Probabilistic Sequence Modeling for Trustworthy IT Servicing by Collective Expert Networks

Kayhan Moharreri, Jayashree Ramanathan, Rajiv Ramnath
Department of Computer Science and Engineering
The Ohio State University, Columbus, OH
{moharrer, jayram, ramnath}@cse.ohio-state.edu

***Abstract*—Within the enterprise the timely resolution of incidents that occur within complex Information Technology (IT) systems is essential for the business, yet it remains challenging to achieve. To provide incident resolution, existing research applies probabilistic models locally to reduce the transfers (links) between expert groups (nodes) in the network. This approach is inadequate for incident management that must meet IT Service Levels (SLs). We show this using an analysis of enterprise 'operational big data' and the existence of collective problem solving in which expert skills are often complementary and are applied in sequences that are meaningful. We call such a network - 'Collective Expert Network' (or CEN). We propose a probabilistic model which uses the content-base of transfer sequences to generate assistive recommendations that improves the performance of CEN by: (1) resolving incidents to meet customer time constraints and satisfaction (and not just minimize number of transfers); (2) conforming to previous transfer sequences that have already achieved their SLs; and additionally (3) address trust in order to ensure adoption of recommendations. We present a two-level classification framework that learns regular patterns first and then recommends SL-achieving sequences on a subset of tickets, and for the remaining directly recommends knowledge improvement. The experimental validation shows 34% accuracy improvement over other existing research and locally applied generative models. In addition we show 10% reduction in the volume of SL breaching incidents, and 7% reduction in MTTR of all tickets.**

***Index Terms*—Classification; Complex Enterprise; Collective Expert Networks; Human-in-the-loop; Knowledge Management; Service Levels; Text Mining; Ticket Resolution Sequence; Trust**

I. Introduction

Within many problem-resolution environments (e.g. emergency response and triage, cloud-based service desks, supply chain resilience, software bug tracking) complex problems must be analyzed and solved within specific time constraints by networks of experts in order to meet the business or the social needs of the community. Here we research a specific case of this general problem using extensive and detailed real-world enterprise data related to IT Service Management (defined by ISO 20,000 standard [12]).

This research focuses on a particular application – the IT service desk (ITSD) and support organization which in our case study *resolve* incidents from a complex data center infrastructure and its service operations. Generally, incidents lead to service loss or disruption. Incidents perceived by customers are logged as *tickets*. Also the smallest unit of problem solving in this study is an 'expert group' consisting of technical individuals with common expertise. Our goal is to develop a statistical learning framework that recommends the best set of transfers to guide expert groups to collectively work on a ticket and meet Service Levels (**SL**). In the real world, SL is a *time-and-satisfaction-based* metric that is defined for and contracted with different lines of business customers. The framework derived from this research performs efficient incident management. The framework is also widely applicable to other service support environments characterized by *a small number of workflows that resolve a majority of the tickets*. In other words, the proposed solution benefits any environment with an observable *Pareto distribution* [10] of tickets over the resolution workflows.

The concept of 'workflow' is about expert groups that commonly work sequentially on a ticket and transfer it along to achieve resolution. Given a ticket, the sequence results in the *resolution of a ticket* and is referred to as a *Resolving Sequence* (**TRS**) for that ticket. A TRS of any ticket can be reflected as a *path* on the Collective Expert Network (**CEN**). We start by assuming that the TRSs captured in the historical incident-resolution database form a digital trace (i.e. the set of transfer sequences) of the best efforts of the expert groups thus far. We will show these efforts often fail to meet service levels on longer transfer sequences. This leaves opportunities for CEN improvement with **automated recommendation assistance**. This work establishes: (a) on frequent paths the SLs are very likely to be met, and (b) the frequent ticket content is associated with frequent paths (learned workflows) and therefore are also likely to successfully meet the SLs. Thus the research method is to make explicit the global knowledge exhibited by the CEN on frequent content and SL-achieving TRSs (i.e. paths that resolve) and use this to prevent ticket misrouting on frequent content. This is accomplished by splitting the digital trace into: (a) a trustworthy set which is used for probabilistic sequence learning and recommendations to the human experts and (b) the remaining unreliable set which is used to signal anomalies in the content to draw early human attention within the resolution process. We implement this with a two-level classification framework that is experimentally shown to: (1) improve the precision of recommendations by 34% over existing content-aware sequence models; (2) improve Mean-Time-To-Resolve by 7%; (3) reduce SL breaches by 10%; and (4) maintain a high level of trust. The validation uses held-out data generated in the production environment of the enterprise.

A. The Enterprise CEN Case

The data for this study has been collected from the IT service support organization of a large insurance company with an online business that serves over 18 million policyholders. Within this environment thousands of incidents are generated daily due to complex (diverse, layered, networked, evolving) hardware and software items called configuration items (CIs). These have to be resolved by the enterprise CEN (in our case with 916 expert groups) for IT Service Support within time constraints established by SL goals. While other current research methods use machine learning applied to this problem, they focus only on reducing the number of transfers and thus Mean-Steps-To-Resolve (MSTR) [7], [14]. However, they have glossed over the Mean-Time-To-Resolve (MTTR) which is critical for meeting SL goals. To illustrate this, data shows that longer paths (i.e. more transfers to resolve) sometimes reflect shorter time to resolve. That is, path length reduction is not the primary goal. Such discrepancies in current research and the needs of the enterprise in deploying a system useful for assisting the ITSD and CEN guided us to conduct a more detailed analysis of the enterprise operational data as the first step. The analysis presented in this paper (i.e. Section II) is therefore necessary to cull the principles that must govern any solution to assistive recommendations making CEN more effective in meeting SLs.

