CIPAT: Latent-resilient Toolkit for Predicting the Performance Impact due to Configuration Tuning

Kartik Patel[†], Changhan Ge[†], Ajay Mahimkar[§], Sanjay Shakkottai[†], Yusef Shaqalle[§]
[†] The University of Texas at Austin, Austin, TX, USA
{kartikpatel, chge, sanjay.shakkottai}@utexas.edu
[§]AT&T Labs, Bedminster, NJ, USA
{mahimkar, yusef.shaqalle}@att.com

Abstract

Cellular service providers (CSPs) aim to optimize network performance and enhance user experience by tuning network configurations. However, this process often requires continuous live network testing, which incurs significant operational costs. In this paper, we focus on predicting the impact of configuration changes using historical data, thereby reducing the need for live network tests. A key challenge in developing such a model is accounting for unobserved external factors (e.g., weather, referred to as *latents*) that can introduce confounding effects between configurations and performance metrics. To address this, we employ intermediate network metrics, called *Mobility, Access, and Traffic (MAT) metrics*, which are influenced by both configurations and latents, and in turn, affect performance metrics. We introduce **C**onfiguration **I**mpact **P**rediction **A**nalysis **T**oolkit (CIPAT), a novel two-stage toolkit developed using a comprehensive real-world dataset from live LTE networks. Our evaluation demonstrates that CIPAT enables CSPs to predict the performance impact of proposed configuration changes with up to 86% accuracy and 85% efficacy, thereby reducing the operational costs associated with configuration tuning.

CCS Concepts

• **Networks** → **Network performance modeling**; **Network management**; **Wireless access points, base stations and infrastructure**; • **Computing methodologies** → **Neural networks**.

Keywords

Network performance modeling, Network management, Machine learning applications

ACM MLNextG '24, Nov. 18 2024, Washington DC, USA

ACM ISBN 979-8-4007-0489-5/24/11.
https://doi.org/10.1145/3636534.3698246

ACM Reference Format:
Kartik Patel[†], Changhan Ge[†], Ajay Mahimkar[§], Sanjay Shakkottai[†], Yusef Shaqalle[§]. 2024. CIPAT: Latent-resilient Toolkit for Predicting the Performance Impact due to Configuration Tuning. In *1st ACM Workshop on Machine Learning for NextG Networks (MLNextG '24), November 18 2024, Washington DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3636534.3698246

1 Introduction

In cellular network management, optimizing configuration parameters is a critical task for Cellular service providers (CSPs) to enhance service quality. These configuration changes are frequent, with thousands of attempts occurring weekly in mid-sized North American cities, as illustrated in Fig. 1.

Traditionally, CSPs rely on a combination of domain knowledge, experience, and vendor input to identify potential configuration changes for live network testing. Once a configuration is selected, it is implemented on a subset of the network, followed by a 5-7 day assessment period to evaluate its impact on Key Performance Indicators (KPIs). Based on the results, the configuration may either be rolled out network-wide or reverted. This process is time-consuming and costly, prompting CSPs to minimize experiments that lead to rollbacks [4]. Although prior works have developed data-driven recommendation engines to identify effective configurations [2–4], they still require extensive testing on the live network.

In this work, we introduce a data-driven toolkit that predicts the impact of configuration changes on KPIs. By leveraging the historical data on configuration changes, our toolkit forecasts the effects of new configurations without deploying them on live network. Therefore, this toolkit enables CSPs to filter out the configuration changes likely to degrade performance, thereby reducing the risk of costly rollbacks.

In principle, predicting the performance impact of configuration changes in complex cellular networks requires sophisticated regression models trained on extensive datasets. Prior research [3] highlights that KPIs are influenced not only by configuration parameters but also by relatively static intrinsic attributes, such as network morphology and hardware (see Table 1). Therefore, a naive approach would involve training a model to directly map configuration parameters and attributes to KPIs.

