Privacy-Preserving Continual Learning for Detecting Concept Drift in Distributed Artificial Intelligence Systems

Sai Doondi Kothapalli[1]

Abstract

Distributed artificial intelligence systems increasingly process data streams whose statistical properties change over time, a phenomenon known as concept drift. In centralised pipelines, researchers typically diagnose drift by inspecting raw observations. In federated and edge deployments, however, regulatory and contractual constraints, along with significant re-identification risks associated with sharing gradients and statistics, preclude such inspection. This article introduces and evaluates a framework for privacy-preserving continual learning that detects and adapts to concept drift without exposing client-level data or unsecured model updates. The framework comprises three components: a differentially private drift-signalling channel that transmits perturbed loss and confidence summaries from clients; a server-side adaptive windowing detector that operates on securely aggregated signals; and a regularised rehearsal mechanism that mitigates catastrophic forgetting during adaptation. We evaluate the proposed design on three streaming benchmarks: energy demand, synthetic rotating hyperplanes, and network intrusion traffic, each distributed across fifty non-identically distributed clients under abrupt, gradual, and recurring drift scenarios. Results indicate that drift is detectable at moderate privacy budgets, with a mean detection delay increase of approximately thirty per cent compared to a non-private baseline. Furthermore, accuracy retention on previously learned concepts improves by up to eleven percentage points relative to naive fine-tuning. The analysis also identifies a critical degradation threshold below a privacy budget of one, where noise variance overwhelms the drift signal and false alarms become prevalent. These findings delineate the practical boundaries for balancing privacy guarantees and drift responsiveness in distributed learning systems.

Keywords: concept drift, continual learning, federated learning, differential privacy, distributed artificial intelligence, catastrophic forgetting, secure aggregation.

1. Introduction

Machine learning algorithms don't need to be trained on the datacenter-level and curated static data sets anymore. Yet each hospital network, each mobile handset, each industrial sensor, the banking system and the educational system have its own observations that it owns and controls and, in turn, trains and refreshes its own models. This setup has inspired the current predominant architectural approach, federated learning, which allows to enhance the shared model by exchanging updates on its parameters instead of records [1]. Such deployments have been documented throughout the Internet of Things, telecommunication and clinical informatics, where issues like bandwidth limitations, jurisdictional constraints, and confidentiality causes make it impractical for data to be centralised [2].

A tricky issue in such deployment which is not treated so much is the temporal validity. Such a joint distribution that maps the inputs to outputs is often non-stationary. The way a consumer acts changes, the tactics of the attacker change, the sensors age, the population changes and the relation a model has learned silently dies. Often referred to as concept drift, it has been widely discussed in the field of stream mining, and there is a great taxonomy of the different types of concept drift, including abrupt, gradual, incremental and recurring [3,4]. The standard solutions (other than what has been said above) require a learner to be able to look at the data it is tracking or, at least, the errors that it produces per instance, and assume that this is possible. That is not the case in a federated scenario. The server does not know anything about the raw stream and, under any reasonably defensible assumption of privacy, the unprotected per-client error stream.

Continual Learning is complementary: To be able to learn new concepts while retaining previously learnt concepts. Field surveys verify the presence of a developed cluster of strategies organised in general categories of regularisation, rehearsal and architectural extension [5,6]. But most research on continual learning focuses on the individual user, who is provided with their own history, and the rehearsal buffers upon which many continual learning methods rely are actually the very type of artifacts privacy regimes detract from. As for the intersection between the two literatures — drift-aware continual learning under distribution — the resulting research has recently been starting to shift attention [7,8] to but in most cases the privacy dimension is rarely considered in terms of concealing the data, rather it is treated as an assertion or the assumption is that the mere absence of the raw data transfer is enough.

This is an hard-to-maintain treatment. The research into gradient inversion has shown that updating the model can be used to replicate the input of training with a relatively high degree of accuracy, first on a small number of data and later under better realistic conditions [9,10]. In medical imaging, reconstruction methods have been demonstrated to work well on deployed federated pipelines but are also dependent on the volume of the batch, the depth of the model and the aggregation strategy [11]. Loss statistics and other confidence distributions that the drift detector needs must be considered as potential disclosure channels, therefore, anything derived from client data and passed

along to a coordinator. The formal means exist for bounding that disclosure: Differential privacy [12]; combining it with deep learning via noised, clipped stochastic gradient descent is well known. The “guarantee” comes at a price: It compromises utility unequally for subgroups, particularly for under-represented subgroups, and they suffer proportionately more of the cost [13].

The tension the article is dealing with is thus concrete as opposed to abstract. Drift detection essentially is a problem of detecting a small, systematic change in a signal over time. The essence of differential privacy is smudging minor, regular modifications in a signal. There is a direct conflict between the two targets, and the area where both can be achieved is poorly-defined.

The stakes are wide-ranging – concerned with learning efforts in various sectors where distributed learning is emerging. As more educational technology platforms adopt predictive analytics without the option of holistic student record sharing across schools, the potential of such systems to inform both administration and instruction relies heavily on their effectiveness in maintaining a high level of accuracy as cohorts of students and courses change over time [14,15]. Can be also found in localized credential and provenance infrastructure where distributed architectures cannot lose integrity as participants and usage evolves over time [16]. In both situations a hidden bias generates a system that is certainly wrong, while a drift response leaks to other systems, creating an accurate but illegal drift response system.

This article has a few points to make. It puts drift signalling in distributed scenarios on a channel design level, instead of a model level, focusing on the quantity that needs to be protected: the detection statistic. It provides a framework, called Privacy-Preserving Continual Drift Adaptation, which includes both client-side signalling using differential privacy and secure aggregation, server-side adaptive windowing, and rehearsal using regularisation. It gives an evaluation on three different datasets, and three different drift regimes, and represents the ratio of privacy budget to forgetting and detection delay. Last, but not least, it finds the threshold value below which the framework becomes non viable, directly relevant to the selection of parameters in real life applications. Throughout, the following approaches are used for comparison: adaptive drift clustering [17]

The methodology is set forth in Section 2, while Section 3 reviews the relevant literature is presented, the problem is formally presented in section 4, the results are presented in section 5, the interpretation of the results is discussed in section 6, limitations are discussed in section 7 and conclusions are drawn in section 8.

2. Methodology

2.1 Research design

The study applies a design and evaluate approach (first principles specify a framework and then it is implemented and brought to controlled experimentation with a baseline). The independent variables include the privacy budget, drift regime and adaptation strategy; the dependent variables include detection delay, false alarm rate, accuracy recovery/drift and retaining previous-concept performance. This organization allows to assign the observed degradation to certain design parameters, and not to any individual data-set.

2.2 System model and threat assumptions

The system includes k number of clients and one coordinating server. All the clients k have their own non-stationary data set and they are involved in the synchronous training rounds after the federated averaging algorithm process [18]. The server should be honest-but-curious – it performs protocol correctly but tries to make inferences on any received message, given the protocol. The clients are presumed not to collude with the server. A pairwise-masking secure-aggregation protocol ensures that the server can't see the sum of the contributions from the clients; the server can only see them contributing altogether but not individually [19]. Differential privacy is used at the level of the individual client prior to aggregation, so it remains unchanged even if the aggregation protocol gets broken. Three values are sent from the client every round: the up-to-date information that was transmitted, but put through the filter of the "clipped and noised" model, a scalar for the loss if the model change, and a low-dimensional histogram of clients' confidence in the model. The two latter ones form the drift-signalling channel. They don't contribute to their overall privacy cost, which is part and parcel of the model update budget, so there is no space in their privacy promise that isn't somehow lessened by adding monitoring capabilities.

2.3 Framework specification

The framework has a total of four modules, which are shown in Figure 1.

Local signal construction: After local training, the mean prediction loss (MPL) over the client k's past observations and a log of max softmax confidences (MSC) represented as b histogram buckets are computed. After local training, the mean prediction loss (MPL) over the client k's past k observations and a log of max softmax confidences (MSC) represented as b histogram buckets are computed. Both are contained by fixed limits (clipped) and the histogram is normalised so as to have finite sensitivity. Each is

subjected to a different level of Gaussian noise added in, which is proportionate to the sensitivity and the per-round privacy value.

