Maintenance of Model Resilience in Distributed Edge Learning Environments

Qiyuan Wang
Computing Science
Glasgow University
q.wang.1@research.gla.ac.uk

Christos Anagnostopoulos
Computing Science
Glasgow University
christos.anagnostopoulos@glasgow.ac.uk

Jordi M Fornes
Computer Science
Lleida University
jordi.mateo@udl.cat

Kostas Kolomvatsos
Inf. & Telecomm.
Thessaly University
kostasks@uth.gr

Andreas Vrachimis
Computing Science
Glasgow University
2380007v@student.gla.ac.uk

***Abstract*—Distributed Machine Learning (DML) at the edge of the network involves model learning and inference across networking nodes over distributed data. One type of model learning could be the delivery of predictive analytics services to formulate intelligent environments, however, those environments heavily rely on real-time inference and are significantly influenced by changes in the underlying data (concept drifts). Moreover, the quality of service and availability in DML environments are directly tied to each node's reliability, since such environments are highly susceptible to the impact of node failures. Even if such challenges can be tackled with distributed resilience mechanisms, their effectiveness and efficiency, due to concept drifts, should be maintained to ensure continuous and sustained quality of service. DML systems operate in dynamic environments, thus, they require their models to be updated according to the novel trends embedded in the new data they encounter. We, therefore, introduce several model maintenance mechanisms to ensure resilient DML systems in the long term when concept drifts emerge. We provide a comprehensive experimental evaluation of our resilience maintenance mechanisms over synthetic and real data showcasing their importance and applicability in edge learning environments.**

***Index Terms*—Edge Computing, Edge Intelligence, model maintenance, resilient Machine Learning**

I. Introduction

The booming developments of the Internet of Things (IoT), in specific, the advances in intelligent sensors, low-energy wireless communication and sensor network technologies, make it possible for a large number of computing devices to be networked through the IoT infrastructure [1], [2]. IoT devices produce an unprecedented amount of data, raising the demand for pushing computation to the edge of the network to save the cost of storing and transferring the data to the Cloud back end and fully utilize the computational power. This gives birth to a new computation paradigm, i.e., Edge Computing (EC), and makes it possible for many applications (e.g., services provided in smart cities) of which the definition and concept are still emerging and have not been reached a consensus by diverse stakeholders [3]. However, those environments are heavily dependent on all kinds of predictive analytics [4] like public transportation analysis [5], human spatial activity pattern prediction [6] and city subway station planning [7]. Of those predictive analytics, the types that rely on real-time inference are significantly influenced. For instance, consider the traffic congestion prediction. Since city traffic is a complex and dynamic system, a Distributed Machine Learning (DML) scheme is indispensable to monitor its status and make predictions accordingly. However, the quality of service and availability for DML systems are directly tied to each consisting node's *reliability*. Specifically, DML systems are highly susceptible to the impact of node failures. To tackle this problem, we proposed a resilience framework to help each node identify its best surrogate nodes and build *enhanced models* therein to serve requests on its behalf with equivalently satisfactory performances in the case of its failure [8]. *Enhanced models* can be seen as the models trained mainly on the model's residing node's data and a small section of signature data (sufficient statistics) chosen by specially designed strategies from the potential failing nodes. Therefore, these *enhanced models* obtained the ability to operate effectively on data from multiple nodes with minimal inter-nodes data transferring. Our experiments showcased *enhanced models'* capabilities to help the system to function reliably even in the most failure-intensive environments within a certain period of time. However, as DML systems oftentimes operate in dynamic environments thus requiring the models to be updated according to novel trends embedded in new data it encounters, we have to identify how the resilience framework behaves in the long term when concept drifts emerge.

Concept drift, also interchangeably called non-stationary data distributions, is a cause of deteriorating predictive model performances [9]. As stationary models were built upon prior knowledge (like Machine Learning (ML) models), they are expected to not handle the unforeseen changes that happen later on well. The standard ways of handling concept drift include detecting them, analysing and adapting the models to the new concept (which could include forgetting mechanisms that make the models forget the old information) and estimating the loss [10]. However, in our context, the influence of concept drifts and the way of handling them might be very different. As *enhanced models* operate on multiple nodes, the concept drifts in one node may only partially impact the models' performance. Furthermore, any modifications made to the models may not affect their performance on all the entailed

nodes equally. Hence, it is a challenge to approach concept drifts related to *enhanced models* than typical setups, in which the drift affects the model and the model treats concept drifts holistically.

In this paper, we investigate the interactions between *enhanced models* and concept drifts. To the best of our knowledge, this issue has not been covered by any previous works. We first built *enhanced models* with different strategies to source training data from multiple nodes and then induced concept drifts (of different types) in nodes to examine the ML models' performance. We also propose various strategies to extract signature information from the drifted data to retrain the models effectively to adapt the model to the new concept and evaluate to impact of the models' performance on data from different sources. In that process, the trade-offs between amount of data transferred among nodes and the effectiveness of concept drift adaption are comprehensively assessed.

The paper is organized as follows: Section II reports on related work and our contribution, Section III formulates the problems, and Section IV demonstrates the challenges of partial concept drift by experiments. Section V introduces our model maintenance strategies while Section VI reports on a performance evaluation and comparative assessment. Section VII concludes the paper.

II. Related Work & Contribution

The domains of learning under concept drift could be divided into three sections: concept drift detection, concept drift understanding, and concept drift adaption [11].

Since concept drift is known to harm model performance, one typical way to carry out *concept drift detection* is by monitoring the model's performance in specific windows. This gives birth to many error-based detection methods like Drift Detection Method (DDM) [12] and ADaptive WINdowing (ADWIN) [13]. DDM maintains a dynamic window and monitors the significant increase in online error rates within it. Depending on its confidence level on the significant increase, it will give out a warning to signal the building of a new model or an indication of drift to signal the replacement of models. ADWIN compares the model's error on two windows of adaptive sizes: one samples history data, and the other samples new data. The average errors of the two windows are used to determine if concept drift exists.

Concept drift *understanding* entails evaluating the severity of concept drift, as it directly affects how concept drift could be mitigated. A small adjustment to the model by incremental ML might suffice for minor concept drift, while significant concept drift could require retraining the model from scratch [11]. For example, Minku et al. characterized the severity of concept drift by measuring the percentage of the input space that has its target class changed before and after the drift [14].

