldinas, J.; Morkevičius, N.; Serkovas, E.; Krištaponis, M. Enhancing the Resilience of a Federated Learning Global Model Using Client Model Benchmark Validation. *Electronics* **2025**, *14*, 1215. https://doi.org/10.3390/electronics14061215

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Federated learning (FL), a type of distributed machine learning (ML), protects user privacy while letting models be trained on client devices. It also brings the benefits of artificial intelligence (AI) to places with sensitive and different kinds of data [1]. This approach emerged primarily for two reasons: first, the lack of adequate data to be stored centrally on the server-side, as opposed to conventional machine learning, because of restrictions on direct data access; and second the need for data privacy by utilizing local data from client devices instead of transferring sensitive information to the server [2]. Many domains are unable to fully utilize its benefits due to primary obstacles: (i) concerns about user data privacy and confidentiality, as well as governing rules; (ii) insufficient data or high computing costs can hinder ML model development and implementation.

Federated learning that is considered trustworthy must be [3] legally compliant and private for handling user data (Privacy); confidential and accurate for implementing systems (Security); reliable and strong for training the model (Robustness); fair in how it handles clients or model inputs (Fairness); and understandable for making decisions

(Explainability). Each essential component of trustworthy federated learning (TFL) was described as follows in [3]:

- Privacy and security refer to how private data in FL are protected from leakage. It is vital to protect the privacy of FL data and model parameters since they are regarded as sensitive information belonging to their owners. This includes preventing the unlawful use of data that can identify individuals or households. The data includes names, ages, genders, face photos, and fingerprints. Commitment to privacy protection is a critical aspect in establishing the reliability of an FL system. This is due to regulatory and legal restrictions, as well as the value of the data.
- Robustness and Resilience refers to the ability of FL to retain stability under harsh conditions, particularly those caused by attackers. This is critical since the real-world situations in which FL systems operate are often complicated and unpredictable. Robustness is an important aspect that determines the performance of FL systems in real applications. A lack of robustness may result in unanticipated or detrimental system behaviors, weakening trustworthiness.
- Fairness and Trust is the term for a set of guiding ideals and moral obligations that demand that FL systems be free from bias, partiality, or discrimination against individuals or groups. This is accomplished by virtue of three different aspects of fairness: attribute fairness, which guarantees similar predictions for similar individuals or keeps the model unaffected by sensitive characteristics like gender or age; performance fairness, which guarantees a uniform distribution of accuracy across clients; and contribution fairness, which rewards clients based on their contributions to the system. These guidelines are essential for advancing reliable and moral FL models that successfully fulfill their stated goals.
- Explainability is the degree to which people can understand and provide an explanation for the choices made or the results obtained by an FL system is known as. This idea basically consists of two parts: post hoc explainability, which is creating external mechanisms to clarify the choices the FL system makes to determine whether the system's results can be explained, and ante hoc explainability, which concentrates on the transparency and understandability built into the FL system design to determine whether each process within the system is explainable. Explainability is essential for an FL system since it builds user trust. Additionally, it encourages more fruitful communication between domain specialists and FL systems, especially in industries like banking and healthcare. This increased comprehension enables the system to produce more reliable and compliant decisions.

While email is a great way to stay in touch with people at little-to-no expense, there is a risk that communications could compromise email and the Internet. Not only is spam email a pain but it also contains phishing links that could steal sensitive information from users by tricking them into clicking on malicious links or engaging in other fraudulent activities. Also, the user might read the entire spam message before identifying it as spam and removing it. There has been investigation and discussion into the problems and unanswered questions concerning the categorization of emails. This article investigates and discusses problems and outstanding difficulties with email classification.

The important contributions of this research are listed below.

- The novelty of the proposed approach to improving the resilience of a global federated learning model is based on the evaluation of the client model using benchmark validation to detect malicious or trustless clients. An update of the client model that exhibits an appropriate benchmark validation score may be used to update the global model. This will protect the global model against a Byzantine attack.

- By giving meaning to an email's metadata, the email domain-specific ontology made it possible to make datasets for email benchmark corpuses and participant updates that were all in the same format and had all the same features.

The remainder of this article is organized as follows. Section 2 discusses related work. Section 3 presents and explains the proposed approach. Section 4 presents and discusses experimental settings and results. Finally, Section 5 presents the discussion, and Section 6 presents the conclusion.

2. Related Work

In this section, an overview of the related work is provided. The first subsection of this section discusses the challenges of preserving the privacy and security of participant data. In the second subsection, it points out the resilience against drop-out clients, attackers that manipulate prediction models by modifying the labeled data's related weights. It then emphasizes approaches to ensure the trained model's resilience against Byzantine attacks and user privacy protection. The third subsection of this section discusses the challenges to ensure fairness and trust, choosing the most appropriate clients to participate in a training process, and global model updates. Finally, the fourth subsection of this section presents a summary of related work.

2.1. Privacy and Security

Confined Gradient Descent (CGD), which was proposed in [4], enables each participant to learn his own private global model and improves federated learning privacy by preventing the exchange of global model parameters. Throughout the training process, the CGD participants rigorously constrain the global models they create locally to themselves. During the training phase, they operate the modifications to their models in a secure collaborative manner. In this way, CGD minimizes information leakage while maintaining the benefits of federation.

To stop privacy leaks, ref. [5] came up with and built a new privacy-preserving federated learning (PPFL) architecture that protects data with Additive Homomorphic Encryption (AHE). The proposed PPFL model uses the Paillier cryptosystem for additive homomorphic encryption to protect encrypted gradients from model poisoning assaults while maintaining user, server, and gradient privacy. The proposed model improves the detection and prevention of poisoning attempts of the encrypted model by using an internal auditor mechanism. The proposed solution to gradient aggregation in PPFL uses the Gaussian mixture model (GMM) and the Mahala Nobis distance (MD), allowing for Byzantine tolerance. This advanced technique effectively manages heterogeneous data and mitigates the risks of targeted and untargeted attacks, ensuring resilience even in settings with hostile users. An efficiency score for the suggested PPFL model is formulated that considers performance and privacy parameters such as computational efficiency, accuracy, and privacy. This provides a clear trade-off for introducing PP approaches into FL, emphasizing the importance of balancing privacy and performance in PPFL.

The paper [6] offers a thorough and motivating overview of privacy-preserving computing protocols, explaining each one's operation, and providing a methodical framework for debating those that already exist. This assessment evaluates the current literature and group privacy-preserving computation methods according to their intended use. These protocols include those based on zero-knowledge proof, homomorphic encryption, differential privacy, trusted execution environments, and secure multi-party computing. The results of the experiments showed that the Swarm Learning (SL) framework performed comparably in terms of accuracy and resilience.

In order to assess how well privacy measurement in FL protects sensitive data privacy while AI and ML models are being trained, the study in [7] surveys the field. In FL, the term "privacy computation" refers to the quantitative techniques used to quantify and guarantee system privacy, with the goal of maintaining the privacy of individual data even in aggregated models. We can comprehend the relationship between privacy and other elements, such as accuracy, communication costs, and computational complexity, by looking at these privacy metrics.

The Social-Aware Clustered Federated Learning (SCFL) system [8] allows trustworthy individuals to build social clusters and aggregate model updates (e.g., gradients) before uploading to the cloud for global aggregation. Mixing model updates in a social group allows opponents to only eavesdrop on the combined outcomes of the social layer, not individual privacy. The authors developed a suite of algorithms, including fair revenue allocation, customizable privacy preservation, and iterative two-sided matching, which converges to Nash-stable equilibrium, to solve the optimal social cluster formation problem among people with social ties as a federation game with transferable utility (FTU).

2.2. Robustness and Resiliency

Resilience against drop-out clients failing to submit model updates for aggregation is a practical necessity for constructing secure FL systems, as certain clients may experience problems such as bad network connections, temporary unavailability, or energy limits [9]. FL services with lightweight, secure and robust aggregation are made possible by a new system design that was proposed in [10]. In contrast to previous research, the proposed system can manage client dropouts while protecting the confidentiality of their secret keys, allowing them to continue participating directly in subsequent rounds. To improve the communication efficiency of the proposed system, the authors investigated and created a novel integration between secure aggregation and quantization-based model compression. Furthermore, useful techniques are suggested to strengthen the security of the constructed system against an actively hostile server.

For FL to function properly, clients must only transmit local model updates (model weights) for global model aggregation. This ensures that no actual data are shared, and the typical lack of direct access to the data gives attackers the chance to manipulate prediction models by modifying the labeled data's related weights, including misclassifying or tampering with the assigned classifications. These attacks aim to compromise the robustness and resiliency of global models. The authors examine the impact of random noise effects and scaling as two methods of data poisoning [11].

Federated learning is vulnerable to system heterogeneity in which clients possess varying capacities for computation and communication. Additionally, the computation speed of clients adversely impacts the scalability of federated learning algorithms, leading to notable delays in their runtime as a result of stragglers or slow devices. A unique federated learning meta-algorithm resistant to stragglers was presented in [12], which uses the statistical properties of the data of the clients to adaptively choose the clients and accelerate the learning process. The main concept of the suggested algorithm is to begin the training process with faster nodes, and then, once the statistical accuracy of the data from the current participating nodes is reached, gradually include the slower nodes in the model training process. The final model for each stage is then used as a warm-start model for the subsequent stage.

The goal of the study [13] was to anticipate the likely outage and resource condition of critical infrastructure agents (CIAs) while maintaining the learning process even in situations when the majority of the agents are unable to complete a particular computational task without exchanging any local data. The FL algorithm was created especially for a

critical infrastructure environment with limited resources. It can help with distributed agent training, select efficient clients by examining their resources, and allow resource-constrained agents to handle a portion of the computation tasks. The authors tested the efficacy of FedAvg and their proposed FedResilience algorithm with prediction tasks for a likely outage, taking into account the varying numbers of agents with different stragglers. They also verified the agents' resource-sharing scope.

Because FL has a number of problems, including sluggish convergence, communication overhead, and adversarial attack susceptibility, especially in Industry 5 scenarios where it is integrated with conventional manufacturing processes and the Internet of Things, a novel FL model based on swarm optimization to improve convergence in a heterogeneous environment in real time was proposed in [14]. Furthermore, in order to evaluate the robustness of the suggested approach in relation to other FL techniques, extensive experiments were conducted on various data poisoning attacks with various scenarios, and a grey wolf optimization-based federated learning scheme to defend against adversarial attacks was developed.

