Predictive Model Resilience in Edge Computing

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

Jordi Mateo Fornes
Computing Science
University of Glasgow
jordi.mateo@udl.cat

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

Kostas Kolomvatsos
Computer Science & Telecomm.
University of Thessaly
kostasks@uth.gr

***Abstract*—Node failure is a commonly seen threat in distributed Machine Learning systems. It is hard to predict having a huge negative impact on system availability to provide e.g., predictive analytics. Considering the benefits obtained from reduced latency and bandwidth overhead in Edge Computing (EC), invocation of the Cloud should be avoided. Hence, finding the best substitute nodes at the network edge to be invoked instead of failing nodes, evidently, builds the system's resilience upon node failures. To achieve this goal, we contribute with a resilience mechanism that relies on several data-mixing strategies that build enhanced models in each node. Such models have satisfactory prediction capabilities to handle failing nodes' predictive tasks, thus, ensuring resilience in predictive services. Furthermore, we propose a graph-driven approach to guide node invocation minimising the performance loss upon node failures. Our performance evaluation and comparative assessment showcase the applicability of our model resilience approach in intelligent EC.**

***Index Terms*—Edge Computing, Edge Intelligence, resilience, Machine Learning**

I. Introduction

The Internet of Things (IoT) prevalence raises the demand for pushing Artificial Intelligence (AI) at the network edge shaping the Edge-AI synergy (Edge AI). As data are produced by IoT devices, local processing and learning at the edge of the network, e.g., local predictive model training for outlier detection, classification, and clustering, cuts down significant costs brought by transmitting and storing data to Cloud [1]. In an Edge-AI environment, *edge nodes* locally build Machine Learning (ML) models to provide predictive services like novelty detection, forecasting and classification. Edge Computing (EC) is aimed to address the limitations of the Cloud in supporting delay-sensitive, real-time decision making, and context-aware services [2]. Due to spatial and temporal differences that edge nodes naturally have, the statistical characteristics of their data significantly vary [3]. Typically, edge nodes equipped with ML models perform predictive tasks solely on local data. Therefore, one node's models could perform well on local data, while not being accurate enough on other nodes' data. This is expected as local models are not intended to be highly generalizable due to training over local data. Though, such a local model learning strategy achieves ideal performance in terms of predictability. However, in real EC systems, nodes can e.g., go offline due to intermittent communication with IoT devices or fail and interrupt their services [4]. This situation leads to nodes' unavailability and Service Level Objective (SLO) violations that can interrupt critical services and seek significant financial losses. In these cases, the Edge-AI environment should not cease its predictive services dealing with the potential lack of generalizability of the nodes' local models. Establishing a resilient distributed mechanism to continue supporting predictive services at high predictive quality standards is deemed appropriate even if failed nodes' services become unavailable.

We focus on an Edge-AI paradigm that selectively chooses data and statistics to be disseminated among peer nodes and, accordingly, adapts their models ensuring resilience in predictive services in case of nodes unavailability. In the case of a (temporarily) unavailable node, its data could be transferred to other nodes for processing and model training or to Cloud imposing extra communication costs and delays. However, due to the local model learning strategy and inherent lack of appropriate model generalizability, the substitute nodes tend to have less than ideal performances when dealing with 'unfamiliar' data coming from failing/unavailable nodes. This turns out to be a challenging problem to deliver robust and resilient services. A baseline solution to tackle this problem would be to add part of (training) data from other nodes to each node. Then, one could anticipate nodes' models' ability to process other nodes' data in light of increasing models' robustness and generalizability. This would render the system resilient for predictive services, where available nodes' models could provide predictive services on behalf of unavailable nodes. However, it is unknown which nodes and data (or which part of data) should be transferred that could obtain satisfactory model accuracy and a high level of resilience. Moreover, statistical diversity in datasets and data split strategies further add uncertainty to that problem.

In this paper, we contribute with a universal approach that aims to investigate data and statistics mixing strategies for nodes to build robust models supporting resilience in predictive services in case of node failures/unavailability. Our method tackles the lack of generalizability of nodes' local models via novel training data and statistics mixing strategies that adapt to the diversity of data characteristics. This yields a trade-off between the locality of modelling and the generalizability of locally adapted models (i.e., their ability to support failing nodes). Furthermore, our method leads to minimizing response time and networking latency to communicate from the edge to the Cloud and vice versa to serve queries/predictive analytics tasks. The experimental results showcase that our resilience strategy help EC systems to maintain close or attain even better predictability performance than baseline solutions (data transfer to Cloud followed by centralized model building) at a fairly high level of node failure probabilities. The paper is organized as follows: Section II reports on the related work and our technical contribution, Section III formulates our problem, while Section IV introduces our predictive model resilience mechanism in Edge-AI environments. Section V reports on the performance evaluation and comparative assessment, and Section VI concludes the paper.