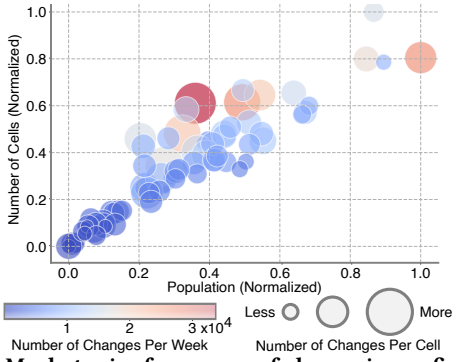

Figure 1: Market-wise frequency of change in configurations

Type	Examples	Type	Examples
Configuration	Transmission power, Antenna tilt angle, Handover threshold	Attribute	Morphology (urban/rural), Hardware/software version, BS type (micro/macro)
MAT Parameter	Reference signal received power/quality (RSRP/Q) Control channel utilization, Traffic volume	KPI	Throughput, Accessibility, Retainability
		Latent Variable	Winter storm, Sports event Lockdown, Holiday, Foliage

Table 1: Examples of the various types of variables

In practice, however, training such models is challenging due to the impact of external factors – such as weather conditions or large-scale events – that can significantly affect KPIs independently of configuration parameters [5]. Hence, these unobserved external factors, or *latent variables*, pose challenges in accurately capturing the dependencies between configurations and KPIs. To mitigate the effect of latents, we use *observable* intermediate network metrics, called *Mobility, Access, and Traffic (MAT) metrics*, which are influenced by both configurations and latents and, in turn, impact KPIs. Consequently, MAT metrics isolate the effect of latents from KPIs (see Fig. 2). *If* all such MAT metrics were known, then the KPIs can be predicted irrespective of the external conditions (snow, holiday). Our prior work [5] demonstrates that MAT metrics are more effective predictors of KPIs than the configuration parameters in the presence of latents.

In this work, we propose **C**onfiguration **I**mpact **P**rediction **A**nalysis **T**oolkit (CIPAT), a novel latent-resilient toolkit designed to predict the performance impact of configuration changes. CIPAT employs a two-stage approach: the first stage models the relationship between configuration parameters and MAT metrics, while the second stage focuses on the relationship between the MAT metrics and KPIs.

Since the MAT metrics are directly impacted by the latent variables, predicting the magnitude of MAT metrics based on the changes in the configuration parameters is challenging. Instead, we predict the direction of change in MAT metrics in the first stage, an approach which can be shown to be robust against latents. Subsequently, we predict the direction of change in KPIs from the direction of change in MAT metrics.

The remainder of this paper is organized as follows: In Section 2, we detail the dataset used in this paper. We then introduce CIPAT in Section 3, followed by the construction of the two-stages of CIPAT in Sections 4 and 5. Finally, the evaluation of CIPAT is presented in Section 6.

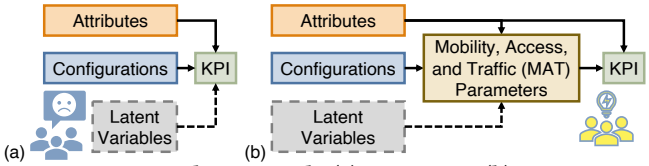

Figure 2: Dependency graph. (a) One-stage. (b) Two-stage.

2 Dataset

A national cellular network is divided into multiple markets. Each market contains multiple base stations (BSs) with each BS operating multiple cells. We collected daily snapshots of cell-level data from a tier-1 CSP for four markets over a span of 20 months. The dataset includes five categories:

(1) *Cell attributes*: These are relatively static parameters defining a cell's operating characteristics. The attributes include the market, morphology (rural/suburban/urban), frequency bands, bandwidth, BS type (macro/micro), hardware and software versions, tower height and the DL MIMO mode (closed loop MIMO/single stream MIMO). We denote the set of attributes by $\mathcal{A}$, and their values by a matrix $\mathbf{A}$.

(2) *Cell Configurations*: These parameters, such as transmitted power, antenna tilt angle and handover thresholds, are tuned to optimize network based on deployment scenarios and requirements. We denote the set of 318 configurations by C and their cell-date-wise values by a matrix $\mathbf{C}$.

(3) *MAT metrics*: These are network metrics monitored by CSPs for each cell, assessing (i) User Mobility-related metrics like distance and speed distributions, as well as attempted and successful handovers; (ii) Cell Access-related metrics like average signal strength; and (iii) Traffic metrics including cell utilization and data volume. We use 56 MAT metrics, denoted by a set $\mathcal{M}$, with the values, denoted by a matrix $\mathbf{M}$.

(4) *Key Performance Indicators (KPIs)*: These are target metrics such as average per-user throughput, call retainability, and call accessibility to measure the quality of service experienced by users.

