Accelerating Causal Inference Based RCA Using Prior Knowledge From Functional Connectivity Inference

Giles Winchester
Department of Informatics
University of Sussex
Brighton, UK
G.Winchester@sussex.ac.uk

George Parisis
Department of Informatics
University of Sussex
Brighton, UK
G.Parisis@sussex.ac.uk

Robert Harper
Moogsoft Ltd
River Reach, 31-35 High Street
Kingston-Upon-Thames, UK
rob@moogsoft.com

Luc Berthouze
Department of Informatics
University of Sussex
Brighton, UK
L.Berthouze@sussex.ac.uk

***Abstract*—A crucial step in remedying faults within network infrastructures is to determine their root cause. However, the large-scale, complex and dynamic nature of modern networks makes causal inference-based root cause analysis (RCA) challenging in terms of scalability and knowledge drift over time. In this paper, we propose a framework that utilises the neuroscientific concept of *functional connectivity* – a graph representation of statistical dependencies between events – as a scalable approach to acquire and maintain prior knowledge for causal inference-based RCA approaches in dynamic networks. We demonstrate on both synthetic and real world data that our proposed approach can provide significant speedups to existing causal inference approaches without significant loss of accuracy. Finally, we discuss the impact of the choice of user-defined parameters on causal inference accuracy and conclude that the framework can safely be deployed in the real world.**

***Index Terms*—Network management, root cause analysis, functional connectivity inference**

I. Introduction

Service interruption and degradation caused by faults in the network, such as in physical or application elements, can be extremely costly to modern service providers and network operators in terms of service-level agreement violations and diminished user experience [2]. To minimise the impact of potential failures, IT operations engineers employ approaches to predict, detect and localise faults before they can cause significant disruption to network operation. In this paper we focus on approaches concerned with localising the origin of a fault, commonly referred to as root cause analysis (RCA). Recent data-driven approaches for RCA can be split into two predominant categories. The first category focuses on applying traditional causal inference methods such as pairwise statistical testing or graph-based algorithms [3]–[6] to operational data. The other category focuses mainly on deep learning (DL) approaches that learn to infer causality from collected operational data [7]–[9]. Both areas face challenges when applied to real world networks, however, in the following work we focus on the category of approaches concerned with traditional causal inference.

For causal inference-based RCA approaches the first challenge is one of scale. Modern networks have seen rapid growth over the past decades, driven by ever-increasing demand. The result of this increase in scale is that state-of-the-art causal inference-based RCA approaches must be able to provide timely fault localisation from vast amounts of collected data [10]. However, approaches that leverage causal inference techniques to deduce causal relationships from collected operational data often incur significant computational overheads as the number of variables grow [4], [11], [12]. One approach in which scalability issues are mitigated in these techniques is as a by-product of the incorporation of costly domain knowledge, often utilised primarily to improve the accuracy of the approaches [13]–[15]. Although, in some cases such prior knowledge is incorporated to address scalability directly [16]. However, the cost associated with expert domain knowledge, both in required expertise and man-hours, is generally underappreciated.

Another important but understudied challenge for causal inference-based RCA is the dynamic nature of modern networks [17]. The constant addition of new network components, software updates and the widespread utilisation of virtualisation, elastic computing and micro-services means that relationships between components of these networks are potentially in a state of constant change. Causal inference approaches, that do not require training, are in principle suited for application in dynamic environments. However, their application is impractical if the time it takes for such algorithms to infer causality is longer than the rate at which things change in the network, a major issue when considering the large computational overhead associated with such approaches. Whilst domain knowledge could help alleviate this problem, it too would have to be updated regularly to maintain

This research was funded by Moogsoft Ltd. and describes patented features of its product.

An early iteration of the framework was discussed in our Network Operations and Management Symposium poster [1].

accurate inference. If the rate of change within the network are sufficiently high, maintaining such expertly derived prior knowledge may become infeasible.

In this paper, we propose enabling causal inference-based RCA to meet the challenges outlined above by providing them access to continuously updated, data-driven, prior knowledge. This knowledge is obtained by leveraging the neuroscientific concept of *functional connectivity* (FC), used to build graph representations of statistical dependencies in the absence of ground truth. To infer FC within network infrastructure we rely on our earlier work [11] that facilitates the inference of such statistical dependencies from collected system log messages. This method for inferring FC addresses the two main challenges of modern networks addressed above. Firstly, its low computational overhead enables FC inference to scale to large networks. Secondly, the devised learning framework [11] allows us to update our inferred graph of statistical dependencies over time, addressing dynamicity.

In Section II, we describe the proposed framework, and present two approaches by which FC knowledge can be leveraged to aid RCA approaches. Additionally, we give details on how FC is inferred and updated from collected system logs in our framework. In Section III we present two state-of-the-art causal inference-based RCA algorithms [3], [18] that will be explored in combination with our approach in this paper. In Section IV we initially present experiments on synthetic data to demonstrate the scalability of our proposed framework, and its ability to adapt to changes over time, as well as further exploring the impact of user-defined parameters in our FC method on results. Furthermore, we explore the application of the proposed framework to a real world dataset in terms of speed-ups and impact on RCA precision. Finally, in Section V we discuss the deployability and additional potential advantages of our framework.

II. Framework

The speed and accuracy of many state-of-the-art causal algorithms benefit from the incorporation of prior knowledge [13]–[16]. However, such knowledge can be difficult to obtain in modern network infrastructures, where interactions can be complex, emergent, and transient. Our method for inferring functional connectivity [11] provides a highly scalable method for gaining knowledge about changing statistical dependencies between network component activities over time. In this paper, we use the knowledge gained from inferring network-wide *functional connectivity* as prior knowledge for causal inference.

In neuroscience, functional connectivity is defined as the set of statistical dependencies among neurophysiological activities [19]. This provides insights into functional (but not necessarily causal) relationships between neurological components without requiring knowledge about the underlying physical neuronal connectivity. In [11], we introduced a method for inferring FC in network infrastructures from sparse (and possibly unreliable) system log messages and showed that the inference process was scalable, making it possible to create and continuously update a knowledge graph of pairwise relationships within even the largest commercial infrastructures.