II. Related Work & Contribution

The reliability of a system is concerned with the system's ability to perform its intended function correctly for a specified time period. At the same time, resilience is aimed at the ability to recover, mitigate and survive a particular failure. Therefore, resilience in a system mainly refers to the system's resistance to external attacks or tolerance to internal failures [5]. We focus on an EC system deployed across nodes being robust in providing predictive analytics services upon node failures. Similarly, such failures could be caused by internal factors like hardware malfunction, energy depletion, link failure and system crash or external adverse environment [6], [7] or even external malicious attacks [8]. Such cases render nodes of the system unavailable until specific solutions and mitigation are in place. Consider an EC system that serves prediction queries using ML models based on local data. System reliability refers here to the system's ability to serve queries/tasks performed by nodes along the time. In contrast, system resiliency is interpreted as the ability to mitigate a failure that compromises system reliability, i.e., when a failure occurs to the node required to serve the query. In our context, we leverage an approach aiming to provide predictive services while nodes become unavailable continuously. The idea is to enhance each node's model(s) by including *unfamiliar* training examples from other nodes' datasets (e.g., akin to adversarial training for improving robustness [9]). This is expected to enhance the generalizability of local nodes' models in light of being capable of providing predictions at the same accuracy as that would be achieved by the unavailable nodes. Our approach offers an ensemble of strategies to tackle queries to unavailable nodes by selecting the most appropriate alternative nodes to process these queries helping to improve the system's resilience. Nevertheless, models from alternative nodes may not be optimal ones in terms of accuracy compared to unavailable node's models. However, these models must be robust in tackling unfamiliar data providing similar accuracy in a system with failures as in the absence of failures.

From the domain adaptation perspective, models adapt to new but similar 'target' domains [10], [11] by solving the problem that training and testing data do not come from the same distributions [12]. Our approach is partially inspired by domain adaptation to ensure resilience upon failing nodes. Our strategies handle the local nodes' models (source domain) by enhancing them with unfamiliar data coming from other nodes (target domains). Particularly, such unfamiliar data are selected to be the most representative of the target domains to ensure high accuracy of the enhanced nodes' models. Such models are used as substitute models in case node failure occurs.

Moreover, our approach reassures consistency of the local models and *controlled* generalizability of the enhanced models tailored to be substitutes for unavailable nodes by avoiding training a global model for all nodes. This would jeopardize the local knowledge derived from local models per node and, evidently, the tailored capacity of the enhanced models to be accurate substitutes to unavailable nodes' local models. This is in principle different from Federated Learning (FL) paradigm fundamentals. In FL, the training processes are repeatedly achieved via data center-nodes communication for model aggregation, thus, deriving a global model for all [13]. Instead, in our approach, we build enhanced models being appropriate to substitute the local models of those nodes being unavailable to tackle incoming predictive analytics tasks. This way, the system supports predictive services upon node failures. To the best of our knowledge, this is a first attempt to introduce strategies for building accurate and tailored substitute models in EC environments that are used upon node failures/unavailability. Our technical contributions are:

1) We introduce a novel and systematic approach to expand the predictability capability of local models over selective training data and statistics across nodes with different strategies and appraise their performances to generalize such models upon node failures.
2) We propose strategies to guide the node invocation prediction services based on the predictability capacity of the enhanced models in the case of node failures to improve systems' resilience.
3) We provide comprehensive experimental evaluation and comparative assessment of our approach over real data against baseline solutions showcasing the resilience achieved in Edge-AI environments.

III. Rationale & Problem Definition

Consider an EC system with n distributed nodes: $\mathcal{N} = \{N_1, \ldots, N_n\}$. Node N_i has its own 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. Without loss of generality, we assume that each N_i has its local model $f_i(\mathbf{x})$ trained on local data D_i.