Because FL random masks are used to protect the local updates, the global server is unable to see their actual values. The ability of hostile (Byzantine) users to affect the global model by altering their local updates or datasets poses a significant challenge to the global model resiliency. The first single server solution for Byzantine-resilient secure federated learning and framework was proposed in [15]. This was carried out to protect user privacy and make sure that the trained model would be able to handle Byzantine faults. This solution is based on a verifiable secure outlier detection strategy. In [16], the authors examined the stochastic convex and non-convex optimization issues for FL at the edge. They demonstrate how to manage heavily tailed data while maintaining maximal statistical error rates, communication efficiency, and Byzantine resilience all at the same time. First, a Byzantine-resilient distributed gradient descent method was described, which can handle heavy-tailed data, while still convergent under conventional assumptions. Another approach was suggested that uses gradient compression techniques to save communication expenses throughout the learning phase to reduce communication overhead.

2.3. Fairness and Trust

The process of choosing the most appropriate clients to participate in a training process and global model updates is one of FL's challenges. Proposed in [17], a trust-based client selection method for FL employs deep reinforcement learning to allow the system to choose the most suitable customers in terms of training time and resource utilization. In addition to the client selection method, a transfer learning mechanism was incorporated to alleviate data shortages in certain areas and compensate for probable learning deficiencies in specific servers. The suggested method uses deep Q-learning to identify IoT devices that maximize trust while achieving the best possible combinations of resource availability.

FL is a concept that aims to enable cloud computing-based IoT devices to train machine learning models cooperatively without transferring data from each device to a single central server. Big volumes of data can be preprocessed at edge nodes using edge computing before being transferred to cloud central servers. This gives edge nodes the computational capacity to maximize cloud resources. The likelihood of coming across untrusted or poorly performing IoT devices is very high due to the vast quantity of deployed IoT devices in edge-based systems. In the event of malicious or compromised IoT devices, these devices would not only result in lengthy processes and execution times but would also increase the number of resources squandered. The advantage of the proposed double deep Q network trust is that it reduces the overestimation of Q values, which facilitates faster and more stable learning [18].

In FL, fairness is essential to address privacy issues, protect model integrity, protect the aggregation process, promote participant collaboration, allow accountability and auditing, and foster user trust. FederatedTrust, an algorithm that measures the trustworthiness of FL models based on the pillars, notions, and metrics provided in the proposed taxonomy, was designed and implemented after a novel taxonomy containing the most pertinent pillars, notions, and metrics was created [19]. FederatedTrust aggregates metrics and pillars dynamically and flexibly based on the validation scenario to calculate global and partial trustworthiness scores.

A brand-new FL technique is presented in which trust is bootstrapped by the service provider in [20]. Specifically, in order to bootstrap trust, the service provider gathers a clean, small training dataset (referred to as the "root dataset") for the learning task and keeps a model (referred to as the "server model") based on it. Every local model update from clients is given a trust score by the service provider at the beginning of each iteration. A local model update is given a lower trust score if its direction differs more from the direction of the server model update. Next, in the vector space, the service provider normalizes the local model updates' magnitudes so that they fall within the same hyper-sphere as the server model update. Ultimately, to update the global model, the service provider calculates the average of the normalized local model updates, weighted by their trust scores.

2.4. Summary of Related Work

The list of state-of-the-art research papers on federated learning is presented in Table 1.

Table 1. List of state-of-the-art research papers on federated learning.

Reference	Privacy Security	Robustness Resilience	Fairness Trust	Advantages	Disadvantages
Yazdinejad et al. [5]	Additive Homomorphic Encryption (AHE)	Byzantine-tolerant gradient aggregation	An efficiency score	An efficiency score that combines privacy, accuracy, and computational efficiency	Computational and communication overhead due to the complexity of the proposed PPFL architecture
Wang et al. [8]	Customizable privacy protection		Hierarchical SCFL framework	Low-cost, feasible, and customized FL services by using the social attributes of the users'	Participants in FL must form social connections
Zheng et al. [10]	Integration between secure aggregation and quantization-based model compression	FL services with lightweight, secure, and resilient aggregation	Minimally trusted hardware	The suggested approach may manage client dropouts while maintaining the confidentiality of their secret keys, preventing them from directly participating in subsequent rounds	The proposed system only assumes minimally trusted hardware with integrity guarantees
Imteaj et al. [13]	On-device learning without sharing any data	Improve the resilience of critical infrastructures through early prediction		The proposed FL-based strategy consists of a local CIA that acts as an intelligent decision-making agent. The FedResilience algorithm provides prediction tasks for potential outages	Privacy was not guaranteed by the partial work that was made possible to gather from agents with limited resources.
So et al. [15]	Use of stochastic quantization	Robust gradient descent approach		A single-server Byzantine-resilient secure aggregation framework is suggested for secure FL	A single server for model update aggregation is not faulty tolerant
Rjoub et al. [17]	Privacy-preserving machine learning		Deep reinforcement learning	A transfer learning mechanism was incorporated to compensate for possible learning deficiencies in some servers and address data shortages in certain regions	The relatively high computation time Cyberattacks could be launched against many parts of the proposed solution.

Table 1. *Cont.*

Reference	Privacy Security	Robustness Resilience	Fairness Trust	Advantages	Disadvantages
Sánchez et al. [19]	Perturbation	Poisoning and Inference	Federation	A novel taxonomy created with the main building blocks—privacy, robustness, fairness, explainability, accountability, and federation FederatedTrust computes global and partial trustworthiness scores by aggregating metrics and pillars	The real-world deployment of FederatedTrust may present challenges such as resource consumption, data leakage, governance, compliance, and scalability
Cao et al. [20]		Byzantine-robust FL method called FLTrust		The service provider manually logs in and collects a small clean training dataset (called the root dataset) for the learning task The defense is performed on the server side	The root dataset must be clean from poisoning The proposed method is effective once the root dataset distribution does not deviate too much from the overall training data distribution to ensure integrity

From the summary presented in Table 1, we can make the following assumptions:

- Statistical heterogeneity difficulty arises from the nonidentically distributed nature of data, as each client has a distinct sample of data that often reflects distinct qualities, patterns, or statistical characteristics;
- The robust and resiliency approach proposed by the other authors is to use Byzantine-robust aggregation rules, which essentially compare the clients' local model updates and remove statistical outliers before applying them to the global model;
- One of the biggest shortcomings of robustness and resiliency is that an attacker can create an adaptable attack by changing the aggregation rule;
- The approaches proposed by other authors [5,10,13,19] tried to use gradient aggregation, lightweight, secure and resilient aggregation, early prediction, poisoning, and inference;
- As stated in [20], for the learning task, the service provider manually gathers a small clean training dataset known as the root dataset. Like how a client maintains a local model, the server maintains a model (referred to as the server model) for the root dataset. The server changes the global model in each iteration considering both the local model updates from the clients and its own server model update.

The research that led to the proposed approach was based on these assumptions. The process involves using benchmark validation to assess the client model, detect malicious or untrustworthy clients, and reject their updates. As a result of this, the global FL model becomes more resilient.

3. Proposed Approach

In this section, we introduce the newly developed approach to enhance statistical heterogeneity, privacy, and resilience challenges. The complexity of federated learning research, which addresses issues such as data privacy, model security, resiliency, client heterogeneity, and effective communication, was emphasized in [21].

3.1. Resolving Statistical Heterogeneity and Privacy Challenges

Since each client has a different subset of data, which frequently reflects unique traits, patterns, or statistical characteristics, the statistical heterogeneity challenge results from the nonidentically distributed nature of data. The task of combining data from several sources to update a global model is made more difficult by this variability. As

a result, each client's training period varies greatly, and the server must wait for the slowest executor to complete its work, which results in a lengthy training period. This is a classic straggler problem [22]. Because statistical heterogeneity affects the performance and generalizability of the global model, it must be addressed. This calls for specialized methods that take these differences into account and reduce them without sacrificing data privacy or communication effectiveness.

We use the email domain-specific ontology we came up with in previous research [23]. This ontology omits any potentially sensitive information from email messages and focuses on their metadata, which is made up of technical details that show how an email travels from sender to recipient. We used the Chi-square feature ranking algorithm, ANOVA (analysis of variance), and Kruskal–Wallis in the first ablation investigation that we published [24]. According to this research, the Chi-square feature ranking method, when combined with 22 predictors, had the best test accuracy for spam recognition. This is why, instead of using all properties of the email domain-specific ontology, we restricted ourselves to just 22 features (listed in Table 2) to prepare the dataset for ML.

Table 2. List of the header fields of email messages to populate the email ontology with data.

Feature	Description	Example
SENT_TS	Sent timestamp assigned by the originator server	1030019517000
CONT_SUBTYPE	Subtype part of the *Content-Type* header field	html
CONT_PARAM	Parameter part of the *Content-Type* header field	charset = "iso-8859-1"
FROM_P	*Display-name*part of the *From* email address	CNET News.com Daily Dispatch
FROM_U	Local part of the *From* email address	Online#3.20777.51-8J4zgE1Uu7_vxsRR.1
FROM_D	Domain part of the *From* email address	newsletter.online.com
TO_U	Local-part part of the first *To* email address	update
TO_D	Domain part of the first *To* email address	list.theregister.co.uk
REPLY_P	Display-name part of the *Reply-to* email address	Daily Dilbert
REPLY_U	Local-part part of the *Reply-to* email address	2.21122.29-GYdCgEWAHESJ.1
REPLY_D	Domain part of the *Reply-to* email address	ummail4.unitedmedia.com
SENDER_U	Local-part part of the *Sender* email address	shifty
SENDER_D	Domain part of the *Sender* email address	spit.gen.nz
OMTA_FROM_H	Host name extracted from the *From-domain* part of the Received field at the originator SMTP server	r-smtp
OMTA_BY_D	Host domain extracted from the *By-domain* part of the Received field at the originator SMTP server	siteprotect.com
OMTA_TS	Timestamp at the SMTP server of the originator	1030149730000
DMTA_NR	Delivery SMTP server's hop number	4
DMTA_FROM_H	Host name extracted from the *From-domain* part of the *Received* field at the delivery SMTP's server	sunu422
DMTA_BY_D	Host domain extracted from the *By-domain* part of the *Received* field at the originator SMTP server	bph.ruhr-uni-bochum.de
DMTA_TS	Timestamp at the delivery SMTP server	1030018641000
DMTA_DELAY	Delay of the message (in milliseconds) at the originator SMTP server	7000
URL1_HOST	Host of the first URL in the message's body	www.ktu.lt

Additionally, there are two features that are not mentioned in Table 2, namely the number of rows in the dataset and the label used for supervised ML.