As our work revolves around maintaining the models' performance in the presence of concept drift, our focus here is *concept drift adaption and resilience maintenance*. Adaptation is handling the concept drift by updating predictive models online during their operation to react to concept drifts [10]. To achieve that, the model has to be maintained to adapt to the new concepts. However, the application scenarios of the existing works in this domain are limited to certain models or are inseparable from specific concept drift detection methods due to their integration with them. For example, the method proposed in [15] approaches concept drift detection and adaption as a whole and relies on two learners: one stable learner that learns long-term information responsible for prediction and one reactive learner that learns recent information used as a concept drift indicator. When the stable learner performs worse than the reactive learner, it indicates the emergence of concept drift and the stable learner gets retrained. Taking the severity of concept drift into consideration, Wang et al [16] proposed a method that can automatically select the best retraining and tuning strategies and find adaptive iterations to maintain a Gradient Boosting Decision Tree (GBDT) according to the severity of the concept drift. However, it is only limited to the GBDT model. Similarly, CVFDT [17], the extension of the Very Fast Decision Tree (VFDT) [18] classifier, along with later extensions [19]–[21], could effectively adapt to new concepts by only updating a part of the model, could not be generally applied to concept adaption problem.

Due to the lack of flexibility and generalizability, none of the current works could be applied to the aforementioned maintenance problem of the *enhanced models* in DML systems. Targeting this issue, we contribute with methods to maintain the performance of *enhanced models* in a DML system that operates on multiple data sources in the case of partial concept drifts (where concept drift only happens in one of the sources). We, therefore, propose strategies that yield different trade-offs between the amount of data needed to be transmitted and the performance of the maintained models and investigate the impact of the maintenance on different sources of data.

III. Rationale & Problem Definition

We report on the fundamentals of our proposed method by reporting on preliminaries from our previous work [8]. Consider a DML system that is providing predictive services in an EC environment and consists of n edge nodes (referred to as *nodes* hereinafter): $\mathcal{N} = \{N_1, \ldots, N_n\}$. While all the nodes are working on the same kind of predictive tasks, each node does work independently on its local data $D_i = \{(\mathbf{x}, y)_\ell\}_{\ell=1}^{L_i}$, with L_i input-output pairs $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$. The input $\mathbf{x} = [x_1, \ldots, x_d]^\top \in \mathbb{R}^d$ is a d-dim. feature vector, which is assigned to output $y \in \mathcal{Y}$ used for regression (e.g., $\mathcal{Y} \subseteq \mathbb{R}$) or classification predictive tasks (e.g., $\mathcal{Y} \subseteq \{-1, 1\}$). In the regression case, given a query input $\mathbf{x}$ to node N_i, the error of the predicted outcome $f_i(\mathbf{x}) = \tilde{y}$ is defined as $\tilde{y} - y$, where y is the actual output. The neighborhood of N_i, $\mathcal{N}_i \subseteq \mathcal{N} \setminus \{N_i\}$, is a subset of nodes which communicate directly with N_i. The collection of the data of all the nodes in $\mathcal{N}_i$ is defined as $\mathcal{D}_i = \{D_j\}, \forall j, j \neq i$. Moreover, we assume each node N_i is equipped with a *local model* f_i that is built purely on D_i.

We first introduced the concept of *enhanced models* in [8]. That is, for each node N_i and its neighbouring nodes $N_j \in \mathcal{N}_i$, we build *enhanced data* $\bar{D}_i^s = D_i \cup \{\Gamma^s(D_j)\}, \forall j, j \neq i,$

in which $\Gamma^s(D_j)$ represents the samples of representative information extracted from N_j with a strategy s. Subsequently, *enhanced model* $\bar{f}_i^s$ is defined as the model trained over $\bar{D}_i^s$ that has its data sampled from neighbouring nodes with strategy s (in which s controls how we sample the information, e.g. with a random sampling strategy, we sample random points within the original data).

For an enhanced model $\bar{f}_i^s$ that operates on D_i and $\mathcal{D}_i$, the underlying distribution of arbitrary neighbouring node's data $D_j \in \mathcal{D}_i$ could change over time (i.e., concept drift happens). Given a certain threshold ε of error that the system could tolerate (i.e., $|f(\mathbf{x}) - \mathbf{y}| - |f(\mathbf{x}') - \mathbf{y}'| > \varepsilon$ means that the model has to be maintained to adapt to the new concept as it induced intolerable error), the errors yielded by all the affected models when facing the same concept drift are not likely to be the same. Therefore, not all the models affected by the drift are necessarily needed to be maintained.

Problem 1: *We seek to investigate how concept drifts of different degrees affect the performances of different kinds of models and most importantly, the enhanced models, distinctly.*

In our investigation, since we are focusing on different kinds of models' reactions to the concept drift, the detection and patterns (as it has been identified in [10], includes sudden/abrupt, incremental, gradual and recurring) of it are irrelevant in the context. But the types/classes of concept drift do play an important role. The existing works do not reach an absolute consensus on the terminology of the types of concept drift. So, in this paper, we are dealing with three types of concept drift that have the definition as follows. Given that $\mathbf{x}$, $\mathbf{y}$ are the original input and output and $\mathbf{x}'$, $\mathbf{y}'$ are the input and output at the time when concept drift occurs, then, the three types of drift are defined based on the associated probability distributions $P(\mathbf{x})$ and $P(\mathbf{y})$, and conditional probability $P(\mathbf{y}|\mathbf{x})$ as follows:

- Virtual Drift: $P(\mathbf{x}) \neq P(\mathbf{x}') \wedge P(\mathbf{y}) = P(\mathbf{y}')$
- Actual Drift: $P(\mathbf{x}) \neq P(\mathbf{x}') \wedge P(\mathbf{y}) \neq P(\mathbf{y}')$
- Total Drift: $P(\mathbf{x}) \neq P(\mathbf{x}') \wedge P(\mathbf{y}) \neq P(\mathbf{y}') \wedge P(\mathbf{y} \mid \mathbf{x}) \neq P(\mathbf{y}' \mid \mathbf{x}')$

When the enhanced model needs to be maintained, this can be achieved by incremental learning or training from scratch.