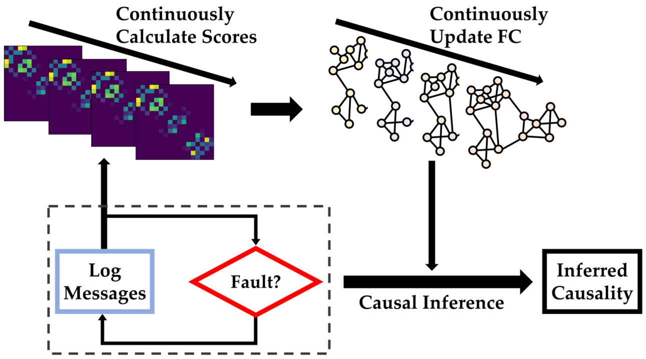

Fig. 1: The proposed framework; continuously updated FC knowledge is used to aid causal inference when a fault is detected.

In this paper, we propose a framework for the generation and application of FC as prior knowledge for causal inference-based RCA. As illustrated in Fig. 1, FC is continuously inferred to regularly update and maintain accurate knowledge graphs of statistical dependencies between network component activities over time using system logs. Once a fault is detected, the current FC is used as prior knowledge to significantly reduce the computational overhead of the chosen causal inference-based RCA approach. Specifically, we propose two methods by which to reduce the computational overhead:

1) Restricting Causal Inference to Functional Edges: A requirement for causal inference is that of some form of propagating failure states that manifests within operational data through time-lagged chronological events. For example, a failure in device A causing a subsequent failure in device B could be observable through a system event in device A followed by some time-lagged system event in device B. Thus, if two nodes within a network do not demonstrate significant statistical dependence between the time at which they emit events, then the nodes are very unlikely to be causally related. Therefore, we propose to restrict causal inference only to nodes in the inferred FC. Restricting the search space in such a manner can significantly reduce the number of hypotheses to be tested by causal inference algorithms.

2) Parallelize Causal Inference on Detected Functional Connectivity Communities: The inferred FC can either consist of a single connected graph, or multiple disconnected components. If the graph consists of sufficiently small disconnected components we can apply causal inference in parallel to each disconnected component, offering the potential for significant speed-ups. If, on the other hand, the FC is a single connected component, or disconnected components are large, the above parallelization cannot be achieved. In this case, assuming that all functional edges have equal likelihood of being a causal relationship, we apply non-overlapping community detection [20] to identify densely connected clusters in the FC. We then apply causal inference to each cluster in parallel. Doing so effectively ignores those edges linking clusters, which may reduce the accuracy of causal inference if those edges happen to be causal edges.

In what follows, we briefly summarise the metric used to detect the presence of a statistical dependence and the FC inference process.

A. Metric of Statistical Dependence [21]

In our method [11] the presence of statistical dependence between device log activities is determined by counting the number of times log events from two devices X and Y occurred within a given time lag σ given a time window T. Both positive and negative lags are considered to address the issue of concurrency and timestamp inaccuracy. Total co-occurrent counts are then compared to the expected values if X and Y were independent and uniformly distributed random variables emitting the same number of events nX and nY over the same time period T [21]. This is used to produce a score S quantifying the likelihood of X and Y being functionally connected, denoted by a functional edge between X and Y. For more details see [11], [21].

B. Model of Time-Varying Connectivity

The above scores are translated into time-varying probabilities of the existence of a functional edge through a learning framework taking into account the underlying variability of relationships over time. Specifically, the data is divided into time windows T of equal size (1 day throughout). At each time window, the scores are used to update the estimate of the probability $p_e(t_w - 1)$ of a functional edge existing between these nodes in the previous time window. The degree to which the scores in a given time window t_w drive up or down the probability from the previous window is determined by two monotonically increasing and decreasing, respectively, sigmoid functions of the score with output in [0;1]. Both functions contain a single free parameter, α and β respectively. A third and final parameter, λ, controls the intrinsic decay of probabilities over time. Such a decay is needed to account for the fact that as probabilities are only updated if at least one of the devices emits a log message, a high probability could persist indefinitely in the absence of any supporting evidence.

Prior to being used for inference, the free parameters of the model are optimised using the available training data. In this optimisation phase, an error criterion is used that minimises surprise, defined as the occurrence of observing scores between nodes X and Y in a given time window that do not match the belief of whether or not a functional edge existed between X and Y in the previous window. Errors are calculated as a function of the distance between $p_e(t_w - 1)$ and a learning threshold, whose value tunes the weighting of false negatives – a below-threshold probability in $t_w - 1$ but a high score in the current time window – and false positives – an above-threshold probability in $t_w - 1$ but a low score in the current time window – thus maximising recall or precision respectively.

Time complexity of the FC inference framework: there are three components to the framework: score calculation, parameter optimisation, FC inference. The time complexity of calculating a score, see Section II-A, is linearly dependent on the maximal delay, σ, over which cross-correlations are calculated. This maximal delay is often set to a small fraction of the total time window over which the scores are being calculated. When all nodes emit events (worst case scenario), the number of score calculations is quadratic in the size of the network due to the pairwise nature of the process. The computational overhead of the parameter optimisation process has two components: calculation of the scores based on the training data and parameter optimisation through gradient descent. The error cost used in the gradient descent is linear in the number of training days and therefore computationally efficient. Thus, the main computation cost for the optimisation phase is the time taken for gradient descent to converge. Whilst this time cannot be predicted, the number of free parameters to optimise is low (3) and our experiments suggest convergence is quick (see Figure in Section 6 for example). It is important to note that this optimisation process captures dynamic changes within the underlying causal structure, and thus as long as the rate of which these changes occur is low enough, no further optimisation is needed. The final component is the FC inference itself. Its time complexity is simply that of calculating the scores for the period considered.

III. Causal Inference Algorithms

To illustrate the potential of our approach, we demonstrate its application with two state-of-the-art approaches for extracting RCA information from system log messages [3], [4]. These two methods were chosen as they are state-of-the-art approaches applied to system log data, allow for the incorporation of prior knowledge, and implementations are open source (implementations can be found at [22] [23]). However, it is important to note that our proposal can be combined with any causal inference based RCA approach that would allow in the incorporation of prior knowledge.