(5) *Derived variables*: These include periodic variables like the day of the week, which account for external periodic influences on the network. These variables are not captured by the CSP but are derived from the dates and attributes.

Notably, the dataset does not include latent variables, such as historical weather data and large-scale events (*e.g.*, sports), which can have a substantial impact on KPIs. Exhaustively listing all latents and measuring their impact is infeasible.

3 CIPAT: Introduction

We introduce CIPAT, designed to predict the performance impact of configuration changes on KPIs purely from the historical data. When a network operator seeks to predict

the impact of changing a specific configuration parameter (e.g., increasing the transmission power `pMax`) on a cell with attribute values **a**, CIPAT facilitates this process through the following stages:

Stage 1: Configuration to MAT metrics. (1) First, CIPAT identifies the MAT metrics that are affected by the chosen configuration parameter in cells with the attribute **a**. (2) Next, CIPAT determines the direction of change in each MAT metric resulting from the configuration change. These directions represent a 56-dimensional orthant with the origin at current values of MAT metrics. Each point within this orthant signifies a possible value of the MAT metrics post-configuration change.

Crucially, the key configuration parameters impacting the MAT metrics and the associated direction of change can be pre-determined from the dataset without requiring the knowledge of latents (see Section 4).

Stage 2: MAT metrics to KPI. CIPAT uniformly samples a large number of points within the orthant. CIPAT then estimates the potential values of KPI for each point in the orthant using a pre-trained model that maps (MAT, attributes) $\rightarrow$ (KPI). Since the relationship between MAT metrics and KPI is not influenced by latents, such a model can be trained from the dataset (see Section 5). By analyzing the fraction of points that predicts an increment/decrement in the KPI, the operator can make a decision on whether to implement the proposed configuration change on the network. If a large number of the sampled points from the orthant results in a KPI decrement, it would imply that a configuration change will change the MAT metrics such that the KPI with the MAT metrics will be lower than the current KPI.

The two-stage process of CIPAT offers several additional advantages. By incorporating domain-specific insights from CSPs, Stage 1 can go beyond a purely data-driven approach, allowing exploration of under-tested configurations. Additionally, the process also enhances prediction explainability. Instead of a direct mapping of configurations to KPIs – which may obscure critical insights if feasible – our approach provides a transparent mechanism through MAT metrics for predicting the effects of configurations changes.

Note that CIPAT can be used to predict the impact of configurations on various cell-level KPIs, including but not limited to throughput, accessibility, and retainability. For the sake of brevity and focused analysis, our subsequent discussion predominantly centers on throughput as the selected KPI. In the subsequent Sections 4 and 5, we describe the construction of each stage of CIPAT from the dataset.

4 Construction of Stage I

In this section, we present the construction of the first stage of CIPAT: determining the relationship between configurations and MAT metrics from the dataset. In the following, we first outline the process of determining key configuration parameters for each MAT metric and then describe the procedure for identifying the direction of change of MATs *w.r.t.* changes in configurations.

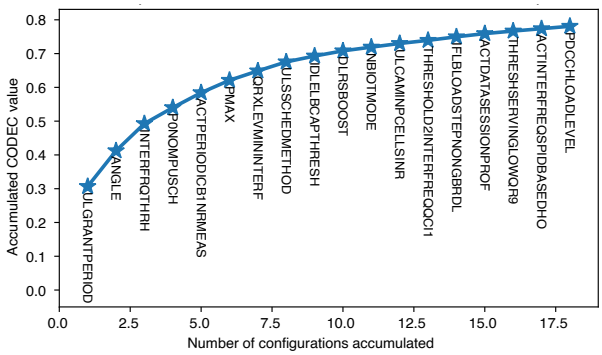

Figure 3: Configurations identified by FOCI for # RRC connection in an LTE network

Step 1: Determining influential configurations: We use the Conditional Dependence Coefficient (CODEC) score and the Feature Ordering by Conditional Independence (FOCI) algorithm, as introduced in [1], to identify the key configuration parameters for each MAT metric. CODEC score measures the proportion of variation in a MAT metric that is explained by both – a subset of configuration parameters and the attributes – beyond what is explained by the attributes alone [1]. The FOCI algorithm is a feature selection method that iteratively selects the configuration parameters to maximize the CODEC score between the chosen configurations and the MAT metric, while conditioned on attributes.