B. Contributions

There are three main contributions towards successful deployment within the enterprise ITSD. (1) The detailed analysis of extensive operational data to motivate the CEN conceptual model appropriate for time-constrained problem solving by expert networks as elaborated in Section II. (2) Using CEN behavior insights obtained from the analysis to develop the principles that must be met by assistive and trustworthy recommendations in a two-level framework as explained in Section IV. And (3) The supervised learning model that meets the principles along with the experimental setup that show performance improvement as presented in Section V. In addition to these core contributions, the rest of the paper consists of the related research in Section III, framework evaluation and comparison in Section VI, and finally conclusions and future work in Section VII.

II. CEN Analysis

In this section we present the terminology and then characterize the causes for poor performance. Existing poor performance despite the enablement provided by current processes and systems provide an understanding of the opportunities for improvement. To avail of these we extract principles for recommendations that address specific causes in a manner consistent with CENs own natural behaviors. With the data analysis below we systematically cover all observed aspects of current CEN performance, CEN behavior characteristics with respect to content and transfers, and principles that guide beneficial assistance.

A. Terminology and Formalism

This formalism has been motivated by our analysis and insights of collective behaviors that exist within the IT service support organization and can be captured by networks [3]. By presenting the conceptual model the insights can later be presented more succinctly.

To start with we define a Collective Expert Network (CEN) on a set of resolved tickets T as a directed graph where expert groups and transfers represent vertices and edges respectively:

$$CEN(T) = (Experts, Transfers) \tag{1}$$

For a ticket $t \in T$ a resolving sequence is $t.rs$:

$$t.rs =< e_{(1)}, e_{(2)}, ..., e_{(k)} > \tag{2}$$

Here $e_{(i)}$ is the ith expert group which was working on t and received t from $e_{(i-1)}$, and transferred it to $e_{(i+1)}$ ($<>$ denotes an ordered list). The last expert group in the sequence achieved resolution and is noted as the 'resolver' ($t.resolver$). Note that the above definition accommodates duplicate elements in the sequence. Therefore, any resolving sequence is a *walk* on the CEN. It is important to note that in network theory the definition of a path does not entail duplicate vertices but here a resolving sequence does. So for simplicity we call any resolving sequence a *path* even though it contains duplicate vertices. The $Experts$ set is defined by the union of expert groups that have worked on at least one ticket in T:

$$Experts = \bigcup_{t \in T \ \wedge \ e_i \in t.rs} e_i \tag{3}$$

$Transfers$ is the set of expert group pairs of the form (a, b) which is a directed edge that belongs to $Transfers$ if there is at least one ticket transferred in T from expert group a to expert group b. Formally:

$$(a, b) \in Transfers \quad if \ \exists t \in T \mid < a, b > \sqsubseteq t.rs \tag{4}$$

Here $< a, b >$ denotes an ordered pair and $\sqsubseteq$ is the notation we use for a 'contiguous subsequence'. Also note that we explicitly add self-loops to indicate resolvers as follows:

$$(a, a) \in Transfers \quad if \ \exists t \in T \mid \ t.resolver = a \tag{5}$$

Considering only edges and vertices as a base definition for CEN, next we enhance the directed graph to make it a *weighted* directed graph. Let the set of all tickets that got transferred from a to b be denoted as T_{ab}, then:

$$T_{ab} = \bigcup_{t \in T \ \wedge \ <a,b> \sqsubseteq t.rs} t \tag{6}$$

Now we define a weight for each edge (a, b) as the count of tickets in T that got transferred along (a, b): $w_{ab} = |T_{ab}|$ Also a self-loop (a, a) is weighted as w_{aa} and evaluates to the count of tickets resolved in a. In order to obtain insights about the CEN, we propose a transformation on the weights introduced above. This transformation yields a *Markov Chain* for the CEN which is a 'memoryless' probabilistic directed graph. The resulting Markov chain is also atypical (compared

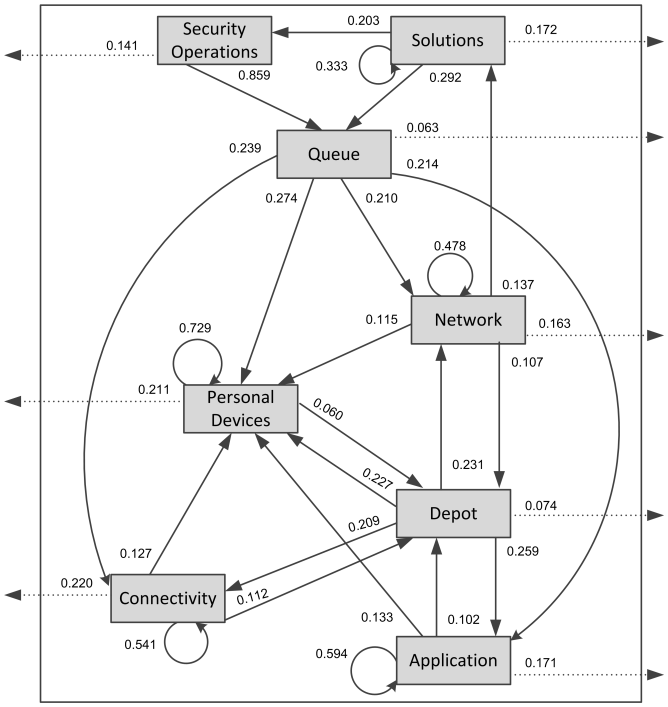

Fig. 1. A strongly connected component of the Collective Expert Network within the enterprise with edge weights as conditional transfer probabilities. Self-loops represent resolution.