A. The PC Algorithm

The PC algorithm [18], [24] is a constraint-based causal discovery algorithm. Starting from a complete graph, it eliminates non-causal edges through conditional independence tests. Once all non-causal edges are removed, remaining edges within the so-called skeleton graph are directed by identifying so-called immoralities, that is the case where for a set of three nodes two nodes are determined to be conditionally independent, but the third node is not in the conditioning set that makes them conditionally independent. Directed edges are further extended by fully directing partially direct paths incident on a collider, exploiting the fact that all immoralities would have been found in the previous step. For more detail see [24].

It has been shown that the PC algorithm can be directly employed to extract operationally exploitable causal relationships from system log data [4]. Additionally, the algorithm also forms the basis of other state-of-the-art approaches, e.g., [25]–[27]. However, because pairwise conditional independence comparisons must be made over all possible conditioning sets, the PC algorithm can incur significant processing times when applied to large-scale networks [11].

B. Topological Hawkes Processes (THP)

Introduced in [3], this method learns a causal structure from network log data whereby the intensity function of an event type, i.e., the summation of the cause event type intensity over different paths, is modelled via a graph convolution on a multivariate Hawkes process. Given a set of observations of events from a set of nodes, hill climbing optimisation is used to learn the optimal likelihood of causal structure. That is, the optimal causal structure of Hawkes processes G_V where the resultant event sequence optimally matches the observed data. Specifically, given a randomly initialised graph G_V at iteration n, the vicinity of the current graph $V(G_V)$, i.e., the set of all possible graphs one step away from G_V that is, G_V with either one edge removed, inserted or reversed in direction, is searched for the causal structure G^*_V with the highest likelihood of explaining the observed discrete data using a Bayesian Information Criterion (BIC) score. This causal graph G^*_V becomes the new causal graph G_V at iteration $n+1$ and the process is repeated until no improvement can be found. For more detail see [3].

The approach has been shown to achieve above state-of-the-art performance [3] when tested on both synthetic and real world data. However, because of the greedy nature of the optimisation process, whereby all possible graphs within the vicinity of G_V are being searched at each iteration, its computational cost grows rapidly with network size, as we show in Section IV.

IV. Results

To demonstrate the potential of our proposed framework, we designed synthetic, and chose real world datasets that elicit the aforementioned challenges met by causal inference based approaches to RCA, namely, size and dynamicity. In all cases results are shown as averages and standard deviation across 10 repeats of each experiment.To characterise causal inference accuracy, we used the F1-score. We did so for two reasons. First, as there are more non-edges than causal in the synthetic data, this metric is more informative about the ability of an approach to recover meaningful RCA information. Second, it is the metric that was used when causal inference approaches were benchmarked on this dataset [3].

A. Stationary Synthetic Data

This dataset was generated from causally related Poisson processes in the absence of noise. This is a classical approach to benchmark state-of-the-art RCA approaches, including the two approaches introduced in Section III. Specifically, we mirrored the three steps used in [3]. First, we randomly generated a directed causal bipartite graph G_N that forms our causal structure. Next, we generated data in the roots (i.e., nodes without parent) in each disconnected graph using a Poisson process with randomly generated rates μ. Finally, we generated events for all other nodes according to the causal structure with randomly generated parameters α, μ where α denotes the causal strength in the intensity function (see Section III-B). In all cases, α took values in the range of [0.03, 0.05] and μ in the range [0.00005, 0.001], derived from real world telecommunication network data [3].

To assess the impact of our proposed framework on the scalability and accuracy of both PC and THP causal inference, we generated 31 days worth of causal data using the above methodology. In all cases, causal inference was carried out on the data from the 31st day. The parameters of the time-varying model of functional connectivity were trained using the data from the first 30 days and used for inference on day 31, with a time window of 1 day, delay of 120s and learning threshold set to 0.2 in all cases. The inferred FC was then used as prior knowledge to either prune or prune and parallelize both RCA algorithms (cf the two approaches to leverage FC described in Section II).

Our results in Fig. 2A and 2B demonstrate that both PC and THP exhibit exponential time complexity, with THP revealing a markedly large computational overhead, taking 72hrs to complete on only a network size of 200 nodes.[1] However, by exploiting the prior knowledge provided by the FC inference, the computational overhead of both algorithms was reduced. This reduction was more pronounced for THP, decreasing the processing time on the largest network by a factor 130, from 61hrs to 2018s on a network of 200 nodes. The reduction was less significant for PC (factor 33 on a network of 1000 nodes), which could be a result of the overall lower time complexity of the algorithm. When using FC prior knowledge for both pruning and community-based parallelization THP saw a reduction of processing time from 61hrs to 9s. However, the PC algorithm only saw an additional 16% reduction from 1959s to 1687s. This suggests that, for the PC algorithm on the network sizes tested, parallelization does not outweigh the cost of the additional community detection and worker initialisation overheads. This could be due to the better utilisation of prior knowledge by the PC algorithm when compared to THP, or perhaps due to the modularity of the inferred FC. Note that the reported times include training time of our model; in Section IV-E we show that the cost of this training is insignificant compared to that of causal inference.

When assessing the accuracy of the causal inference, Fig. 2C and 2D, by comparing the inferred edges with the known ground truth causality structure, we found that for PC, the use of FC did not result in any significant change (with the overall F1-score being low in all cases). On the other hand, for THP, we found that both with and without community-based parallelization, the inclusion of FC-based prior knowledge improved the F1-score. Specifically, the application of FC knowledge appeared to reduce the loss of accuracy incurred

[1] We restricted the application of THP with no FC prior knowledge only to networks of up to 200 nodes. This is because larger sizes of network would have led to prohibitively long run times for the causal inference algorithm.

