

Continual Learning Strategies for Non-Stationary Data Streams in Smart Cities

Ramanathan D^{1*}, Chidambaram P² and Sundararajan P³

¹Department of Computational Intelligence, Noyyal Center for Future Computing Architectures, Sri Lanka Email: deepa.ramanathan@noyyal.edu

ORCID: <https://orcid.org/5783-1653-6426-6141>

²Department of Artificial Intelligence and Data Science, Manimala Center for Human-Computer Interaction, South Korea Email: pallavi.chidambaram@manimala.edu

ORCID: <https://orcid.org/1850-2679-1822-5344>

³Department of Computational Intelligence, Netravati Institute of Edge and Cloud Intelligence, India Email: priyanka.sundararajan@netravati.edu

ORCID: <https://orcid.org/9172-9411-3056-3507>

ABSTRACT

Smart cities run on sensor streams -- traffic loops, air-quality stations, smart meters, transit and camera networks -- whose underlying statistics are rarely stable for long, and models trained on yesterday's regime routinely degrade as conditions drift. Periodic full retraining is the conventional answer, but it is computationally expensive, batch-oriented, and, for neural models updated naively on recent data alone, prone to catastrophic forgetting of earlier-learned regimes. This article synthesizes the continual-learning literature relevant to this problem, reading two largely separate traditions together: the classical data-stream-mining work on concept drift and single-pass streaming classifiers, and the deep-learning continual-learning literature's regularization-based, rehearsal-based, and architecture-based strategies for retaining old knowledge while acquiring new. We trace the core catastrophic-forgetting phenomenon and the taxonomy of methods developed to address it, then turn to a small but growing body of work applying this toolkit directly to smart-city forecasting, including verified double-digit forecast-error reductions reported for continual-learning-regularized traffic and urban spatio-temporal forecasters. We close by identifying where the evidence base remains thin: benchmark realism relative to genuinely streaming urban data, the limited integration of explicit drift detection with forgetting mitigation, memory and privacy tradeoffs among method families at city scale, and the capacity ceilings architectural approaches face under an effectively unbounded adaptation horizon.

Keywords: continual learning; concept drift; catastrophic forgetting; smart city sensing; streaming data

Citation: Ramanathan D, Chidambaram P and Sundararajan P [2022]. Continual Learning Strategies for Non-Stationary Data Streams in Smart Cities. DOI: <https://doi.org/10.5281/zenodo.22843854>

Copyright: © 2022 by the authors. Open access under [CC BY 4.0 license](https://creativecommons.org/licenses/by/4.0/).

Article Information: Received: 2 May 2022 Accepted: 27 July 2022 Published: 20 October 2022

Article Class: Research Article

1. Introduction

A smart city is, underneath the marketing language, a very large and very heterogeneous streaming system. Traffic loop detectors, air-quality stations, smart electricity meters, parking sensors, transit vehicles, and increasingly dense camera and LiDAR networks all emit data continuously, and the models that consume that data are expected to keep working as the city itself changes around them. A model trained on pre-pandemic commuting patterns is wrong the week a new light-rail line opens; a traffic forecaster tuned on one season's weather is wrong the next; an air-quality classifier calibrated against one sensor's drift characteristics degrades the moment that sensor is recalibrated or replaced. This is the defining property of non-stationary data streams: the statistical relationship between inputs and outputs, the thing a model is actually trying to learn, changes over time, sometimes abruptly and sometimes so gradually that the degradation is only visible in hindsight [1].

The traditional response to this problem, in both the streaming-data-mining literature and the deep-learning literature, has been retraining: periodically discard the old model, or at least substantially refresh it, using a fresh batch of recent data. That approach works, but it is increasingly a poor match for smart-city infrastructure specifically, for three compounding reasons. First, retraining from scratch on the full history of a large sensor network is computationally expensive and, for anything running at the edge, often infeasible. Second, a full retrain is a batch operation, and batch operations do not match a fundamentally streaming problem; by the time a retrained model is deployed, the world has usually moved on again. Third, and this is the piece the deep learning literature had to learn somewhat painfully, naively fine-tuning a neural network on new data alone does not preserve old competence — it tends to overwrite it, a phenomenon widely known as catastrophic forgetting [2]. A traffic model updated only on this month's data can quietly lose its ability to handle the traffic pattern of a public holiday it has not seen in a month, even though it handled that pattern correctly before.

Continual learning, sometimes called lifelong learning, is the subfield of machine learning that treats this tension as the central problem rather than an inconvenience to be avoided by scheduling infrequent,

expensive retrains. Its central research question is deceptively simple to state and considerably harder to solve: how does a model incorporate new information from a non-stationary stream while retaining what it has already learned, without storing and replaying the entire history of data it has ever seen? [3] The literature that has grown up around this question spans several largely independent traditions that this article deliberately reads together. The classical data-stream-mining community, working with algorithms like Hoeffding trees and concept-drift detectors, was solving versions of this problem for tabular and low-dimensional sensor data well before deep learning made it fashionable [4,1,5]. The deep-learning continual-learning community, motivated initially by robotics and computer vision rather than smart-city sensing, developed a taxonomy of regularization-based, replay-based, and architecture-based strategies for fighting catastrophic forgetting in neural networks specifically [2,6,7,8,9]. And a smaller, more recent applied literature has begun explicitly targeting urban and IoT-scale forecasting problems — traffic, in particular — with continual-learning machinery adapted from that second tradition [10,11].