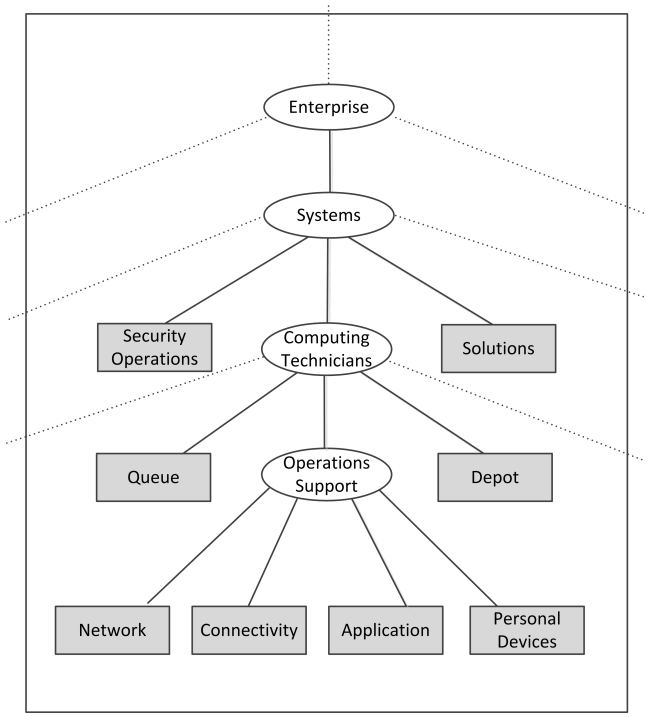

Fig. 2. Enterprise Taxonomy tree Associated with the Connected Component of the CEN of Figure 1

with [14]) since it contains self loops characterizing resolvers. w'_{ab} is the probability that a ticket was transferred to b after that the ticket was received at a. Formally that is evaluated as:

$$w'_{ab} = P(b \mid a) = \frac{w_{ab}}{\sum_{[c \in Experts \wedge (a,c) \in Transfers]} w_{ac}} \quad (7)$$

Also w'_{aa} can be interpreted as the probability that a resolves a ticket after receiving it. To illustrate the Markov representation of the CEN in our case, the Tarjan algorithm [18] was used to obtain strongly connected components. Figure 1 illustrates a strongly connected component of the CEN in which each vertex is reachable from any other vertex. Note that low-frequency edges ($w_{ij} < 60$) were removed upfront to focus on dominant transfer patterns. Some of the insights from this are: (1) the expert group 'Queue' almost evenly distributes all of its tickets among 'Connectivity', 'Personal Devices', 'Network' and 'Application'. (2) 'Queue' does not resolve any tickets. (3) 'Application' resolves more than half (0.594) of all the tickets it receives, and transfers almost a third of its non-resolved tickets (0.133) to 'Personal Devices' which is then more than 70% likely to get resolved at 'Personal Devices'. Thus the figure captures dynamics of workflows from resolving sequences illustrating that the nodes of a CEN play specific roles in a more global context. For example, 'Depot' does not resolve tickets, it appears to mediate among four other groups. More detailed insights about roles are discussed next.

Enterprise Taxonomy (i.e. ET) is a view of the CEN constructed from transfer labels obtained within the enterprise system. These labels were found to be related as a tree. Each internal node of this tree represents a conceptual scope of responsibility (abstract role) and each leaf represents an expert group (concrete role in the CEN). In this structure, a child is a subarea of its parent. In knowledge representation terms, if a child and its parent are both internal nodes they have a 'part-of' relationship. Otherwise the child is a leaf and has an 'instance-of' relationship. Figure 2 illustrates an ET corresponding to the CEN of Figure 1.

Transfer distance: For each ticket t we define transfer distance $t.td$ which is the average pairwise distance on the taxonomy tree between consecutive pairs of expert groups in $t.rs$. Formally:

$$t.td = \frac{1}{\mid t.rs \mid -1} \sum_{j=1}^{|t.rs|-1} d_{ET}(e_{(j)}, e_{(j+1)}) \quad (8)$$

where d_{ET} is the pairwise distance function on the ET. Later in Section II the transfer distance is used for further analysis.

Note that the ET tree and its association patterns make explicit the important role of collective problem-solving behavior. The CEN has different skill groups working hard to resolve the tickets by exhibiting *collective intelligence*. That is, groups of individuals working together display intelligent behavior that transcends individual contributions as in [4].

Returning back to the CEN view, an incident has two phases, (1) the *discovery* phase by the end of which main characteristics of the event is captured in a ticket, and (2) the *resolution* phase in which the expert groups identify the problem and restore the service to meet SL. The attributes collected by the end of the discovery phase are:

- $Content$: Ticket content is text explaining the incident in natural language. This field is a summary of what is reported by the end-user. (denoted as $t.c$ for a ticket t).
- $Priority$: Ticket priority is decided based on severity and urgency of the reported incident to the customer (denoted as $t.p$). Priority is an integer between 1 to 4, where priority 1 signifies the most business-critical class of tickets.
- $Service-Time$ (ST): Each ticket has an assigned service time defined as a part of its SL which indicates the time limit by which the ticket has to be resolved to maintain minimal impact to the business. The Service Time of a ticket is decided based on ticket priority (denoted as $t.st$). Details are also provided in Table I.