The simplest and most effective method for semantically interpreting a given email message metadata is to utilize the semantic parser, which is software that uses semantic annotations to transform an email message metadata into an ontological representation.

The proposed approach is to resolve the challenges of statistical heterogeneity and privacy depicted in Figure 1.

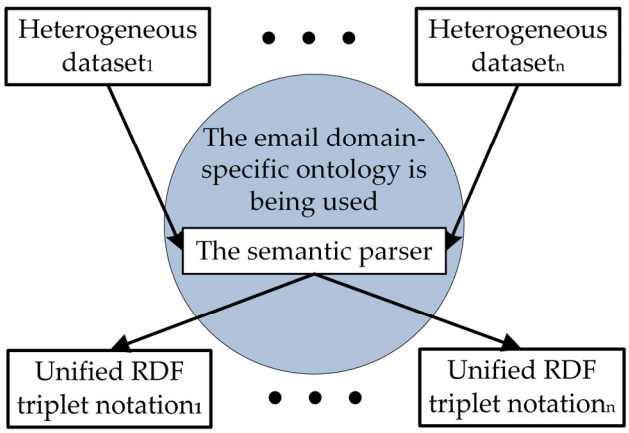


Figure 1. The proposed approach aims to resolve the challenges of statistical heterogeneity and privacy.

RDF triplet notation in a subject–predicate–object format is used to represent all annotations, metadata, and other significant data. A group of predicates that indicate the relationships between subjects and objects makes up each message.

3.2. Resolving Robustness and Resiliency Challenges

In addition to training global models, it is essential to create FL systems that offer privacy assurances and are resilient against various types of attackers [25]. The quantity and diversity of information generated at the organization's edge will continue to expand at previously unheard-of rates as the number of advanced computer devices and applications rises. It is very difficult to defend FL against several robust attacks [26]. First, only local gradients are available on the server side, which is where the defense can only be implemented. This makes numerous backdoor defense techniques created in the centralized machine learning system invalid. Second, the security strategy must be resistant to attacks that use both model and data poisoning. Most current robust defenses use gradient aggregation techniques, which were primarily created to protect against untargeted Byzantine assaults. Geometric median-based robust federated aggregation (RFA) [27] and robust learning rate [28] are two dedicated protection approaches that have been studied against both data poisoning and model poisoning threats.

The proposed method was intended to be used in federated environments where the federation participants are small organizations or enterprises that have their own server infrastructures and act as enterprise-level email providers (see Figure 2). If the organization can host and support local email servers, it should likely allocate the necessary computational resources for local model training to mitigate spam messages and enhance the overall quality of the email service for its users.

As shown in Figure 2, the suggested approach to improve the resilience of the global federated learning model requires that FL clients create a unified client dataset using the semantic parser that takes advantage of the email domain-specific ontology. The client model creation process starts with a heterogeneous client dataset. Clients use the semantic parser to generate a unified client dataset to create local models. A semantic representation of the email message's metadata is used by the semantic parser to protect the privacy of data in emails to generate a dataset and populate the email domain-specific ontology. Clients train local models and upload local ML models to update the global model. For

each client that participates in FL, an ML model is trained and then uploaded to the cloud, as depicted in Figure 3.

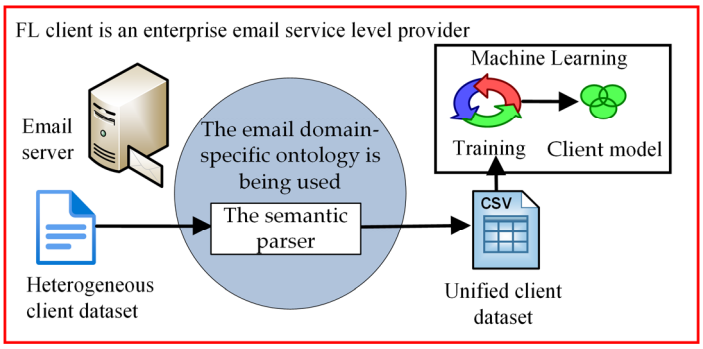


Figure 2. The proposed approach is to create a client ML model.

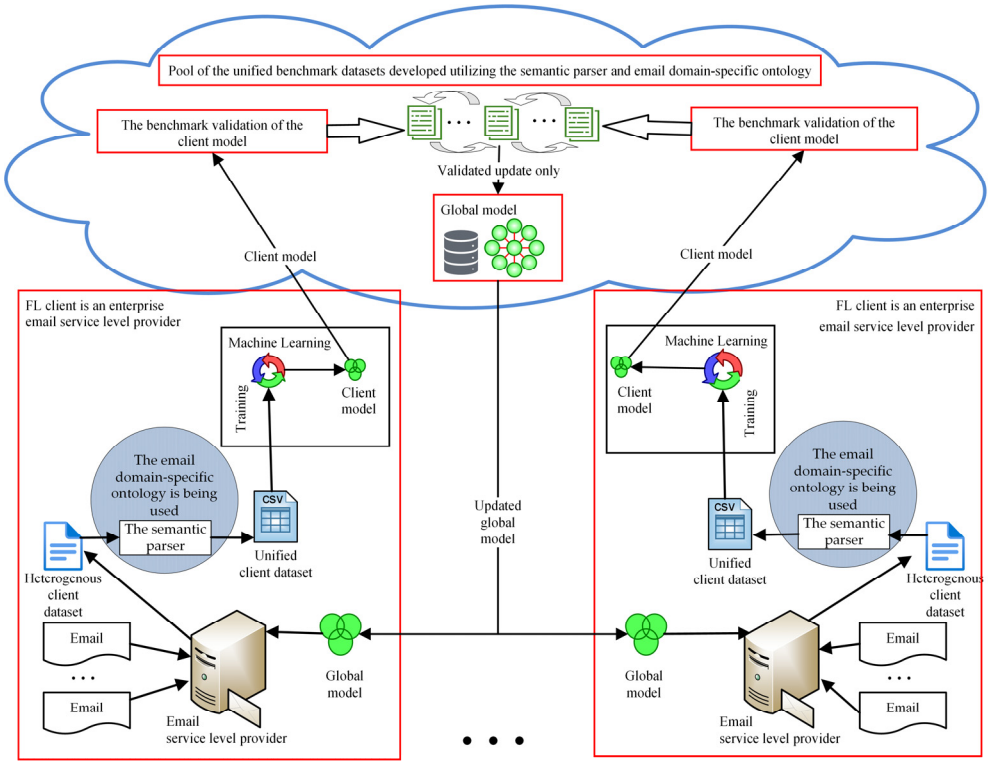


Figure 3. The proposed approach to improving the resilience of a global FL model.

To address the challenges of robustness and resilience, we propose using a pool of benchmark datasets in the cloud to apply the validation process to client models and select validated updates only that exhibit an appropriate benchmark validation score. When the cloud server receives a local model update from the client (refer to Figure 3), it uses a pool of the unified benchmark datasets developed utilizing the semantic parser and the email domain-specific ontology and initiates the validation process. The validation process gathers performance metrics from all benchmark datasets.

The main characteristics traditionally used to evaluate machine learning methods are *accuracy*, *precision*, *recall*, and the *F1 score*. The accuracy is calculated using the following equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (1)$$

where *TP* is a number of true positive predictions, *TN* is a number of true negative predictions, *FP* is a number of false positive predictions, and *FN* is a number of false negative predictions.

The *precision* is the fraction of instances marked as positive that are genuinely positive.

$$Precision = \frac{TP}{TP + FP}, \tag{2}$$

The fraction of true positive predictions marked as positive, or the percentage of true positives predicted as positive, is known as *recall*.

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

The *F1 score* is the harmonic mean of precision and recall, which equally assesses the importance of Type I and Type II errors.

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN}. \tag{4}$$

If the dataset is balanced, the accuracy gives the most meaningful results, but if the dataset is imbalanced, the *F1 score* is more meaningful. Although the *F1 score* works well with unbalanced datasets, it is not suitable for FL as a generic evaluation metric. That is because it does not take actual negative predictions into account; therefore, it cannot identify true negative rates [29].

The Matthews correlation coefficient (*MCC*) provides an effective approach for addressing dataset imbalance. For the purpose of measuring the performance of the classifier in both balanced and imbalanced scenarios, it was demonstrated that *MCC* is more robust and dependable [30] than the performance metrics that were previously mentioned. The *MCC* is the sole binary classification metric that yields a high score exclusively when the binary predictor accurately identifies the majority of both positive and negative data instances. This coefficient can take values on the interval $[-1, +1]$, where -1 and $+1$ represent perfect misclassification and perfect classification, respectively.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}. \tag{5}$$

Concordance Probability (*CP*) is an effective strategy for determining the optimal threshold point [31]. The ideal cut-off value is determined by multiplying the specificity and sensitivity to yield the highest number. The specificity is calculated using the following equation:

$$Specificity = 1 - \frac{FP}{FP + TN}. \tag{6}$$

The sensitivity is calculated using the following equation:

$$Sensitivity = \frac{TP}{TP + FN}. \tag{7}$$

The *CP* is calculated using the following equation:

$$CP = Specifity \times Sensitivity \tag{8}$$

For efficient client model validation, some integral model performance metrics are required. We suggest a two-part evaluation that finds the average *MCC* and *F1 score* achieved during client model validation using benchmark datasets to judge the quality of

the FL participant's updates. Subsequently, compare the average *MCC* and *F1 score* to the associated threshold *MCC* and *F1 score* metrics. In certain instances, updating the global model is considered suitable and robust when the score surpasses the threshold value. The client model update is discarded if the *F1 score* falls below the threshold, which is calculated using Equation (8). As only validated client model updates are permitted and utilized to update the global model, client model benchmark validation ensures the resilience of the global model. The global model is accessible to FL clients after a successful upgrade.

4. Experimental Settings and Results

In this section, we provide the experimental results obtained when evaluating the proposed approach.