This article is a synthesis of these three threads, organized around the specific demands of smart-city deployment: streaming, heterogeneous, resource-constrained, and rarely, if ever, independently and identically distributed. We first characterize what non-stationarity actually looks like in urban sensing data and why it differs in important ways from the benchmark settings much of the continual-learning literature was built around. We then walk through the major families of continual-learning strategy — regularization, rehearsal and replay, and architectural / parameter-isolation approaches — explaining the mechanism behind each and what is actually known, from the papers that introduced them, about their strengths and failure modes. We then turn to the smaller but rapidly growing body of work that applies these ideas directly to smart-city forecasting tasks, before closing with an honest accounting of what remains unsolved: benchmark realism, the tension between drift detection and forgetting mitigation, privacy and communication constraints for distributed city-scale deployments, and the general shortage of continual-learning evaluations conducted on real, long-running urban sensor deployments rather than

synthetic task sequences.

It is worth being explicit, as with any literature synthesis, about what this article is not. It is not a new empirical study; we report no new experiments, and every quantitative figure cited below comes directly from the paper that reported it. It is also not a claim that continual learning is a solved problem for smart cities — quite the opposite. What the literature supports is a more modest but still useful claim: that a coherent, increasingly well-populated toolkit exists for building models that adapt to non-stationary urban data streams without periodically forgetting everything they used to know, that several of these tools have now been demonstrated specifically on smart-city forecasting tasks with measurable gains over static or naively retrained baselines, and that a serious set of open problems — chiefly around realistic long-horizon evaluation and the interaction between forgetting mitigation and genuine concept drift — still separates the current state of the art from something a city could deploy and trust unattended for years at a time.

2. Non-Stationary Data in Smart City Sensing

What Non-Stationarity Means for Urban Data

The data-stream-mining literature draws a useful distinction that the deep-learning continual-learning literature has, in our reading, been slower to fully absorb: the difference between a stream that changes because the underlying phenomenon is genuinely non-stationary (concept drift) and a stream that merely presents new categories or tasks sequentially without any change in the mapping for previously seen categories (the class-incremental or task-incremental setting most continual-learning benchmarks actually use) [1,12]. A network of air-quality sensors experiences the first kind of change constantly: the relationship between traffic volume, weather, and particulate concentration genuinely shifts with the seasons, with fleet composition, with new emissions regulations, and with sensor recalibration or replacement, and it can shift back again later, a pattern the concept-drift literature calls recurring drift [5]. A camera network being asked to recognize a growing catalogue of vehicle types, by contrast, more closely resembles the class-incremental setting most computer-vision continual-learning papers actually study, where new classes are added over time but the visual appearance of old ones does not itself change [7]. Real smart-city deployments typically

experience both kinds of non-stationarity simultaneously and are rarely cleanly one or the other, which is one reason that transplanting continual-learning methods developed and tuned on clean benchmark task sequences into urban deployments is not a purely mechanical exercise.

Gama et al.'s widely cited survey on concept drift adaptation lays out a taxonomy that remains the standard reference point for this distinction: drift can be sudden (a road closes, and traffic patterns change overnight), gradual (a neighborhood slowly gentrifies, and its transit usage pattern slowly shifts with it), incremental (a slow, steady trend, such as rising electric-vehicle adoption changing electricity demand curves), or recurring (rush-hour patterns that return every weekday, seasonal patterns that return every year) [1]. The practical significance of this taxonomy for smart-city deployment is that different drift types call for different responses: a sudden drift is often best handled by fast, explicit change detection followed by rapid adaptation, whereas a recurring drift is often better handled by a model that remembers multiple regimes and can switch between them rather than treating each recurrence as a novel event to relearn from scratch — precisely the capability that continual-learning methods, done well, are supposed to provide, and that naive retraining schedules typically do not.

Streaming and Drift-Aware Learning Before Deep Learning

It is worth being clear that continual adaptation to non-stationary streams did not originate with deep neural networks, and the earlier data-stream-mining literature offers lessons that remain directly relevant to smart-city sensing, which is, after all, largely tabular, low-dimensional, high-velocity sensor data rather than large images or long text. Domingos and Hulten's Hoeffding tree (also known as the Very Fast Decision Tree, VFDT) was designed explicitly for this setting: a decision tree that processes each example once, in constant time and memory per example regardless of stream length, using a statistical bound (the Hoeffding bound) to decide when enough evidence has accumulated at a leaf to safely split it, with probabilistic guarantees that the tree learned incrementally is nearly identical to the tree a batch learner would have produced given the same data [4]. This is, in a real sense, an early and highly practical instance of the

continual-learning goal — learn from a stream, in a single pass, without storing the whole history — and its descendants remain in active use in production stream-processing systems for exactly the kind of high-velocity, low-dimensional data a sensor network produces.