To find the most influential configuration parameters for a MAT metric on the cells with attribute value **a**, we first create a conditional dataset containing cells with these attributes. We then apply FOCI algorithm [1], using the MAT metric as the target and the configuration set C as the feature set. The FOCI algorithm identifies the configurations that maximize the CODEC score, thus, identifying the configurations that have maximum influence on the MAT metric.

For example, Fig. 3 illustrates the important configurations identified by FOCI and their associated cumulative CODEC score for the number of Radio Resource Control (RRC) connections (a MAT metric) of cells in the LTE network. The algorithm highlights several key configuration parameters, such as transmitted power (`pMax`) and antenna tilt (`Angle`), which are crucial for cell coverage. Additionally, handovers related parameters (`InterFreqThr`, `qRxLevMin-InterFreq`), the physical layer parameters such as reference signal boost (`DlRsBoost`), and load balancing-related parameters (`IdleLB-CapThresh`, `PDCCHLoadLevel`) also significantly affect the number of RRC connections of a cell. Furthermore, derived periodic features such as the day of the week also play significant roles. These findings align with the expectations and practices of CSPs.

We note that the CODEC score achieved in Fig. 3 is 0.75, which is below the maximum possible value of 1. This suggests that some variability in the MAT metric remains unexplained by the configurations, indicating that the latent variables also significantly impact the MAT metrics. Therefore, predicting the *magnitude* of the MAT metrics purely based on configuration parameters is challenging. Instead, we focus on determining the *direction of change* in the MAT metrics due to changes in the configuration parameter.

Step 2: Identifying the direction of change: We aim to determine whether changes in the configurations and MAT metrics move in the *same* direction (both increases or decreases together), or in the *opposite* directions (an increase in configuration value decreases the MAT metric, or vice versa). We begin by outlining the dataset processing steps to identify the changes in configurations and their associated changes in MAT metrics. Next, we describe the method for determining the direction of change for each configuration-MAT metric pair with a fixed attribute value.

(a) Data processing. For this analysis, we use daily snapshots of configurations, and compute differences between consecutive dates. We define a configuration change matrix, $\Delta\mathbf{C}$, where $\Delta\mathbf{C}(i, c)$ represents the amount of change in the configuration c in the i-th configuration change event (indexed by date-cell pair of the event).

With the configuration change matrix established, the next step is to asses the change in MAT metrics due to each configuration change. To do this, we compare the median MAT metrics (taken over a 7-day interval) before and after each configuration change event and determine the direction of change (positive or negative) in the MAT metrics. If the relative change is within a 3% margin, we consider the MAT metric unaffected to avoid considering the normal fluctuations as significant changes. We denote the MAT change matrix by $\Delta\mathbf{M}$, where $\Delta\mathbf{M}(i, m) \in \{+1, -1, 0\}$ denotes the direction of the change in the MAT metric m for i-th event.

(b) Attribute-wise change direction identification. Finally, to estimate the direction of change in MAT metrics due to configuration changes, we focus on the sign of the weights in a linear model mapping $\Delta\mathbf{C} \rightarrow \Delta\mathbf{M}$, which helps us determine the direction of change instead of the magnitude.

In practice, we observe that the direction of change of a MAT metric *w.r.t.* a configuration change may vary depending on the cell attributes. For example, the number of RRC connections in an LTE network is impacted by the tilt angle (as shown in Fig. 3). However, its impact on the RRC connection can differ based on the morphology; rural sites might see increased connections with increasing tilt angle due to better coverage, while urban sites might not be affected[2]. Thus, identifying the direction of change as a function of attributes is crucial.

To measure the direction of change in MAT metrics, we first partition the dataset by the attribute values associated with the cell of each configuration change event. Let $\mathbf{a} \in \mathcal{U}(\mathbf{A})$ denote an attribute value. Then for each partition, the subsets of $\Delta\mathbf{C}$ and $\Delta\mathbf{M}$ are denoted by $\Delta\mathbf{C}_\mathbf{a}$ and $\Delta\mathbf{M}_\mathbf{a}$.