4.1. Dataset Selection

Upon analyzing publicly accessible email benchmark datasets, we selected four of them, which contain not only texts of messages, but also full email metadata. The concise definition of the chosen datasets is presented in the subsequent subsections.

4.1.1. SpamAssasin

This dataset [32] comprises email metadata obtained from the "SpamAssassin public mail corpus". All message headers are fully recreated. The address obfuscation was executed, and in certain instances, the hostnames were substituted with "spamassassin.taint.org" (which possesses a legitimate MX record). The corpus comprises 6047 messages, with an approximate spam ratio of 31%. The messages comprise: 500 plus 1397 "SPAM" messages sourced from non-spam-trap origins; 2500 plus 1400 messages categorized as "EASY HAM", which are non-spam messages easily distinguishable from spam; and 250 non-spam messages classified as "HARD HAM", which closely resemble typical spam in various aspects.

4.1.2. CDMC2010

This corpus includes one of the datasets suggested for the 2010 ICONIP Cybersecurity Data Mining Competition [33]. This dataset comprises a collection of email messages designated to test spam filtering systems. All message headers are retained in their entirety. Address obfuscation was executed, with hostnames in certain instances substituted with "csmining.org" (which possesses a legitimate MX record), among other modifications. For evaluation reasons, the designated section referred to as "TRAINING" was utilized. The original dataset included 4327 messages, with 2949 classified as "HAM" and 1378 as "SPAM". All 2948 "HAM" messages were parsed without errors during the analysis using the proposed ontology. Only 1290 messages from the "SPAM" category were properly parsed; the remaining messages exhibited discrepancies with standards and were rejected. For example, the local address contains an illegal character, etc.

4.1.3. ENRON-SPAM

This corpus [34] is derived from two distinct sources. The spam component was gathered from four distinct sources using traps (e.g., email addresses displayed on the Web in a manner that is evident to humans but not to crawlers, indicating they should not be utilized). The ham segment of the corpus was derived from email communications of six Enron users extracted from the Enron dataset [35]. The original communications were altered to conceal personal information. The Enron corpus lacks certain "low-level" SMTP protocol metadata that is typically included in standard email communications, such as information provided by the Mail Transfer Agent (MTA). The initial dataset comprises 19,088 "HAM" messages and 32,988 "SPAM" mails. Throughout the semantic parsing

procedure, 17,121 "HAM" messages and 31,369 "SPAM" messages were processed without errors.

4.1.4. TREC07p

Trec07p [36] is the publicly available corpus supplied to participants of the TREC 2007 Spam Track event. This dataset comprises 75,419 email messages, of which 25,220 are classified as "HAM" and 50,199 as "SPAM". These messages represent the entirety of communications sent to a certain server from 8 April 2007 to 6 July 2007. Throughout the semantic parsing procedure, 25,217 "HAM" messages and 50,021 "SPAM" messages were processed without any mistakes.

4.1.5. Assessment of Dataset and Preliminary Global Model Development

Figure 4 shows the process of adopting the email domain-specific ontology to create an initial global model.

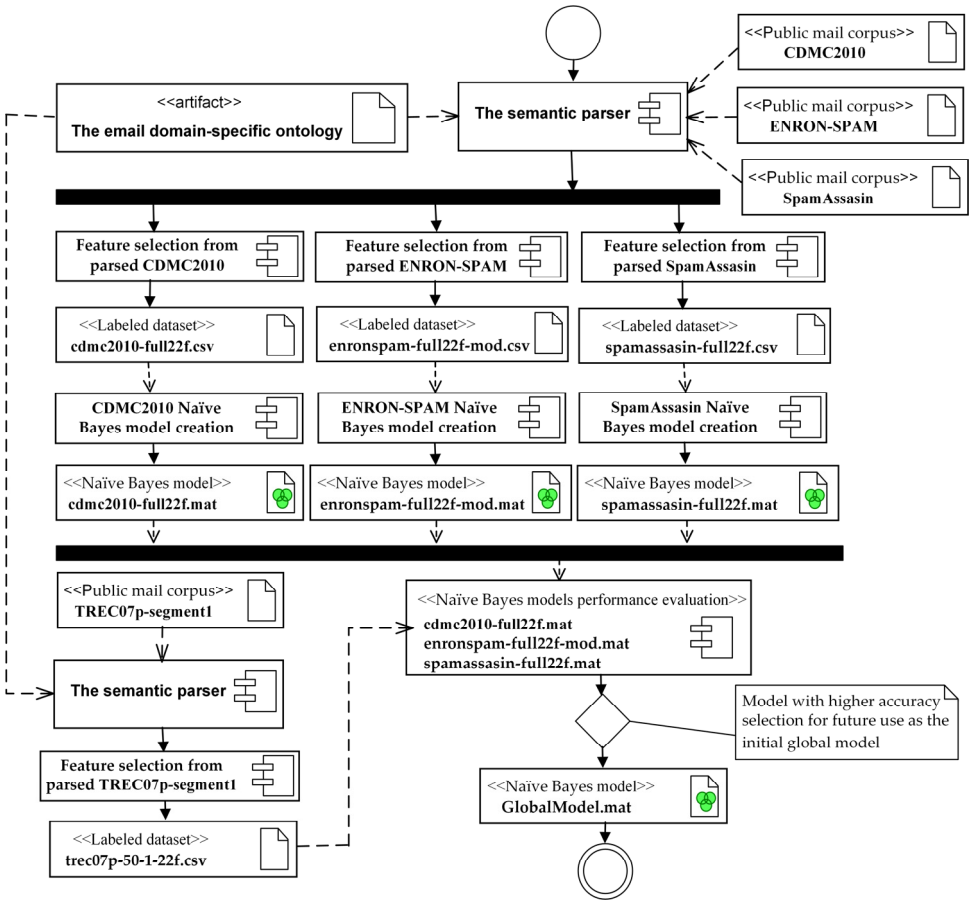


Figure 4. The process of adopting the email domain-specific ontology to create an initial global model.

The semantic parser uses semantic annotations to convert an email message's metadata into the email domain-specific ontological representation. The semantic parser generates the triplets according to the email domain-specific ontology schema. In the second stage, we select the predictor features of the triplet and create a labeled dataset, saving it in a comma-separated file. A labeled dataset is created for every selected public email corpus: CDMC2010, ENRON-SPAM, SpamAssassin, and TREC07p-segment1. Next, Naïve Bayes models called CDMC2010, ENRON-SPAM, and SpamAssasin were made and tested on the

TREC07p-segment1 labeled dataset. The model that did the best was chosen as the initial global model.

The number of messages selected from the public mail corpuses for our experiments and included in the labeled datasets that were created using the semantic parser and the email domain-specific ontology is provided in Table 3. Email metadata features were parsed and saved in the labeled CSV files according to Table 2.

Table 3. The number of records from the public mail corpuses selected for the experiment.

Dataset	File Name	The Number of Records	The Percentage of All Records	The Number of SPAM Records	The Percent of SPAM Records	The Number of HAM Records	The Percent of HAM Records
CDMC2010	cdmc2010-full22f.csv	4279	100%	1331	31.11%	2948	68.89%
ENRON-SPAM	enronspam-full22f-mod.csv	48,490	100%	31,369	64.69%	17,121	35.31%
SpamAssasin	spamassasin-full22f.csv	5961	100%	1816	30.46%	4145	69.54%
TREC07p -full	Not used	75,225	100%	50,008	66.48%	25,217	33.52%
TREC07p-segment1	trec07p-50-1-22f.csv	37,612	50%	24,936	66.30%	12,676	33.70%
TREC07p-segment2	trec07p-50-2-22f.csv	37,612	50%	25,071	66.66%	12,541	33.34%

Table 3 illustrates that the datasets are heterogeneous and have varying numbers of records. Taking into account the number of records in the selected datasets, we propose to partition TREC07p into two segments, as it is the largest dataset. On the contrary, we must use the dataset for testing purposes and to generate client model updates. The first segment was utilized for model testing, while the second segment was used for client models updates. According to the information presented in [37], Naive Bayes is frequently the first method that is attempted while working with spam filters and sentiment analysis. Naïve Bayes, a high-bias/low-variance classifier, outperforms logistic regression and closest neighbor algorithms for training models with sparse data. The Naïve Bayes is also an excellent option when CPU and memory resources are limited. Because Naïve Bayes is so simple, it rarely overfits the data and can be taught rapidly. It also works effectively when the classifier is updated with new data on a continuous basis. We chose to use the ML Naïve Bayes classifier and the MATLAB R2024b function *fitcnb* to set up and create the initial global model for the reasons already mentioned. The simplified pseudo-code (see Algorithm 1) of the global model creation is presented below.

The performance evaluation of the benchmark models created to select an initial global model for further experiments is provided in Table 4.

Table 4. Performance evaluation of the benchmark models created.

Model	The Number of Training Records	Accuracy	Precision	Recall	F1 Score	MCC
cdmc2010	4279	0.80	0.72	0.81	0.77	0.53
enronspam	48,490	0.51	0.39	0.33	0.35	−0.28
spamassasin	5961	0.78	0.77	0.76	0.77	0.53
Global model		0.80	0.72	0.81	0.77	0.53

The graphical representation of the performance evaluation of the benchmark models depicted in Figure 5.

Algorithm 1. The pseudo-code of the initial global model creation.

Input:
Datasets: CDMC2010, ENRON-SPAM, SpamAssasin, TREC07p-segment1.
Creation:

1. Using the MATLAB function *readtable* which reads column-orientated data from a text file and produces a table, read all records from files:
    a. cdmcdata=readtable("cdmc2010-full22f.csv")
    b. enrondata=readtable("enronspam-full22f-mod.csv")
    c. assadata=readtable("spamassasin-full22f.csv")
    d. trecdata=readtable("trec07p-50-1-22f.csv")
2. Create ML models (cdmc2010, enronspam, spamassasin) using the MATLAB function *fitcnb* for Naïve Bayes classification.
3. Create compacted ML models using the *compact* MATLAB function. The compacted model does not include the training data, whereas the trained model does in its X and Y characteristics.
4. Classify observations with the Naïve Bayes classifier using MATLAB function *predict* that provides, using the learnt Naïve Bayes classification model Mdl, a vector of predicted class labels for the predictor data in the table or matrix X.
5. Use compact trained Naïve Bayes models and TREC07p-segment1 as the dataset for model testing.
6. Analyze the obtained test result and select the model with higher accuracy for future use as an initial global model.