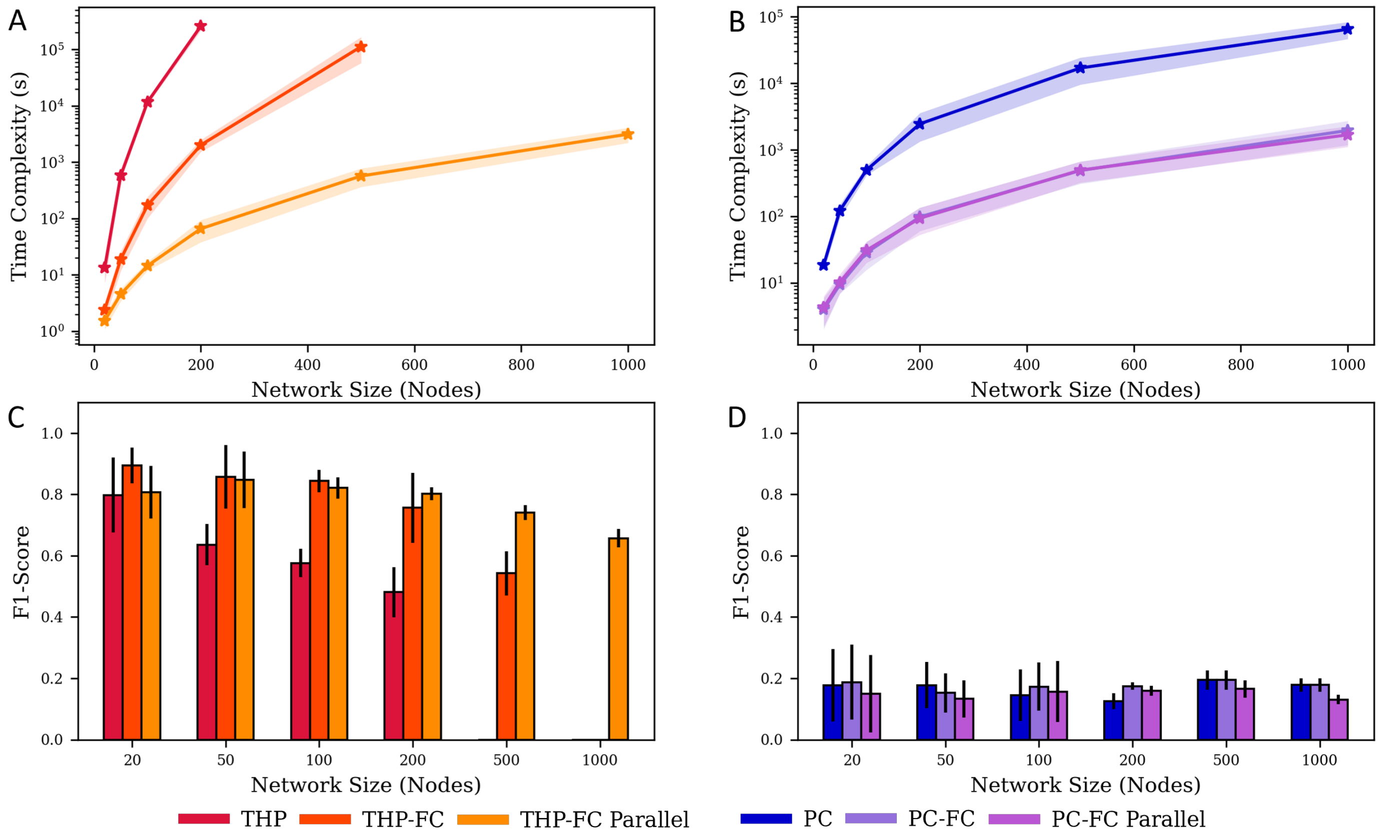

Fig. 2: Average time complexity (logarithmic scale) and F1-Score comparison of THP (panels A and C) and PC (panels B and D) on networks of different size, either applied standalone (red and blue for THP and PC respectively) or when leveraging FC knowledge for pruning (dark orange and dark purple for THP and PC respectively) or community-based parallelization (light orange and light purple for THP and PC respectively).

when deploying THP on larger network sizes (50-200), (Fig. 2C). As the decrease in accuracy can still be observed in both FC methods at even larger network sizes (500 and 1000), it is likely that this gain is due to THP reaching the maximum number of iterations. This suggests that FC prior knowledge reduces the number of iterations required to reach a solution, and thus enables the THP algorithm to find the optimal solution across a wider range of network sizes.

B. Dynamic Synthetic Data

To introduce dynamicity in the synthetic dataset, we added a new parameter controlling the proportion of causal edges to be rearranged each day, δ = [0.0,0.5]. Specifically, the three steps above were repeated to generate one day's worth of data, then a subset of nodes, N_V, where $|N_V| = \delta \cdot |V(G_N)|$, with degree at least one, were randomly selected. For each node in N_V, a random edge was chosen to be removed and reattached to another node in G_N to create a new causal graph G^*_N, used for the next iteration. This process was repeated until the maximum number of days was reached. As in Section IV-A, we generated 30 days' worth of training data with causalities changing as per the value of δ and used these 30 days to train the parameters of our time-varying functional connectivity model. To characterise the impact of changes in causal structure on the effectiveness of our proposed framework, we used the train model to predict the functional connectivity over the next 10 days and used each day's predicted FC as prior knowledge for both causal inference methods. We then calculated the averages of both time complexity and F1-scores across all 10 days. Due to the high time complexity of the THP algorithm without FC prior knowledge, and because we now carried out inference over 10 days, we limited our analysis to a network size of 100 nodes so that we could provide results with THP.

The time-complexity results shown in Fig. 3A and 3B demonstrate significant speed-ups for both causal inference algorithms in all scenarios and suggest that these speed-ups are robust to changes in the underlying causal structure, i.e., not decreasing with increased dynamicity. When comparing the inferred edges to the known ground truth causality on each day, Fig. 3C, we found that for small to moderate (1% to 10%) changes in the underlying causal structure, we saw a small decrease in performance when combining FC to prune or parallelize THP as compared to the results with the static dataset. However, accuracy was still superior to the performance of the standalone THP algorithm. Additionally, we found that the use of FC prior knowledge in the presence of