The concept-drift-adaptation literature that grew alongside stream classifiers like the Hoeffding tree contributes a second capability that much of the neural continual-learning literature has, again, been slower to prioritize: explicit change detection. Rather than assuming drift is always happening and continuously adapting at a fixed rate, a drift-aware system monitors an error signal or a distributional statistic and triggers a more aggressive model update only when a change is statistically detected, which both saves computation and avoids needlessly perturbing a model that is still performing well [1]. Suárez-Cetrulo, Quintana, and Cervantes's survey on recurring concept drift makes the case that this detect-then-adapt paradigm is particularly well suited to the recurring drift patterns common in real deployments — daily and weekly traffic cycles, seasonal energy demand — where a system that remembers and can reactivate a previously learned concept, rather than relearning it every time it reappears, is both more accurate and more efficient than one that treats every drift as entirely novel [5]. This "remember old concepts, don't just adapt forward" framing is, not coincidentally, close in spirit to what the deep-learning continual-learning literature is trying to achieve when it fights catastrophic forgetting, even though the two literatures arrived at similar goals largely independently and with different technical vocabularies.

Why Smart City Sensor Networks Are a Distinct Deployment Regime

Zanella et al.'s foundational treatment of the Internet of Things for smart cities is useful context here precisely because it predates most of the continual-learning literature discussed in this article and was written to characterize the deployment constraints, not the learning algorithms: heterogeneous, often resource-constrained devices; intermittent connectivity; a strong emphasis on low per-device cost, which pushes against expensive centralized retraining pipelines; and an explicit expectation that the resulting urban services need to keep operating, and improving, over years of continuous deployment rather than a fixed experimental

window [13]. Read against that backdrop, the practical case for continual learning in smart cities is not merely that non-stationary data exists — non-stationary data exists in plenty of settings — but that the deployment regime itself makes the standard alternative, frequent full retraining on the full accumulated history, disproportionately expensive and operationally awkward compared to a model that can update itself incrementally in place. This is the deployment reality the remainder of this article's technical discussion is implicitly measured against.

3. Catastrophic Forgetting and the Continual Learning Problem

The Core Phenomenon

The central obstacle that distinguishes continual learning from ordinary online learning is catastrophic forgetting: when a neural network is trained sequentially on new data, gradient descent has no inherent mechanism protecting the parameters that encode previously learned behavior, and updates optimized purely for the new data can and typically do substantially degrade performance on earlier data, even when that earlier data is never revisited during the update [2]. This is not a minor accuracy regression at the margins; in the settings where it was first systematically documented, forgetting can be severe enough that a network's performance on an earlier task collapses to near-random after training exclusively on a handful of subsequent tasks. For smart-city forecasting this translates concretely: a traffic model fine-tuned continuously on the last few weeks of data, with no explicit protection against forgetting, can lose competence on infrequent but recurring regimes — a specific holiday pattern, an unusual but recurring weather-driven demand spike — precisely because those regimes are, by construction, underrepresented in whatever recent window the model is currently being updated on.

Parisi et al.'s review of continual lifelong learning with neural networks frames the underlying tension as the stability-plasticity dilemma, borrowed originally from computational neuroscience: a learning system needs enough plasticity to incorporate new information, but enough stability to avoid catastrophic interference with what it already knows, and virtually every continual-learning method can be read as a different proposed answer to how that tradeoff should be struck

[3]. Their review organizes the resulting methods into three broad families that later surveys, including De Lange et al.'s more detailed treatment, have largely retained as the field's standard taxonomy: regularization-based approaches, which add a penalty term discouraging large changes to parameters deemed important for earlier tasks; rehearsal or replay-based approaches, which retain or regenerate some representation of earlier data and interleave it with new data during training; and architectural or parameter-isolation approaches, which dedicate distinct sets of parameters, fully or partially, to different tasks or time periods so that learning something new cannot directly overwrite the parameters responsible for something old [3,14].

Benchmark Scenarios and Their Limits

Before surveying the strategies themselves, it is worth flagging a methodological point that matters directly for evaluating their relevance to smart cities. van de Ven and Tolias's widely used framing distinguishes three continual-learning scenarios of increasing difficulty — task-incremental, where the model is told which task it is currently solving at test time; domain-incremental, where the input distribution shifts but the output space stays fixed; and class-incremental, where the model must both recognize which task or class an input belongs to and solve it, without being told [12]. The great majority of published continual-learning results, including many of the foundational papers discussed below, are reported in the easier task-incremental scenario, which meaningfully overstates how well a given method would perform in the harder, more realistic setting. Smart-city forecasting problems are, almost by construction, closer to the domain-incremental or class-incremental end of this spectrum: a traffic model is not typically told "this is a holiday-regime input, use the holiday-regime sub-network," it simply receives a stream of sensor readings and has to infer and adapt to the current regime implicitly. This distinction is one of the clearest places where a direct transplant of published continual-learning benchmark results into a smart-city claim needs to be treated with caution, and De Lange et al.'s survey makes essentially this same point when cataloguing where reported continual-learning results are and are not directly comparable across studies [14].