Output:
Compacted models cdmc2010-full22f.mat, enronspam-full22f-mod.mat, spamassasin-full22f.mat, GlobalModel.mat
End the initial global model creation algorithm

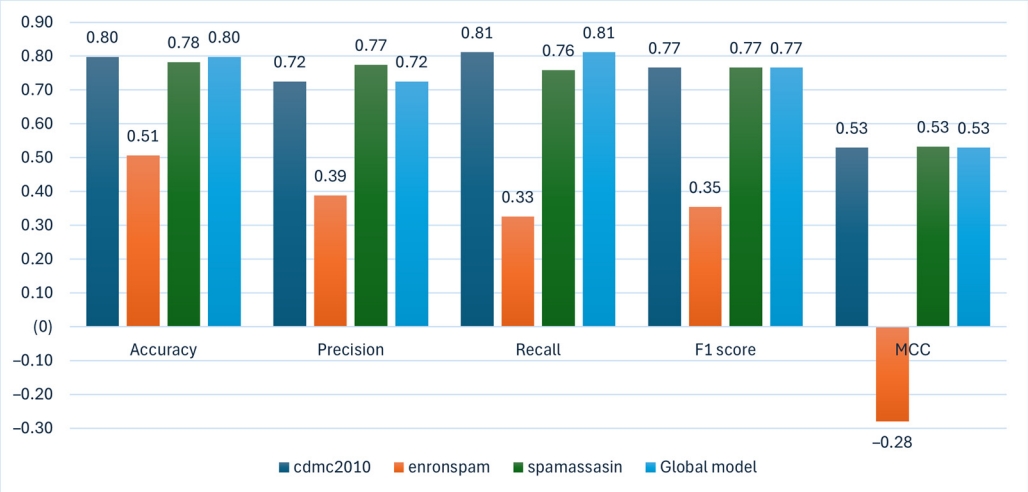


Figure 5. The graphical representation of the performance evaluation of the benchmark models.

Given that the ecdmc2010 model exhibits better performance metrics, it was designated as the initial global model for future applications and client updates.

4.2. Evaluation of Model Resilience: Experimental Results of the Proposed Approach

This section simulates two scenarios: first, all clients are trustworthy, and all model updates generated by them are reliable; second, clients are malicious and have intentionally modified all model updates.

4.2.1. Assessment of the Proposed Approach When Clients Are Trustworthy

The number of records of the public mail corpuses selected for the experiment included in the labeled datasets is provided in Table 3. To assess the proposed approach, client model updates were generated utilizing partitions from the TREC07p-segment2 dataset. The dataset was divided into 4K records and stored in ten CSV files, as detailed in Table 5.

Table 5. The number of records in the client updates for the experiment.

File Name of the Client Updates	The Number of Records	The Number of SPAM Records	The Percentage of SPAM Records.	The Number of HAM Records	The Percentage of HAM Records
CL4K01.csv	4096	2709	66.14%	1387	33.86%
CL4K02.csv	4096	2733	66.72%	1363	33.28%
CL4K03.csv	4096	2705	66.04%	1391	33.96%
CL4K04.csv	4096	2761	67.41%	1335	32.59%
CL4K05.csv	4096	2782	67.92%	1314	32.08%
CL4K06.csv	4096	2719	66.38%	1377	33.62%
CL4K07.csv	4096	2728	66.60%	1368	33.40%
CL4K08.csv	4096	2707	66.09%	1389	33.91%
CL4K09.csv	4096	2717	66.33%	1379	33.67%
CL4K10.csv	748	510	68.18%	238	31.82%
In total	37,612	25,071	66.66%	12,541	33.34%

Table 5 shows that the client models were created from the TREC07p-segment2, which contains a total of 37,612 data, comprising 25,071 SPAM and 12,541 HAM records. Figure 3 shows that the client model must be validated against all benchmark datasets before being applied to the global model. Only significant client model updates with high *F1 scores* and *MCC*s may be used as the global model update. In this situation, we use reliable labeled records from the TREC07p-segment2. This guarantees that all client model updates may be implemented correctly to the global model. The following is a simplified version of the pseudo-code (see Algorithm 2) that is used to validate the client's model.

The result of the client's local model *F1 scores* obtained during benchmark validation is presented in Table 6.

The result of the client's local model *MCC* scores obtained during the benchmark model validation is presented in Table 7.

Algorithm 2. The pseudo-code for the validation of the client model update.

Input:
Naïve Bayes classification models of participants: from CL4K01 to CL4K10;
Benchmark datasets for benchmark client model validation: cdmc2010.csv, en-ronspam.csv, spamassasin.csv, TREC07p-segment1.csv.
Creation:

1. Classify observations with the Naïve Bayes classifier using the MATLAB function *predict* that provides, using the learnt Naïve Bayes classification model, a vector of predicted class labels for the predictor data.
2. Use compact trained Naïve Bayes models from CL4K01 to CL4K10 to validate client models.
3. Use benchmark datasets for cross-validation: cdmc2010.csv, enronspam.csv, spamassasin.csv, TREC07p-segment1.csv.
4. Analyze the test result obtained, calculate the average *F1 score* and the average *MCC*.
5. Select client models with an *F1 score* and *MCC* exceeding the threshold for future applications.
6. Update the global model using selected client models.

Output:
Compacted updated Global model GlobalModel.mat
End the client's model updates testing algorithm

Table 6. *F1 scores* obtained during the benchmark validation of the client's models.

Client Model	F1 Score				
	cdmc2010	Enronspam	Spamassasin	TREC07p Segment1	AVERAGE
CL4K01	0.6280	0.6033	0.6528	0.9700	0.7135
CL4K02	0.6396	0.6054	0.6786	0.9704	0.7235
CL4K03	0.6257	0.5932	0.6522	0.9615	0.7081
CL4K04.	0.6378	0.6020	0.6670	0.9698	0.7192
CL4K05	0.6232	0.6036	0.6529	0.9681	0.7120
CL4K06	0.6204	0.6003	0.6450	0.9628	0.7071
CL4K07	0.6143	0.6042	0.6398	0.9669	0.7063
CL4K08	0.6232	0.6036	0.6485	0.9608	0.7090
CL4K09	0.6324	0.6027	0.6581	0.9641	0.7143
CL4K10	0.6132	0.3366	0.6657	0.9825	0.6495

Table 7. *MCC* scores obtained during the benchmark validation of client's models.

Client Model Update	MCC				
	cdmc2010	Enronspam	Spamassasin	TREC07p Segment1	AVERAGE
CL4K01	0.1635	0.1691	0.2431	0.9401	0.3789
CL4K02	0.1968	0.1758	0.3128	0.9409	0.4066
CL4K03	0.1457	0.1486	0.2312	0.9230	0.3622
CL4K04.	0.1822	0.1736	0.2776	0.9396	0.3933
CL4K05	0.1431	0.1703	0.2389	0.9362	0.3721
CL4K06	0.1309	0.1603	0.2097	0.9257	0.3566
CL4K07	0.1202	0.1720	0.1997	0.9338	0.3564
CL4K08	0.1431	0.1700	0.2247	0.9216	0.3649
CL4K09	0.1657	0.1675	0.2511	0.9282	0.3781
CL4K10	0.2253	−0.3042	0.3314	0.9650	0.3044

We use Equation (8) and the values of the global model *TP*, *FN*, *FP*, and *TN* values (see Figure 6) to find the 0.6574 *CP* threshold for the *F1 score*, which is then used to ensure the update of the client model is acceptable.

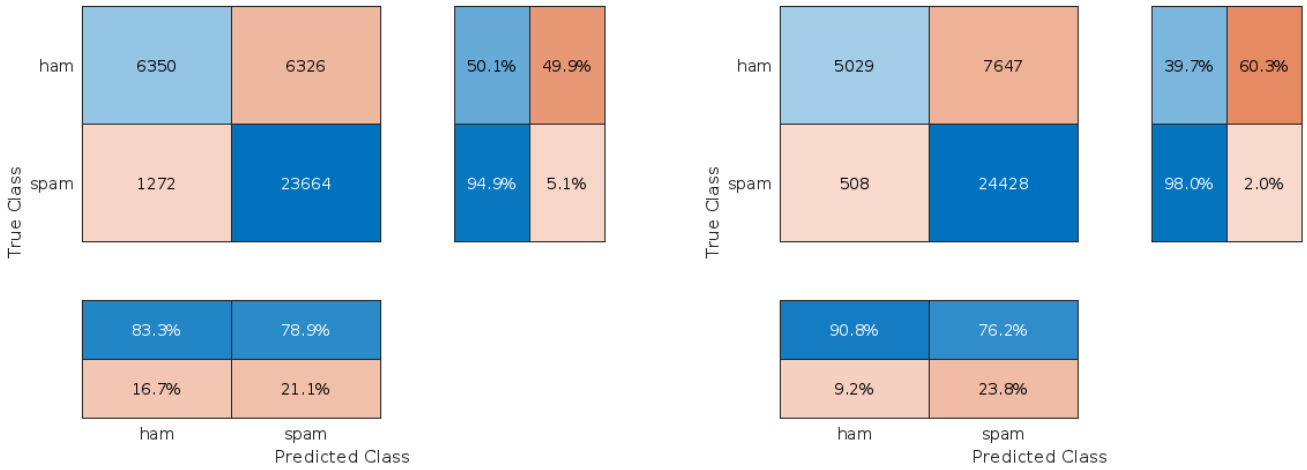


Figure 6. The confusion matrix of the global model: (**a**) before applying client's model updates; (**b**) after client's model updates were applied.

Looking at the results of the benchmark validation procedure for the client model (see Table 7 for more details) and the *F1 score* threshold for the *MCC*, it was decided that a threshold of 0.32 was most likely to be correct.

This section presents an experiment evaluating the suggested approach under the condition that all clients are trustworthy, utilizing client model updates without any malicious alterations. Although the CL4K10 client is classified as trusted, his model did not exceed the *F1 score* and *MCC* thresholds and is inapplicable for the global model update. All remaining updates are appropriate and implemented in the global model. The global model confusion matrixes before applying the local client model updates (a) and after the updates (b) are presented in Figure 6. We employed a standard plot produced by the MATLAB function *confusionchart* to present the confusion matrix. The confusion matrix displays the total number of observations in each cell.