Since multiple configurations may be adjusted simultaneously [2], we use a linear model to isolate the effects of each configuration on MAT metrics. Although the relationship may not be strictly linear, the sign of the model's weights are sufficient to determine the direction of the relationship (*same* or *opposite*). Formally, let $\mathbf{W}_\mathbf{a}$ be the weight matrix where $\mathbf{W}_\mathbf{a}(c, m)$ denotes the impact of a configuration c on the MAT metric m for a given attribute value $\mathbf{a}$. Then,

$$\mathbf{W}^*_\mathbf{a} = sgn\left(\underset{W \in \mathbb{R}^{|\mathcal{C}|\times|\mathcal{M}|}}{\arg\min} \|\Delta\mathbf{C}_\mathbf{a} W - \Delta\mathbf{M}_\mathbf{a}\|_2\right). \quad (1)$$

where $\mathbf{W}^*_\mathbf{a}(c, m)$ provides the relative direction of the change for each configuration-MAT metric pair given the attribute value $\mathbf{a}$.

Note that, determining the direction of change in MAT metrics due to configuration changes without the knowledge of latents requires an underlying assumption that the direction of change remain consistent regardless of underlying latents. This assumption can be validated by considering known changes in the latent variables.

5 Construction of Stage II

In this section, we discuss the design of the Deep Neural Network (DNN)-based regression model and training methodology used to construct the second stage of CIPAT.

DNN architecture: A fundamental aspect of our DNN design is that the relationship between KPIs and MAT metrics can vary based on the attributes of the cell. To address these variations, we construct a DNN model with a shared representation followed by tunable heads tailored to specific attribute values.

Our model takes MAT metrics and attributes as inputs and categorizes the attributes into two groups: (1) Mask attributes, which can influence the relationship between MAT metrics and KPIs in different ways, necessitating separate models; and (2) Feature attributes, which exert a lesser influence and can serve as supplementary features. The classification of attributes into mask and feature categories is performed heuristically.

The base of our model takes MAT metrics alongside the one-hot encoded feature attributes to form a shared representation. This base comprises seven hidden layers, each consisting of fully connected layers activated by the leaky ReLU function. Subsequently, the base branches into sub-channels, called heads. Each value of mask attributes is paired with a personalized head, which is responsible for tuning the shared representation according to the specific values of the mask attributes. Each head consists of two hidden layers and a fully

connected output layer, all using leaky ReLU activations. Ultimately, a mask selector, derived from the Kronecker product of the one-hot encoded mask attribute values, determines the final output by choosing the corresponding head. Notably, backpropagation through the entire model affects only the weights of the selected head and the shared representation.

Training methodology: We use two loss functions in training. Let $\mathbf{y}, \hat{\mathbf{y}}$ denote the true KPI and predicted KPI respectively. First, we use Mean Square Error (MSE) loss given by $\text{MSE}(\mathbf{y}, \hat{\mathbf{y}}) = ||\mathbf{y} - \hat{\mathbf{y}}||^2$. Second, we use the Wasserstein distance as a loss to ensure that the predicted KPI conforms to the distribution of the ground truth. The Wasserstein loss is calculated as follows: Let $y_{(i)}$ denote i-th sample after sorting the vector $\mathbf{y}$. Then, $\mathcal{W}_1(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n} |y_{(i)} - \hat{y}_{(i)}|$. Jointly, the training loss function is given by $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \text{MSE}(\mathbf{y}, \hat{\mathbf{y}}) + 0.5\mathcal{W}_1(\mathbf{y}, \hat{\mathbf{y}})$. Finally, we use a standard MSE loss for evaluation of the regression model.

We choose 5×10^{-4} as the initial learning rate. To accelerate the convergence, we decrease the learning rate by $\times 0.9$ if the validation loss remains roughly unchanged for 5 consecutive epochs. We use the batch size of 256 data samples.

6 Evaluation

In this section, we use a post-facto performance analysis to show the effectiveness of CIPAT as a pre-filter for CSP.

In our analysis, we use the dataset detailed in Section 2. We filter the dataset to only consider 189 out of 318 configurations which were changed at least once over 490 days. We set the DL throughput as the target KPI.

Evaluation metrics: To evaluate CIPAT as a filter for rejecting the configuration changes that degrade the cell-level KPI, we use two criteria:

(1) *Accuracy:* If a configuration change is going to degrade cell-level throughput, CIPAT should be able to identify it. High accuracy implies CIPAT can filter *most* configuration changes that result in throughput degradation.
(2) *Efficacy:* If CIPAT suspects a configuration change to degrade cell throughput, the change should result in a degradation with high probability. High efficacy implies CIPAT *only* filters configuration changes that results in a throughput degradation.