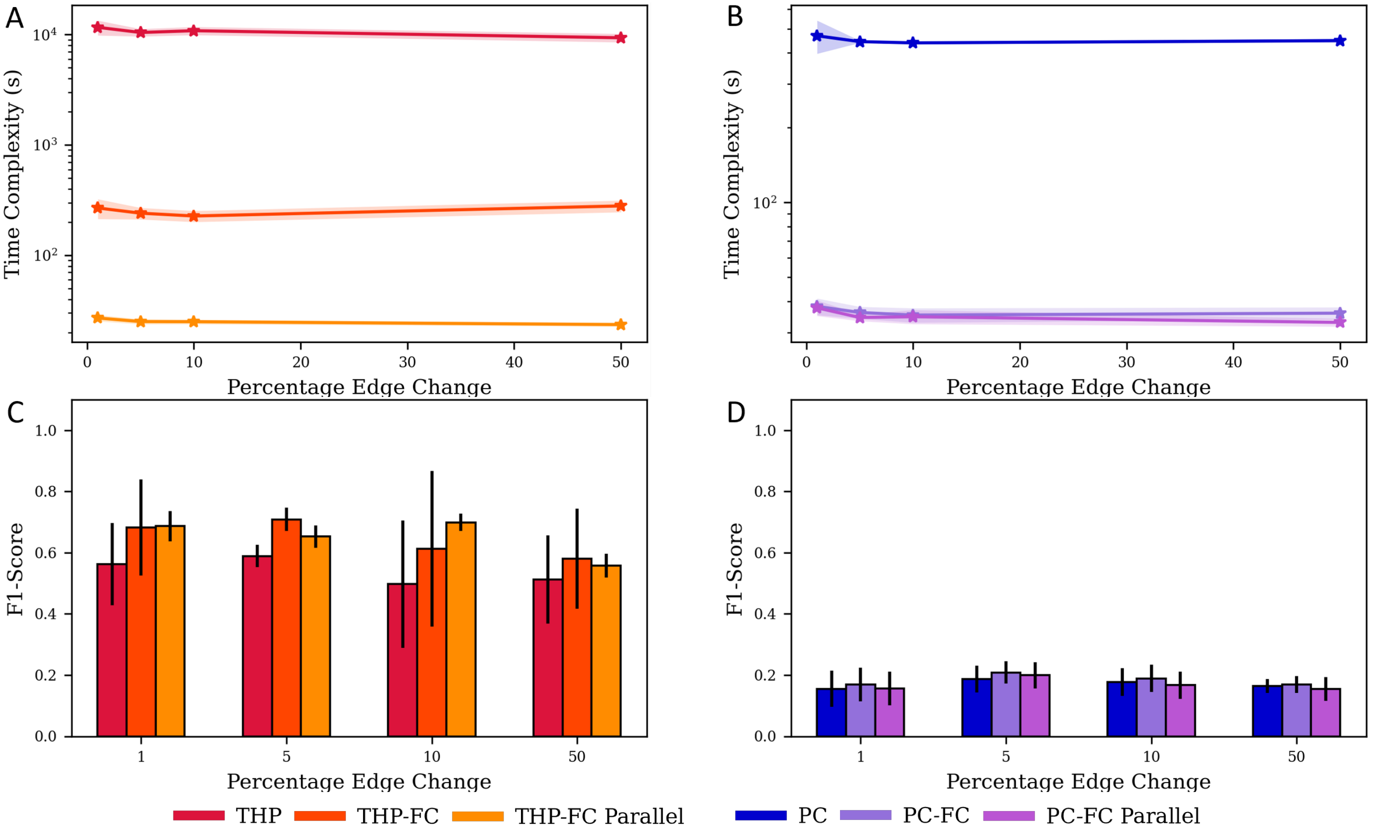

Fig. 3: Average time complexity (logarithmic scale) and F1-Score comparison of THP (panels A and C) and PC (panels B and D) when applied standalone (red and blue respectively), or combined with FC knowledge either for pruning (dark orange and dark purple for THP and PC respectively) or community-based parallelization (light orange and light purple for THP and PC respectively), with differing degrees of dynamicity.

severe changes in underlying causal structure (50%) resulted in a small further loss of accuracy, still remaining superior to the performance of standalone THP. When applying FC prior knowledge to PC, we observed a insignificant change in accuracy across all conditions as compared to the results obtained with the static dataset (see Fig. 3D). This suggests that the PC algorithm may be more robust to less accurate or incomplete prior knowledge.

C. Sensitivity to Chosen Threshold

Our method for inferring FC involves two user-defined parameters. The first parameter is the learning threshold (see Eq. 5 in [11]) which alters the degree to which true positives (TPs) or false positives (FPs) are penalised, see Section II-B for the definition of true and false positives. The second parameter is the probability threshold, which determines whether a functional edge is created between two nodes given an edge probability. The impact of this threshold is opposite to that of the learning threshold, namely, a high threshold will minimise FPs, whereas a low threshold will maximise TPs. The choice of these thresholds can affect the inferred FC, and thus impact performance when combined with the RCA approaches. To characterise the potential impact of choosing a particular threshold on the performance of the FC-aided causal inference, we carried out a sensitivity analysis of probability threshold on causal inference accuracy. We considered three different scenarios: a small network (20 nodes), a larger network (100 nodes), and a network with dynamic causal structure (20 nodes, 10% of changes per day). In all cases, average time complexity and F1-Scores were displayed as fractions, that is, relative to the time complexity and F1-Scores of the baseline algorithms run on the same scenarios.

Fig. 4 shows that when using FC prior knowledge solely to constrain causal inference (be it THP or PC - *dark solid lines for all panels*), the choice of probability threshold has little negative impact on the relative accuracy compared to baseline approaches. However, in both cases, stricter thresholds are required to gain significant speed-ups in processing time (*dark dashed lines for all panels*). This is because when the threshold is low, the FC probability matrix is very dense, and thus does not substantially add to the base assumption that all causal relationships are possible. The sensitivity to the threshold parameter, both for F1-score and time complexity, remains similar across the three scenarios tested, suggesting that the above observations are not contingent on either network size or dynamicity of the underlying causal structure.