The rows of the confusion matrix represent the actual class, while the columns denote the expected class. Diagonal cells represent accurately classified observations, while off-diagonal cells denote inaccurately classified observations. A row-normalized summary presents the percentages of accurately and inaccurately identified observations for each actual class. A column-normalized summary presents the percentages of accurately and inaccurately identified observations for each projected class.

While analyzing the column-normalized summary one can observe that the updated global model has increased the percentage of true ham messages among all messages, which are classified as ham (from 83.3 to 90.8%). This is an expected result because, in this scenario, all local clients were trustworthy and they used correct data for local model training; therefore, the global model was improved after aggregation of model updates.

On the other hand, the percentage of true spam messages among all messages classified as spam decreased slightly from 78.9 to 76.2%. A similar situation could be observed when analyzing the row-normalized summary values. The percentage of all true spam messages classified as spam increased from 94.9 to 98.0% after updates to the global model were applied. The percentage of all true ham messages classified as ham decreased from 50.1 to 39.7%. One can observe that the percentage increase in one summary row or column is followed by the decrease in the other one. This is caused by the limitations of the

composite scores (i.e., *F1 score*), which are known to cause contradictory results when used for optimization and performance evaluation of the prediction models. On the other hand, the primary purpose of the classification model is to recognize spam messages, which means that the primary characteristic of the model is the ability to recognize spam messages, which is achieved successfully in 98% of all cases.

An attacker may alter labels from SPAM to HAM and vice versa to compromise the model resilience and execute a Byzantine attack. This scenario will be simulated in the next section.

4.2.2. Assessment of the Proposed Approach When Clients Are Malicious

Client model updates were generated using partitions of the TREC07p-segment2 dataset to assess the proposed approach. The dataset was partitioned into 4K records and stored in ten CSV files. We modified the class labels in the CSV files to simulate malicious participants who adversely impacted clients' local models. Details of these modifications are presented in Table 8 (inversion of Table 5).

Table 8. The groups of messages in the malicious client updates for experiment.

File Name of the Client Updates	The Number of Records	The Number of SPAM Records	The Percentage of SPAM Records	The Number of HAM Records	The Percentage of HAM Records
CL4KM01.csv	4096	1387	33.86%	2709	66.14%
CL4KM02.csv	4096	1363	33.28%	2733	66.72%
CL4KM03.csv	4096	1391	33.96%	2705	66.04%
CL4KM04.csv	4096	1335	32.59%	2761	67.41%
CL4KM05.csv	4096	1314	32.08%	2782	67.92%
CL4KM06.csv	4096	1377	33.62%	2719	66.38%
CL4KM07.csv	4096	1368	33.40%	2728	66.60%
CL4KM08.csv	4096	1389	33.91%	2707	66.09%
CL4KM09.csv	4096	1379	33.67%	2717	66.33%
CL4KM10.csv	748	238	31.82%	510	68.18%
In total	37,612	12,541	33.34%	25,071	66.66%

Table 5 shows that client updates are created from the TREC07p-segment2, which contains a total of 37,612 records, comprising 25,071 SPAM and 12,541 HAM records. Table 8 shows that malicious client updates are created from the inverted TREC07p-segment2, where there are a total of 37,612 records, comprising 25,071 HAM and 12,541 SPAM records. Figure 3 shows that the client model must be validated against all benchmark datasets before being applied to the global model. Only significant client model updates with high *F1 scores* and *MCC*s may be used as global model updates. In this scenario, we utilize maliciously labeled records from the TREC07p-segment2, to simulate the Byzantine attack, ensuring that all client model updates can be appropriately applied to the global model.

Table 9 presents the results of the malicious client's local model *F1 scores* obtained during client model benchmark validation.

In the section before this one, the *F1 score* threshold was determined to be 0.6574. Considering the findings shown in Table 9, it is possible to conclude that there is no local model update that could be applied to the global model.

The result of the malicious client's local model *MCC* obtained during benchmark model validation is presented in Table 10.

Table 9. *F1 scores* of the malicious client's local model obtained during client model benchmark validation.

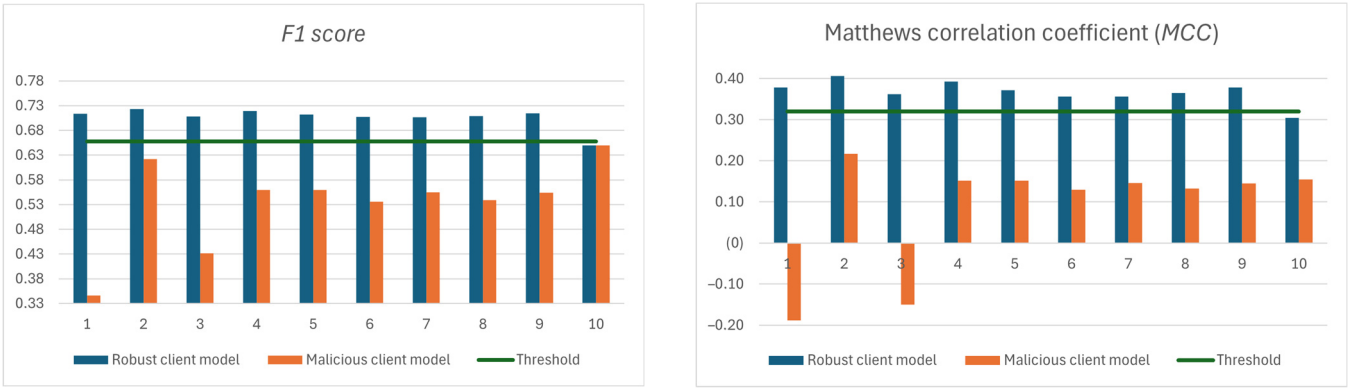
Client Model Update	F1 Score				
	cdmc2010	**Enronspam**	**Spamassasin**	**TREC07p Segment1**	**AVERAGE**
CL4KM01	0.4237	0.3904	0.3436	0.2263	0.3460
CL4KM02	0.5775	0.6789	0.5230	0.7082	0.6219
CL4KM03	0.5281	0.0005	0.5315	0.6643	0.4311
CL4KM04.	0.4759	0.9994	0.4280	0.3364	0.5599
CL4KM05	0.4759	0.9994	0.4280	0.3364	0.5599
CL4KM06	0.3909	0.9990	0.4160	0.3385	0.5361
CL4KM07	0.4650	0.9990	0.4164	0.3376	0.5545
CL4KM08	0.3957	0.9991	0.4223	0.3383	0.5388
CL4KM09	0.4639	0.9990	0.4176	0.3359	0.5541
CL4KM10	0.6132	0.3366	0.6657	0.9825	0.6495

Table 10. *MCC* score of the malicious client's local model obtained during the benchmark validation.

Client Model Update	MCC				
	cdmc2010	**Enronspam**	**Spamassasin**	**TREC07p Segment1**	**AVERAGE**
CL4KM01	−0.0395	−0.0602	−0.1060	−0.5465	−0.1881
CL4KM02	−0.0108	0.5578	0.006	0.3163	0.2173
CL4KM03	0.0216	−0.9988	0.0509	0.3275	−0.1497
CL4KM04.	−0.0161	0.9987	−0.0548	−0.3222	0.1514
CL4KM05	−0.0161	0.9987	−0.0548	−0.3222	0.1514
CL4KM06	−0.0871	0.9981	−0.0735	−0.3204	0.1293
CL4KM07	−0.0237	0.9980	−0.0677	−0.3212	0.1464
CL4KM08	−0.0825	0.9982	−0.0658	−0.3206	0.1323
CL4KM09	−0.0252	0.9981	−0.0685	−0.3243	0.1450
CL4KM10	0.2253	−0.3042	0.3314	0.365	0.1544

In Section 4.2.1 that came before this one, a minimum score of 0.32 was decided on for the *MCC* score. Considering the findings shown in Table 10, it is possible to conclude that there is no local model update that could be applied to the global model.

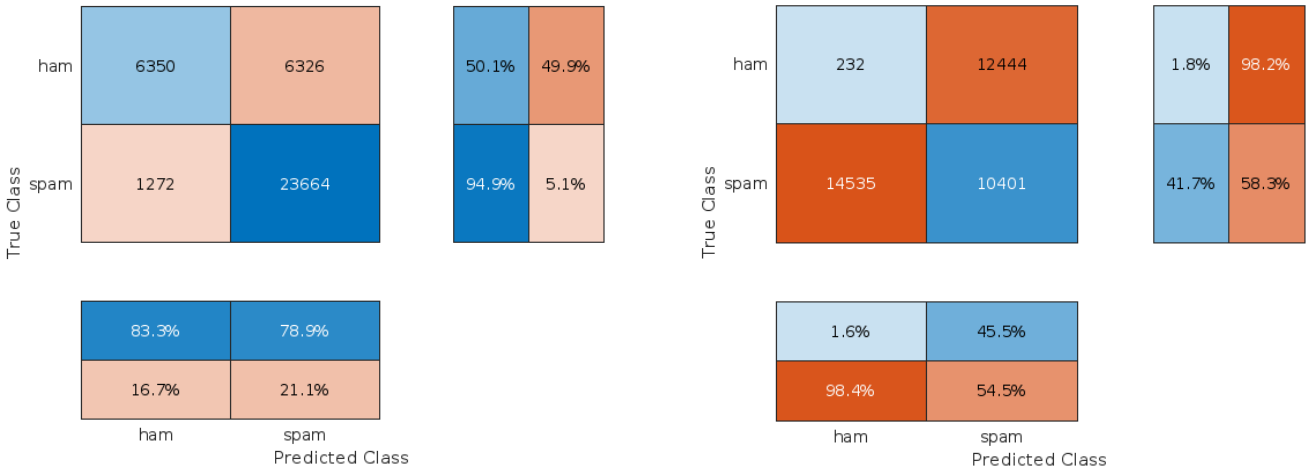
Through the utilization of the proposed approach, only reliable clients' local model updates that have been validated through benchmark model validation can be applied to the global model. A comparison of the *F1 scores* obtained and the *MCC* measures of the client models shown in Figure 7.



(**a**) (**b**)

Figure 7. Comparison of the *F1 scores* obtained and the *MCC* measures of the client's models: (**a**) *F1 score*; (**b**) Matthews correlation coefficient (*MCC*).