Problem 2: *Since the enhanced model is built with data from other nodes and operates on multiple nodes, we seek strategies to extract statistical information from the new drifted data to maintain the model effectively and efficiently.*

Let the enhanced model $\bar{f}_i$ reside on node N_i and a concept drift occur on node N_k. Consider also a neighbouring node N_j whose data D_k are not drifted; $\{N_k, N_j\} \in \mathcal{N}_i$. The $\bar{f}_i$ model has been built on node N_i in case of the N_k and/or N_j failures. Consider now the *maintained* enhanced model $\bar{f}_i'$ using either incremental learning or training from scratch using the drifted data $D_k\prime$ of node N_k. We then obtain the two prediction errors (losses $\mathcal{L}$) by assessing the performance of $\bar{f}_i'$ to drifted data $D_k\prime$ (from node N_k, $\mathbb{E}\mathcal{L}(\bar{f}_i'(D_k\prime))$, and to non-drifted data D_j (from node N_j), $\mathbb{E}\mathcal{L}(\bar{f}_i'(D_j))$. The objectives of the maintainability process are then:

O1: Minimize $\mathbb{E}\mathcal{L}(\bar{f}_i'(D_k\prime))$ (for node N_k).

O2: Minimize $\mathbb{E}\mathcal{L}(\bar{f}_i'(D_j))$ (for node N_j).

O3: Reduce inter-node data transfer between nodes N_i and N_k during enhanced model maintenance.

IV. Partial Concept Drift Impact

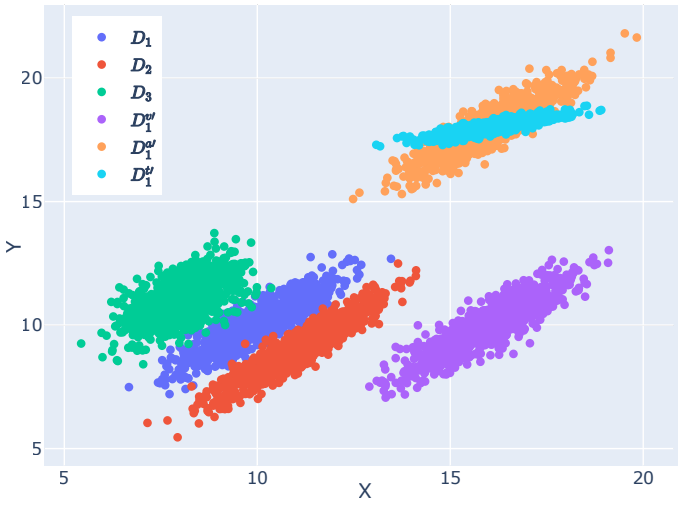

Fig. 1. Distributions of the artificially generated data

A. Investigation Scenario Set-up

To simulate concept drift in a controlled manner and provide insights on the impact of the concept drifts in the enhanced models, we created an artificial dataset that is comprised of three nodes $\{N_1, N_2, N_3\}$ with data retrieved by the corresponding Gaussian distributions while manipulating the means and the covariance matrices. The results are shown in Figure 1. In this setup, D_1 serves as the baseline to generate D_2 and D_3. Given the covariance matrix and the means adopted to generate D_i are denoted by cov_i and $\mathbf{c}_i$, cov_2, $\mathbf{c}_2$ and cov_3, $\mathbf{c}_3$ are acquired by adding a 20% to 30% random noise to cov_1, $\mathbf{c}_1$. Concept drift was only introduced to D_1, so drifted data of three types were generated based on this distribution. For virtual and actual drifted D_1 (**denoted with $\boldsymbol{D_1^{v\prime}}$ and $\boldsymbol{D_1^{a\prime}}$**), the covariance matrices used are the same with cov_1. Following the definition, the virtual and actual drifted D_1 have their centres shifted along the x-axis and both the x- and y-axis, respectively. The total drifted D_1 (**denoted with $\boldsymbol{D_1^{t\prime}}$**) shares the same centre as the actual drifted D_1 to control the variables while having a different covariance matrix. Specifically, drifted data were shifted to where it barely has any intersections with D_1, D_2 and D_3 in x-axis to avoid the case in which the same input corresponds to different outputs to make it possible for good retraining results.

B. Building Enhanced Models

When building the enhanced models, we are focusing on those that reside on node N_2. As the main purpose of the enhanced models is to serve as surrogate models for other nodes, $\bar{f}_2$ could help us to gain insights into the enhanced model's performance on a drifted node (N_1) and a normal

node (N_3). To understand how different strategies used to build the enhanced model affect its performance on the drifted data, we built the enhanced model with two different strategies mentioned in our previous paper [8], that is Global Sampling (GS) strategy and Centroid Guided (CG) strategy. GS is based on random sampling over the data of its neighbouring nodes. For each node $N_j \in \mathcal{N}_i$, we acquire $\Gamma(D_j)$ by randomly sampling $\alpha|D_j|$ points. Here α represents the mixing rate that directly controls the proportion of data being selected in the sampling process. The enhanced data is then produced by $\bar{D}_i = D_i \cup \Gamma(D_j)$, $\forall j, j \neq i$. As for CG, instead of sampling real data to build the enhanced data, we are using the centroids (cluster heads) acquired by applying vector quantization (clustering) to D_j. That is, for each $N_j \in \mathcal{N}_i$, we quantize the entire D_j into K clusters with respect to α and $|D_j|$ (i.e. $k = \alpha|D_j|$). Then, for each cluster, we have a centroid (cluster head) $\mathbf{w}_{jk}$ that represents the centre of the cluster. $\Gamma(D_j)$ is consequently defined by:

$$\Gamma(D_j) = \cup_{k=1}^{K}\{\mathbf{w}_{jk}\}. \quad (1)$$

C. Effects of Concept Drift on Enhanced Model's Performance

We focus on an abrupt concept drift on the dataset, e.g., N_1, by concatenating D_1 and three types of drifted D_1 ($D_1^{v\prime}$, $D_1^{a\prime}$ and $D_1^{t\prime}$) and applied $\bar{f}_2^{GS}$ and $\bar{f}_2^{CG}$ on them. We then compare the results with the local model f_1 to see if they are influenced by the concept drift any differently. Specifically, to rule out the influence of different ML models, for each $\bar{D}_i$, we feed it to a Support Vector Regression (SVR) model and a Gradient Boosting Regression (GBR) model (all local models are built with SVR by default).