Privacy accounting: Tracking of composition across rounds is done using the Rényi differential privacy accountant [22] which gives almost as tight bounds as the naive sequential composition for the Gaussian mechanism [20] and can be converted to the traditional formulation of Dwork and colleagues [21] for reporting. The entire budget is split between the model update and the signalling channel: Without explicit specification 80% will go towards the update and 20% towards the signalling channel, as determined by preliminary tuning.

Server-side drift detection: The server has an adaptive window over the loss signal (known as the ADWIN procedure), which means that it will have a variable length window and will shorten it when any two sub-windows have means that are statistically different [22]. In aggregated data it includes known noise variance, which causes an overestimate of the detector's confidence parameter, in order to maintain the nominal false alarm rate. A secondary detector which uses the drift detection method of Gama and colleagues presets the confidence histogram for distributional displacement [23] and states a drift if either of the detectors fire, and check the results again in the next round. This two-stage process piggy-backs on the initial alarm, meaning there is some additional delay but the noise gives a solid “noise alarm” in which “noise detected” is non-negotiable.

Continual adaptation: Confirmed, the server starts an adaptation phase. The clients are restarted with an elastic weight consolidation penalty based on the parameters that existed before the drift with weights of the parameters based on estimated Fisher information, whereby parameter directions with more information are stabilised [24]. In parallel, each client keeps in memory a small number of logits from previous rounds, and uses an experience replay consistency term based on dark samples, which prevents storing raw inputs and thus does not create any additional disclosure surface [25]. The relative stability versus plasticity is controlled by a single coefficient, which is determined for each dataset, and is fixed for each privacy setting.

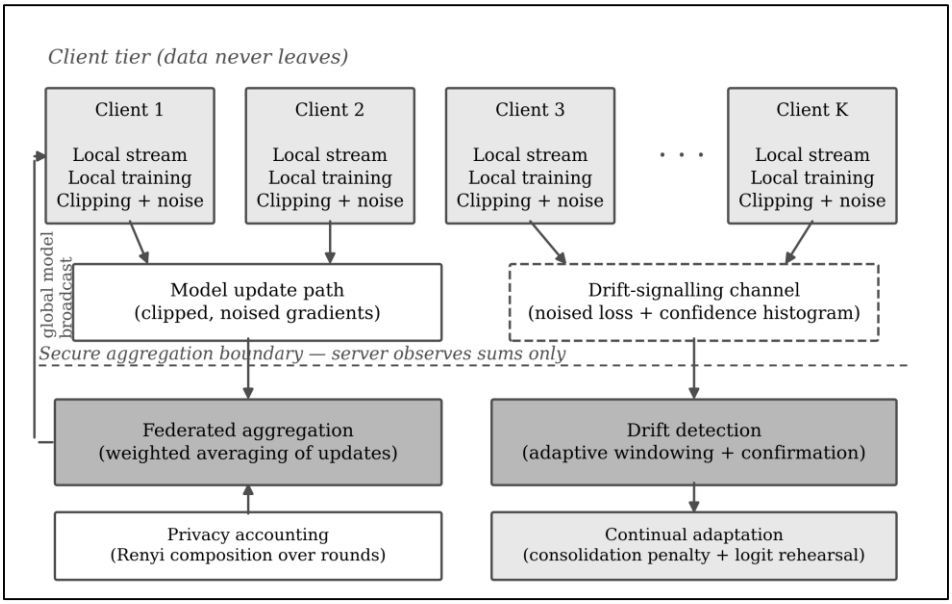

Figure 1. *Architecture of the Privacy-Preserving Continual Drift Adaptation framework, showing the separation between the model update path and the differentially private drift-signalling channel, the secure aggregation boundary, and the server-side detection and adaptation logic.*

2.4 Baselines

There are four comparators implemented. The first of these is a federated averaging without drift handling, which is common practice. The second is federated averaging with unconstrained retraining on detection with unprotected signals which gives the upper bound when there are no privacy or protection restrictions. The third, a federated continual learning approach based on the weighted inter-client transfer formulation [26] was used to transfer parameters selectively among the different clients. The fourth imposes a proximal regularisation term on limit the client divergence, meanwhile not being able to alleviate heterogeneity fully, but not providing any mechanism to fight the drift [27]. All of the baselines run both with and without differential privacy, allowing the marginal effect of differential privacy on the measurement to be isolated [28].

2.5 Datasets and drift injection

Three data sets are used, details in Table 1. The first is the Australian electricity market dataset, which is a well-known stream benchmark, where the price direction is forecasted by determining the demand and transfer variables, and natural drift results from changes in market regulations, and seasonal fluctuations [29]. This represents the second, an artificial stream known as the SEA generator which allows cities to manipulate the class boundary threshold for drift and thus the frequency and intensity of drift [30]. The third is an up-to-date network intrusion dataset with labeled benign and attack traffic from different attack families to enable realistic drift construction by

changing the mix of attack type changes over time [31]. The data is distributed among 50 clients through a Dirichlet allocation with concentration parameter 0.5 which is a common approach to producing realistic label skew [32]. Three drift regimes are constructed. Abrupt drift takes the place of the generating concept on a particular round. Gradual drift is going from one concept to the other in between span of forty rounds. Recurring drift alternates between two concepts used before, with the emphasis on importance placed on the properties of retention for the continual learning component. Drift is also made variable in its extent: the global drift (which affects all clients at once) is different than the partial drift (which affects a random 40 per cent of clients, prescribed by the heterogeneous situation that is found in real deployments and applications).

Table 1. *Characteristics of the three evaluation datasets and their associated drift constructions.*

Dataset	Records	Features	Task	Drift origin	Clients	Rounds
Electricity market	45,312	8	Binary	Natural; market and seasonal	50	300
SEA generator	100,000	3	Binary	Synthetic; threshold change	50	300
Network intrusion	225,745	78	Multi-class (6)	Semi-synthetic; attack reweighting	50	400

2.6 Evaluation metrics

There will be five metrics reported. Detection delay is the number of rounds required between the true drift point and when it is detected, and false alarm rate the percentage of stationary rounds which are incorrectly reported as drift. Post-drift accuracy is the average of the precision (accuracy) of 20 shots after adaptation. Retention is able to measure the accuracy of a held-out set from the pre-drift version of the concept after adaptation, which quantifies catastrophic forgetting; backward transfer measures the net reduction in the accuracy of pre-drift versions of concepts when evaluated on held-out sets that they have learned later in training, with negative scores representing forgetting.

2.7 Implementation and statistical procedure

Models are 3-layer multilayer perceptrons that prevent architecture from taking the place of protocol effects. The stream simulation and baseline detectors are based on the conventions of the massive online analysis framework [33]. Numbers given are means and 95% confidence interval for 10 seeds per configuration. Paired t tests with Holm–Bonferroni correction and Cohen's d values are used for comparisons and Cohen's d is reported if the difference is significant.