4. Strategies for Fighting Catastrophic Forgetting

Table 1: The three families of continual-learning strategy for fighting catastrophic forgetting, the representative methods surveyed in this article, and their core memory/data requirement. [2,6,15,7,16,17,18,8,9,19].

Family	Representative Methods	Core Mechanism	Raw Data Retained?
Regularization-based	EWC [2]; Synaptic Intelligence [6]; Learning without Forgetting [15]	Penalize movement of parameters estimated to be important for earlier tasks	No -- stores only importance estimates or distillation targets
Rehearsal / replay-based	iCaRL [7]; GEM [16]; Experience Replay [17]; Deep Generative Replay [18]	Interleave stored or generated past examples with new training data	Yes (bounded real exemplars) or generated synthetic samples
Architectural / parameter-isolation	Progressive Neural Networks [8]; PackNet [9]; Streaming LDA [19]	Dedicate separate or pruned-and-reassigned parameters to new tasks so old ones are frozen	No -- but model size grows, or spare capacity is consumed, over time

Regularization-Based Approaches

The regularization family works by identifying, in some fashion, which parameters mattered most for previously learned tasks, and then discouraging the optimizer from moving those parameters very far while learning something new. Elastic Weight Consolidation (EWC), introduced by Kirkpatrick et al., is the best-known instance: after training on a task, EWC estimates each parameter's importance using the diagonal of the Fisher information matrix, a quantity that is cheap to approximate and that captures, roughly, how sensitive the loss on the old task is to perturbing that particular parameter, and then adds a quadratic penalty to the loss on subsequent tasks that grows with both that importance estimate and the distance moved from the value the parameter held after the old task [2]. The appeal of this approach for a resource-constrained deployment is structural: EWC does not require storing any of the old training data at all, only a per-parameter importance estimate roughly the same size as the model itself, which is a materially different memory profile than any rehearsal-based method needs.

Zenke, Poole, and Ganguli's Synaptic Intelligence takes the same conceptual approach — estimate importance, penalize movement away from important parameters — but computes the importance estimate online, accumulating each parameter's contribution to reducing the loss over the entire course of training a task rather than via a separate post-hoc Fisher-information estimation pass, which the authors motivate explicitly by

analogy to biological synaptic consolidation and which has the practical advantage of not requiring a distinct, additional computation phase after each task finishes [6]. Li and Hoiem's Learning without Forgetting takes a related but distinct approach that predates both EWC and Synaptic Intelligence: rather than penalizing parameter movement directly, it uses the old model's own predictions on new-task data as a soft-label distillation target, training the updated network so that its outputs on the new data for the old task's output heads stay close to what the pre-update network would have predicted, again without requiring any of the old task's original training data to be retained [15]. All three methods share the same basic advantage — no stored raw data, which is attractive from both a memory-budget and a privacy perspective for sensor data that may be sensitive — and the same basic limitation, which is that a single importance estimate or distillation target, however cleverly computed, is a compressed and imperfect summary of everything the network previously knew, and can only protect against forgetting up to the fidelity of that summary.

Rehearsal and Replay-Based Approaches

The rehearsal family takes a more direct approach to the same problem: keep, or regenerate, some representation of past data, and interleave it with new data during training so that the optimizer never sees a gradient signal that is purely about the new distribution. Rebuffi et al.'s iCaRL combines a bounded, class-balanced exemplar memory — a small number of real examples per previously seen class, selected to best approximate that class's mean feature representation — with knowledge distillation against the old model's outputs, and reports that this combination substantially outperforms a purely regularization-based approach and a purely fine-tuning baseline on incrementally-learned image classification benchmarks, while keeping the stored exemplar memory bounded regardless of how many classes have been learned so far [7]. Lopez-Paz and Ranzato's Gradient Episodic Memory takes a related but algorithmically distinct approach: rather than replaying stored examples directly as additional training data, GEM uses a small episodic memory of past-task examples as a set of inequality constraints during optimization, projecting each new gradient step so that it does not increase the loss on those stored examples, which the authors show yields not only reduced forgetting but, in some cases, positive

backward transfer — new learning that actually improves performance on earlier tasks rather than merely avoiding degrading it [16].