TABLE I
PERFORMANCE OF DIFFERENT MODELS

	RMSE			
Model	D_1	$D_1^{v\prime}$	$D_1^{v\prime}$	$D_1^{v\prime}$
f_1	0.47	1.29	8.06	7.93
$\bar{f}_2^{GS}(SVR)$	1.73	1.69	7.57	7.41
$\bar{f}_2^{CG}(SVR)$	1.69	1.68	7.57	7.41
$\bar{f}_2^{GS}(GBR)$	1.54	2.19	6.26	6.12
$\bar{f}_2^{CG}(GBR)$	1.50	2.17	6.29	6.15

$D_1^{v\prime}$ corresponds to virtual drifted D_1
$D_1^{a\prime}$ corresponds to actual drifted D_1
$D_1^{t\prime}$ corresponds to total drifted D_1

The performances yielded by all the models in terms of Root-Mean-Square Error (RMSE) are shown in Table I. We also plotted the point-wise absolute errors of them to visualize the trends. From Figures 2 to 6, we can observe that all the models have similar trends when they are facing the same kind of concept drift (the **light red** regions on the right of the figures represent the drifted data). In this setup, the impact of concept drift did not seem to be affected much by the strategies used to build the enhanced models (Figures 3 to 6), as GS and CG provided similar results. Compared to the local model f_1 (Figure 2), being more generalized, the enhanced models performed better on more severe drift types ($D_1^{a\prime}$ and $D_1^{t\prime}$) while being worse on $D_1^{v\prime}$. Note that enhanced models

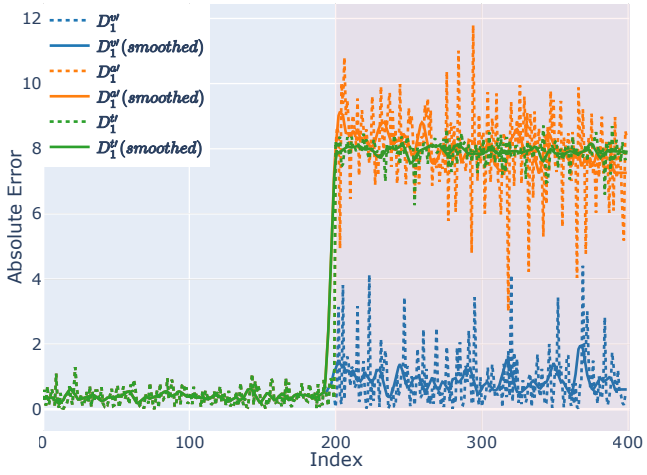

Fig. 2. The performance of f_1 on D_1, $D_1^{v\prime}$, $D_1^{a\prime}$ and $D_1^{t\prime}$

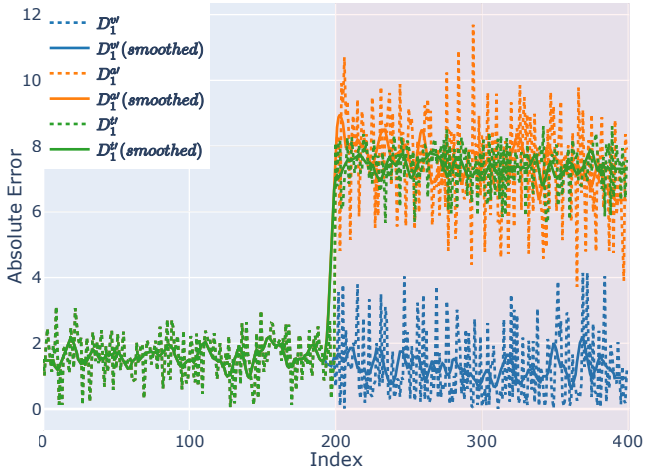

Fig. 3. The performance of $\bar{f}_2^{GS}(SVR)$ on D_1, $D_1^{v\prime}$, $D_1^{a\prime}$ and $D_1^{t\prime}$

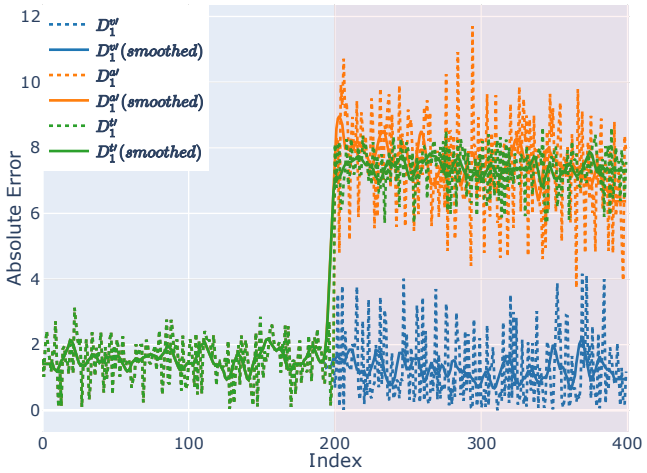

Fig. 4. The performance of $\bar{f}_2^{CG}(SVR)$ on D_1, $D_1^{v\prime}$, $D_1^{a\prime}$ and $D_1^{t\prime}$

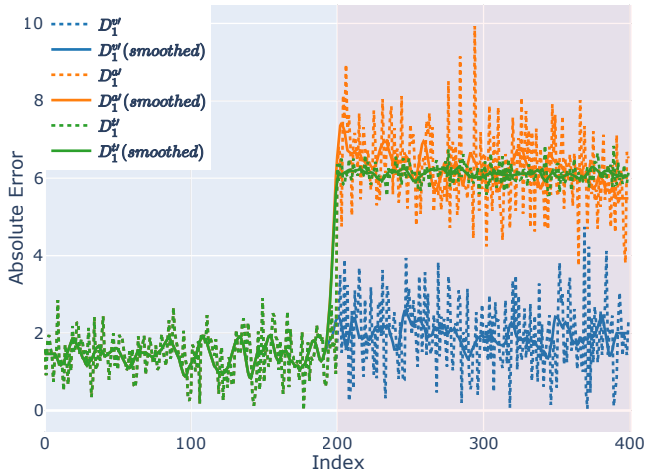

Fig. 5. The performance of $\bar{f}_2^{GS}(GBR)$ on D_1, $D_1^{v\prime}$, $D_1^{a\prime}$ and $D_1^{t\prime}$