3. Literature Review

3.1 Privacy mechanisms in distributed learning

The initial attacks were surveyed and became known as inference on aggregation server and poisoning by malicious clients and leakage due to repeated participation as main attack vectors [34]. Later, defensive structures were classified into three different categories: cryptographic, perturbation and hybrid, and each type had an intermediate spot on the axes of computational cost, communication overhead and utility loss [35]. The most widely used perturbation is that of differential privacy; particular attention has been paid to local and central trusted models for adapting its use from database releasing to federated learning [36] and to the adaptation of local vs central trusted models [36, 38, 44]. Modest noise on the other hand in the context of secure multiparty computation results in other schemes that limit the utility loss that occurs if the data are randomly altered without additional post-processing [37] and later empirical results showed that the level of accuracy loss measured in the context of poisoning attacks due to local differential privacy is stronger than other methods that do not incur noise [38]. Since then, practitioner guidance has coalesced around parameter selection and reporting, one of the most common of which is that the budgets are often published without the parameters in the auxiliary files that are required to interpret the budget. The need for such defenses, along with a host of attack research, furthers them. Membership inference and property inference/reconstruction are two different forms of threats and countermeasures that are explicitly listed as such in comprehensive writings such as privacy and robustness in federated learning [40]. Recent research has shown that a server that is unwilling to honor this aspect of the protocol could manipulate the model parameters so that it is possible for nearly perfect reconstruction of the inputs at the client to be achieved without secure aggregation, and therefore without server cooperation [41]. The essential idea behind a shared gradient leaking information is not specific to the federated framework, but rather comes before it, and is the basis for subsequent theoretical analyses in which members of the data set were proven to be learnable even when only exposed to a subset of representations in the learning task [43]; in addition, the foundational study of white-box analyses showed that even under

standard training setups, deeply parameterized models were vulnerable to inference across membership in the cohort, for properties not included in the learning task [44]. All of this literature and this position taken here can be summarized with the notion that any statistic derived from a client (as this includes a rate drift signal) is sensitive.

3.2 Heterogeneity and optimisation in federated systems

In addition to drift, there is a source of difficulty, namely, differences in the data across clients which are independent of drift. The survey of non-identically distributed federated learning mentions aggregation losses, client convergence in the optimization sense, and instability in the convergence [45] propelling the conceptual applications of the two as intertwined and not separable issues [46]. Algorithmic are variance reduced (VR) local updates to correct client gradient bias [47] and adaptive server side optimisers [48]. In a different way to talk about personalisation, it is not the goal of a single, global model, but rather the approach of creating specific different models to meet specific communications goals, such as the goal of addressing statistical, systemic and architectural types of heterogeneity in the field [49] [50]. The difference is important because all signal which is aggregated may be misinterpreted as continuous heterogeneity and the experimental design is set in accordance with that.

3.3 Concept drift detection and adaptation

Foundational surveys established the drift taxonomy and the canonical detector families, distinguishing error-rate-based, distribution-based and ensemble-based approaches [4]. Reviews from the current decade refine this picture: one influential treatment reframes detection in terms of downstream model degradation rather than distributional change per se, arguing that shifts leaving the decision boundary intact should not trigger costly adaptation [51]. Complementary reviews consolidate the detector landscape and highlight the difficulty of evaluation under unlabelled or delayed-label conditions [52], while dedicated surveys of recurring drift examine concept storage and reuse strategies that avoid relearning previously encountered states [53]. Theoretical work questions whether drift is well posed absent a specified learning task, clarifying when detection is meaningful [54].

Extension to distributed settings is comparatively recent. One line of work adapts detection to federated networked systems, showing that aggregated performance signals can localise drift to particular network segments [56]. Another treats drift as a client-clustering problem, maintaining multiple global models and assigning clients by the concept they currently exhibit, thereby accommodating drift that affects only part of the population [8]. Adaptive schemes modifying client participation and learning rates improve recovery speed without architectural change [17], and routing inference to the most appropriate of several models addresses drift at serving rather than training time

[55]. These contributions share the assumption that the detection signal may be transmitted unprotected, an assumption the present work relaxes.

3.4 Continual learning under distribution

All continuous learning surveys focus on the three-fold taxonomy of regularisation, rehearsal and parameter isolation [5,6]. Recent evaluations suggest that the field has a focus on benchmarks that has generated techniques that are inappropriate for deployment, which should be assessed instead against memory, compute and privacy limits [57]. The most challenging formulation is class-incremental learning, for which a lot has been said, and generally exemplar retention is the most effective intervention [58,78]; this is in danger of conflicting with concerns about privacy, since exemplars of information are the information themselves. Using selection strategies to maximise buffer occupancy by gradient diversity does not reduce disclosure risk, but it does reduce storage cost [59]. There are ways in which we tackle both points in federated continual learning. In [26] the weighted inter-client transfer formulation breaks down the parameters into a global, a task-adaptive, and a client-specific contribution, thus allowing to select selectively transfer without interference. Class-aware gradient compensation [60] and relation distillation, exemplar-free distillation through synthesised data [61] and generating reconstructed prior-task representations [62] have all been explored to address federated class-incremental learning (CIL). A recently published work bundles all these advancements under the topic of knowledge fusion [63]: distribution of only reliable parts of a model's knowledge improves its utility and privacy [64]. The current approach is to build around it but with different focus: not to determine a series of prescribed activities, but simply identify (at the tested levels of privacy) that the task has changed at all.

3.5 Application domains and residual gaps

Distributed learning has essentially been implemented most quickly at places where one has the most sensitive data. The clinical applications are well covered and regulations are considered as well as failure modes of medical imaging pipelines [65,75,77]. Cybersecurity also has a strong appearance as the cross-organisational nature of intrusion detection is negated by the need to disclose other applications while its requirements are limited by resource (privacy) and adaptation (cost). It would seem that, as pointed out by more general surveys, most deployments make the assumption of stationarity which is not often justified by the environments described [69]. Weighted schemes are a core method in aggregation and have been extensively treated systemically [73,74], various decentralized variants of aggregation that lack a coordinating server have been considered [76], and aggregation methods which consider generalisation, robustness and fairness are benchmarked [72]. The lack of a multi-label setting makes multi-label recognition with temporal non-stationarity a general open problem, whereas delayed labels or missing labels necessitate further

processing, which constitutes a bigger challenge since it eliminates the error signal that most detectors depend on [71, 70]. This article is therefore focused on addressing the gap. The literature for privacy covers the model updates, but has not addressed the monitoring of signals; Literature on drift assumes that one can obtain access to the signals, and literature on federated continual learning assumed that a new concept is not a matter of inference, but known. It has been shown that, so far, there is no treatment that measures the value of drift responsiveness that is put on offer for a unit (or a level) of formal privacy guarantee.

4. Problem Statement

Suppose that there are k different clients, denoted k, with different joint distributions from which k labelled pairs are streamed as a consequence of a round, t, of the experiment. For client k, concept drift happens at time t = t_{k} if the conditional distribution of the label, given the input is different from the previous round, as long as there is no change in the input distribution. Such real drift changes the ideal decision boundary while such virtual drift, which changes only the input marginal does not necessarily affect predictive performance, and also does not need adaptation [51]. At every round, the coordinating server needs to determine if changes should be made to the system. Only aggregated messages are available for use that have a differential privacy guarantee for any individual record of a client. Formally, the mechanism for generating the round-t message should have the property that, if the data used by all the clients is either a or b for the two nearest datasets, then the ratio of the probability of the round-t message that it would originate from dataset a to its probability that it would originate from dataset b is within a factor of the exponentially-neighbouring neighbourhood of the privacy parameter, or additive failure allowance. The detection problem is thus to build the detection rule over a time series of noisy aggregates to have minimal expected delay while containing the false alarm rate within a set constraint given that the amount of noise governed by the privacy constraint is not governed by the estimation problem. There are three properties to make this formulation hard. Enforcing privacy increases adversarially the variance of the quantity being monitored: tighter privacy leads to a greater variance that corresponds to a greater variation of SNR. While every human implements a real drift of unknown magnitude, adversarially, and in a way that increases the variance when the privacy is tightened. The budget is fixed, and will be used monotonically, so the more often a detector queries the signal the less noisy the observations will be per query; there is an optimum monitoring frequency, which is not necessarily the maximum! Costs of adaptation: once the false alarm begins the retraining process, uses up communication and compute, can decrease performance on the correct, remaining concepts if consolidation anchors are changed early. The cost function is thus not symmetric or sequence-independent provided no change points are

present or happen at the same point in the sequence, as typical change-point formulations aren't able to handle. The present study tackles three research questions formulated from the above formulation. How close and loose can a formal privacy guarantee be on a drift-signalling channel to an unprotected channel in terms of (signal) Detection Delay & False Alarm rate? Can a (normalised) rehearsal process that does not store raw exemplars offer a satisfactory degree of retention to justify privacy constrained adaptation? How much privacy users could spend at the framework before becoming an ineffective and unfunded operation, defined as having its potential detection performance at or above chance?