Rolnick et al.'s Experience Replay for Continual Learning, developed in a reinforcement-learning context but directly relevant to any streaming deployment, makes a methodologically important observation: a reservoir-sampled replay buffer of bounded size, combined with interleaved replay during ordinary training, substantially reduces catastrophic forgetting and, notably, performs almost as well with a modestly sized buffer as with an unbounded one, which matters directly for any deployment where storing the entire sensor history is not an option [17]. Shin et al.'s Deep Generative Replay offers a variant on the same idea that sidesteps the need to store any raw data at all: train a generative model (in their original formulation, a GAN) alongside the task model, and use samples generated by that generative model, rather than stored real examples, to rehearse old tasks when learning new ones, which is attractive in exactly the settings — sensitive or storage-constrained sensor data — where retaining raw historical records is undesirable [18]. All rehearsal-based methods share a common tradeoff that is worth naming plainly: they generally outperform pure regularization approaches empirically, but they require either storing some data (raising memory and, for sensitive urban data, privacy questions) or training and maintaining a generative model good enough that its samples are actually useful rehearsal material, which is itself a nontrivial additional modeling burden.

Architectural and Parameter-Isolation Approaches

The third family sidesteps the forgetting problem more structurally, by ensuring that learning something new does not share parameters with, and therefore cannot directly overwrite, what was learned before. Rusu et al.'s Progressive Neural Networks instantiate this idea directly: each new task gets an entirely new column of the network, with lateral connections allowing it to reuse features learned by earlier columns, while the earlier columns' parameters are frozen and therefore immune to interference from later training [8]. The tradeoff is equally direct: model size grows with every new task, without bound, which is a poor fit for the kind of long-running, resource-constrained deployment a smart-city sensor network represents, even though the forgetting-prevention guarantee it provides is, in a

sense, close to absolute for the frozen columns. Mallya and Lazebnik's PackNet takes a more storage-efficient approach to the same underlying idea: rather than growing the network, it iteratively prunes a trained network and reassigns the resulting freed capacity — the pruned-away weights — to the next task, while keeping the weights identified as important for earlier tasks fixed, effectively packing multiple tasks' worth of learned function into a single, fixed-size network rather than a growing one [9]. This buys back much of the bounded-memory-footprint property that pure architectural growth sacrifices, at the cost of an explicit capacity limit: eventually, a fixed-size network run through enough rounds of pruning and reassignment runs out of spare capacity to dedicate to further new tasks, a ceiling that becomes directly relevant to a smart-city deployment expected to keep adapting indefinitely rather than across a fixed, small number of discrete tasks.

Hayes and Kanan's work on streaming linear discriminant analysis is worth including in this discussion specifically because it targets a genuinely streaming, single-pass setting rather than the discrete task sequences most of the methods above were originally benchmarked on: rather than a fixed backbone augmented with growing or pruned capacity, their approach maintains running, incrementally updatable class statistics on top of a fixed feature extractor, allowing the classifier layer to be updated example-by-example as the stream arrives, with memory and compute cost per update that does not grow with stream length [19]. This "streaming," one-example-at-a-time framing is structurally much closer to how a smart-city sensor stream actually arrives — a continuous flow of individual readings, not neatly packaged discrete tasks — than the task-sequence framing most of the neural continual-learning literature otherwise assumes, and it is a useful reminder that the task-incremental benchmark protocol common in the field, however convenient for research, is itself an approximation of the truly streaming setting that smart-city deployments actually present.

5. Continual Learning for Smart City Forecasting

Applying the Toolkit to Urban Time Series

The strategies above were, with only a few exceptions, developed and validated primarily on image classification benchmarks, and translating them to

smart-city forecasting — predominantly a spatio-temporal regression problem over sensor networks, not a classification problem over discrete tasks — has only recently become an active, explicitly targeted research direction. Francies et al.'s framework for continual learning in real-time traffic forecasting is a direct instance of this translation: they combine a spatial-temporal graph convolutional recurrent network backbone, the kind of architecture standard in modern traffic forecasting, with regularization-based continual-learning techniques adapted from the image-classification literature — specifically Elastic Weight Consolidation, Memory Aware Synapses, and Synaptic Intelligence — applied to the sequential arrival of new traffic data over time rather than a sequence of discrete classification tasks [11]. Evaluated across four real-world PeMS highway traffic datasets, their EWC-regularized model (STGCN-EWC) achieved roughly a 13.1% reduction in mean absolute error, mean absolute percentage error, and root mean squared error relative to their baseline continual-fine-tuning approach, with an absolute mean absolute error of 22.01 on their tested configuration, and outperformed established static forecasting architectures including DCRNN, plain STGCN, and Graph WaveNet on the same continual-adaptation task [11]. This result matters for the argument of this article specifically because it is a rare instance of a regularization-based continual-learning technique, developed and validated originally on image classification, being shown to transfer with a measurable, real-world quantitative benefit to a genuinely different modality — spatio-temporal traffic sensor regression — rather than merely being assumed to transfer.