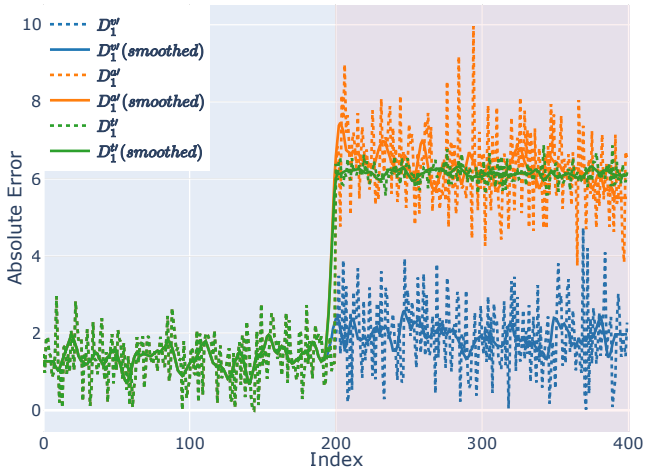

Fig. 6. The performance of $\bar{f}_2^{CG}(GBR)$ on D_1, $D_1^{v\prime}$, $D_1^{a\prime}$ and $D_1^{t\prime}$

did not perform well on D_1, indicating the relatively huge difference between D_1, D_2 and D_3, as they do have a low level of overlap. Thus, from the perspective of the enhanced models, D_1 and $D_1^{v\prime}$ are very similar. They did provide very close results on D_1 and $D_1^{v\prime}$. Furthermore, as an ensemble model, GBR (Figures 5 and 6) demonstrated better generalizability, and performed noticeably better on $D_1^{a\prime}$ and $D_1^{t\prime}$ as compared to SVM (Figures 3 and 4). From this investigation, we see that if we use the relative drops in performance to trigger the model retraining, for $D_1^{v\prime}$, the retraining for f_1 does not necessarily mean the retraining for the enhanced models that operate on D_1 need to be retrained as well. However, for $D_1^{a\prime}$ and $D_1^{t\prime}$, the reaction of the local model f_1 is well in line with the enhanced models.

V. Model Maintainability Strategies

To adapt the models to the new concept, we sample from the drifted data and retrain the models with the samples. The rationale for this process is identical to the strategies with those used to extract signature information and train the enhanced models at first. The only difference is, instead of sampling from the initial data D, we sample from drifted data D'. In this way, GS and CG could be directly used for the maintenance of the enhanced models. Moreover, we are also introducing the Mock Data (MD) strategy and the Enhanced Centroid Guided (ECG) strategy to experiment with how we reduce the amount of inter-node data transmission while maintaining the effectiveness of the retraining. Specifically, for MD, the inter-node transmission of real data can be avoided as MD only requires the transmission of the local model and some statistical information. That is, given the collection of all the input-output pairs of D_j are denoted with $\mathcal{X}_j$ and $\mathcal{Y}_j$, we calculate the average of $\mathcal{X}_j$ as $\boldsymbol{\mu_j} = \frac{\sum_{m=1}^{|D_j|} \mathbf{x}_m}{|D_j|} \in \mathbb{R}^d$ and the standard deviation of $\mathcal{X}_j$ as $\boldsymbol{\sigma_j} = \sqrt{\frac{1}{|D_j|}\sum_{m=1}^{|D_j|}(\mathbf{x}_m - \mu_j)^2} \in \mathbb{R}^d$. We also compute the Standard Error of the Mean (SEM) on $\mathcal{Y}_j$ as:

$$\bar{\boldsymbol{\sigma}}_{\boldsymbol{j}} = \frac{\sqrt{\frac{1}{|D_j|}\sum_{m=1}^{|D_j|}(\mathbf{y}_m - \frac{\sum_{m=1}^{|D_j|}\mathbf{y}_m}{|D_j|})^2}}{\sqrt{|D_j|}} \in \mathbb{R}^d. \quad (2)$$

Such statistical information and the local model f_j are then sent to N_i to build enhanced models therein. N_i then samples $\alpha|D_j|$ vectors from the Gaussian distribution of $\mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2)$ to obtain the fabricated training inputs $\hat{\mathcal{X}}_j$. The corresponding training outputs are acquired by $\hat{\mathcal{Y}}_j = f_j(\hat{\mathcal{X}}_j) + \boldsymbol{\epsilon_j}$. In which $\boldsymbol{\epsilon_j}$ is the added random noise sampled from Gaussian distribution of $\mathcal{N}(0, \bar{\boldsymbol{\sigma}}_j^2)$. In this way, we have:

$$\Gamma(D_j) = \{(\hat{\mathcal{X}}_j, \hat{\mathcal{Y}}_j) : \hat{\mathcal{X}}_j \sim \mathcal{N}(\boldsymbol{\mu}_j, \boldsymbol{\sigma}_j^2), \hat{\mathcal{Y}}_j = f_j(\hat{\mathcal{X}}_j) + \epsilon_j\} \quad (3)$$

With MD, we expect the amount of inter-node transmission to be cut down by an evident margin with a larger value of $\alpha|D_j|$. While with a smaller value of it, considering the size of the model, MD may not help in reducing the amount of data transmitted.

The ECG works similarly to CG. However, instead of quantizing D_j into $\alpha|D_j|$ clusters directly, we introduce a new parameter called *intensity* λ to control the number of clusters as well as the duplication process later. That is, we define the number of clusters $K = \frac{\alpha|D_j|}{\lambda}$. N_j first quantizes D_j into K clusters and the centroids $\{\mathbf{w}_{jk}\}$ are sent to N_i. Then, for each centroid $\mathbf{w}_{jk}$ in $\{\mathbf{w}_{jk}\}$, N_i sample $\lambda - 1$ points $(\hat{\mathbf{x}}, \hat{\mathbf{y}})$ from $\mathcal{N}(\mathbf{w}_{jk}, \boldsymbol{\sigma}_j^2)$. Here, the standard deviation $\boldsymbol{\sigma}_j$ could be set to an arbitrary value as long as it is small enough to ensure the sampled point does not fall far from the original distribution of D_j. Then, the final sample used to maintain the model is obtained by aggregating all the centroids and the sampled points.

$$\Gamma(D_j) = \cup_{k=1}^{K}\{\mathbf{w}_{jk} \cup \{(\hat{\mathbf{x}}, \hat{\mathbf{y}}) \sim \mathcal{N}(\mathbf{w}_{jk}, \boldsymbol{\sigma}_j^2)\}\}. \quad (4)$$

VI. Experimental Evaluation

We provide a comprehensive experimental evaluation to examine how the proposed model maintenance process in a DML system affects the models' performance on the new concept as well as the data from unchanged nodes. We

furthermore investigate how cutting down the amount of inter-node data transmission influences the maintenance process.