The attributes generated after the resolution phase are:

- $Time-To-Resolve$ (TTR): The time duration taken to resolve the ticket (denoted as $t.ttr$).
- $Resolving-Sequence$ (TRS): As defined above this is the sequence of expert groups in the order they worked on the ticket (denoted as $t.rs$ or simply TRS).

SL goal or (*SL*) for incident management is to achieve ticket resolution in collaboration with the customer and achieve this within the predefined service time of each ticket. In other words for each ticket the objective is: $t.ttr \leqslant t.st$. In an aggregate level, we define a SL metric called *Breach Ratio* on a ticket set which measures the ratio of the tickets in the set that did not meet their service time to all of the tickets. Assuming $\mathbb{1}$ denotes the indicator function which takes a conditional statement and returns 1 if the statement evaluates to true, formally for a ticket set T' we define Breach Ratio as:

$$T'.breachRatio = \frac{\sum_{t \in T'} \mathbb{1}(t.ttr > t.st)}{|T'|} \tag{9}$$

B. Current CEN Performance

Incident Context and CENs Digital Trace: The digital trace of CEN problem solving has 7250 distinct paths that resolved incidents that were generated from over 7400 Configured Items (CIs) in the IT infrastructure. The operational data from the enterprise analyzed here consisted of 149,000 user-perceived tickets reported over a period of 13-months and resolved by 916 unique expert groups through 267,721 transfers.

The priority levels are set between P1 (highest priority and impact) to P4 (low priority and impact). The SL goal is more relaxed for lower priorities. If the SL goal is not met the ticket is said to breach the SL. Table I depicts that the CEN more often resolves highest priority tickets within SL goals, and SL breaches occur more with lower priority tickets.

Collective problem solving and performance: Longer TRSs cause more difficulties in SL compliance. To prove this, our objective is to examine the TRS length of the tickets against their breach ratio. Given our ticket set, Figure 3 demonstrates (1) $P(|t.rs| = TRS\_length)$ that is the probability distribution of the length of TRSs, and (2) $P(t.ttr > t.st \mid |t.rs| = TRS\_length)$ that is the breach ratio of tickets conditioned on their TRS length.

TABLE I
PRIORITY OF TICKET RELATED TO THE BREACH RATIOS

Priority	Business Impact	Service Time	% of all	Breach Ratio%
1	Significant	14 hrs	10.8%	6.1%
2	Moderate	34.5 hrs	45.1%	7.6%
3	Minor	46 hrs	25.5%	8.1%
4	Negligible	115 hrs	18.6%	10.0%

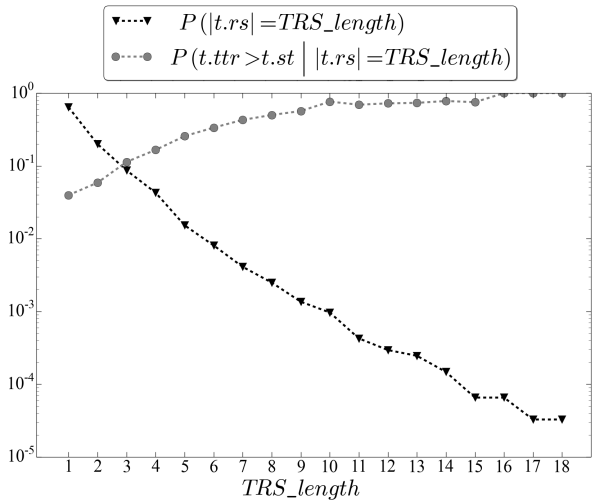

Fig. 3. Distribution of tickets per TRS length and Breach ratio of tickets per TRS length

Observations: (1) The CEN is able to resolve most of its tickets via short TRSs that is $P(1 \leqslant |t.rs| \leqslant 4) = 0.79$. More generally, Figure 3 illustrates an *exponential decay* in the volume of tickets as the TRS length increases. In other words, a transfer chosen by the CEN to be executed on a ticket is highly expected (with the probability greater than 0.5) to resolve that ticket. This establishes a *power law* [16] distribution. Here for our ticket set, we experimentally found the power law function that represents the probability of a ticket being resolved by a TRS of length h where $h \in \mathbb{N}$:

$$P_{resolve}(|t.rs| = h) = 0.56\ e^{(-0.82(h-1))} \tag{10}$$

(2) Also per Figure 3 as the TRS length increases, the probability of the tickets breaching their SL increases. Although longer TRSs are unlikely to occur, they are highly likely to breach their SL. This presents an opportunity for improvement by avoiding wrong transfers which are the leading cause of longer resolving sequences, thus saving many tickets from the inevitable SL breaches.

CEN transfer knowledge has semantic dependency associations: We found that there are semantic association patterns in the ET of the CEN. To show this we contrast two CEN views - the network view (Figure 1) and the taxonomy-based view (Figure 2). With Figure 2 we found the ET tree labels identify expert groups based on semantics of the knowledge that they possess related to: (1) a technology or application, (2) a region of the physical facility, (3) a major enterprise project, (4) a mediator or resolution role, or (5) a virtual node representing a collection of sub nodes. The labels have

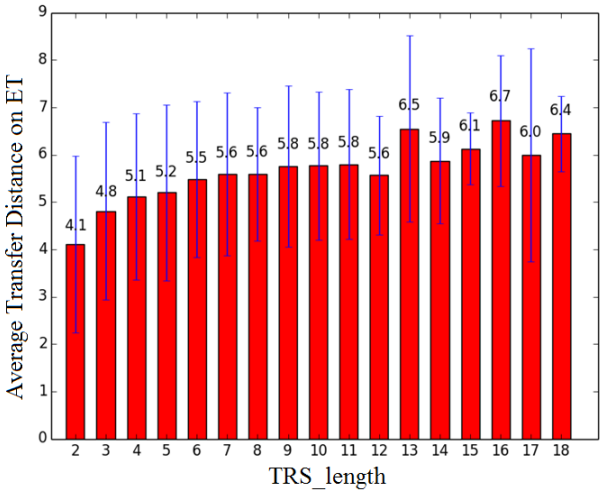

Fig. 4. Average Transfer Distance on ET grouped by $t.rs$ length.