To access the proposed approach, an experiment was conducted in which all malicious local model updates were applied to the global model, simulated by the Byzantine attack. The confusion matrices before the updates of the maliciously modified local models of the client (a) and after the global model (b) are shown in Figure 8.



(**a**) (**b**)

Figure 8. The confusion matrix of the global model: (**a**) before applying malicious client model updates; (**b**) after malicious client model updates were applied.

While analyzing the confusion matrices, one can clearly see that malicious client model updates significantly compromise the global model. The row-normalized summary shows that the percentage of true spam messages that were correctly classified as spam decreased from 94.9% to only 41.7%. The percentage of true ham messages that were classified as ham also decreased from 50.1 to 1.8%. Column-normalized summaries are even worse. It shows that the percentage of true ham messages among messages classified as ham dropped from 83.3 to 1.6%. The percentage of true spam messages among all messages classified as spam dropped from 78.9 to 45.5%. These results clearly illustrate how malicious clients could compromise the global model by providing harmful model updates obtained by training local models on purposely altered data.

The results of this scenario clearly show that the use of client models validation scores for approval of the global model update is essential in FL environments where the possibility of malicious clients capable of performing Byzantine attacks is significant.

4.2.3. Assessment of the Proposed Approach to Protect Against Backdoor and Model Inversion Attacks

An adversary conducts a model inversion attack (MIA) when he or she leverages the trained model to retrieve its training data. MIAs essentially include frequently querying a model with known inputs and collecting its results. With just two pieces of information the target model and a basic understanding of non-sensitive features MIAs have successfully recovered local social networks, private information, and realistic images [38]. As a measure against MIA overfitting, early stopping is used to stop training the model as soon as its performance on a validation set stops improving, based on a predetermined patience parameter [39]. While MIAs can target as few as one client, backdoors intentionally misclassify just classes with certain properties in order to poison all client models. According [40] it is possible to launch a backdoor assault, in which only the clients that are targeted (victims) receive a backdoored model, while the clients that are not targeted receive a clean model.

In our proposed method, two defenses against MIAs and backdoor attack strategies are used:

- We employ email domain-specific ontology that excludes any potentially sensitive information from email messages and instead concentrates on their metadata, which includes technical elements that show how an email travels from sender to recipient;
- To ensure that updates from malicious or trustless clients do not compromise a global FL model's resilience, client models are evaluated using benchmark validation prior to making any updates;
- It is possible to use the semantic representation of the email message's metadata for the classification of encrypted email messages without knowing of the decryption key. The S/MIME email encryption standard cryptographically protects only the body of the messages, whereas the header fields remain in plaintext because the SMTP servers need to deliver the message correctly. In such a case, only the metadata of the attached files and links to external resources cannot be extracted and populated. That is, classification is possible on the SMTP servers of the service provider without compromising the confidentiality of the final user's data.

5. Discussion

The organization of ML research is largely influenced by benchmark datasets. They act as a gauge of progress toward common objectives and arrange researchers around related projects. There are various public mail corpuses that could be used to create benchmark datasets, such as CDMC2010, ENRONSPAM, SpamAssasin, TREC07p that were used in our research, as mentioned in Table 2. All headers had to be fully reproduced because the semantic parser and the email domain-specific ontology were used to make a benchmark dataset with email metadata only from headers to preserve email message privacy. This was the main factor used to choose the benchmark dataset. We chose the four email corpuses described above based on this criterion. We were unable to identify any other benchmark datasets that were appropriate.

The novelty of the proposed approach to improving the resilience of a global federated learning model is based on the evaluation of the client model using benchmark validation to detect malicious or trustless clients. An update of the client model that exhibits an appropriate benchmark validation score may be used to update the global model. This will protect the global model against a Byzantine attack.

To resolve the robustness and resiliency challenges, we propose using a pool of benchmark datasets on the cloud to apply the process for validating client models and select validated updates only that exhibit an appropriate benchmark validation scor. When

the cloud server receives a local model update from the client (refer to Figure 3), it uses a pool of the unified benchmark datasets developed utilizing the semantic parser and the email domain-specific ontology and initiates the validation process. The validation process gathers performance metrics from all benchmark datasets.

For efficient client model validation, some integral model performance metrics are required. We suggest a two-part evaluation that finds the average *MCC* and *F1 score* achieved during client model validation using benchmark datasets to judge the quality of the FL participant's updates and subsequently, compare the average *MCC* and *F1 score* with the associated threshold *MCC* and *F1 score* metrics. In certain instances, updating the global model is considered suitable and robust when the score surpasses the threshold value. The client model update is discarded if the score falls below the threshold. As only validated client model updates are allowed and utilized to update the global model, client model benchmark validation ensures the resilience of the global model.

We started by simulating trusted client models using the TREC07p dataset partitions, since the spam and ham labels remained unchanged. We used concordance probability to evaluate the client's models to find the best threshold of 0.6574 for the *F1 score*; for the *MCC*, a 0.32 threshold was chosen. Our experimental results demonstrate that the measures we chose to use can reliably forecast how trustworthy a client model will be and how well it can be used to update a global model.

The second experiment involved replicating models from malicious or untrusted clients with the TREC07p dataset partitions, in which the spam and ham labels were modified. The spam label was reclassified as ham, while the ham label was reclassified as spam. Our experimental results indicate that the selected measures *F1 score* and *MCC* may consistently predict malicious behavior of the client model and prevent it from updating the global model, thus ensuring the resilience of the global model.

The experimental results were compared with state-of-the-art research (SOTA), as presented in Table 11.

Table 11. Experimental results compared to state-of-the-art research.

Research	Approach	Performance Metrics	Algorithm	Validation	F1	MCC
Yazdinejad et al. [5]	Additive Homomorphic Encryption (AHE)	Target accuracy, other label accuracy, overall Accuracy	Detecting anomalies by comparing local gradients' cosine similarity with benign gradients	The server-held validation dataset	–	–
Wang et al. [8]	Customizable privacy protection	Accuracy	A distributed federation game with transferable utility	–	–	–
Zheng et al. [10]	Integration between secure aggregation and quantization-based model compression	Accuracy	Empowering Federated Learning with secure and efficient aggregation	The cloud server only learns the aggregate model update without knowing individual model updates	–	–
Imteaj et al. [13]	On-device learning without sharing any data	Accuracy	FedResilience algorithm	The server initializes a global model	–	–
So et al. [15]	Use of Stochastic Quantization	Accuracy	The multi-Krum algorithm in a quantized stochastic gradient setting	Single-server solution for Byzantine-resilient secure federated learning	–	–
Rjoub et al. [17]	Privacy-preserving machine learning	Precision, Recall, F1 score	Deep reinforcement learning (DRL) scheduling algorithm	The server aggregates parameters to derive a global aggregate model	+	–
Proposed	Integrates FL, email domain ontology, a semantic parser, and a collection of benchmark datasets from heterogeneous email corpuses	Accuracy, Precision, Recall, F1 score, MCC	Naïve Bayes classifier	Enhanced global model resilience through client model benchmark validation on the cloud	+	+

The findings indicate the feasibility of implementing the proposed approach in companies providing email services. The proposed approach is designed for use in a federated environment, where participants consist of small organizations or enterprises that maintain their own server infrastructures and function as enterprise-level email service providers. In this scenario, it is not essential to pursue the objective of reducing computational overhead. If an enterprise can host and support local email servers, it should be able to allocate the computational resources necessary for training of the local model to mitigate spam messages and improve the overall quality of the email service offered to its users.

6. Conclusions

There are many obstacles that FL faces, one of which is the process of selecting the clients who are the most suitable to participate in a training process and global model updates. Since random masks are used in FL to protect local updates, the worldwide server is unable to view the true values of these updates. One of the most critical challenges to the resilience of the global model is the fact that hostile users, also known as Byzantine users, have the ability to influence the global model by modifying their local updates or datasets.

To achieve robustness and resilience, the other authors have suggested employing Byzantine-robust aggregation rules. These rules include gradient aggregation, lightweight secure and resilient aggregation, early prediction, poisoning, and inference. The fact that an adversary might develop an adaptable attack by altering the aggregation rule is one of the most significant drawbacks of robustness and resiliency.

In summarizing our research, we can conclude as follows:

- This article presents an investigation of the issues associated with email classification, along with a new strategy to improve resilience that is based on benchmark validation;
- Federated learning, the email domain-specific ontology, the semantic parser, and benchmark datasets derived from publicly available email corpuses are incorporated into the approach that was suggested;
- Using the email domain-specific ontology to extract metadata from an email message enables the creation of datasets with identical features derived from the benchmark email corpora and model updates provided by the client. This approach enables the utilization of benchmark models to verify the accuracy of client model updates;
- Meanwhile, the semantic parser serves to safeguard the confidentiality of email messages because only email message metadata are used.
- All client models have their performance metrics collected during the client model benchmark validation. In our experiment, the *F1 score* and the *MCC* were the primary metrics used to assess the client model and its applicability to update the global model.

Feature work could investigate additional applications of the proposed approach, such as investigating the optimization value of all performance metrics obtained during validation, to enhance the precision of evaluating the applicability of the client models.

Author Contributions: Conceptualization, A.V., J.T. and N.M.; methodology, A.V., J.T., N.M. and E.S.; software, N.M.; validation, J.T., N.M. and M.K.; formal analysis and investigation, J.T., N.M. and E.S.; data curation, N.M. and M.K.; writing—original draft preparation, A.V., J.T. and N.M.; writing—review and editing, A.V., J.T., N.M., E.S. and M.K.; visualization, J.T., E.S. and M.K.; supervision, A.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work was conducted as part of the execution of the project "Mission-driven Implementation of Science and Innovation Programs" (No. 02-002-P-0001), funded by the Economic Revitalization and Resilience Enhancement Plan "New Generation Lithuania".