Fig. 4 shows that when applying FC prior knowledge to

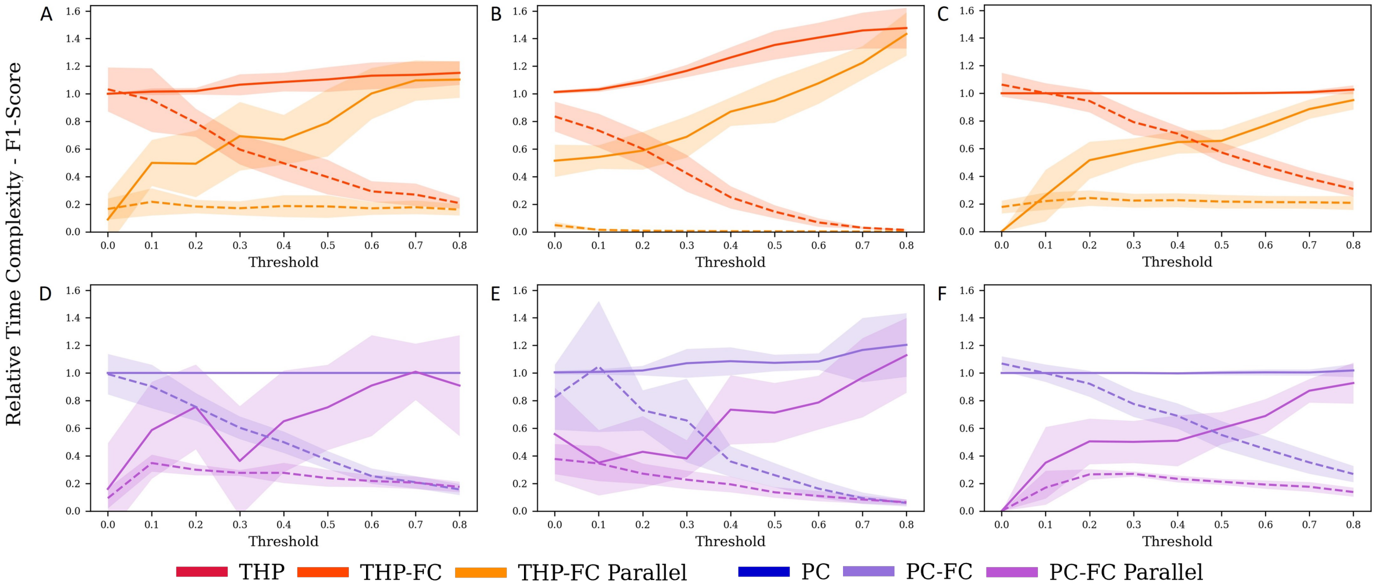

Fig. 4: Average time complexity (dashed lines) and F1-Score (solid lines), relative to the baseline algorithms across three scenarios: a small network (panels A and D), a larger network (panels B and E), and a network with dynamic causal structure (panels C and F). Dark orange and dark purple are for the combination of FC with THP and PC respectively for pruning. Light orange and light purple are for the combination of FC with THP and PC respectively for pruning and community-based parallelization. The user-defined probability threshold is shown on the x-axis.

both constrain and parallelize causal inference, we see opposite sensitivity of the F1-score (*light solid lines for all panels*) and time complexity (*light dashed lines for all panels*) to the probability threshold than in the case of applying FC prior knowledge for constraint alone. We now see an increased sensitivity of F1-Score to the choice of the probability threshold, demonstrating a reduction in accuracy when compared to the baseline algorithms in some cases. However, it should be noted that excluding the worst-case scenarios of thresholds between 0 and 0.2, we do not completely degrade causal inference, in both cases remaining above 0.5 relative F1-Score. On the other hand, we observe a decreased sensitivity of time complexity to the choice of probability threshold, demonstrating significant relative speed-ups in all cases. This suggests that even when FC provides minimal information on statistical dependencies, community detection-based parallelization can still provide substantial speed-ups. However, as is observed in Fig. 4, parallelization in these cases comes at the cost of relative inference accuracy to baseline algorithms, perhaps due to the difficulty of obtaining realistic communities from dense FC structure. Once again, the sensitivity to the threshold parameter, both for F1-score and time complexity, remains similar across the three scenarios tested, suggesting our results are not contingent on either network size or dynamicity of the underlying causal structure.

Due to space constraints, full results from the sensitivity analysis for the learning threshold are not reported here but can be found at [28]. These results demonstrate that the learning threshold can generally be safely set to any value below 0.5.

D. Application to Real-world Data

There is limited availability of datasets for which ground truth regarding the underlying causal structure is available. Here, we constructed a real-world dataset by re-purposing a publicly available dataset [23], namely, data from a metropolitan cellular network, with a total of 48572 network events collected from 439 network devices over a 6-day period. This dataset was provided with expertly curated ground truth causal relationships between 24 event types (see [3], [23] for more detail). Because our FC inference approach relies on the timing, rather than the type of events, we redefined the nodes in the network as (device, type) pairs rather than just devices. In other words, a device emitting 3 types of events was treated as 3 different event-emitting nodes. For this reason, the ground truth causal structure also needed to be adapted. Thus, if two nodes emitting events of types causally related (as per the provided ground truth) were predicted to be causally related (by causal inference), the corresponding edge was counted as a true positive. On the contrary, if causal inference predicted edges between nodes that emitted event types not causally related (as per the provided ground truth), such edges were counted as false positives. As this setup did not allow us to determine whether the lack of a predicted edge indicated a false negative or a true negative, we could only calculate precision. However, considering our aim to demonstrate that our framework can accelerate causal inference based RCA algorithms without negatively impacting their accuracy, a partial ground truth is better than none.