Our rationale is based on the idea of training enhanced substitute models on nodes by introducing our strategies used in case of failures. In each strategy $s \in \mathcal{S} = \{S_1, \ldots, S_{|\mathcal{S}|}\}$, certain training data and/or statistics on a node come from neighboring nodes; we coin these externally received training data as *unfamiliar* data (or statistics). A strategy s results in a set of *enhanced* local models $\{\tilde{f}_i^s\}$ on node N_i, which are expected to be more generalizable than the local model f_i in terms of predictability due to the fact that they attempt to capture the statistical features of unfamiliar data from neighboring nodes $N_j \in \mathcal{N}_i$. The enhanced models of N_i will be used to provide predictive services in case of failures of nodes $N_j \in \mathcal{N}_i$. Given an unavailable node N_j (having a local model f_j) with a prediction query input $\mathbf{x}$, we seek an alternative available node N_i (with enhanced models $\{\tilde{f}_i^s\}$), such that the prediction of the most appropriate model $\tilde{f}_i^{s^*}$ on node N_i for query $\mathbf{x}$ is as accurate as that of the node N_j, i.e., $\tilde{f}_j^{s^*}(\mathbf{x}) \approx f_j(\mathbf{x})$. In that case, N_i invokes its enhanced model $\tilde{f}_i^{s^*}$ for servicing prediction requests directing to node N_j for as long N_j remains unavailable.

Problem 1: *We seek for the best mixture of enhanced models $\{\tilde{f}_i^s\}_{s\in\mathcal{S}}$ across all available nodes $N_i \in \mathcal{N} \setminus \{N_j\}$ and strategies $\mathcal{S}$ to be used in order to achieve the same quality of predictions as that of the failing node N_j ensuring resilience without engaging data transfer to the Cloud. Our objective is to minimize:*

$$\mathcal{J}_j(\mathcal{S},\mathcal{N}) = \min_{(s,N_i)\in(\mathcal{S}\times\mathcal{N}), i\neq j} \mathbb{E}[(\tilde{f}_i^s(\mathbf{x}) - f_j(\mathbf{x}))^2]. \quad (1)$$

Let us focus on N_i with local model f_i and neighbors $N_j \in \mathcal{N}_i$. Given a strategy $s \in \mathcal{S}$, we get specific data and/or statistics of subsets of the datasets $\{D_j\}$, $\Gamma(\{D_j\})$, and include them to D_i, as it will be elaborated later. This results in an enhanced training dataset $\bar{D}_i = D_i \cup \Gamma(\{D_j\}), j \in \mathcal{N}_i$. Then, we use $\bar{D}_i$ to train the enhanced model $\tilde{f}_i^s$ for strategy s. Different strategies yield different enhanced models in N_i. The difference lies on how we select subsets of D_j or specific statistics from N_j to train the enhanced models in N_i. $\Gamma(\{D_j\})$ can be either real data or certain statistics derived from other nodes' datasets which are used to generate the $\bar{D}_i$ per strategy s. Once the enhanced models $\{\tilde{f}_i^s\}_{s\in\mathcal{S}}$ in N_i are built, then, a methodology for selecting the best strategy s for N_i is introduced given the unavailability of N_j. Once N_j receives a predictive service request and is unavailable, then the system advises on the most appropriate substitute N_i given the performance of its enhanced model $\tilde{f}_i^s$ per strategy s.

IV. Predictive Model Resilience Strategies

A. Global Sampling Strategy (GS)

Consider node N_i and its neighbors $N_j \in \mathcal{N}_i$. GS is based on random sampling node N_j's dataset, i.e., $\Gamma(D_j) \subset D_j$. N_i receives samples from neighbors' datasets and expands its dataset as $\bar{D}_i = D_i \cup \{\Gamma(D_j)\}$, $\forall j, j \neq i$. The size of sample $|\Gamma(D_j)|$ and sample mixing rate $\alpha = \frac{|\Gamma(D_j)|}{|D_j|} \in (0,1)$ is controlled by N_i, which affects the generalizability of enhanced model $\tilde{f}_i^S$.

B. Guided Sampling Strategies

GS gives a relatively average summary of D_j, making it ideal for datasets distributed evenly, i.e., it does equally count all the samples. However, sampled data can convey a variety of characteristics, thus, feeding the enhanced models with such data without taking into account these characteristics cannot boost the generalizability of the enhanced models. To allow control on the sampling process of $\{D_j\}$, we introduce guided sampling strategies that exploit data clustering to selectively capture data which will be included during the training of the enhanced models. We rely on vector quantization (clustering) of $D_j, \forall j$ in light of exploiting the information derived by the corresponding clusters (representatives). Such representatives, a.k.a., centroids $\mathbf{w}_{jk}$, $k = 1, \dots, K$, partition D_j into K disjoint subsets $D_j \equiv \cup_{k=1}^K \{D_{jk}\}$ with $\cap_{k=1}^K D_{jk} = \emptyset$. The way such centroids are used yields in certain variants.