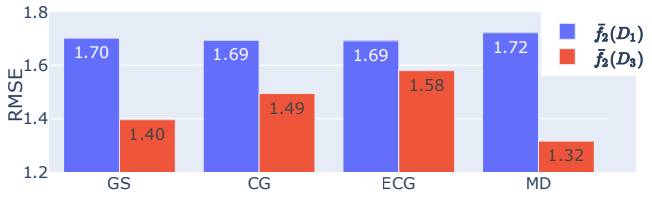

Fig. 7. The performance of $\bar{f}_2^{GS}$, $\bar{f}_2^{CG}$, $\bar{f}_2^{ECG}$, $\bar{f}_2^{MD}$ on D_1 and D_3

A. Synthetic Dataset

As shown in Figure 7, though the enhanced models of N_2 built with different strategies performed similarly on D_1, they did yield noticeably different results on D_3. To control the variables in these experiments regarding enhanced model maintenance, we (i) build all the enhanced models with GS strategy, and (ii) use SVR to build all the models. Hence, we first built an enhanced model $\bar{f}_2^{GS}$. Then, for each type of the concept drift in N_1, we extract input-output pairs from the drifted data with all four strategies (GS, CG, ECG, MD) and retrain $\bar{f}_2^{GS}$ to get the four maintained models, respectively.

Since our focus here is the enhanced models' performance on D_1' and D_3, we plotted their performance *before* and *after* the maintenance with different kinds of drifted data, shown in Figures 8-13. Note: even though all the maintained models are maintained on the enhanced model built with GS strategy, the 'before maintenance' results are yielded by enhanced models built with different strategies to demonstrate more comprehensive results.

In Figure 8, by comparing the blue bars and red bars, we could see that the performance of $\bar{f}_2$ barely dropped after it encounters the virtual drifted D_1 ($D_1^{v\prime}$), which is oftentimes the sign of the model not needing maintenance. However, as the purple bars indicate, the maintenance with $D_1^{v\prime}$ could help to cut down the error on $D_1^{v\prime}$ ($\mathbb{E}\mathcal{L}(\bar{f}_2'(D_1^{v\prime}))$) almost in half while maintaining the performance on D_1 ($\mathbb{E}\mathcal{L}(\bar{f}_2'(D_1))$, indicated by green bars) and D_3 ($\mathbb{E}\mathcal{L}(\bar{f}_2'(D_3))$, shown in Figure 11) at the same level. This is the case where the model maintenance should be triggered, but could not be triggered by the performance drop yielded by the enhanced models. In this case, the local model f_1 was able to provide a more accurate representation of the need for model maintenance.

The enhanced models' performance before/after the maintenance with actual and total drifted data are very close. As shown in Figures 9 and 10, before the maintenance, the $\bar{f}_2$ models built with all four strategies could not handle $D_1^{a\prime}$ and $D_1^{t\prime}$ well, generating results with several times of error as compared to $\mathbb{E}\mathcal{L}(\bar{f}_2(D_1))$. However, after maintenance, the enhanced models once again reduced the error on $D_1^{a\prime}$ and $D_1^{t\prime}$ to a very low level, almost one-third of $\mathbb{E}\mathcal{L}(\bar{f}_2(D_1))$, which are strong evidence showing the effectiveness of the maintenance with both $D_1^{a\prime}$ and $D_1^{t\prime}$. Moreover, as shown in Figures 12 and 13, this maintenance does not ruin the models' performance on D_3, which means that after maintenance, $\bar{f}_2'$ could still serve as a valid surrogate model in the case of the failures of N_1 and N_2.

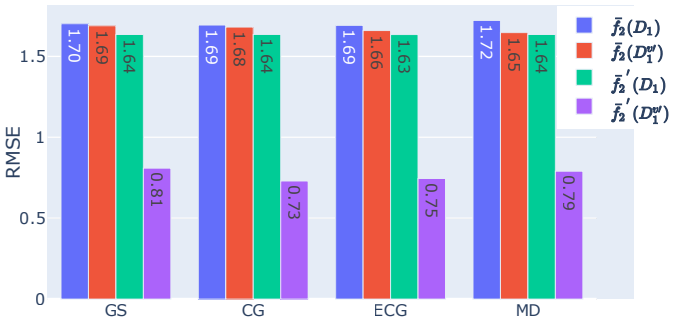

Fig. 8. The performance of $\bar{f}_2$, $\bar{f}_2'$ (retrained with $D_1^{v\prime}$) on D_1 and $D_1^{v\prime}$

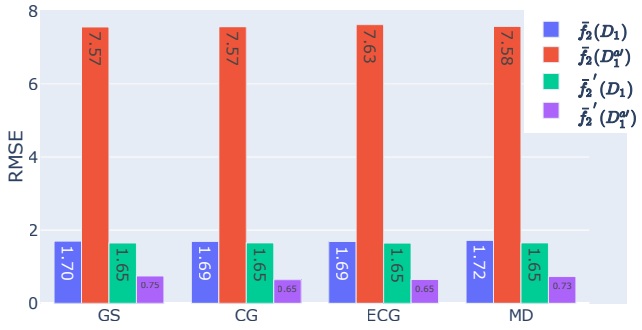

Fig. 9. The performance of $\bar{f}_2$, $\bar{f}_2'$ (retrained with $D_1^{a\prime}$) on D_1 and $D_1^{a\prime}$

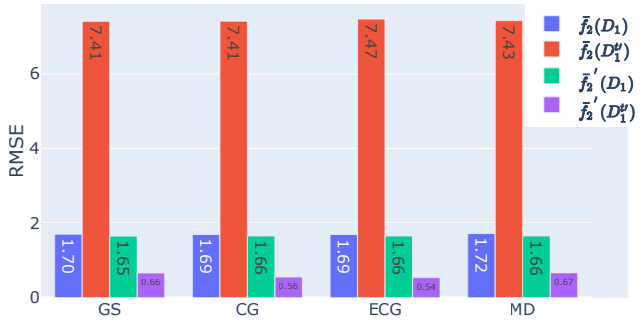

Fig. 10. The performance of $\bar{f}_2$, $\bar{f}_2'$ (retrained with $D_1^{t\prime}$) on D_1 and $D_1^{t\prime}$