We trained our FC inference model on the first 5 days of the re-purposed dataset and inferred the functional connectivity

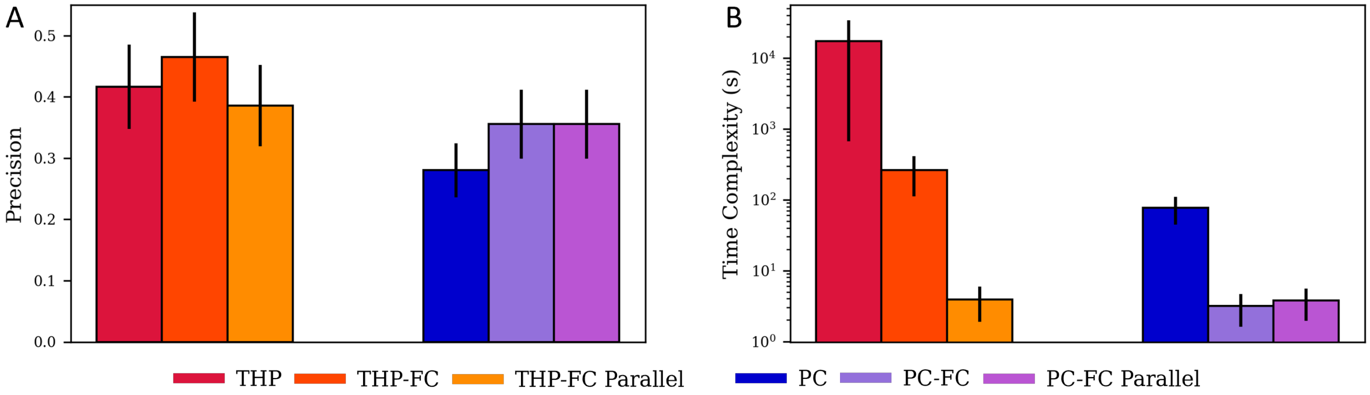

Fig. 5: Average precision (panel A) and time complexity (panel B) of both THP and PC applied either standalone (red and blue for THP and PC respectively), or with FC knowledge either for pruning (dark orange and dark purple for THP and PC respectively) or community-based parallelization (light orange and light purple for THP and PC respectively), when applied to 4-hour windows across 1 day's worth of data.

over the final day. We then ran each of the two causal inference methods (THP and PC) in 2hr windows across the entirety of the last day's worth of data. A window length of 2hrs was chosen because it was the largest length before THP became too computationally expensive to run without any prior knowledge. We used the largest window possible because both THP and PC have been shown to perform better when the number of log messages is maximised [3]. This is unsurprising since certain causal interactions occur on different timescales, sufficient data must be available to capture the most delayed interactions, and thus maximise inference accuracy. Fig. 5A shows that in all cases, using FC as prior knowledge to THP does not significantly impact the accuracy of the method. However, it does reduce the computational overhead of the algorithm, Fig. 5B, from 4.8hrs per 2h window of data to 4 seconds and 4.4 minutes when using FC with and without parallelization, respectively. It is important to note that the average time complexity of the THP algorithm is greater than the duration (2h) of the time window considered. Thus, whereas real-time execution of the standalone method on such a network would not be possible in the real world, the inclusion of FC prior knowledge would make it possible. Furthermore, because larger sample sizes would be possible, increased accuracy might also be achievable.

A similar picture emerges with the PC algorithm, namely, providing speed-ups without any significance change in precision, which remains lower than that achieved with THP.

E. Functional Connectivity Inference Time Complexity

In this section, we provide evidence that the training time (which was included in the time complexities reported previously) is negligible compared to the speed-ups achieved with the RCA approaches. The training time is made up of two components: the calculation of scores for each day of training data considered and the optimisation step. Figure 6 shows both components when training the model over 30 days' worth of data for networks of various sizes. Whilst the time spent in the optimisation phase itself (light green) increases with network size, it does so with a small and approximately constant rate of increase. The time spent calculating the scores (dark green; combined over 30 days) increases in a quadratic manner (in the worst case, i.e., when all nodes emit events) with network size, consistently with the fact that it entails pairwise calculations. Note that for the largest network size considered here, the total time is $650s$ which is very small compared to the speed-ups reported in previous sections. Additionally, if the rate of change of dynamics within the network do not significantly vary, no further retraining is required.

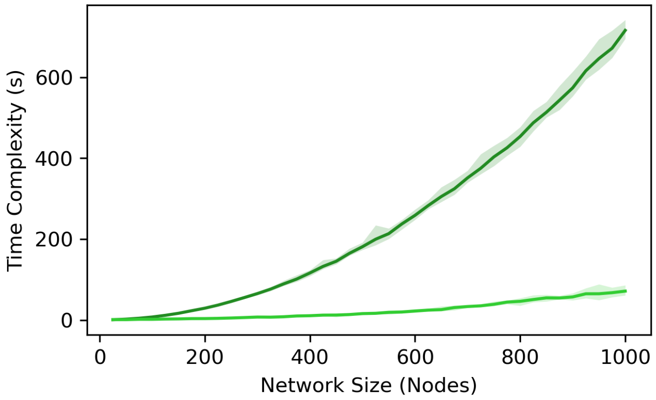

Fig. 6: Average time complexity for score calculation phase (dark green) and optimisation phase (light green) in FC inference method.

V. Discussion

The key premise of the proposed framework is that the scalability of suitable causal inference based RCA approaches can be substantially improved by incorporating prior knowledge informed by functional connectivity inference, i.e., the identification of statistical dependencies in the activity of an infrastructure's components. Such inference is computationally inexpensive and its use as prior knowledge can come at little or no cost accuracy wise. Our framework is general in so far as it can be applied to any causal inference based RCA approach that accepts prior knowledge as input. Whilst the computational cost of training our approach scales with network size, model training is an operation that does not need to be repeated unless the underlying causal structure (i.e., how the components interact) dramatically alters, and, even then, the computational cost is significantly smaller than the speed-ups achieved. Our results demonstrate that our framework can enable causal inference-based RCA to scale to sizes of networks where their standalone deployment is currently not feasible. This is true even when the underlying causal structure changes over time.

Our sensitivity analysis presents the two approaches for leveraging FC as distinct but synergistic. When speed is to be maximised above all else, community detection-based parallelization can provide significant speedups for all parameter selections, at the cost of reduction in inference accuracy. Inversely, when causal inference accuracy is paramount, FC knowledge can be used without parallelization to achieve speed-ups (albeit smaller ones) without loss of accuracy. Thus, it might be worth considering a staged deployment of each method. For example, whilst the user-defined parameters are still being tuned to maximise performance, FC without parallelization may be deployed to provide "good enough" speed-ups whilst maintaining user-confidence in the accuracy of the RCA inference. Once thresholds have been satisfactorily tuned, and the risk of significantly degrading RCA inference accuracy is reduced, operators could switch to FC with parallelization to maximise the potential speed-ups.