Data Availability Statement: The email domain-specific ontology is publicly available at doi: 10.17632/tgm39cfggr.1 (https://data.mendeley.com/datasets/tgm39cfggr/1, accessed on 27 February 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Qi, P.; Chiaro, D.; Guzzo, A.; Ianni, M.; Fortino, G.; Piccialli, P. Model aggregation techniques in federated learning: A comprehensive survey. *Future Gener. Comput. Syst.* **2024**, *150*, 272–293. [CrossRef]
2. Mothukuri, V.; Parizi, M.R.; Pouriyeh, S.; Huang, Y.; Dehghantanha, A.; Srivastava, G. A survey on security and privacy of federated learning. *Future Gener. Comput. Syst.* **2021**, *115*, 619–640. [CrossRef]
3. Zhang, Y.; Zeng, D.; Luo, J.; Fu, X.; Chen, G.; Xu, Z.; King, I. A Survey of Trustworthy Federated Learning: Issues, Solutions, and Challenges. *ACM Trans. Intell. Syst. Technol.* **2024**, *15*, 1–47. [CrossRef]
4. Zhang, Y.; Sun, R.; Shen, L.; Bai, G.; Xue, M.; Meng, M.H.; Li, X.; Ko, R.; Nepal, S. Privacy-Preserving and Fairness-Aware Federated Learning for Critical Infrastructure Protection and Resilience. In Proceedings of the ACM Web Conference 2024 (WWW '24), Singapore, 13–17 May 2024; Association for Computing Machinery: New York, NY, USA, 2024; pp. 2986–2997. [CrossRef]
5. Yazdinejad, A.; Dehghantanha, A.; Karimipour, H.; Srivastava, G.; Parizi, R.M. A Robust Privacy-Preserving Federated Learning Model Against Model Poisoning Attacks. *IEEE Trans. Inf. Forensics Secur.* **2024**, *19*, 6693–6708. [CrossRef]
6. Chen, J.; Yan, H.; Liu, Z.; Zhang, M.; Xiong, Z.; Yu, S. When Federated Learning Meets Privacy-Preserving Computation. *ACM Comput. Surv.* **2024**, *56*, 319. [CrossRef]
7. Jagarlamudi, G.K.; Yazdinejad, A.; Parizi, R.M.; Pouriyeh, S. Exploring privacy measurement in federated learning. *J. Supercomput.* **2024**, *80*, 10511–10551. [CrossRef]
8. Wang, Y.; Su, Z.; Pan, Y.; Luan, T.H.; Li, R.; Yu, S. Social-Aware Clustered Federated Learning with Customized Privacy Preservation. *IEEE/ACM Trans. Netw.* **2024**, *32*, 3654–3668. [CrossRef]
9. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
10. Zheng, Y.; Lai, S.; Liu, Y.; Yuan, X.; Yi, X.; Wang, C. Aggregation Service for Federated Learning: An Efficient, Secure, and More Resilient Realization. *IEEE Trans. Dependable Secur. Comput.* **2023**, *20*, 988–1001. [CrossRef]
11. Shabbir, A.; Manzoor, H.U.; Ahmed, R.A.; Halim, Z. Resilience of Federated Learning Against False Data Injection Attacks in Energy Forecasting. In Proceedings of the 2024 International Conference on Green Energy, Computing and Sustainable Technology (GECOST). Miri Sarawak, Malaysia, 17–19 January 2024; pp. 245–249. [CrossRef]
12. Reisizadeh, A.; Tziotis, I.; Hassani, H.; Mokhtari, A.; Pedarsani, R. Straggler-Resilient Federated Learning: Leveraging the Interplay Between Statistical Accuracy and System Heterogeneity. *IEEE J. Sel. Areas Inf. Theory* **2022**, *3*, 197–205. [CrossRef]
13. Imteaj, A.; Khan, I.; Khazaei, J.; Amini, M.H. FedResilience: A Federated Learning Application to Improve Resilience of Resource-Constrained Critical Infrastructures. *Electronics* **2021**, *10*, 1917. [CrossRef]
14. Yamany, W.; Keshk, M.; Moustafa, N.; Turnbull, B. Swarm Optimization-Based Federated Learning for the Cyber Resilience of Internet of Things Systems Against Adversarial Attacks. *IEEE Trans. Consum. Electron.* **2024**, *70*, 1359–1369. [CrossRef]
15. So, J.; Güler, B.; Avestimehr, A.S. Byzantine-Resilient Secure Federated Learning. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 2168–2181. [CrossRef]
16. Tao, Y.; Cui, S.; Xu, W.; Yin, H.; Yu, D.; Liang, W.; Cheng, X. Byzantine-Resilient Federated Learning at Edge. *IEEE Trans. Comput.* **2023**, *72*, 2600–2614. [CrossRef]
17. Rjoub, G.; Wahab, O.A.; Bentahar, J.; Cohen, R.; Bataineh, A.S. Trust-Augmented Deep Reinforcement Learning for Federated Learning Client Selection. *Inf. Syst. Front.* **2024**, *26*, 1261–1278. [CrossRef]
18. Rjoub, G.; Wahab, O.A.; Bentahar, J.; Bataineh, A. Trust-driven reinforcement selection strategy for federated learning on IoT devices. *Computing* **2024**, *106*, 1273–1295. [CrossRef]
19. Sánchez, P.M.S.; Celdrán, A.H.; Xie, N.; Bovet, G.; Pérez, G.M.; Stiller, B. FederatedTrust: A solution for trustworthy federated learning. *Future Gener. Comput. Syst.* **2024**, *152*, 83–98. [CrossRef]
20. Cao, X.; Fang, M.; Liu, J.; Gong, N.Z. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping. In Proceedings of the 2021 Network and Distributed System Security (NDSS) Symposium, San Diego, CA, USA, 21–25 February 2021. [CrossRef]
21. Ji, S.; Tan, Y.; Saravirta, T.; Yang, Z.; Liu, Y.; Vasankari, L.; Pan, S.; Long, G.; Walid, A. Emerging trends in federated learning: From model fusion to federated X learning. *Int. J. Mach. Learn. Cybern.* **2024**, *15*, 3769–3790. [CrossRef]
22. Tang, Z.; Chu, X.; Ran, R.Y.; Lee, S.; Shi, S.; Zhang, Y.; Wang, Y.; Liang, A.Q.; Avestimehr, S.; He, C. Fedml parrot: A scalable federated learning system via heterogeneity-aware scheduling on sequential and hierarchical training. *arXiv* **2023**, arXiv:2303.01778.
23. Venčkauskas, A.; Toldinas, J.; Morkevičius, N.; Sanfilippo, F. Email Domain-specific Ontology and Metadata Dataset. *Mendeley Data* **2024**. [CrossRef]

24. Venčkauskas, A.; Toldinas, J.; Morkevičius, N.; Sanfilippo, F. An Email Cyber Threat Intelligence Method Using Domain Ontology and Machine Learning. *Electronics* **2024**, *13*, 2716. [CrossRef]
25. Sikandar, H.S.; Waheed, H.; Tahir, S.; Malik, S.U.R.; Rafique, W. A Detailed Survey on Federated Learning Attacks and Defenses. *Electronics* **2023**, *12*, 260. [CrossRef]
26. Lyu, L.; Yu, H.; Ma, X.; Chen, C.; Sun, L.; Zhao, J. Privacy and Robustness in Federated Learning: Attacks and Defenses. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 8726–8746. [CrossRef] [PubMed]
27. Herath, C.; Rahulamathavan, Y.; Liu, X. Recursive Euclidean Distance-based Robust Aggregation Technique for Federated Learning. In Proceedings of the 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), London, UK, 19–21 May 2023; pp. 1–6. [CrossRef]
28. Ozdayi, M.S.; Kantarcioglu, M.; Gel, Y.R. Defending against Backdoors in Federated Learning with Robust Learning Rate. In Proceedings of the 2021 AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, pp. 9268–9276. [CrossRef]
29. Gencturk, M.; Sinaci, A.A.; Cicekli, N.K. BOFRF: A Novel Boosting-Based Federated Random Forest Algorithm on Horizontally Partitioned Data. *IEEE Access* **2022**, *10*, 89835–89851. [CrossRef]
30. Chicco, D.; Jurman, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genom.* **2020**, *21*, 6. [CrossRef]
31. Kaur, N.; Singh, H. An empirical assessment of threshold techniques to discriminate the fault status of software. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 6339–6353. [CrossRef]
32. Spamassassin. Available online: https://spamassassin.apache.org/old/publiccorpus/ (accessed on 20 December 2024).
33. CDMC2010. ICONIP 2010: 17th International Conference on Neural Information Processing. Available online: http://www.wikicfp.com/cfp/servlet/event.showcfp?eventid=10627&copyownerid=5571 (accessed on 20 December 2024).
34. Metsis, V.; Androutsopoulos, I.; Paliouras, G. Spam Filtering with Naive Bayes-Which Naive Bayes? In Proceedings of the 2006 Third Conference on Email and Anti-Spam CEAS, Mountain View, CA, USA, 27–28 July 2006; Volume 17, pp. 28–69.
35. Klimt, B.; Yang, Y. The Enron Corpus: A New Dataset for Email Classification Research. In Proceedings of the Machine Learning: ECML 2004, Pisa, Italy, 20–24 September 2004; Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; pp. 217–226.
36. Cormack, G.V. TREC 2007 Spam Track Overview Conference. In Proceedings of the 2007 Text Retrieval Conference, Gaithersburg, MD, USA, 6–9 November 2007; Available online: https://api.semanticscholar.org/CorpusID:2848551 (accessed on 20 December 2024).
37. Choosing the Right Classification Model. Available online: https://www.mathworks.com/campaigns/offers/next/choosing-the-best-machine-learning-classification-model-and-avoiding-overfitting.html (accessed on 20 December 2024).
38. Zhou, Z.; Zhu, J.; Yu, F.; Li, X.; Peng, X.; Liu, T.; Han, B. Model Inversion Attacks: A Survey of Approaches and Countermeasures. *arXiv* **2024**, arXiv:2411.10023. Available online: https://arxiv.org/abs/2411.10023 (accessed on 27 February 2025).
39. Parikh, R.; Dupuy, C.; Gupta, R. Canary Extraction in Natural Language Understanding Models. *arXiv* **2022**, arXiv:2203.13920. Available online: https://arxiv.org/abs/2203.13920 (accessed on 27 February 2025).
40. Abad, G.; Paguada, S.; Ersoy, O.; Picek, S.; Ramírez-Durán, V.J.; Urbieta, A. Sniper Backdoor: Single Client Targeted Backdoor Attack in Federated Learning. *arXiv* **2023**, arXiv:2203.08689. Available online: https://arxiv.org/abs/2203.08689 (accessed on 27 February 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.