1) Nearest Centroid Guided Strategy (NCG): In NCG strategy, we quantize only the input space $\mathcal{X} \subseteq \mathbb{R}^d$ of D_j. The centroids $\mathbf{w}_{jk} \in \mathcal{X}$ convey representative information to the enhanced model's input. The number of clusters K depends on the size $L_i = |D_j|$ and mixing rate α. Each of the centroids $\mathbf{w}_{jk}$ is used to select the m closest input-output pairs $(\mathbf{x}, y) \in D_{jk}$ from the k-th cluster with L_{jk} pairs. Such m pairs represent the input data subspace in each cluster. We select training examples representing the input space of D_j across all clusters obtaining the sample $\Gamma(D_j) = \{\cup_{k=1}^K \Gamma(D_{jk})\}$:

$$\Gamma(D_{jk}) \quad = \quad \{(\mathbf{x}, y)_\ell \in D_{jk} : d_{(\ell)} = \|\mathbf{x} - \mathbf{w}_{jk}\|\}, \quad (2)$$

with $d_{(\ell)}$ be the ℓ-th order statistic of the Euclidean distance between input vector $\mathbf{x}$ and centroid $\mathbf{w}_{kj}$, for $\ell = 1, \dots, m < L_{jk}$; note $d_{(1)} = \min\|\mathbf{x} - \mathbf{w}_{jk}\|$ and $d_{(L_{jk})} = \max\|\mathbf{x} - \mathbf{w}_{jk}\|$.

2) Centroid Guided Strategy (CG): In CG strategy instead of selecting the nearest pairs to centroids $\mathbf{w}_{jk}$ w.r.t. input, we select the centroids of clusters that partition the input-output space $\mathcal{X} \times \mathcal{Y}$ of D_j, i.e., centroids $\mathbf{w}_{jk} \in \mathbb{R}^{d+1}$ are samples:

$$\Gamma(D_j) = \cup_{k=1}^K \{\mathbf{w}_{jk}\}. \quad (3)$$

$\{D_j\}$ samples contain only representatives across the input-output space. CG strategy is adopted to applications with restrictions in data privacy as it avoids evidently actual data transfer among nodes.

3) Weighted Guided Strategy (WG): The unfamiliar samples $\Gamma(\{D_j\})$ for N_i's enhanced model $\tilde{f}_i$ might negatively affect its generalizability due to certain anomalies. If they exist, they cause enhanced models to fit abnormally. This problem worsens when $\Gamma(\{D_j\})$ are contaminated by a large amount of anomalies. To tackle this challenge, we introduce a weighted guided strategy to eliminate the probability of selecting anomalous samples based on the cluster density. Similar to CG, data clustering in WG quantises both input $\mathcal{X}$ and output $\mathcal{Y}$ of D_j. Smaller clusters in size are more likely to contain anomalies, thus, we assign higher probabilities of selecting samples from relatively bigger clusters than smaller ones. We define this probability p_{jk} to be proportional to the number of input-output pairs L_{jk} in cluster D_{jk}, i.e., $p_{jk} = \frac{L_{jk}}{\sum_{\kappa=1}^K L_{j\kappa}}$. Hence, given a rate α of the data size $|D_j|$, we randomly select $\alpha \cdot p_{jk}$ samples from cluster D_{jk} along with centroid $\mathbf{w}_{jk}$, i.e.,

$$\Gamma(D_j) = \cup_{k=1}^K \{\mathbf{w}_{jk} \cup \{(\mathbf{x}, y) \in D_{jk} : |D_{jk}| = \alpha \cdot p_{jk}\}\}. \quad (4)$$

V. Performance Evaluation

Experimental Setup: We test our strategies in a realistic EC environment using the real dataset [14] collected during the experiments of our project GNFUV[1]. The dataset contains readings of temperature and humidity from sensors mounted on four Unmanned Surface Vehicles (USVs), i.e., four edge nodes, monitoring the sea surface in a coastal area in Athens, Greece. The local data $D_i, i = 1, \dots, n = 4$, recorded by the USVs exhibit different distribution while bearing some spatiotemporal correlation. We used 'temperature' as the input variable $x \in \mathbb{R}$ to predict the output variable 'humidity' $y \in \mathbb{R}$. We adopted the Support Vector Regression (SVR) regression model in our experiments. Note: other ML models could be also adopted, which does not spoil the evaluation methodology.