5. Results

5.1 Detection performance across privacy budgets

The non-linearity is conspicuous as the relationship between detection delay and shorten privacy budget is still monotonically increasing. Figure 2 shows mean detection delay and false alarm rate as a function of the total privacy budget for abrupt global drift on the network intrusion dataset. The effect of the delay on budgets ten to three is rather inconsequential, adding about forty-five per cent to become 4.2 to 6.1 rounds, which is still acceptably small for most deployments. At a budget less than one, the behaviour takes more than 19.4 rounds to respond and the false alarm rate climbs to 0.148, about one false alarm every seven rounds of the detector. These alarms are ignored by the two-stage confirmation procedure, which causes the alarm rate to undergo a sharp spike when noise variance gets close to the mean loss due to drift, but will not recover the underlying signal if this happens.

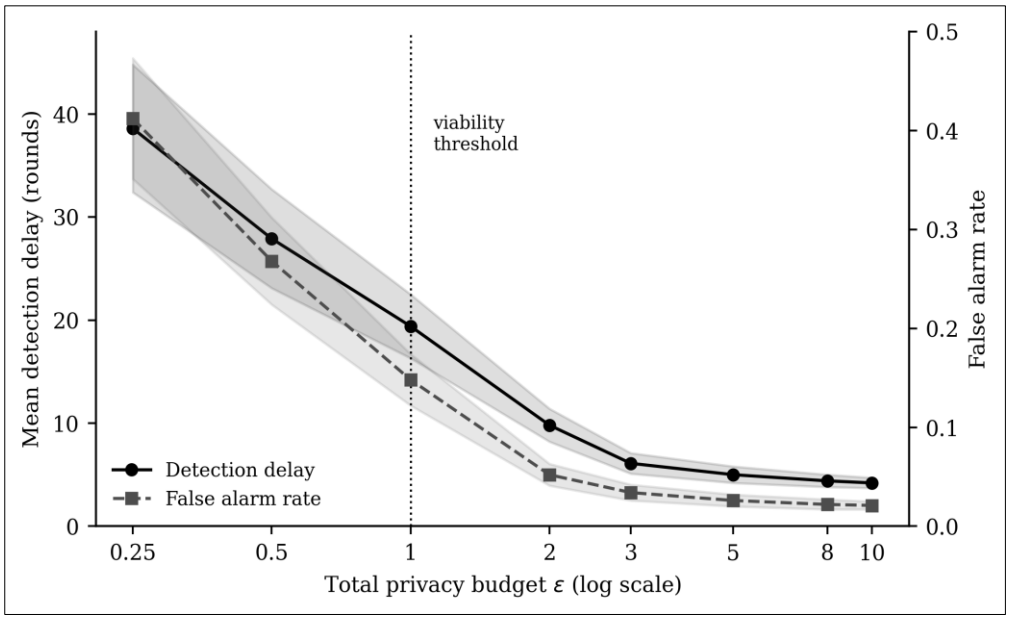

Figure 2. *Mean drift detection delay (rounds, left axis) and false alarm rate (right axis) as a function of the total privacy budget, for abrupt global drift on the network intrusion dataset. Shaded bands indicate ninety-five per cent confidence intervals across ten seeds. The vertical reference line marks the empirical viability threshold.*

The drift regime has a significant impact on this picture. The abrupt drift is a big sudden drop in aggregate loss, and also the farthest from noise. Distributing the same total change over 40 rounds is significantly more difficult: the mean delay over 10 seeds for the delay of unity is 14.7 rounds versus 6.1 for the abrupt delay. Recurring drift falls somewhere between; an average of 2.3 rounds are shorter to detect the second and subsequent occurrences, due to the fact that a consolidated parameter from the previous occurrence yields a sharper loss response. Partial drift, which occurs in 40 per cent of clients, can only be detected reliably if the budget exceeds 5: the greater the proportion of clients that experience partial drift, the weaker the affected subpopulation's signal is in the overall sample.

5.2 Adaptation quality and retention

Table 2 presents the mean accuracy, retention, and backward transfer at a set privacy budget of five for the proposed framework along with the four baselines, for the three datasets tested, in the recurring drift setup. Without any drift handling, the federated averaging configuration will not update much and hence preserves the accuracy of the previous concept but it has the lowest accuracy in post-drift scenarios. When no constraints are imposed on retraining, the best accuracy after the drift task is obtained, yet the worst retention score is also obtained, in such a way that there is a backward transfer of −14.6 percentage points. The proposed framework recovers 92.4 per cent of

the accuracy of the unconstrained retraining method after the drift, and reduces forgetting by over 66%.

Table 2. *Post-drift accuracy, prior-concept retention and backward transfer under recurring drift at a privacy budget of five, averaged across three datasets over ten seeds. Values in parentheses are ninety-five per cent confidence intervals.*

Method	Post-drift accuracy (%)	Retention (%)	Backward transfer (pp)	Detection delay
Federated averaging, no adaptation	71.3 (±1.4)	88.1 (±0.9)	−2.1 (±0.6)	not applicable
Unconstrained retraining, no privacy	89.7 (±1.1)	74.2 (±1.7)	−14.6 (±1.3)	3.4 (±0.5)
Weighted inter-client transfer	84.2 (±1.6)	83.9 (±1.2)	−6.8 (±1.1)	8.9 (±1.4)
Proximal regularisation	79.6 (±1.8)	85.4 (±1.0)	−4.9 (±0.9)	10.2 (±1.7)
Proposed framework	82.9 (±1.3)	86.7 (±1.1)	−4.2 (±0.8)	7.3 (±1.1)

Differences between the proposed framework and the proximal baseline in post-drift accuracy are significant ($p = 0.003$, Cohen’s $d = 1.12$), as are the retention differences from unconstrained retraining ($p < 0.001$, $d = 1.84$). The comparison with weighted inter-client transfer is more nuanced: that method attains marginally higher post-drift accuracy but responds more slowly, lacking an explicit detection mechanism and reacting only once degradation becomes apparent through the loss of transfer benefit.

Figure 3 traces accuracy over rounds for a representative run on the SEA generator under recurring drift. All adaptive methods decline sharply at the drift point and then recover; the differences lie in the depth of the trough, the speed of recovery and, most tellingly, the accuracy achieved when the original concept returns at round 200. Methods without consolidation relearn that concept from scratch, whereas the proposed framework recovers it within four rounds.

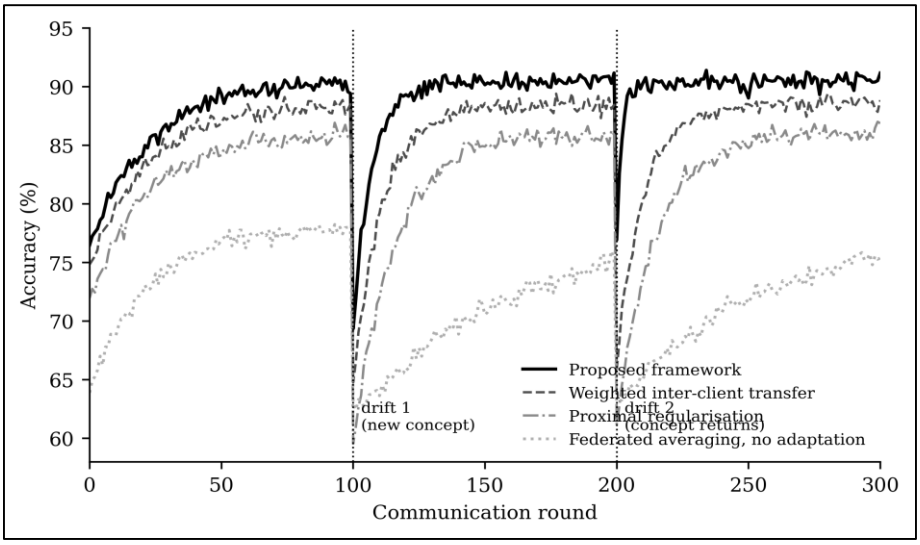

Figure 3. *Accuracy trajectory over 300 communication rounds on the SEA generator under recurring drift, comparing the proposed framework with three baselines. Drift points occur at rounds 100 and 200; the second transition returns the stream to the original concept.*

5.3 Budget allocation between model and signal

A unique design conundrum is the partitioning of the overall privacy budget into that used for the model update and that for the drift signal. We now report the results of changing this split in the electricity dataset while keeping the total budget constant at 5.Now we report the results of varying this split, when the overall budget remains the same (which is always 5 in total). If too little is allocated to the signal, detection will be slow, and if too much is allocated to the signal, the signal will degrade as a result of the adaptation itself, even if it has been done in the correct time frame. The best value is 15–25 per cent and the response surface is quite level at those levels; this is also good for practitioners because tight control is not necessary.