Accuracy and efficacy are essentially analogous to recall and precision if we treat CIPAT as a binary classifier. Thus, denoting the event that the observed KPI is degraded due to a configuration change by $\mathcal{D}$, and the event that CIPAT predicts throughput degradation due to a configuration change by $\hat{\mathcal{D}}$, we can define the accuracy and efficacy of CIPAT as $\mathbb{P}(\hat{\mathcal{D}}|\mathcal{D})$ and $\mathbb{P}(\mathcal{D}|\hat{\mathcal{D}})$, respectively.

Evaluation methodology: To evaluate the accuracy and efficacy of CIPAT, we concentrate on four diverse markets in the US, containing densely populated urban, suburban regions in plains and urban, suburban, rural regions with diverse geography. We use the data of 260 consecutive days for all cells in each market as a training dataset, and follow the procedures outlined in Sections 4 and 5 to construct the toolkit for each market.

We then define another 230 days period as the test period. We select cell-date pairs from the test period that meet three criteria: (1) at least one configuration change occurred, (2) there was at least a 10% change (improvement or degradation) in cell-level throughput, and (3) there was at least a 10% change (improvement or degradation) in BS-level throughput. We use 10% threshold to filter out fluctuations in KPIs that are not attributable to configuration changes.

We process each configuration change event and the associated cell-date pair in the test period through CIPAT as follows: The first stage predicts the direction of the change in all MAT metrics due to the configuration change. Using these predicted directions, the first stage constructs an orthant with the origin at the current MAT metric values for the given date. This orthant is further constrained by two conditions: (1) the number of attempted mobility management-related procedures (*e.g.*, handovers, RRC session setup) must be higher than the number of successes, and (2) the range of each MAT metric must not exceed the historically observed ranges for the given attributes. CIPAT then uniformly samples 20,000 points from the orthant, representing the possible values of MAT metrics post-configuration change. Then the pre-trained DNN predicts the cell-level throughput at each sampled point and determines the direction of predicted throughput change for each point. Finally, the majority vote is used to determine the predicted direction of throughput change associated with the configuration change. We compare the predicted direction with the observed direction of throughput change and calculate the accuracy and efficacy.

Accuracy of CIPAT ($\mathbb{P}(\hat{\mathcal{D}}|\mathcal{D})$): To measure the accuracy of CIPAT, we count the number of configuration changes from the test data that are correctly identified as degrading throughput. We then divide this by the total number of degradations observed in the test data. The resulting ratio represents the accuracy of CIPAT and its values for different markets are presented in Table 2. CIPAT achieves up to 85% accuracy in detecting throughput-degrading configuration changes. However, its performance varies across markets, with Market D exhibiting lower accuracy (around 70%). We believe this is due to the diverse landscapes within this market, encompassing both densely populated and unpopulated areas as well as plains and mountains. Utilizing hourly aggregated MAT metrics instead of daily aggregates and modeling the impact of neighboring cells might improve accuracy in such scenarios.

Market & Market attributes	Accuracy: $\mathbb{P}(\hat{\mathcal{D}}\|\mathcal{D})$		Efficacy: $\mathbb{P}(\mathcal{D}\|\hat{\mathcal{D}})$			
	CIPAT	One stage	All changes (CIPAT)	All changes (One-stage)	Novel changes (CIPAT)	Novel changes (One-stage)
(A) Densely populated metropolitan downtown, plains	**0.85**	0.62	**0.82**	0.71	**0.95**	0.67
(B) Densely populated suburban, plains	0.74	**0.88**	**0.86**	0.78	**0.93**	0.75
(C) Densely populated urban and sparsely populated mountains and deserts	**0.70**	0.67	**0.78**	0.53	**0.76**	0.54
(D) Sparsely populated regions with mountains, plateaus, and deserts	**0.78**	0.56	**0.78**	0.52	**0.81**	0.51

Table 2: Accuracy and efficacy of CIPAT and one-stage model across markets: CIPAT performs better than the one-stage model, particularly in markets with significant seasonal user dynamics. This highlights the benefit of the two-stage approach using MAT metrics in addressing the impact of underlying latent variables.