A. Model Performance Assessment

Upon nodes (USVs) failure, our method seeks the most appropriate substitute node N_i and strategy $s \in \mathcal{S} = \{\text{GS, NCG, CG, WG}\}$ given a mixing rate α to train the enhanced models on the substitute node. We devised a systematic approach to access the performance of the enhanced models trained with different parameters for each node adopting grid search for tuning. For each node N_i and strategy $s \in S$, we set mixing rates $\alpha \in \{0.02, \dots, 0.2\}$. The corresponding datasets $\bar{D}_{i,s}$ are built based on D_i and the $n-1$ $\Gamma(\{D_j\})$ derived from $\{D_j\}$ for each strategy s. Then, we trained the enhanced SVR models $\tilde{f}_i^s$ for each $\bar{D}_{i,s}$ and evaluated the models' performance in terms of the Root Mean Square Error (RMSE) between actual output y and model prediction $\hat{y}$.

For each node N_i, the data we evaluated the models on include the enhanced data $\bar{D}_i$ and the raw data D_j from the neighboring nodes. We obtain an insight into the influences brought by applying our approach to the node's capabilities to handle its own data and data from others. The evaluation was conducted using 3-fold cross-validation and the data included in $\{D_j\}$ are excluded from the models' evaluation on D_j. Moreover, to compare and contrast the influence brought by our approach, we evaluated the performance of (i) the (Global) Cloud model, i.e., the model that was trained on all the nodes' data (D_G) transferred from USVs to Cloud (denoted by f_G), which serves as the baseline model, and (ii) the local models, i.e., models trained only on local data D_i (denoted by f_i) on the same kind of data they were trained with. We obtained $f_G(D_G)$ (the baseline) and four local models $f_i(D_i), i = 1, \dots, 4$. We expect prediction error for some $\tilde{f}_i^s(D_j)$ to fall above $f_G(D_G)$ and for some to fall below $f_G(D_G)$. Prediction error below or close to that of $f_G(D_G)$ is desired as it indicates that the parameters corresponding to these models provide more accurate predictions than the Cloud. Furthermore, the accuracy of $\tilde{f}_i^s(D_j)$ is expected to be less than $f_i(D_i)$ as local models have ideal performance on local data. Due to space limitations, we only provide in Figure 1 the results of the enhanced model $\tilde{f}_3^s$ of node N_3; similar results are obtained for the rest nodes. Figure 1 shows the performance (RMSE) of local, Cloud and enhanced models against different mix-rate values across all strategies. One could observe that N_3 is not a good candidate as a substitute node for N_1 and N_4 because the errors of $\tilde{f}_3^s(D_1)$ and $\tilde{f}_3^s(D_4)$ are above the baseline $f_G(D_G)$. However, as evidenced, N_3 is an appropriate substitute for node N_2, if N_2 fails. As with NCG and CG, for certain α values, the corresponding enhanced model $\tilde{f}_3$ of N_3 outperformed the baseline on D_2 (accuracy is higher than that of $f_G(D_G)$). This indicates that given our strategies, when N_2 fails, N_3 can take over N_2's predictive tasks without needing to transfer to N_2's data to Cloud (for building a new model therein). More importantly, substitute N_3 gives better predictions than the Cloud. This denotes the resilience capacity of the system adopting our strategies.