Table 3. *Effect of privacy budget allocation between model updates and the drift-signalling channel, at a total budget of five on the electricity dataset.*

Signal allocation (%)	Detection delay (rounds)	False alarm rate	Post-drift accuracy (%)	Steady-state accuracy (%)
5	16.8	0.019	74.1	81.9
10	11.2	0.024	77.6	81.6
20	7.9	0.031	80.3	80.8
30	7.1	0.048	79.4	78.9
50	6.8	0.087	75.2	74.3

The results also show an interaction between client participation. If less than the total number of clients are involved in a particular round the effective noise on the aggregate will rise because the number of independent perturbations that will be averaged is reduced. This change in participation rate is important because it affects the threshold of the viable budget, which moves from about unity to about 2.5 when going from twenty percent to twenty-five percent[68].

5.4 Regime comparison and statistical validation

The results of the above tests are combined and presented together in figure 4. The difference in regime and budget is directly comparable between the two panels of (a) and (b) with panels (a) and (c) reporting detection delay over a range of four drift constructions and five privacy budgets; panel (a) shows the steepest relation between delay and budget, dropping the detection delay by a gradient of 48.0 rounds at budget 0.5 and a gradient of 10.8 rounds at budget 10 when compared in abrupt global drift, a range more than twice as great. A zero-cost selecting-for-stability choice involves high amounts of baseline retention but low plating rates, while the opposite extreme involves high amount of learning while being highly constrained (see panel (b) below), it isn't possible for neither of these extremes to happen in the proposed framework, and it's precisely at that point that the aggregates identified in table 2 get in the way.

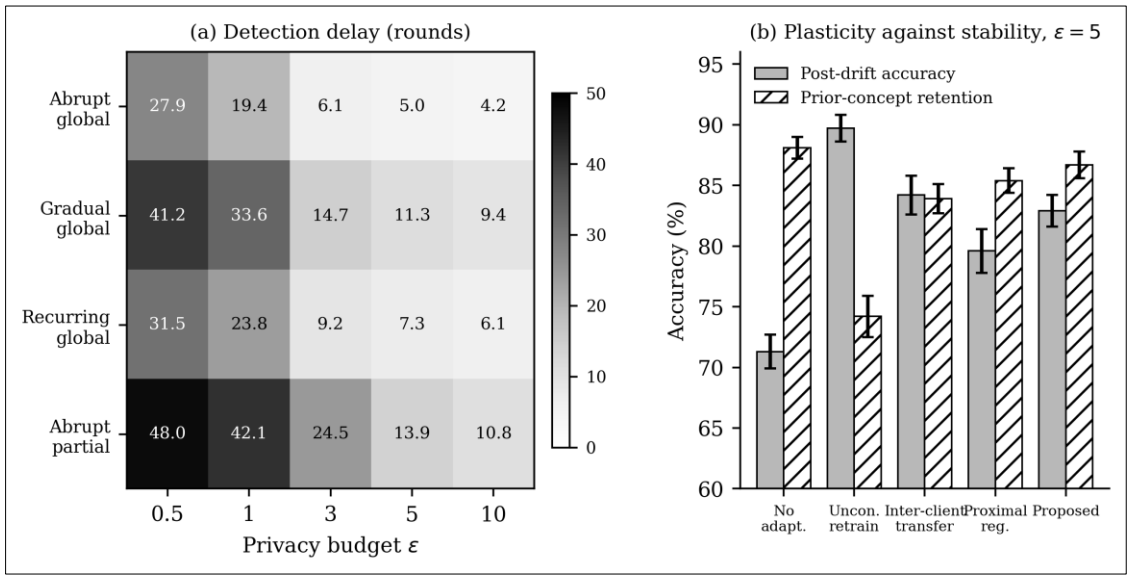

Figure 4. *(a) Mean detection delay in rounds across four drift constructions and five privacy budgets; darker cells denote longer delay. (b) Post-drift accuracy and prior-concept retention for the five configurations at a privacy budget of five, with error bars showing ninety-five per cent confidence intervals across ten seeds.*

Formal comparisons supporting these observations are reported in Table 4, and they are grouped together on a seed-by-seed (Holm–Bonferroni) basis using the six contrasts.

Results are clear and distinctly better than the non-adaptive baseline both on the post-drift and retention accuracy component of the framework, and its advantage over unconstrained retraining is also large. The difference in its post drift accuracy compared to unconstrained retraining was also significant, being about what's to be expected in the accuracy price of restraining the update; but when there is recurring drift, it is clearly an acceptable price for retaining 12.5 points of accuracy. The biggest impact in the study comes from the budget contrast, with an interpretation of the viability threshold as a regime boundary appearing more acceptable than a point on a smooth curve.

Table 4. *Paired comparisons across ten seeds under recurring drift, with Holm–Bonferroni correction applied across six contrasts. Positive differences favour the first-named condition.*

Contrast	Metric	Mean diff.	95% CI	t(9)	p (adj.)	d
Proposed vs. no adaptation	Post-drift accuracy	+11.6 pp	[9.4, 13.8]	11.84	<0.001	2.41
Proposed vs. unconstrained retraining	Retention	+12.5 pp	[10.1, 14.9]	11.21	<0.001	1.84
Proposed vs. unconstrained retraining	Post-drift accuracy	-6.8 pp	[-8.5, -5.1]	-8.94	<0.001	1.97
Proposed vs. proximal regularisation	Post-drift accuracy	+3.3 pp	[1.5, 5.1]	4.06	0.003	1.12
Proposed vs. inter-client transfer	Detection delay	-1.6 rounds	[-2.8, -0.4]	-2.98	0.016	0.94
Budget 5 vs. budget 1	Detection delay	-14.4 rounds	[-17.1, -11.7]	12.06	<0.001	3.10

6. Main Discussion

6.1 Interpreting the privacy–responsiveness exchange

The main empirical result is that there is no simple trade-off between privacy and drift responsiveness, but more of a regime change. Within medium budgets, the cost of privacy is an incremental and linear extension in detection delay which is suitable in a system that is retrained daily or weekly. When the participation budget is crossed (near unity) the system is not degraded gracefully, but it fails, apparently driving unwanted adaptations which are detrimental since they reset the consolidation anchor and result in forgetting. Finally, instead of having a privacy parameter as one would expect to dial up and down in reaction to legal or reputational pressures, a deployment should be regarded as periodic retraining of the system on a schedule rather than an adaptive

maintenance of the system. This is in line with an assumption commonly used in the applied literature, which often suggests a budget of some kind without adhering to any monitoring functions that may be required by a system [39]. It might be possible to provide enough funding for training, but a lack of funding for detection may not be considered sufficient, as detection requires some criterion for signal-to-noise, whereas training requires only something like the average gradient is in the right general direction.