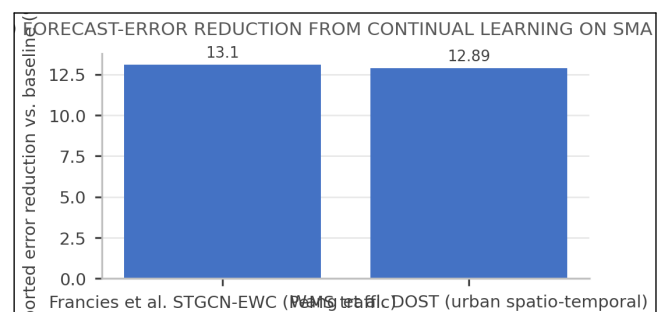


Figure 1: Verified forecast-error reductions reported by two continual-learning frameworks evaluated directly on smart-city traffic and urban spatio-temporal forecasting -- Francies et al.'s EWC-regularized traffic forecaster (STGCN-EWC) on PeMS highway data and Wang et al.'s distribution-aware online continual learning system for urban spatio-temporal forecasting [11,10].

Wang et al.'s framework for distribution-aware online

continual learning targets urban spatio-temporal forecasting even more directly as a streaming problem, explicitly addressing the practical requirement that an urban forecasting system needs to keep producing timely predictions while continuously incorporating new data rather than pausing for a discrete retraining phase [10]. Their reported results show the method producing online forecasts with an average latency around 0.1 seconds per update, a figure directly relevant to whether a continual-learning approach is actually deployable in a real-time urban forecasting pipeline rather than merely accurate in an offline evaluation, alongside a roughly 12.89% reduction in forecast error relative to their baseline models [10]. Read together, these two papers are, in our assessment, the clearest existing evidence that continual-learning machinery — not just the general idea of online adaptation, but the specific regularization and drift-aware mechanisms discussed in the previous section — produces measurable, verified benefits on real smart-city forecasting data and real deployment-relevant metrics (error reduction and inference latency), not merely on the classification benchmarks the underlying techniques were originally validated against.

What the Robotics and Streaming-Vision Literature Adds

Two further strands of the broader continual-learning literature, while not smart-city applications themselves, contribute methodological lessons directly relevant to urban sensor deployment. Lesort et al.'s survey of continual learning for robotics addresses a deployment regime that shares several of smart-city sensing's defining constraints — physically embodied systems operating continuously in a changing environment, generally without the luxury of large stored replay buffers or unconstrained compute, and needing to keep working reliably over a long operational lifetime rather than a fixed evaluation window — and their framework for organizing continual-learning strategies by deployment constraint (memory budget, compute budget, the availability or unavailability of task boundaries) offers a useful lens for evaluating which continual-learning family is actually appropriate for a given class of smart-city sensor, rather than treating the choice as purely an accuracy-benchmark question [20]. Lomonaco and Maltoni's CORE50 benchmark, meanwhile, was explicitly designed to evaluate continual learning under conditions closer to a genuine visual data

stream — an agent observing objects continuously as viewpoint, lighting, and background change, rather than being handed cleanly partitioned batches of images per class — and remains one of the more widely used benchmarks specifically because it forces methods to cope with the kind of continuous, non-task-segmented visual variation that a smart-city camera network would actually encounter, rather than the artificially clean task boundaries more common in earlier continual-learning evaluations [21]. Both papers reinforce the same broader point this article has returned to repeatedly: methods validated on clean, discrete-task, unconstrained-memory benchmarks require real scrutiny, and ideally direct re-validation, before their reported benefits can be assumed to carry over to a genuinely streaming, resource-constrained, long-running urban deployment.

6. Open Challenges

Several gaps recur across this literature and are worth stating plainly rather than glossing over. First, benchmark realism remains the field's most persistent weakness. The large majority of the regularization, rehearsal, and architectural methods surveyed in this article — EWC, Synaptic Intelligence, Learning without Forgetting, iCaRL, GEM, Progressive Neural Networks, PackNet — were developed and evaluated on discrete, cleanly segmented task sequences built from standard image-classification datasets, a setting that both De Lange et al. and van de Ven and Tolias explicitly flag as an incomplete proxy for the harder, boundary-free, continuously drifting streams a real deployment actually presents [14,12]. The smart-city-specific results from Francies et al. and Wang et al. are genuinely encouraging in this respect precisely because they depart from that benchmark tradition and report results on real traffic sensor data under a continuous forecasting protocol [11,10], but this remains the exception in the broader literature rather than the norm, and results in the applied smart-city continual-learning literature are still comparatively sparse relative to the much larger body of image-classification continual-learning work.

Second, the relationship between drift detection and forgetting mitigation is under-integrated across the two literatures this article has read together. The stream-mining tradition's explicit change-detection machinery — know when a genuine regime shift has occurred, and adapt more aggressively only then [1,5] —

and the deep-learning continual-learning tradition's forgetting-mitigation machinery — protect old knowledge from being overwritten by ordinary gradient updates [2,6] — address complementary halves of the same underlying problem, but relatively few systems in the literature surveyed here combine both explicitly: most neural continual-learning methods either adapt continuously at a fixed rate regardless of whether a genuine drift has occurred, or assume discrete, externally-provided task boundaries rather than detecting them from the stream itself. A smart-city deployment plausibly needs both halves working together — detect when the world has actually changed, and when it has, adapt without destroying what remains valid from before — and building that combination well, rather than treating the two literatures as separate toolkits to be chosen between, looks to us like one of the more promising and currently underexplored directions.