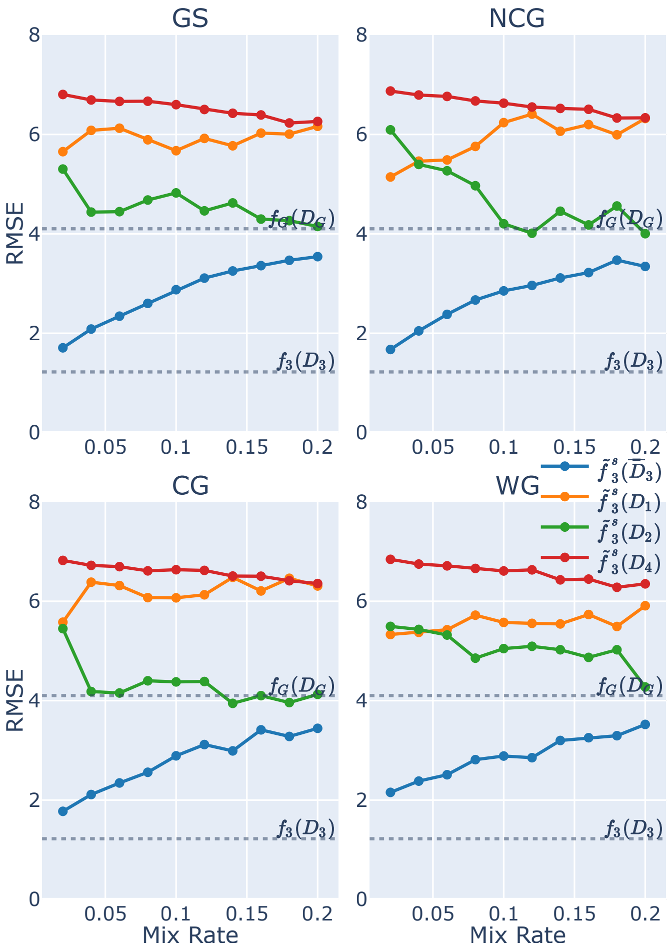

Fig. 1. Sensitivity analysis of the α (mix rate) on the performance evaluation of the local model $f_3(D_3)$, Cloud model $f_G(D_G)$ and the enhanced models $\tilde{f}_3^s(D_i)$ in N_3 under the four strategies (GS,NCG,CG,WG).

B. Deployment to Intelligent Edge Computing

We have first identified the best strategy s and mix-rate α for every pair of potentially failing node N_i and potentially substitute node N_j. Then, in an EC system with n nodes ($n = 4$ USVs in our scenario), we obtain adequate information

[1] http://www.dcs.gla.ac.uk/essence/funding.html#GNFUV

to guide the invocation of substitute nodes in the case of node failures. We introduce a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ to guide the node invocation and visualize the guidance as illustrated in Figure 2. The vertices $\mathcal{V}$ represent edge nodes; we introduce one extra vertex referring to Cloud (G). A directed edge $e_{ij}^{\epsilon,s} \in \mathcal{E}$ starts from node N_i and ends with N_j attached with an RMSE value ϵ and strategy $s \in \mathcal{S}$ (the mixing rate α was omitted for clearance). The semantics of $e_{ij}^{\epsilon,s}$ is that: if a predictive task request is received to failing node N_j, then a potential substitute node N_i could, at its best, provide an RMSE ϵ from its enhanced model $\tilde{f}_i^s$ given the best selected strategy s. For instance, the edge e_{42} in Figure 2 indicates that upon N_2's failure, the best substitute node N_4 can invoke its enhanced model $\tilde{f}_4^{GS}$ given the best strategy GS obtaining RMSE 2.13. The edges e_{12} and e_{32} indicate N_1 and N_3 can serve N_2's requests when it fails both with NCG as the best strategy obtaining RMSE 7.98 and 3.95, respectively. Thus, the second best substitute for N_2 is N_3 offering the second lowest RMSE. If the best substitute N_4 is unreachable (or e.g., overloaded), then N_3 can be reached next and so on. If none of the candidate substitutes are reachable, then the request goes to Cloud G as a last resort. A *recursive* edge $e_{ii}^{\epsilon,s}$ indicates the RMSE achieved by N_i's enhanced model over its best strategy s. The graph is disseminated to all nodes for localized decision-making. In our experiments, we investigate the system's performance when the best substitute is available to serve the directed requests from the failing nodes.

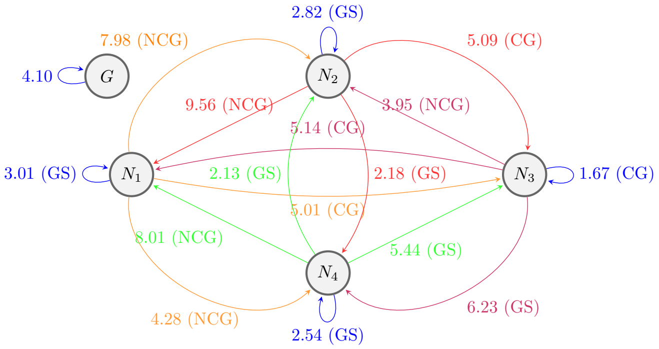

Fig. 2. Directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ guiding the decision making for the most appropriate substitute node and strategy upon node failures.