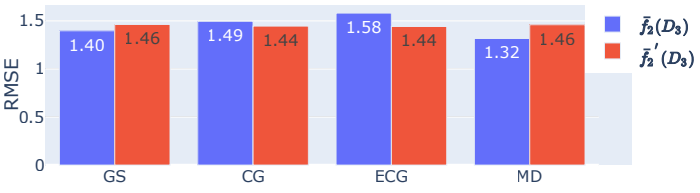

Fig. 11. The performance of $\bar{f}_2$, $\bar{f}_2'$ (retrained with $D_1^{v\prime}$) on D_3

To investigate how the ECG strategy achieves a reduction in inter-node data exchanges in the maintenance process, we

Fig. 12. The performance of $\bar{f}_2$, $\bar{f}_2'$ (retrained with $D_1^{a\prime}$) on D_3

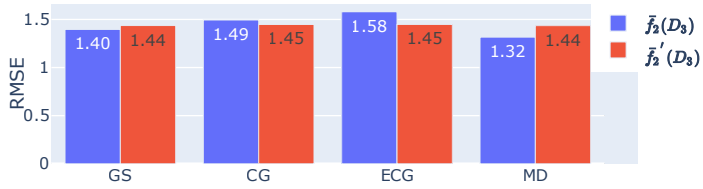

Fig. 13. The performance of $\bar{f}_2$, $\bar{f}_2'$ (retrained with $D_1^{t\prime}$) on D_3

experimented with extracting data used to perform maintenance with the ECG strategy of different intensities λ and compared it with the other three strategies. Specifically, for MD, as the model size is at an equivalent scale of all the data points needed to be transferred, it is considered to transfer the same amount of data as GS and CG. The results are shown in Figures 14 and 15. In these figures, the bottom left area means lower error and a smaller amount of data transferred. As shown in Figure 14, for $D_1^{v\prime}$, $D_1^{a\prime}$ and $D_1^{t\prime}$, the lowest errors are all yield by ECG with *only* 5% to 20% of the data transferred, which showcased ECG's potential in performing models maintenance effectively. Moreover, ECG is also very helpful for not deteriorating the maintained models' performance on D_3. As showcased in Figure 15, all the best results in terms of RMSE are acquired with ECG. Although holistically speaking, the maintained models' performance on D_3 is not sensitive to what strategy we use (as the maximum and minimum RMSE only differ around 0.02).

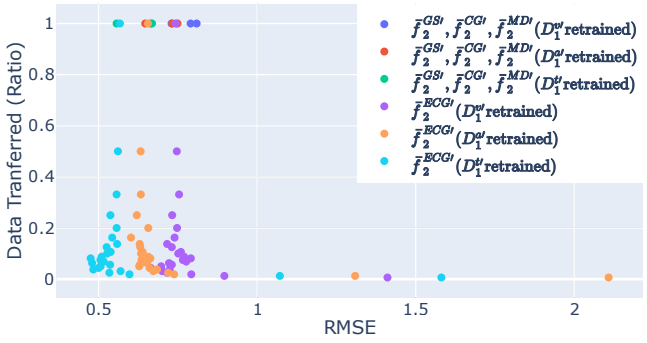

Fig. 14. The performance of $\bar{f}_2'$ on $D_1^{v\prime}$, $D_1^{a\prime}$ and $D_1^{t\prime}$ given different ratio of the amount of data transferred

B. Real Dataset

We also evaluate our resilience framework over real multi-node datasets adopted by [22]. The dataset consists of mobile sensors readings over four Unmanned Surface Vehicles

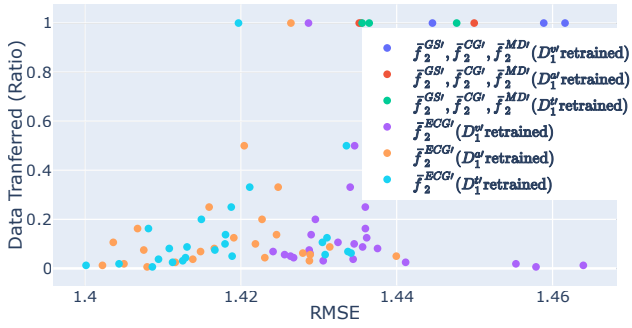

Fig. 15. The performance of $\bar{f}_2'$ on D_3 given different ratio of the amount of data transferred

(USVs), floating over the sea surface in a testbed in Athens, Greece. Each USV (node) records the measurements such as humidity and temperature of the sea surface, each of which represents an edge node within a DML environment. For our experiments, the local data gathered by two of the USVs are employed notated by D_1 and D_2. Using as input the temperature variable $x \in \mathbb{R}$, we are seeking to predict the humidity which acts as our output variable $y \in \mathbb{R}$.

Similarly, a concept drift only happens in D_1 local data, encompassing the three different drift types mentioned, virtual, actual and total. Using all four model maintenance strategies presented, enhanced models are constructed for each node, evaluating their response to concept drift. Subsequently, we retrain the $\bar{f}_2$ for all the strategies against all the drifted types, thus, maintaining high predictability analytics performance over the environment. A series of enhanced models are constructed using the ECG strategy by varying the number of cluster centroids transferred over the network while maintaining a constant number of data samples used for the training of the enhanced models. Therefore, we can evaluate whether our approach sacrifices performance for less transferred data.

Figure 16 illustrates the RMSE performance of the $\bar{f}_2$ enhanced model after maintenance over drifted D_1 data types using the different maintenance strategies. One could observe that the severity of the drift types correlates with the increase of the RMSE value, regardless of the value of the data transferred ratio. However, as evidenced, the data compression ratio over the ECG strategy is not associated in a proportional manner with the overall performance of the enhanced model. The results demonstrate that the enhanced model's performance is adversely affected by either a very small or a very large number of cluster centroids used. In the latter, the high number of cluster centroids and hence the small number around them, overfit the enhanced models. While, in the former case, there are too few cluster centroids to adequately capture the characteristics of the dataset, leading to under-fitting. Accordingly, the best trade-off lies within the 0.1 data transmission ratio, which scores a lower RMSE for each type of drift. In this way, our framework achieves similar performance with GS, CG, and MD strategies while transferring 10 times fewer data in the network.