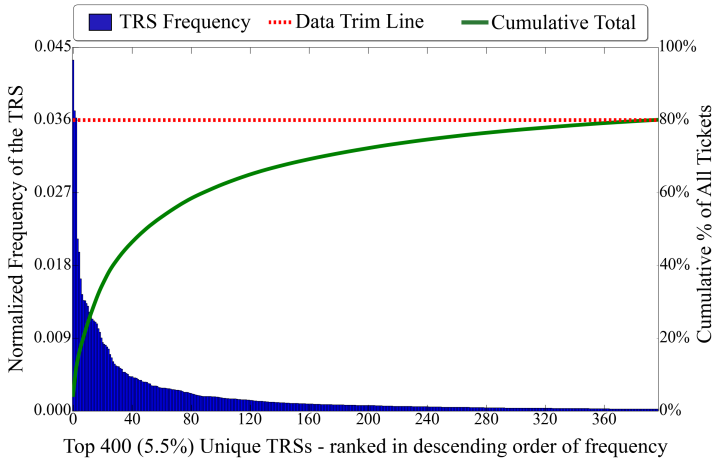

Fig. 5. Normalized frequency of paths – Pareto Chart

emerged over time and are locally used by humans interacting with the workflow routing menu of the enterprise system (without any assistance). The labels were found to form the ET tree that makes explicit (in Figure 2) the knowledge associations that are not shown in the view of Figure 1. With this ET tree as the basis we also found that as length of $t.rs$ increases, the average $t.td$ also increases as shown in Figure 4. This implies that tickets with longer TRSs are more likely to have long-distance transfers on the ET and this signals increased incident complexity due to expertise needed from distanced subtrees.

Observations: (1) On more frequent and shorter paths when SLs are met they are also more likely aligned to the ET tree structure (i.e. low average transfer distance). This shows that the unassisted CEN problem solving is naturally of a collective nature because specific sequences of labels of the ET tree have become learned tacit transfer knowledge. The longer paths are less frequent and they imply greater transfer distance on the ET tree. This together with the semantic understanding of the ET tree tells us that the increase in transfers is due to complex or less frequent content needing collective knowledge utilized from groups with very different skills on the tree. And these long paths more often fail to meet SLs, thus, making the opportunity for improvement a little more explicit.

(2) We also observed within some of the SL-compliant TRSs, the nodes repeat. The entire TRS applies knowledge based on the workflow needed and what collective knowledge is needed next. To further illustrate with a simple case consider the following ticket content: "Application X is not able to connect to database D". The t.rs for this is the following sequence: 'Connectivity' $\rightarrow$ 'Database Administration' $\rightarrow$ 'Connectivity'. The TRS thus has global characteristics, e.g. connectivity group needs to execute problem solving *twice*: the first time partially contributing to problem solving as a 'collective contributor'; and the second time as 'resolver' who also integrates and tests the solution. The SL is achieved collectively by the entire TRS working start-to-finish, and not simply by local fast-working groups. *This means that the Markov property which makes local statistical learning feasible does not generally hold in the CEN context.*

(3) Not all groups act as resolver groups and many TRSs contain wasteful transfers due to lack of global transfer assistance with the total problem faced by the CEN.

Note the points (1)-(3) above together provide the motivation for conceptualizing the entire TRS as a collective sequence with global workflow characteristics.

Is there a viable business opportunity? The aspects of analysis above point towards and opportunity for the assistive model to help with the potential misroutable tickets where the SL breaches. We must establish that there are enough such cases and there is adequate performance improvement. The business rationale required was in the form of potential improvement in performance versus resources needed to achieve that improvement. The performance improvement metrics identified were : (1) improve Mean-Time-To-Resolve; (2) reduce SL breaches; (3) reduce the number of transfers for specific priorities; and (4) maintain a high level of trust to ensure the system is used and the investment is beneficial.

C. Digital trace characteristics – content & transfer knowledge

In the previous subsection we established that the CEN could better benefit the business from recommendation assistance on longer transfer sequences that are (1) more likely to breach SL goals, and (2) that the entire sequence has global associations that are tacit and also difficult for the CEN to exhibit. Given the observations, the next related questions are: (1) Is there machine learnable regularity exhibited in the paths of the CEN; (2) How are the content and the paths related? and (3) How do we ensure that the CEN trusts the recommendations?

Regularity of the paths: Many of the paths are very common reflecting the fact that the CEN's digital trace of *collective problem solving is not erratic*. The related analysis is in Figure 5. This figure also shows that the *Pareto Principle* holds: 5.5% of paths resolve 80% of the tickets. This skewed distribution of the tickets over the paths helps identify the

subset of the paths that overcomes the challenge of *data sparsity*, leading to effective machine learning on that subset. Next we found that frequent content was also associated with frequent paths that are more successful.