With the directed graph, the system is fully operable in EC environments where node failures happen. To further better understand the benefits brought by our approach, we evaluated the system's performance with full, half and no guidance at all with node failing probability p from 0% to 100%. To avoid the "chain reaction" of the substitute nodes failing one after another and allowing clearer insight into the system's behavior, we assume that when a node fails, its substitute node(s) will work. Specifically, at every predictive task request to a node N_i, we draw with probability p its status. If N_i is not failed, then the node processes the request locally using its local model. Otherwise, i.e., N_i fails/unavailable, then, we consider the following node assignment resilience policies:

1) **Random Substitute Assignment:** The request is assigned to a randomly chosen non-failing node, which locally processes the task using its local model. This is a baseline policy to investigate what happens without the graph guidance of our approach and without invoking the enhanced models on substitute nodes (*zero guidance*).
2) **Random Substitute Assignment with best Enhanced Model:** The request is assigned to a randomly chosen non-failing node, which locally processes the task using its best enhanced model given the provided graph (*half guidance*).
3) **Guided Substitute Assignment:** The request is assigned to the most appropriate substitute node as per graph, which locally processes the task using its best enhanced model as per graph (*full guidance*).

Remark: in Figure 2, none of the edges are associated with WG. Compared to other strategies, WG did not achieve a single best on arbitrary pairs of failing N_i and substitute N_j. WG is designed for special anomalous datasets yielding suboptimal performance. Our test results using the Local Outlier Factor (LOF) algorithm do conform with that, as the resulting anomaly rate is 7.6%, which is normal for LOF.

The prediction accuracy results of the above-mentioned resilience policies are shown in Figure 3 against node failure probability p. We compare the results obtained from these assignment policies including: (i) the Cloud-based assignment policy, i.e., sending the predictive tasks to the Cloud (which maintains a global model trained over all nodes' data) and (ii) the average non-failing nodes assignment policy, where we obtain the average prediction by invoking the best enhanced models (with the best selected strategy) over the expanded datasets of *all* non-failing nodes, i.e., $\tilde{f}_0 = \frac{1}{n-1}\sum_{i=1}^{n-1} \tilde{f}_i^s(\bar{D}_i)$. One could observe that the system's predictability with the guided substitute assignment always outer performs the Cloud-based policy *even* when p reaches 100%. This indicates that our approach helps the system to maintain better performance than the Cloud even when a node is unavailable all the time. Moreover, when $p < 20\%$, the system performance is quite close to the best local enhanced models. This denotes that our resilience method supports the system to maintain performance equivalent to the cases of no failures at all, especially when p is at a relatively low level. This further proves its potency in improving the system's resilience in EC environments. Furthermore, the performance of the random substitute assignment with the best enhanced model (half guidance) helps to keep the system predictability capacity at relatively higher levels than that of the baseline until p reached around 50%. That is, even if nodes fail half of the time, exploiting the enhanced models of the non-failing nodes provides better predictability than directing the requests to the Cloud. In addition, by comparing the half guidance with the random substitute assignment (zero guidance), the former was able to reduce the RMSE by half. This evidences that our resilience approach endows the system with a fair amount of flexibility: even if we do not direct the predictive task requests to the best substitute node due to reasons like load balance all the time, it still can contribute to boosting the system's predictability performance via using the enhanced models from randomly chosen substitute nodes. Evidently, the full guidance policy exploits the full information in graph $\mathcal{G}$, thus, resilience is achieved without needing requests to be directed to Cloud even if with high node failure probabilities.

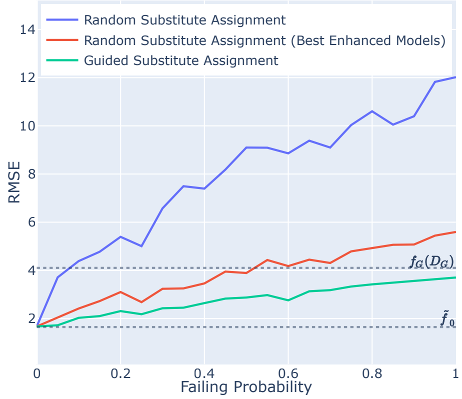

Fig. 3. System performance against node failure probability p across different node assignment resilience policies.

We investigated the impact of the substitute assignment policies on the number of invocations on each node, i.e., extra load. Figure 4 shows that with the full guidance policy, node extra loads are relatively balanced. Although we expect node imbalance given tasks workloads; requests to failing node N_i are directed to the same best substitute N_j. If multiple nodes have the same best substitute node N_j, upon their failures, all the requests will be directed to N_j, thus, causing imbalanced loads. In systems that are susceptible to node imbalance, this could be alleviated e.g., by directing a percentage $\beta\%$ of requests to the best substitute node and directing the rest $(1-\beta)\%$ to the second best substitute node. Evidently, the challenge to find the optimal β to achieve load balance and system performance is on our future research agenda.