Efficacy of CIPAT ($\mathbb{P}(\mathcal{D}|\hat{\mathcal{D}})$): To evaluate the efficacy of CIPAT, we analyze its ability to predict severe degradation scenarios. We focus on configuration changes causing more than 10% decline in BS-level throughput. In this setting, we count the number of configuration changes that are correctly identified as a potential degradation and divide this by the total number of degradation predicted by CIPAT to calculate the efficacy. The results are presented in Table 2.

Efficacy of CIPAT in predicting the impact of untested configuration changes: Table 2 also highlights CIPAT's performance in predicting the impact of never-tried configuration changes, which is a crucial challenge for CSPs. This task is significantly challenging due to the lack of historical data for such settings. We define a configuration value on a cell X as "novel" if it has not been previously tested on any cell with identical attributes to X. We then filter the test set to include only these novel configuration values and re-calculate the efficacy of CIPAT on this filtered dataset. This estimates CIPAT's effectiveness in predicting the impact of untested configurations and shows the generalization capability of CIPAT. The results are presented in Table 2.

Comparison with the one-stage model: We train a one-stage model that use the configurations, the proposed change in the configuration values, and cell attributes to predict potential KPI degradation. The accuracy and efficacy of this model are presented in Table 2. Our analysis reveals that CIPAT outperforms the one-stage model by 10-40%, particularly in markets characterized by significant seasonal user dynamics. Moreover, CIPAT exhibits notably superior performance in predicting the impact of novel configuration changes (i.e., changes that were not observed in the training dataset, but occur in the test dataset). Such changes can be interpreted as a form of "out-of-distribution" changes, meaning that the test environment requires the model to predict about scenarios not present in the training dataset. This phenomenon aligns with a well-established principle in causal inference: understanding relationships prevents the learning algorithm from fitting models to spurious correlations, thereby enhancing model performance on out-of-distribution samples. In our context, CIPAT leverages an extensive collection of MAT metrics and offers a systematic process for learning relationships between configurations and KPI through MAT metrics. Consequently, it mitigates the risk of learning spurious correlations between configurations and KPI in the presence of latent variables. This crucial distinction from one-stage model (which makes no efforts in avoiding spurious relations) offers the improvement in the efficacy even for unobserved configuration changes, thus, indicates the generalization capability of CIPAT.

7 Conclusion

In this paper, we introduced CIPAT, a data-driven toolkit designed to aid CSPs in configuration tuning by filtering out potentially detrimental configuration changes. By leveraging observable intermediate network metrics, or MAT metrics, CIPAT can be constructed solely from historical data, eliminating the need for prior knowledge of underlying external factors. Our evaluation demonstrated CIPAT's accuracy and efficacy in predicting KPI impacts, even for novel, untested configuration changes, thus demonstrating generalization capability of the toolkit. By proactively identifying configurations that may degrade KPIs and result in a rollback, CIPAT can empower CSPs to reduce the number of rollbacks on the live network, thus, the cost of network management.

Acknowledgements

This work is supported by NSF grants CNS-2107037 and CNS-2112471. Changhan Ge thanks Prof. Lili Qiu at UT Austin.

References

[1] Mona Azadkia and Sourav Chatterjee. 2021. A simple measure of conditional dependence. *The Annals of Statistics* 49 (2021), 3070 – 3102.

[2] Changhan Ge, Zihui Ge, Xuan Liu, and et al. 2023. Chroma: Learning and Using Network Contexts to Reinforce Performance Improving Configurations. In *Proc. of ACM MobiCom '23* (Madrid, Spain). ACM, New York, NY, USA, Article 42, 16 pages.

[3] Ajay Mahimkar, Zihui Ge, Xuan Liu, and et al. 2022. Aurora: conformity-based configuration recommendation to improve LTE/5G service. In *Proc. of ACM IMC '22*. New York, NY, USA, 83–97.

[4] Ajay Mahimkar, Ashiwan Sivakumar, Zihui Ge, Shomik Pathak, and Karunasish Biswas. 2021. Auric: using data-driven recommendation to automatically generate cellular configuration. In *Proc. of ACM SIGCOMM'21*. ACM, New York, NY, USA, 807–820.

[5] Kartik Patel, Changhan Ge, Ajay Mahimkar, Sanjay Shakkottai, and Yusef Shaqalle. 2024. Predicting the Performance of Cellular Networks: A Latent-resilient Approach. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (Washington, DC, USA). ACM, New York, NY, USA, 3 pages.