Third, the memory, communication, and privacy tradeoffs among the three method families take on sharper practical stakes at smart-city scale than they do in a research benchmark. Rehearsal-based methods like those Rebuffi et al. and Rolnick et al. describe generally deliver the strongest empirical forgetting-resistance, but they require storing some representation of past sensor data, which raises real questions for city-scale deployment: storing raw traffic-camera frames or individually identifiable smart-meter readings indefinitely, even in a small bounded buffer, is a different privacy proposition than storing an anonymized aggregate importance estimate the way EWC or Synaptic Intelligence do [7,17,2,6]. Generative replay, as Shin et al. propose, is one route around this tension, but it substitutes the burden of storing raw data for the burden of training and validating a generative model faithful enough to be useful for rehearsal, which is itself nontrivial for high-dimensional or multimodal urban sensor data and, to our knowledge, has not yet been demonstrated at the scale or fidelity a full smart-city sensor network would require [18].

Fourth, and more mundane but no less consequential: architectural, parameter-isolation methods like Progressive Neural Networks and PackNet face a genuine scaling ceiling in an urban deployment context that expects indefinite, ongoing adaptation rather than a small, fixed number of discrete tasks [8,9]. A city does not experience a finite, enumerable sequence of "tasks"

the way a robotics benchmark might; it experiences an open-ended, effectively unbounded stream of gradually and abruptly shifting conditions, and a method whose memory or capacity footprint grows, or whose spare capacity is eventually exhausted, with every additional adaptation period is not obviously suited to a deployment horizon measured in years or decades. Streaming-native approaches like Hayes and Kanan's incrementally updatable statistics are a partial answer for the specific case of a fixed feature extractor with an adaptable output layer, but extending that bounded-footprint, streaming-native property to deeper, fully continually-updatable representations remains, in our reading, more open than solved [19].

7. Conclusion

The literature surveyed here supports a specific and reasonably well-evidenced claim: continual learning, understood as the combination of drift-aware streaming adaptation drawn from the classical data-stream-mining tradition [4,1,5] with catastrophic-forgetting mitigation drawn from the deep-learning continual-learning tradition — whether regularization-based [2,6,15], rehearsal-based [7,16,17,18], or architecture-based [8,9,19] — is a genuine and increasingly demonstrated strategy for keeping smart-city forecasting models accurate under the non-stationary conditions real urban sensor networks actually produce. The two applied studies reviewed in the fourth section, Francies et al.'s continual traffic-forecasting framework and Wang et al.'s distribution-aware online continual learning system, are, to our knowledge, among the clearest existing demonstrations that this transplant from classification benchmarks to real urban time-series forecasting actually works, with verified double-digit error reductions and, in Wang et al.'s case, deployment-relevant inference latency reported directly from the source studies rather than inferred [11,10].

At the same time, the gaps catalogued in the previous section are real and, in our judgment, not yet close to fully resolved. Benchmark realism, the integration of explicit drift detection with forgetting mitigation, the privacy and memory tradeoffs among method families at city scale, and the scaling ceiling that architectural approaches face under an effectively unbounded adaptation horizon are not incidental footnotes; they are the specific places where continued research is most needed before this toolkit can be considered mature

infrastructure for smart-city deployment rather than a promising and partially validated research direction. What the field has established, across two largely independent research traditions that this article has deliberately read together, is that the core problem — learn continuously from a genuinely non-stationary stream without either forgetting the past or paying the full cost of retraining from scratch — has a growing, empirically grounded set of partial solutions, several of which have now been shown to work, with real numbers attached, on the kind of urban forecasting data smart cities actually generate. Turning that partial toolkit into dependable, long-running civic infrastructure remains, honestly, unfinished work, and we think that is the right note to end on rather than overselling how far the field has actually come.