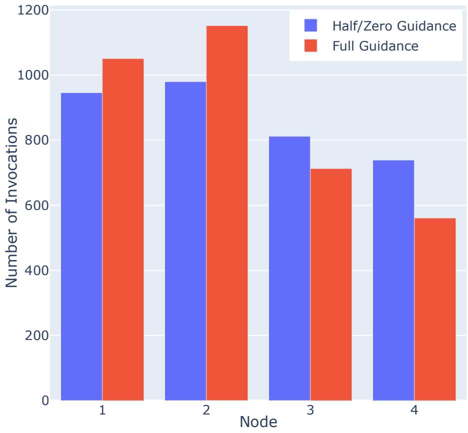

Fig. 4. Extra load per node in the system with full guidance assignment policy and half/zero guidance policy.

VI. Conclusions

We propose a predictive model resilience framework relying on strategies to build enhanced models handling requests on behalf of failing nodes. Our framework seeks the best strategy for pairs of failing and substitute nodes to guide invocations upon failures. The best strategies are represented in a directed graph. We assess the system performance over certain node assignment guidance policies and compared it with baseline approaches over real data in a realistic EC environment. Our framework maintains the system's predictability performance higher than the baselines even with high failure probability and offers flexibility in load balancing problems.

Acknowledgment

Jordi Mateo's work was supported by EU grant Ajuts de requalificació per a professorat universitari promoted by the Ministeri d'Universitats (NextGenerationEU), Intelligent Energy Europe (IEE) program, and Ministerio de Economía y Competitividad (PID2020-113614RB-C22) funded by MCIN/AEI/10.13039/501100011033-Ministerio de Ciencia e Innovación nad Agencia Estatal de Investigación. He thanks Glasgow University's fellowship in the Essence: Data Science & Distributed Computing[2] group.

References

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, 2016.

[2] J. Ren, Y. Pan, A. Goscinski, and R. A. Beyah, "Edge computing for the internet of things," *IEEE Network*, vol. 32, no. 1, pp. 6–7, 2018.

[3] M. A. J. et al., "An ai-enabled lightweight data fusion and load optimization approach for internet of things," *Future Generation Computer Systems*, vol. 122, pp. 40–51, 2021.

[4] J. Wang, S. Pambudi, W. Wang, and M. Song, "Resilience of iot systems against edge-induced cascade-of-failures: A networking perspective," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6952–6963, 2019.

[5] K. A. Delic, "On resilience of iot systems: The internet of things (ubiquity symposium)," *Ubiquity*, vol. 2016, no. February, pp. 1–7, 2016.

[6] J. Beutel, K. Römer, M. Ringwald, and M. Woehrle, *Deployment Techniques for Sensor Networks*, 10 2009, pp. 219–248.

[7] K. et al., "Diagnostic powertracing for sensor node failure analysis," in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 117–128. [Online]. Available: https://doi.org/10.1145/1791212.1791227

[8] S. Shao, X. Huang, H. E. Stanley, and S. Havlin, "Percolation of localized attack on complex networks," *New Journal of Physics*, vol. 17, no. 2, p. 023049, feb 2015. [Online]. Available: https://doi.org/10.1088/1367-2630/17/2/023049

[9] A. Shafahi, M. Najibi, Z. Xu, J. Dickerson, L. S. Davis, and T. Goldstein, "Universal adversarial training," *AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, pp. 5636–5643, Apr. 2020.

[10] H. Daumé III, "Frustratingly easy domain adaptation," *arXiv preprint arXiv:0907.1815*, 2009.

[11] X. Ge, F. Chen, C. Shen, and R. Ji, "Colloquial image captioning," in *IEEE International Conference on Multimedia and Expo*. IEEE, 2019, pp. 356–361.

[12] A. Farahani, S. Voghoei, K. Rasheed, and H. R. Arabnia, "A brief review of domain adaptation," *Advances in Data Science and Information Engineering*, pp. 877–894, 2021.

[13] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process Mag*, vol. 37, no. 3, pp. 50–60, 2020.

[14] N. Harth and C. Anagnostopoulos, "Edge-centric efficient regression analytics," in *2018 IEEE EDGE*. IEEE, 2018, pp. 93–100.

[2] http://www.dcs.gla.ac.uk/essence/