6.2 The value of signal separation

To treat the drift signal as a separate channel that is allocated with its own budget is beneficial in two ways: It is to treat the drift signal as a separate channel by having its own budget allocation because of the 2 following points: The first is accounting transparency – in that, because the sensitivity of the signal is limited by what its built is, the cost of this signal's privacy can be calculated exactly, instead of being limited in a loose fashion. Both are shown in Table 3; the second is allocative flexibility. Methods that account for drift based on divergence of updates combine 3 different types of divergence: real drift, persistent client divergence and stochastic divergence, and don't differentiate between them without extra assumptions [46]. The dedicated channel is not heterogeneous as it is not used for monitoring a disagreement between clients but rather a change in an aggregate over time. The cost of the design decision should be clearly expressed: the cost of each round is budget, and this cost doesn't pay off, in continuous models, with any improvement in the model, so it reduces the overall learning that can be achieved within a lifetime budget. On one hand, the results are suggestive of this being defensible because otherwise they would follow a model with a confidence in an outdated concept to perform and consequently place themselves in an uncomfortable position; on the contrary side they will differ above all when the penalty for outdated predictions is not so marked.

6.3 Retention without exemplars

As such, the logit-level rehearsal retains memory nearly as well as exemplar storage is found in the literature thus far on class-incremental there have been numerous instances of exemplar retention proving to be the best single intervention [58,78], an obstacle where access to raw observations is blocked. Results are consistent with new exemplar-free, data-free federated learning strategies [61,62] for rehearsal, indicating that it is mainly a constraint on function, rather than providing information to store as memories. Most of the benefit will be maintained when the outputs (and not the inputs) are stored, and the disclosure of raw data surface reduces. Unfortunately, neither are there guarantees to eliminate risk, as there is distributional information that is contained in logits and is also not removed from the accounting of the privacy dimension here.

6.4 Partial drift and the aggregation problem

It is arguably the most significant detrimental effect of degraded detection which is a partial drift. Real systems don’t have uniform drift: a region-wide regulatory change applies only to a subset of systems; an adversary is deployed in the area, or the systems have an update to their firmware, that affects only a subset of systems; aggregation, the mechanism that provides privacy amplification, is what reduces the power of the subset. The tension is overt and not secondary. Clustering methods help combat it by using multiple models and clients to be distributed across the various models [8]; however in order to perform assignment, per-client signals must be present that are informative enough to reintroduce the disclosure risk aggregation was designed to mitigate. A possible solution is hierarchical aggregation which involves grouping the clients at regional or organizational level, and monitoring the drift at the group level in exchange for weaker amplification but stronger localisation. This is not assessed here, and is the most promising continuation.

6.5 Implications for applied deployments

These findings have different applicability for the different sectors. Coherts in clinical environments change rather slowly and the training period is long, and thus, penalty as a result of the delay observed at moderate budgets does not appear to be a concern and privacy as a strong posture cannot be compromised [65,75]. The calculus of intrusion detection is reversed: adversarialdrift is deliberate and it's fast; a twenty-round delay might correspond to the success of an attack; and below three budgets are not suitable for this application [66,67]. In between is education analytics. As cohorts, curricula and delivery modes shift, and as the institutions undergoing the change are often not allowed to pool student data, systems supporting administrative decision making and instructional planning must be valid as cohorts, for example [14,15]. The moderate-budget regime described here seems to be appropriate for this context because on the one hand there are academic terms during which the regular budget can be adapted, and on the other hand the impact of a few days' delay is relatively small. Distributed verification infrastructure is related in this way; as participation patterns have been changing, maintaining accurate models of the legitimate participation in the system is critical to the integrity of the system, and without centralisation of the records of participants.

7. Challenges and Limitations

7.1 Methodological limitations

The generalities of these findings were subject to a few limitations. The evaluation follows a multi-layer perceptrons approach with tabular streams, while excluding any architectural confounds while preserving the possibility to explain the results for models with larger capacities on unstructured data, which would push up the viability threshold as more noise was expected with these larger models. The drift constructions cover the mainstream taxonomy but vary in their natural occurrence and lack of control or their construction, which is artificial and simplified; these do not fully reflect the compound type drifts encountered in production. The threat model is based on the assumption that there are an honest-but-curious server and non-colluding clients.

In adversarial deployments it may be optimistic, as evidence shows a malicious server can break the secure aggregation with model parameters.In adversarial deployments, it may be optimistic as it is shown that the secure aggregation can be broken by a malicious attacking server with model parameters. No countermeasure against poisoning is included and there is a denial-of-service attack concept that requires special study, 'spurious adaptation' [40] that can be induced by a malicious client changing the drift signal. Last but not least, the study assumes that labels are available with the client, very soon after the prediction. This is a usual assumption in the drift literature but it is often not met in reality: labels are late, partly or completely absent [71]. This error based signal is not present under delayed labelling and it has to be detected by input distribution monitoring where virtual drift is detected regardless of the actual signal being present resulting in higher false alarm rate.

7.2 Systemic and practical challenges

There are a number of challenges which are intrinsic in the problem itself and not in this study. Exhaustion of budgets over a prolonged implementations lifetime is still not addressed: after continuous monitoring during years, a system undergoes privacy loss over time and there are no principles to suggest periodic renewals should be due to data turnover, nor because it is weaker, how often. This is a restriction that is overcome by decentralised aggregation architectures [76]. The other crucial verification problem is also a formidable one. A client is not easily able to verify that the server is following its budget as stated, and server is not easily able to verify that its client is applying the noise as stated. The area of cryptographic attestation of correct noise generation; it makes the protocol protocol communication intensive in addition to it being an active topic [73]. Both fairness and privacy interact with drift, amplifying the effects of enhanced fairness. The cost is even worse for sub-populations that are involved in the data with a low frequency [13] and drift adaptation is the one that will benefit the "big dog" as the aggregate signal is most dominated by the largest group of concepts. This situation

might lead one to believe that the minority group that is drifting would be doubly disadvantaged – hearing updates that are of lesser quality, and adapted to more slowly. This interaction has begun to emerge in practice as benchmarks that target generalisation, robustness and fairness together that apply in a federated setting [72], but there are very few studies focused on the generalisation robustness and fairness aspects of non-stationarity, especially regarding the detection delay at the subgroup level, which is not measured in the present study. The overhead of the framework is also significant - secure aggregation is effected by the number of clients; consolidator needs Fisher information estimates and reference parameters for each individual client; and rehearsal reservoir uses memory. These costs can also be impractical on edge devices with limited resources, and solutions for aggregating and compressing data provide only a partial remedy at best [74].

8. Conclusion

In this article, we see a problem emerging at the intersection of three active research areas that, while it can be solved to some degree in each of the areas alone, is addressed adequately in none.If a distributed learning system is not allowed to look at its data or any unprotected statistic derived from this data, how does it know if its data have changed? In the proposed framework detection is a channel design problem, and a bounded-sensitivity signal is securely aggregated, the consolidation/rehearsal process is adaptive, and monitoring is by means of a adaptive windowing detector. The evaluation results in three important findings to be of practical value. First, drift detection with enough formal privacy guarantees can be done at moderate budgets and at budgets below about unity there will be a detectable increase in detection delay, but there will not be a gradual loss of privacy; the loss of privacy when the budget is below about unity is abrupt, and at these budgets systems should not be called drift-aware. Second, there is a way to retain previous concepts, which does not involve holding on to raw exemplars, that recovers most of the gains of being able to store raw exemplars, without the disclosure surface that they bring. Third, aggregation does a poor job at detecting drift in the subset of clients, this is true for partial drift, and this is structural: they are the same mechanism that is used to achieve privacy amplification that is used to diminish the localised signal. Three directions follow. It is possible to see hierarchical aggregation as a systematic way of compromising between amplification and localisation, and it is a direct extension. But there is almost no work done in detecting it under the absence of marking or delayed marking scenario, and the need for detection exists when ground truth is costly or time-consuming. Empirically, there might be fairness concerns at the subgroup level which could be determined through subgroup level analysis. Improvements on these would make distributed systems closer to being both lawful and accurate, as the world of which they are an instance evolves.

