Predictive Model Resilience in Edge Computing

Mr Qiyuan Wang

Supervisor: Dr Christos Anagnostopoulos

School *of* Computing Science
Essence: Data Science & Distributed Computing Lab

Introduction

For **Distributed Machine Learning (ML)** systems operating in **Edge Computing (EC)**, data are collected and processed at the edge nodes to address the limitations of the Cloud in supporting delay-sensitive, real-time decision-making, and context-aware services [1]. However, nodes can go offline due to various reasons. We would not want the system to cease its service or send the failing nodes' data to the Cloud. It would be ideal to let the neighboring nodes process the data. Due to spatial and temporal differences that edge nodes naturally have, statistical characteristics of their data significantly vary [2] and one node trained on its local data usually cannot process other nodes' data very well. **We propose a universal framework that investigates data and statistics mixing strategies for nodes to build robust models supporting resilience in predictive services in case of node failures**.

Rationale

Our rationale is based on the idea of **training enhanced substitute/surrogate models on nodes** by introducing our strategies used in case of failures. Consider an EC system with n distributed nodes: $\mathrm{N}=\{N_1, \dots, N_n\}$, node N_i has its own local data D_i. In each strategy $s \in S = \{s_1, \dots, s_{|S|}\}$, certain training data (***unfamiliar* data**) on a node come from neighboring nodes $\{N_j\}$. A strategy s results in a set of enhanced local models $\{\widetilde{f_i^s}\}$ 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 $\{N_j\}$. The enhanced models of N_i will be used to provide predictive services in case of failures of nodes $N_j \in \{N_j\}$.

Methods & Results

Strategies S

Global Sampling Strategy (GS):
GS is based on random sampling on node N_j's data, i.e.,$\Gamma(D_j) \subset D_j$. N_i receives samples from neighbors' data and expands its data as $\overline{D_i} = D_i \cup \{\Gamma(D_j)\}$. The size of sample $|\Gamma(D_j)|$ is controlled by mixing rate α.

Nearest Centroid Guided Strategy (NCG):
We quantize (cluster) only the input space of D_j into K clusters $\{D_{jk}\}$ and get the centroids w_{jk}. The numbers of cluster K depends on $|D_i|$ and α. Each w_{jk} is used to select the m closest input-output pairs $(x, y) \in D_{jk}$ to get $\Gamma(D_{jk})$:

$$\Gamma(D_j) = \{\cup_{k=1}^{K} \Gamma(D_{jk})\}$$

Centroid Guided Strategy (CG):
The clustering process of CG is quite similar to NCG, the difference is that CG quantize input-output space, and w_{jk} are used directly:

$$\Gamma(D_j) = \{\cup_{k=1}^{K} w_{jk}\}$$

CG strategy is suitable for privacy-sensitive scenarios as it avoids evidently actual data transfer among nodes.

Weighted Guided Strategy (WG):
In the clustering process, smaller clusters are more likely to contain anomalies, thus, we assign a higher probability of selecting samples from relatively bigger clusters than smaller ones. We define this probability p_{jk} to be proportional to $|D_{jk}|$. We randomly select $\alpha \cdot p_{jk}$ samples from cluster D_{jk} along with centroid w_{jk}:

$$\Gamma(D_j) = \cup_{k=1}^{K} \{w_{jk} \cup \{(x, y) \in D_{jk} : |D_{jk}| = \alpha \cdot p_{jk}\}\}$$

Evaluation & Results

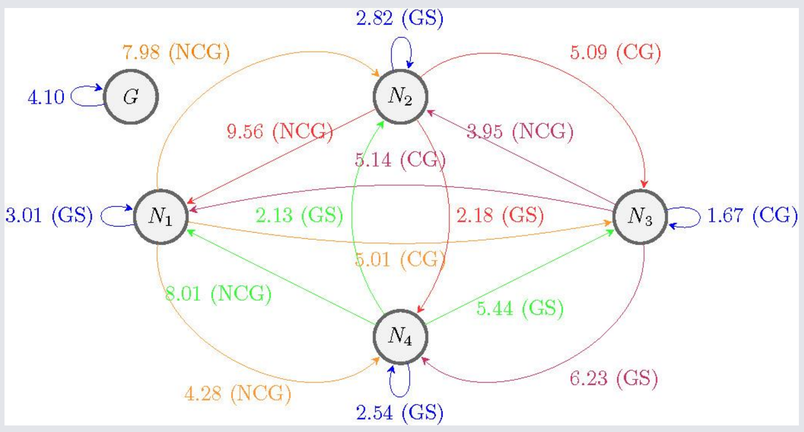

Figure 1.
Directed graph $G(V, E)$ guiding the decision making for the most appropriate substitute node and strategy upon node failures.

We experimented in a realistic EC environment with the dataset of our project GNFUV*. With it, we simulated 4 nodes and run evaluations on arbitrary pairs of nodes and with all 4 strategies. We gathered the results and produced a directed graph (shown in Figure 1). The semantics of edge $e_{ij}^{\epsilon,s}$ is that: if a predictive task request is received at failing node N_j , then a potential substitute node N_i could, at its best, provide an RMSE ϵ from its enhanced model $\widetilde{f_i^s}$ given the best-selected strategy s.

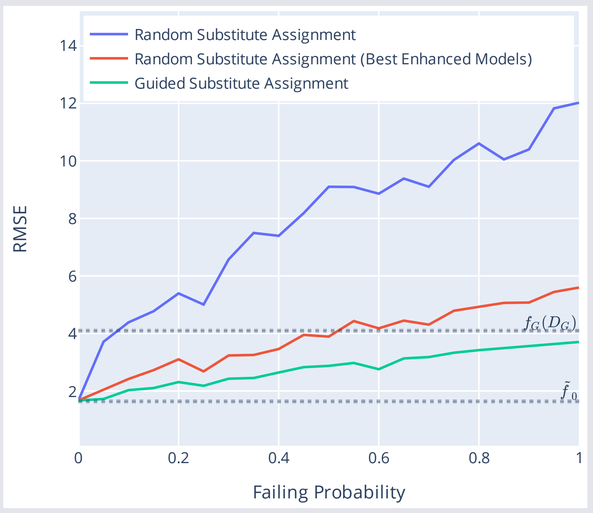

Figure 2.
System performance against node failure probability p across different node assignment resilience policies.

*http://www.dcs.gla.ac.uk/essence/funding.html#GNFUV

Discussion

Form the directed graph, one could see many of the results are below the baseline result got with the Cloud (global model trained with all the data, represented by G in Figure 1), which is very promising. To compare how the ML system would operate with/without our framework, we also simulated multiple node failure probabilities p and tested the system's performance with full/half/zero guidance of our framework (shown in Figure 2). It is evident that compared to zero guidance (blue line), full guidance (green line) helped to cut down the RMSE significantly. Moreover, with full guidance of our framework, the system could always attain results better than the baseline even when p is close or equals 1 (one node is always failing).

Conclusions

Our framework seeks the **best strategy for pairs of failing and substitute nodes to guide service invocations upon failures**. The best strategies are represented in a directed graph. It maintains system's predictability performance higher than the baselines even with high failure probability.

References

1. 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.
2. Mian Ahmad Jan et al. 2021. An AI-enabled lightweight data fusion and load optimization approach for Internet of Things. Future Generation Computer Systems 122 (2021), 40–51.