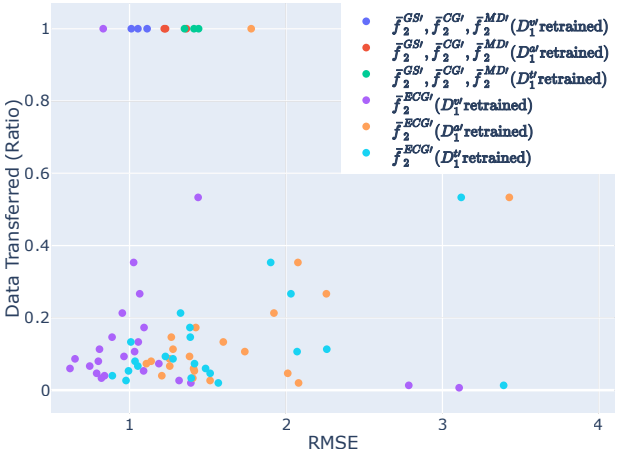

Fig. 16. The performance of $\bar{f}_2'$ on $D_1^{v\prime}$, $D_1^{a\prime}$ and $D_1^{t\prime}$ given different ratio of the amount of data transferred over the GNFUV dataset.

VII. Discussion & Conclusions

We investigated the model maintenance problem in the case of concept drift in DML environments. We introduced enhanced models to build resilience to node failures for models operating in such environments. Such models are trained on and operate over data from multiple nodes' data. The challenge we tackled is to maintain their performance in the presence of drifts as they may emerge in several data sources that the enhanced models operate on.

We conducted experiments to explore how the enhanced models react to different types of concept drifts and compared the results against the performance of nodes' local models. It turned out that, by being generalizable, the enhanced models' performance was hardly affected by simple concept drifts and got impacted less by harder concept drift types compared to local models' performances. We also found this trend is barely affected by what kind of model we adopted and what strategies we used to build the enhanced model. This makes it possible for concept drifts to be handled in a unified way, i.e., model retraining with the drifted data. Therefore, we performed model maintenance by retraining the enhanced models with information extracted from different kinds of drifted data with multiple strategies proposed. The results showcased the effectiveness of maintenance as in all of the scenarios tested, the maintained models' performance on the drifted data was significantly improved while not ruining the performance on the other node's data. This indicates that, after maintenance, the enhanced model could gain the ability to operate on the new concept while not losing the capability to operate on the node where no concept drift happens. Furthermore, the proposed ECG demonstrated its potential in reducing the amount of inter-node data transfer while achieving better results in maintaining the model. This further reduced the cost of performing model maintenance.

References

[1] E. Welbourne, L. Battle, G. Cole, K. Gould, K. Rector, S. Raymer, M. Balazinska, and G. Borriello, "Building the internet of things using rfid: the rfid ecosystem experience," *IEEE Internet computing*, vol. 13, no. 3, pp. 48–55, 2009.

[2] L. Li and J. Liu, "An efficient and flexible web services-based multidisciplinary design optimisation framework for complex engineering systems," *Enterprise Information Systems*, vol. 6, no. 3, pp. 345–371, 2012.

[3] C. Yin, Z. Xiong, H. Chen, J. Wang, D. Cooper, and B. David, "A literature survey on smart cities," *Science China Information Sciences*, vol. 58, no. 10, pp. 1–18, 2015.

[4] R. Brauneis and E. P. Goodman, "Algorithmic transparency for the smart city," *Yale JL & Tech.*, vol. 20, p. 103, 2018.

[5] A.-R. A. Audu, A. Cuzzocrea, C. K. Leung, K. A. MacLeod, N. I. Ohin, and N. C. Pulgar-Vidal, "An intelligent predictive analytics system for transportation analytics on open data towards the development of a smart city," in *Conference on Complex, Intelligent, and Software Intensive Systems*. Springer, 2019, pp. 224–236.

[6] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[7] J. Bram and A. McKay, "The evolution of commuting patterns in the new york city metro area," *Current Issues in Economics and Finance*, vol. 11, no. 10, 2005.

[8] Q. Wang, J. M. Fornes, C. Anagnostopoulos, and K. Kolomvatsos, "Predictive model resilience in edge computing," in *IEEE 8th World Forum on Internet of Things*, 2022.

[9] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen, and F. Petitjean, "Characterizing concept drift," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, 2016.

[10] J. a. Gama, I. Žliobaitundefined, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, mar 2014. [Online]. Available: https://doi.org/10.1145/2523813

[11] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2018.

[12] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Brazilian symposium on artificial intelligence*. Springer, 2004, pp. 286–295.

[13] A. Bifet and R. Gavaldà, *Learning from Time-Changing Data with Adaptive Windowing*, pp. 443–448. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611972771.42

[14] L. L. Minku, A. P. White, and X. Yao, "The impact of diversity on online ensemble learning in the presence of concept drift," *IEEE Transactions on knowledge and Data Engineering*, vol. 22, no. 5, pp. 730–742, 2009.

[15] S. H. Bach and M. A. Maloof, "Paired learners for concept drift," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 23–32.

[16] K. Wang, J. Lu, A. Liu, Y. Song, L. Xiong, and G. Zhang, "Elastic gradient boosting decision tree with adaptive iterations for concept drift adaptation," *Neurocomputing*, vol. 491, pp. 288–304, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231222003320

[17] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2000, pp. 71–80.

[18] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, 2001, pp. 97–106.

[19] J. Gama, R. Rocha, and P. Medas, "Accurate decision trees for mining high-speed data streams," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 523–528.

[20] H. Yang and S. Fong, "Incrementally optimized decision tree for noisy big data," in *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, ser. BigMine '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 36–44. [Online]. Available: https://doi.org/10.1145/2351316.2351322

[21] ——, "Countering the concept-drift problems in big data by an incrementally optimized stream mining model," *Journal of Systems and Software*, vol. 102, pp. 158–166, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121214001526

[22] N. Harth and C. Anagnostopoulos, "Edge-centric efficient regression analytics," in *2018 IEEE International Conference on Edge Computing (EDGE)*, 2018, pp. 93–100.