References

1. Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., et al. (2021) 'Advances and open problems in federated learning', *Foundations and Trends in Machine Learning*, 14(1–2), pp. 1–210.
2. Nguyen, D.C., Ding, M., Pathirana, P.N., Seneviratne, A., Li, J. and Poor, H.V. (2021) 'Federated learning for internet of things: a comprehensive survey', *IEEE Communications Surveys and Tutorials*, 23(3), pp. 1622–1658.
3. Lu, J., Liu, A., Dong, F., Gu, F., Gama, J. and Zhang, G. (2019) 'Learning under concept drift: a review', *IEEE Transactions on Knowledge and Data Engineering*, 31(12), pp. 2346–2363.
4. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M. and Bouchachia, A. (2014) 'A survey on concept drift adaptation', *ACM Computing Surveys*, 46(4), pp. 1–37.
5. De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G. and Tuytelaars, T. (2022) 'A continual learning survey: defying forgetting in classification tasks', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7), pp. 3366–3385.
6. Wang, L., Zhang, X., Su, H. and Zhu, J. (2024) 'A comprehensive survey of continual learning: theory, method and application', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8), pp. 5362–5383.
7. Casado, F.E., Lema, D., Criado, M.F., Iglesias, R., Regueiro, C.V. and Barro, S. (2022) 'Concept drift detection and adaptation for federated and continual learning', *Multimedia Tools and Applications*, 81(3), pp. 3397–3419.
8. Jothimurugesan, E., Hsieh, K., Wang, J., Joshi, G. and Gibbons, P.B. (2023) 'Federated learning under distributed concept drift', *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, PMLR 206, pp. 5834–5853.
9. Zhu, L., Liu, Z. and Han, S. (2019) 'Deep leakage from gradients', *Advances in Neural Information Processing Systems*, 32, pp. 14747–14756.
10. Geiping, J., Bauermeister, H., Dröge, H. and Moeller, M. (2020) 'Inverting gradients: how easy is it to break privacy in federated learning?', *Advances in Neural Information Processing Systems*, 33, pp. 16937–16947.
11. Hatamizadeh, A., Yin, H., Molchanov, P., Myronenko, A., Li, W., Dogra, P., et al. (2023) 'Do gradient inversion attacks make federated learning unsafe?', *IEEE Transactions on Medical Imaging*, 42(7), pp. 2044–2056.
12. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K. and Zhang, L. (2016) 'Deep learning with differential privacy', *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318.
13. Bagdasaryan, E., Poursaeed, O. and Shmatikov, V. (2019) 'Differential privacy has disparate impact on model accuracy', *Advances in Neural Information Processing Systems*, 32, pp. 15479–15488.
14. Sharwani, A.E., Melo, R. and Rabbi, I. (2025) 'The role of artificial intelligence in education: transforming administration and leadership for revolutionising education', *Adhyayan: A Journal of Management Sciences*, 15(1), pp. 10–16.

15. Sharwani, A.E. (2024) 'Modernising the US educational system by elevating teaching methods and student performance through human–computer integration, data analytics, and other innovative technologies', *2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, 2, pp. 1–6.
16. Melo, R. and Sharwani, A.E. (2024) 'Blockchain-enabled academic credential verification in the USA', *Adhyayan: A Journal of Management Sciences*.
17. Panchal, K., Choudhary, S., Mitra, S., Mukherjee, K., Sarkhel, S., Mitra, S. and Guan, H. (2023) 'Flash: concept drift adaptation in federated learning', *Proceedings of the 40th International Conference on Machine Learning*, PMLR 202, pp. 26931–26962.
18. McMahan, H.B., Moore, E., Ramage, D., Hampson, S. and Arcas, B.A. (2017) 'Communication-efficient learning of deep networks from decentralized data', *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, PMLR 54, pp. 1273–1282.
19. Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H.B., Patel, S., Ramage, D., Segal, A. and Seth, K. (2017) 'Practical secure aggregation for privacy-preserving machine learning', *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191.
20. Mironov, I. (2017) 'Rényi differential privacy', *2017 IEEE 30th Computer Security Foundations Symposium*, pp. 263–275.
21. Dwork, C., McSherry, F., Nissim, K. and Smith, A. (2006) 'Calibrating noise to sensitivity in private data analysis', *Theory of Cryptography Conference*, pp. 265–284.
22. Bifet, A. and Gavaldà, R. (2007) 'Learning from time-changing data with adaptive windowing', *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 443–448.
23. Gama, J., Medas, P., Castillo, G. and Rodrigues, P. (2004) 'Learning with drift detection', *Advances in Artificial Intelligence – SBIA 2004*, pp. 286–295.
24. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., et al. (2017) 'Overcoming catastrophic forgetting in neural networks', *Proceedings of the National Academy of Sciences*, 114(13), pp. 3521–3526.
25. Buzzega, P., Boschini, M., Porrello, A., Abati, D. and Calderara, S. (2020) 'Dark experience for general continual learning: a strong, simple baseline', *Advances in Neural Information Processing Systems*, 33, pp. 15920–15930.
26. Yoon, J., Jeong, W., Lee, G., Yang, E. and Hwang, S.J. (2021) 'Federated continual learning with weighted inter-client transfer', *Proceedings of the 38th International Conference on Machine Learning*, PMLR 139, pp. 12073–12086.
27. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A. and Smith, V. (2020) 'Federated optimization in heterogeneous networks', *Proceedings of Machine Learning and Systems*, 2, pp. 429–450.
28. Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q.S. and Poor, H.V. (2020) 'Federated learning with differential privacy: algorithms and performance analysis', *IEEE Transactions on Information Forensics and Security*, 15, pp. 3454–3469.

29. Harries, M. (1999) *SPLICE-2 comparative evaluation: electricity pricing*. Technical Report. Sydney: University of New South Wales.
30. Street, W.N. and Kim, Y. (2001) 'A streaming ensemble algorithm (SEA) for large-scale classification', *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 377–382.
31. Sharafaldin, I., Lashkari, A.H. and Ghorbani, A.A. (2018) 'Toward generating a new intrusion detection dataset and intrusion traffic characterization', *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, pp. 108–116.
32. Hsu, T.M.H., Qi, H. and Brown, M. (2019) 'Measuring the effects of non-identical data distribution for federated visual classification', *arXiv preprint*, arXiv:1909.06335.
33. Bifet, A., Holmes, G., Kirkby, R. and Pfahringer, B. (2010) 'MOA: massive online analysis', *Journal of Machine Learning Research*, 11, pp. 1601–1604.
34. Mothukuri, V., Parizi, R.M., Pouriyeh, S., Huang, Y., Dehghantanha, A. and Srivastava, G. (2021) 'A survey on security and privacy of federated learning', *Future Generation Computer Systems*, 115, pp. 619–640.
35. Yin, X., Zhu, Y. and Hu, J. (2021) 'A comprehensive survey of privacy-preserving federated learning: a taxonomy, review, and future directions', *ACM Computing Surveys*, 54(6), pp. 1–36.
36. El Ouadrhiri, A. and Abdelhadi, A. (2022) 'Differential privacy for deep and federated learning: a survey', *IEEE Access*, 10, pp. 22359–22380.
37. Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R. and Zhou, Y. (2019) 'A hybrid approach to privacy-preserving federated learning', *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pp. 1–11.
38. Naseri, M., Hayes, J. and De Cristofaro, E. (2022) 'Local and central differential privacy for robustness and privacy in federated learning', *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.
39. Ponomareva, N., Hazimeh, H., Kurakin, A., Xu, Z., Denison, C., McMahan, H.B., Vassilvitskii, S., Chien, S. and Thakurta, A.G. (2023) 'How to DP-fy ML: a practical guide to machine learning with differential privacy', *Journal of Artificial Intelligence Research*, 77, pp. 1113–1201.
40. Lyu, L., Yu, H., Ma, X., Chen, C., Sun, L., Zhao, J., Yang, Q. and Yu, P.S. (2024) 'Privacy and robustness in federated learning: attacks and defenses', *IEEE Transactions on Neural Networks and Learning Systems*, 35(7), pp. 8726–8746.
41. Boenisch, F., Dziedzic, A., Schuster, R., Shamsabadi, A.S., Shumailov, I. and Papernot, N. (2023) 'When the curious abandon honesty: federated learning is not private', *2023 IEEE 8th European Symposium on Security and Privacy*, pp. 175–199.
42. Shokri, R. and Shmatikov, V. (2015) 'Privacy-preserving deep learning', *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1310–1321.
43. Melis, L., Song, C., De Cristofaro, E. and Shmatikov, V. (2019) 'Exploiting unintended feature leakage in collaborative learning', *2019 IEEE Symposium on Security and Privacy*, pp. 691–706.