An advantage of our framework not explored in this paper is its ability to capture so-called emergent network interactions [29]. Such interactions may be a consequence of co-located virtual machines competing for resources in multi-tenant servers with heterogeneous hardware specifications, or through interplay between cascading failure recovery that trigger unplanned network-wide interactions. Being transient, heterogeneous, and complex, they are likely to be extremely difficult for domain experts to understand *a priori*, and thus, less likely to be represented in prior knowledge. Because our approach extracts knowledge about statistical dependencies between network components independently of whether operational meaning can be readily attributed to them, we expect our method to be able to capture and provide knowledge on such emergent behaviours.

References

[1] G. Winchester, G. Parisis, and L. Berthouze, "Exploiting functional connectivity inference for efficient root cause analysis," in *IEEE/IFIP NOMS*, 2022.

[2] J. Chen, X. He, Q. Lin, H. Zhang, D. Hao, F. Gao, Z. Xu, Y. Dang, and D. Zhang, "Continuous incident triage for large-scale online service systems," in *IEEE/ACM ASE*, 2019.

[3] R. Cai, S. Wu, J. Qiao, Z. Hao, K. Zhang, and X. Zhang, "THP: Topological hawkes processes for learning causal structure on event sequences," 2021. [Online]. Available: https://arxiv.org/abs/2105.10884

[4] S. Kobayashi, K. Otomo, K. Fukuda, and H. Esaki, "Mining causality of network events in log data," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, 2018.

[5] H. Wang, P. Nguyen, J. Li, S. Kopru, G. Zhang, S. Katariya, and S. Ben-Romdhane, "GRANO: Interactive graph-based root cause analysis for cloud-native distributed data platform," *VLDB Endowment*, vol. 12, no. 12, 2019.

[6] Q. Wang, L. Shwartz, G. Y. Grabarnik, V. Arya, and K. Shanmugam, "Detecting causal structure on cloud application microservices using granger causality models," in *IEEE CLOUD*, 2021.

[7] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *ACM CCS*, 2017.

[8] P. Zhou, Y. Wang, Z. Li, X. Wang, G. Tyson, and G. Xie, "LogSayer: Log pattern-driven cloud component anomaly diagnosis with machine learning," in *IEEE/ACM IWQoS*, 2020.

[9] C. You, Q. Wang, and C. Sun, "sBiLSAN: Stacked bidirectional self-attention lstm network for anomaly detection and diagnosis from system logs," in *IntelliSys*, 2021.

[10] A. Messager, G. Parisis, R. Harper, P. Tee, I. Z. Kiss, and L. Berthouze, "Network events in a large commercial network: What can we learn?" in *IEEE/IFIP NOMS*, 2018.

[11] A. Messager, G. Parisis, I. Z. Kiss, R. Harper, P. Tee, and L. Berthouze, "Inferring functional connectivity from time-series of events in large scale network deployments," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, 2019.

[12] M. Wienöbst, M. Bannach, and M. Liśkiewicz, "Extendability of causal graphical models: Algorithms and computational complexity," in *Uncertainty in Artificial Intelligence*, 2021.

[13] P. Chen, Y. Qi, P. Zheng, and D. Hou, "CauseInfer: Automatic and distributed performance diagnosis with hierarchical causality graph in large distributed systems," in *IEEE INFOCOM*, 2014.

[14] H. Yan, L. Breslau, Z. Ge, D. Massey, D. Pei, and J. Yates, "G-RCA: A generic root cause analysis platform for service quality management in large ip networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, 2012.

[15] J. Qiu, Q. Du, K. Yin, S.-L. Zhang, and C. Qian, "A causality mining and knowledge graph based method of root cause diagnosis for performance anomaly in cloud applications," *Applied Sciences*, vol. 10, no. 6, 2020.

[16] S. Kobayashi, K. Otomo, and K. Fukuda, "Causal analysis of network logs with layered protocols and topology knowledge," in *CNSM*, 2019.

[17] A. Dusia and A. S. Sethi, "Recent advances in fault localization in computer networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, 2016.

[18] P. Spirtes and C. Glymour, "An algorithm for fast recovery of sparse causal graphs," *Social Science Computer Review*, vol. 9, no. 1, 1991.

[19] K. Friston, "Functional and effective connectivity: A review," *Brain connectivity*, vol. 1, 01 2011.

[20] V. Traag, L. Waltman, and N. J. van Eck, "From louvain to leiden: guaranteeing well-connected communities," *Scientific Reports*, vol. 9, 03 2019.

[21] A. Messager, N. Georgiou, and L. Berthouze, "A new method for the robust characterisation of pairwise statistical dependency between point processes," 2019. [Online]. Available: https://arxiv.org/abs/1904.04813

[22] K. Shima, "pcalg." [Online]. Available: https://github.com/keiichishima/pcalg

[23] Huawei-Noah, "gCastle." [Online]. Available: https://github.com/huawei-noah/trustworthyAI/tree/master/gcastle

[24] M. Kalisch and P. Bühlmann, "Estimating high-dimensional directed acyclic graphs with the pc-algorithm," *J. Mach. Learn. Res.*, vol. 8, 2007.

[25] Y. Meng, S. Zhang, Y. Sun, R. Zhang, Z. Hu, Y. Zhang, C. Jia, Z. Wang, and D. Pei, "Localizing failure root causes in a microservice through causality inference," in *IEEE/ACM IWQoS*, 2020.

[26] K. Zhang, M. Kalander, M. Zhou, X. Zhang, and J. Ye, "An influence-based approach for root cause alarm discovery in telecom networks," 2021. [Online]. Available: https://arxiv.org/abs/2105.03092

[27] S. Kobayashi, K. Shima, K. Cho, O. Akashi, and K. Fukuda, "Comparative causal analysis of network log data in two large ISPs," in *IEEE/IFIP NOMS*, 2022.

[28] G. Winchester. [Online]. Available: https://github.com/gwinch97/CNSM_3D_sensitivity_analysis

[29] P. Garraghan, R. Yang, Z. Wen, A. Romanovsky, J. Xu, R. Buyya, and R. Ranjan, "Emergent failures: Rethinking cloud reliability at scale," *IEEE Cloud Computing*, 2018.