Observations: Regularity of global network knowledge is exhibited by frequent paths (refer to as 'Routine' paths) that mainly resolve certain frequent content. From the machine learning standpoint, the goal is to choose a subset of all the paths, which contains frequent and well-separated paths (classes) for a multiclass classification algorithm resulting in a generalizable trustworthy classifier. From the standpoint of benefiting the CEN practically, the goal is to provide recommendations of routine paths that contain high-performing global patterns and thus prevent tickets with frequent content from taking suboptimal 'non-routine' paths.

D. Machine learning goals: trustworthy recommendations with Routine-Content and Human-in-the-Loop with Non-Routine-Content

With the potential for beneficial assistance established above along with the business motivation, we next formulate the machine learning research problem to address the observations immediately above.

The goal is to develop assistive recommendations where the machine itself (1) determines the conditions under which it can learn and recommend based on some trustworthiness criteria; (2) learns the global network knowledge in terms of TRSs that can assist the CEN to meet SL; and finally (3) flags where trustworthy recommendations are not possible and in these cases increases the reliance on human problem solving (i.e. without recommendations) and put effort into dynamic knowledge creation. Thus this approach requires the machine to *differentiate* between the **Routine (R)** problem solving where it learns and recommends to meet SLs more effectively; from the **Non-routine (NR)** where the human experts do better to achieve SL and recommendations are not trustworthy.

Heuristics were adapted from [9] to label a subset of paths as Routine (R-TRS) and the complement set as Non-routine (NR-TRS). In Section IV, all of the tickets will be used to train the top-level R/NR classifier, and routine paths only will be used to train the second-level multi-class classifier.

The breach ratio of the R-TRS class was found to be almost one-fourth of the NR-TRS class (2.26% to 8.25%). That is if a recommendation correctly saves a ticket from a non-routine path by recommending a R-TRS then there is 72% *reduction* in its likelihood to breach. In our study, 14% of all the tickets had regular content that got misrouted to a NR-TRS and the recommendation system could save these cases. This leads us to the conclusion that the expected breach ratio reduction overall through beneficial recommendation is $14\% \times 72\% = 10\%$.

E. Principles for improvement

According to [14], [7], [2], the existing research uses statistical inference models to perform target prediction. These studies trained their models on a set of triples of the form $< t.c, current - expert, next - expert >$ to locally infer next most likely expert. In all of those studies, experimental reduction in the MSTR is reported for a set of test tickets. However, the real world enterprise deployment goal that is meeting SL is particularly overlooked. The analysis in the previous subsections has established the following principles and our novel research goals:

1) *Business performance is related to improved MTTR:* That is, in contrast to MSTR of previous research, MTTR is a customer-facing measure and needed for ITSM. We need an objective function maximizing likelihood of meeting service times.
2) *Assistive recommendations must be consistent with previous CEN behaviors along the entire TRS:* As presented earlier, we note that R-TRS's are well-defined workflows throughout which incremental contributions are made in the context. Thus the machine learning and recommendation must be on the entire TRS. (i.e. No conditional independence assumption)
3) *Trust is not achieved by noisy transfers:* Noisy transfer sequences with low probabilities for achieving SL goals are not to be used for machine learning and are to be filtered out through the R/NR classification. This will need a first level for Routine (R)/Non-Routine(NR) inference and the second level for actual path recommendation to improve resolution within SLs.
4) *Trustworthiness of recommendations must be considered:* At the User Interface, the presentation of the specific recommendations must be followed by the percentage of times it led their colleagues to successful resolution on that content. The human is also notified when a trustworthy recommendation is not available and the knowledge base must be improved.
5) *Experiments must demonstrate improvement where the CEN struggles with content:* The CEN is actually performing as well as it can on frequent and high priority tickets and the SLs are being met. Thus experiments in the later Section explicitly show that there is enough *other opportunity* to improve SL related to lower priorities or the longer TRSs due to (1) poor transfer knowledge, (2) content that is truly complex or new and the transfer knowledge is not explicit, and (3) lack of resources or training [17].

III. Related research

Collaborative problem solving is leveraged today in many collaboration ecosystems. Question-answer microblogs such as Stack Overflow, Quora, and WebMD have focused on taking advantage of wisdom of the 'qualified crowd' in order to answer questions in respective domains. Other systems go further to exploit content expertise and network knowledge in complex problem solving. Examples include medical systems such as TriageLogic and InXite focus on resolving complex medical cases through ***collective*** collaboration between care providers. Work as a Service (WaaS) research in [11] proposes a hub to achieve responsiveness and address unpredictability.