44. Nasr, M., Shokri, R. and Houmansadr, A. (2019) 'Comprehensive privacy analysis of deep learning: passive and active white-box inference attacks against centralized and federated learning', *2019 IEEE Symposium on Security and Privacy*, pp. 739–753.
45. Zhu, H., Xu, J., Liu, S. and Jin, Y. (2021) 'Federated learning on non-IID data: a survey', *Neurocomputing*, 465, pp. 371–390.
46. Criado, M.F., Casado, F.E., Iglesias, R., Regueiro, C.V. and Barro, S. (2022) 'Non-IID data and continual learning processes in federated learning: a long road ahead', *Information Fusion*, 88, pp. 263–280.
47. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S. and Suresh, A.T. (2020) 'SCAFFOLD: stochastic controlled averaging for federated learning', *Proceedings of the 37th International Conference on Machine Learning*, PMLR 119, pp. 5132–5143.
48. Reddi, S., Charles, Z., Zaheer, M., Garrett, Z., Rush, K., Konečný, J., Kumar, S. and McMahan, H.B. (2021) 'Adaptive federated optimisation ', *Proceedings of the 9th International Conference on Learning Representations*.
49. Tan, A.Z., Yu, H., Cui, L. and Yang, Q. (2023) 'Towards personalized federated learning', *IEEE Transactions on Neural Networks and Learning Systems*, 34(12), pp. 9587–9603.
50. Ye, M., Fang, X., Du, B., Yuen, P.C. and Tao, D. (2024) 'Heterogeneous federated learning: state-of-the-art and research challenges', *ACM Computing Surveys*, 56(3), pp. 1–44.
51. Bayram, F., Ahmed, B.S. and Kassler, A. (2022) 'From concept drift to model degradation: an overview on performance-aware drift detectors', *Knowledge-Based Systems*, 245, 108632.
52. Agrahari, S. and Singh, A.K. (2022) 'Concept drift detection in data stream mining: a literature review', *Journal of King Saud University – Computer and Information Sciences*, 34(10), pp. 9523–9540.
53. Suárez-Cetrulo, A.L., Quintana, D. and Cervantes, A. (2023) 'A survey on machine learning for recurring concept drifting data streams', *Expert Systems with Applications*, 213, 118934.
54. Hinder, F., Vaquet, V. and Hammer, B. (2024) 'One or two things we know about concept drift: a survey on monitoring in evolving environments', *Frontiers in Artificial Intelligence*, 7, 1330257.
55. Mallick, A., Hsieh, K., Arzani, B. and Joshi, G. (2022) 'Matchmaker: data drift mitigation in machine learning for large-scale systems', *Proceedings of Machine Learning and Systems*, 4, pp. 77–94.
56. Manias, D.M., Shaer, I., Yang, L. and Shami, A. (2021) 'Concept drift detection in federated networked systems', *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6.
57. Verwimp, E., Aljundi, R., Ben-David, S., Bethge, M., Cossu, A., Gepperth, A., et al. (2024) 'Continual learning: applications and the road forward', *Transactions on Machine Learning Research*.
58. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D. and van de Weijer, J. (2023) 'Class-incremental learning: survey and performance evaluation on image classification', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5), pp. 5513–5533.

59. Aljundi, R., Lin, M., Goujaud, B. and Bengio, Y. (2019) 'Gradient based sample selection for online continual learning', *Advances in Neural Information Processing Systems*, 32, pp. 11816–11825.
60. Dong, J., Wang, L., Fang, Z., Sun, G., Xu, S., Wang, X. and Zhu, Q. (2022) 'Federated class-incremental learning', *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10164–10173.
61. Zhang, J., Chen, C., Zhuang, W. and Lyu, L. (2023) 'TARGET: federated class-continual learning via exemplar-free distillation', *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4782–4793.
62. Babakniya, S., Fabian, Z., He, C., Soltanolkotabi, M. and Avestimehr, S. (2023) 'A data-free approach to mitigate catastrophic forgetting in federated class incremental learning for vision tasks', *Advances in Neural Information Processing Systems*, 36, pp. 70725–70738.
63. Yang, X., Yu, H., Gao, X., Wang, H., Zhang, J. and Li, T. (2024) 'Federated continual learning via knowledge fusion: a survey', *IEEE Transactions on Knowledge and Data Engineering*, 36(8), pp. 3832–3850.
64. Shao, J., Wu, F. and Zhang, J. (2024) 'Selective knowledge sharing for privacy-preserving federated distillation without a good teacher', *Nature Communications*, 15, 349.
65. Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H.R., Albarqouni, S., et al. (2020) 'The future of digital health with federated learning', *npj Digital Medicine*, 3, 119.
66. Alazab, M., RM, S.P., Parimala, M., Maddikunta, P.K.R., Gadekallu, T.R. and Pham, Q.V. (2022) 'Federated learning for cybersecurity: concepts, challenges, and future directions', *IEEE Transactions on Industrial Informatics*, 18(5), pp. 3501–3509.
67. Belenguer, A., Navaridas, J. and Pascual, J.A. (2022) 'A review of federated learning in intrusion detection systems for IoT', *arXiv preprint*, arXiv:2204.12443.
68. Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.C., Yang, Q., Niyato, D. and Miao, C. (2020) 'Federated learning in mobile edge networks: a comprehensive survey', *IEEE Communications Surveys and Tutorials*, 22(3), pp. 2031–2063.
69. Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J. and Zhang, W. (2023) 'A survey on federated learning: challenges and applications', *International Journal of Machine Learning and Cybernetics*, 14(2), pp. 513–535.
70. Truong, N., Sun, K., Wang, S., Guitton, F. and Guo, Y. (2021) 'Privacy preservation in federated learning: an insightful survey from the GDPR perspective', *Computers and Security*, 110, 102402.
71. Gomes, H.M., Grzenda, M., Mello, R., Read, J., Le Nguyen, M.H. and Bifet, A. (2022) 'A survey on semi-supervised learning for delayed partially labelled data streams', *ACM Computing Surveys*, 55(4), pp. 1–42.
72. Huang, W., Ye, M., Shi, Z., Wan, G., Li, H., Du, B. and Yang, Q. (2024) 'Federated learning for generalization, robustness, fairness: a survey and benchmark', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12), pp. 9387–9406.
73. Qi, P., Chiaro, D., Guzzo, A., Ianni, M., Fortino, G. and Piccialli, F. (2024) 'Model aggregation techniques in federated learning: a comprehensive survey', *Future Generation Computer Systems*, 150, pp. 272–293.

74. Li, Z., Lin, T., Shang, X. and Wu, C. (2023) 'Revisiting weighted aggregation in federated learning with neural networks', *Proceedings of the 40th International Conference on Machine Learning*, PMLR 202, pp. 19767–19788.
75. Guan, H., Yap, P.T., Bozoki, A. and Liu, M. (2024) 'Federated learning for medical image analysis: a survey', *Pattern Recognition*, 151, 110424.
76. Sun, T., Li, D. and Wang, B. (2023) 'Decentralized federated averaging', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4), pp. 4289–4301.
77. Nguyen, D.C., Pham, Q.V., Pathirana, P.N., Ding, M., Seneviratne, A., Lin, Z., Dobre, O. and Hwang, W.J. (2022) 'Federated learning for smart healthcare: a survey', *ACM Computing Surveys*, 55(3), pp. 1–37.
78. Zhou, D.W., Wang, Q.W., Qi, Z.H., Ye, H.J., Zhan, D.C. and Liu, Z. (2024) 'Class-incremental learning: a survey', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12), pp. 9851–9873.