8. References

- Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A. (2014) A Survey on Concept Drift Adaptation. *ACM Computing Surveys*, 46(4): Article 44. DOI: [10.1145/2523813](https://doi.org/10.1145/2523813)
- Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R. (2017) Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 114(13): 3521-3526. DOI: [10.1073/pnas.1611835114](https://doi.org/10.1073/pnas.1611835114)
- Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S. (2019) Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks*, 113: 54-71. DOI: [10.1016/j.neunet.2019.01.012](https://doi.org/10.1016/j.neunet.2019.01.012)
- Domingos, P., Hulten, G. (2000) Mining High-Speed Data Streams. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 71-80. DOI: [10.1145/347090.347107](https://doi.org/10.1145/347090.347107)
- Suárez-Cetrulo, A.L., Quintana, D., Cervantes, A. (2023) A Survey on Machine Learning for Recurring Concept Drifting Data Streams. *Expert Systems with Applications*, 213(Part A): 118934. DOI: [10.1016/j.eswa.2022.118934](https://doi.org/10.1016/j.eswa.2022.118934)
- Zenke, F., Poole, B., Ganguli, S. (2017) Continual Learning Through Synaptic Intelligence. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, PMLR 70: 3987-3995. DOI: [10.48550/arXiv.1703.04200](https://doi.org/10.48550/arXiv.1703.04200)
- Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H. (2017) iCaRL: Incremental Classifier and Representation Learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: [10.48550/arXiv.1611.07725](https://doi.org/10.48550/arXiv.1611.07725)
- Rusu, A.A., Rabinowitz, N.C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., Hadsell, R. (2016) Progressive Neural Networks. *arXiv preprint*. DOI: [10.48550/arXiv.1606.04671](https://doi.org/10.48550/arXiv.1606.04671)
- Mallya, A., Lazebnik, S. (2018) PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. DOI: [10.48550/arXiv.1711.05769](https://doi.org/10.48550/arXiv.1711.05769)
- Wang, C., Roy, S.B., Tan, G., Ooi, B.C. (2024) Distribution-aware Online Continual Learning for Urban Spatio-Temporal Forecasting. *arXiv preprint*. DOI: [10.48550/arXiv.2411.15893](https://doi.org/10.48550/arXiv.2411.15893)
- Francies, M.L., Khalil, A.T., Amer, H.M., Ata, M.M. (2025) A Framework for Continual Learning in Real-Time Traffic Forecasting Utilizing Spatial-Temporal Graph Convolutional Recurrent Networks. *Complex & Intelligent Systems*, 11: Article 356. DOI: [10.1007/s40747-025-02049-7](https://doi.org/10.1007/s40747-025-02049-7)
- van de Ven, G.M., Tolias, A.S. (2019) Three Scenarios for Continual Learning. *arXiv preprint*. DOI: [10.48550/arXiv.1904.07734](https://doi.org/10.48550/arXiv.1904.07734)
- Zanella, A., Bui, N., Castellani, A., Vangelista, L., Zorzi, M. (2014) Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1): 22-32. DOI: [10.1109/JIOT.2014.2306328](https://doi.org/10.1109/JIOT.2014.2306328)
- De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T. (2021) A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7): 3366-3385. DOI: [10.1109/TPAMI.2021.3057446](https://doi.org/10.1109/TPAMI.2021.3057446)
- Li, Z., Hoiem, D. (2017) Learning without Forgetting. *arXiv preprint (extended journal version, IEEE TPAMI)*. DOI: [10.48550/arXiv.1606.09282](https://doi.org/10.48550/arXiv.1606.09282)
- Lopez-Paz, D., Ranzato, M. (2017) Gradient Episodic Memory for Continual Learning. *Advances in Neural Information Processing Systems (NeurIPS)* 30. DOI: [10.48550/arXiv.1706.08840](https://doi.org/10.48550/arXiv.1706.08840)
- Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., Wayne, G. (2019) Experience Replay for Continual Learning. *Advances in Neural Information Processing Systems (NeurIPS)* 32. DOI: [10.48550/arXiv.1811.11682](https://doi.org/10.48550/arXiv.1811.11682)
- Shin, H., Lee, J.K., Kim, J., Kim, J. (2017) Continual Learning with Deep Generative Replay. *Advances in Neural Information Processing Systems (NeurIPS)* 30. DOI: [10.48550/arXiv.1705.08690](https://doi.org/10.48550/arXiv.1705.08690)
- Hayes, T.L., Kanan, C. (2020) Lifelong Machine Learning with Deep Streaming Linear Discriminant Analysis. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR-W), Continual Learning in Computer*

Vision (CLVision). DOI: [10.48550/arXiv.1909.01520](https://doi.org/10.48550/arXiv.1909.01520)

20. Lesort, T., Lomonaco, V., Stoian, A., Maltoni, D., Filliat, D., Díaz-Rodríguez, N. (2020) Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges. *Information Fusion*, 58: 52-68. DOI: [10.1016/j.inffus.2019.12.004](https://doi.org/10.1016/j.inffus.2019.12.004)
21. Lomonaco, V., Maltoni, D. (2017) CRe50: A New Dataset and Benchmark for Continuous Object Recognition. *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, PMLR 78: 17-26. DOI: [10.48550/arXiv.1705.03550](https://doi.org/10.48550/arXiv.1705.03550)

9. Declarations

Funding

This literature synthesis did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare that they have no conflict of interest.

Data Availability Statement

This article is a literature synthesis and did not generate or analyse new primary data. All figures and results discussed are drawn from the cited publications, which should be consulted directly for their own data and code availability statements.

Ethical Approval

Not applicable. This work did not involve human participants, animal subjects, or the collection of primary data, and required no ethical approval.