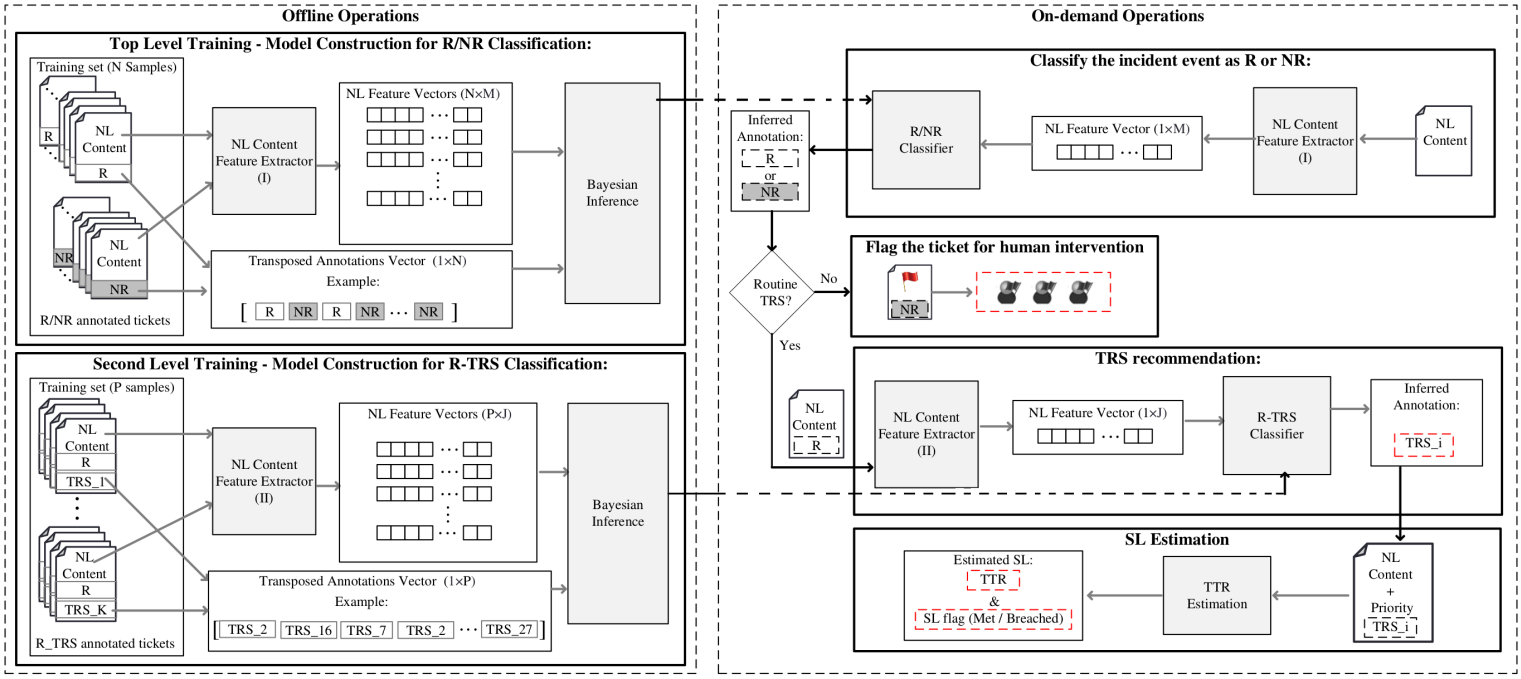

Fig. 6. Dynamic CEN recommendation framework

In general, however, the design of statistical models to guide CEN groups to ***collectively achieve SL*** has not been addressed.

In the fields of Computer Supported Cooperative Work and Social Networks, coordination mechanisms that address the increasing complexity of collaboration has been extensively studied [1]. As mentioned previously, pure inference models [14], [7], [2], [8] focus on showing improvement by making assumptions that are not valid for improving SLs. Chief among these limiting assumptions are: knowledge applications are locally determined, MSTR improvement goals suffice (and MTTR remains unaddressed), and a lack of knowledge improvement strategies that treat the real world as static. Other approaches to enhancing the knowledge management through community aware strategies are in [6]. The use of event logs to reconstruct the process model as executed has been studied by [19], [20] under the topic of process mining. The applications explored include process conformance and data provenance. These methods are relevant for extracting span time, queue time, repeating patterns, etc. for better TRS predictions. Finally, on-demand real time score and recommendation systems are becoming increasingly popular. These systems are most effective where critical decisions are to be made in massive-scale within limited periods of time, and otherwise can get heavily impacted by constrained and error-prone human performance. Their applications range from intelligent decision support systems [21], to automated response assessment [15].

IV. Enterprise CEN Deployment

The two-level framework is illustrated in Figure 6. The model developed is divided into offline training (left), and on-demand recommendations (right). Offline training includes computationally intensive operations and they lead to construction of the classification models. Formal details of training are in Section V. On-demand recommendations apply the classifiers on the unlabeled data and recommend actions for achieving SL goal. Formalization details of recommendations and their validation are given in Section VI.

Offline Operations – Top Level Training: Here we use a Bayesian binary classifier that takes Natural Language (NL) content, and identifies whether it is associated with highly frequent paths (marked as Routine). Labeling strategy (R/NR) for the paths is adapted from [9]. As shown in Figure 6, NL features are extracted from training tickets and then used along with their R/NR annotations to perform Bayesian inference. Thus the top-level R/NR classifier is constructed for on-demand use.

Offline Operations – Second Level Training: Next we use a Bayesian multiclass classifier that takes NL content and identifies a Routine path that is most likely to resolve the incident. Here Bayesian inference is only performed for tickets with routine resolving sequences. NL features are extracted from those tickets ($t.c$) and are annotated with their associated TRSs. Thus the second-level R-TRS classifier is constructed for on-demand use. We will discuss and address the underlying challenges of dealing with skewed class distribution in Section V.

On-demand Operations: On the right of Figure 6 we show the two-level application of the method on an unlabeled ticket. First we determine if the NL content of the ticket is associated with either R or NR using the top level R/NR classifier. Second, if it is associated with R then the second level R-TRS classifier is applied to provide the path recommendation for the CEN. Also SL estimation is performed for the recommended

path. If the content is associated with NR class then it is flagged and turned over to the CEN for resolving without assistance. In Figure 6 within the on-demand operations box, all of the dotted boxes are denoting predicted values. In Section VI we discuss the validation and SL advantages of the framework.

V. Experiments using the two-level classification framework

Based on the existence of a strong relationship between frequent content and routine paths, we proceeded to build the classifiers. The learning algorithm we leveraged is the Transformed Weight-normalized Complement Naïve Bayes (TWCNB) [13] for both top and second level classifiers of the Framework introduced in Figure 6. This algorithm is designed to perform on skewed training data, and it incorporates effective weight normalization and feature transformations. Further rationale for selecting this method follows.

Training and classification (R/NR and TRS recommendation): We modified TWCNB for path (R-TRS) classification as follows. Let:

1) $\vec{t}$ be the training set of routine tickets that previously got resolved by an R-TRS: $\vec{t} = (\vec{t_1}, \vec{t_2}, ..., \vec{t_n})$ and t_{ij} is the frequency of the j_th word of the dictionary in ticket $\vec{t_i}$.
2) $\vec{RS} = (r\vec{s}_1, r\vec{s}_2, ..., r\vec{s}_n)$ be the resolving sequences corresponding to each of the training tickets.
3) $C = \{C_1, C_2, ..., C_s\}$ be the set of distinct paths.
4) $\vec{test} = (f_1, f_2, ..., f_m)$ be a test ticket where f_j is the frequency of the j_th word of the dictionary in the test ticket.

Then *train* and *predict*:

$$\omega = R\text{-}TRS\_Training(\vec{t}, \vec{RS}) \tag{11}$$

$$Predicted\_label(\vec{test}) = \arg\min_{c \in C} \sum_{j=1}^{m} f_j \cdot \omega(j \mid \overline{c}) \tag{12}$$

Algorithm 1 R-TRS_Training $(\vec{r}, \vec{RS})$

1: **for** $j = 1$ to m **do**
2: $\quad IDF_j = \log \dfrac{n}{\sum_{k=1}^{n} \delta_{kj}}$
3: $\quad$ **for** $i = 1$ to n **do**
4: $\quad\quad TF_{ij} = \log(t_{ij} + 1)$
5: **for** $j = 1$ to m **do**
6: $\quad$ **for** $i = 1$ to n **do**
7: $\quad\quad NC_{ij} = \dfrac{TF_{ij} \cdot IDF_j}{\sqrt{\sum_{k=1}^{m} (TF_{ik} \cdot IDF_k)^2}}$
8: **for** $j = 1$ to m **do**
9: $\quad$ **for** $h = 1$ to s **do**
10: $\quad\quad \hat{P}(j \mid \overline{C_h}) = \dfrac{\lambda + \sum_{k:rs_k \neq c_h}^{n} NC_{kj}}{m\lambda + \sum_{k:rs_k \neq c_h}^{n} \sum_{p=1}^{m} NC_{kp}}$
11: $\quad\quad \omega(j \mid \overline{C_h}) = \dfrac{\log \hat{P}(j \mid \overline{C_h})}{\sum_{k=1}^{m} \log \hat{P}(k \mid \overline{C_h})}$
12: **Return** ω

The function call $R\text{-}TRS\_Training(\vec{t}, \vec{RS})$ is elaborated by Algorithm 1 which performs the training. It uses a set of transforms for term frequencies adapted from [13]. These transforms resolve different poor modeling assumptions of Naïve Bayes classifier including skewed word and class distribution. ω is the transformed weighted normalization function over $P(j \mid \overline{c})$ where j can be the index of any word in the corpus dictionary, and $\overline{c}$ can be complement of any class in the data set (distinct paths in this case). Some details of Algorithm 1 are: Line 2 constructs inverse document frequency transformation where $\delta_{kj} = 1$ if the j_th word of the dictionary is in ticket $\vec{t_k}$, otherwise $\delta_{kj} = 0$. Line 3: n is the number of tickets in the training set. Line 4: constructs term frequency transformation. Line 7: provides the length norm, where m is the size of the corpus dictionary. Line 9: s is the cardinality of the set C. Line 10: builds a smoothed probability function that estimates the probability of j_th word of the dictionary not in the class C_h. Line 11: is the log weight normalization of $\hat{P}(j \mid \overline{C_h})$.

Experimental process overview: For both classifiers Figure 6 we extracted features from the NL content and the text was first transformed to vectors with weighted normalized values as discussed in the 'dampening the effect of skewed data bias' section V. We dropped the stop words and removed low-frequency words, thus reducing the dimensions of our feature vectors to 4623. Next we randomly sampled 80% of $< t.c, t.rs >$ tuples (i.e. 119200 tickets) for end-to-end model training and 20%(i.e. 29800 tickets) for validation. That 80% was used to train the top level R/NR classifier, and the routine portion of it (i.e. 35776 tickets or 24% of all tickets) was used to train the second level R-TRS classifier. The training on each level was validated by 10-fold cross validation (i.e. rotation on 90%, 10% splits). After tuning parameters of each of the classifiers separately, we observed significant performance by both classifiers in isolation. Then we measured the overall performance of the sequentially combined classifiers by using the 20% validation set.

A. Performance evaluation with respect to SL

Given our goal of achieving trustworthy recommendations we opted to increase the reliability at the expense of reducing the number of tickets for which assistive recommendations were presented. Assume the ground truth labels, *actual-R* and *actual-NR*. The human experts are capable to handle (1) all actual-NR tickets, and (2) actual-R tickets that got misclassified as NR. On the other hand, it is unfavorable for trust if an actual-NR ticket is misclassified as R, and is further recommended with an R-TRS. Therefore, in this application domain ***the precision of the top-level classifier and the accuracy of the second-level classifier are more important for the overall performance than the coverage***. In particular from a SL achievement perspective, it is notable that the recall of the top level classifier is not as important as its precision since false negatives (misclassified routine tickets) will nevertheless get routed through the CEN and addressed directly by human experts (i.e. without recommendations). The performance of

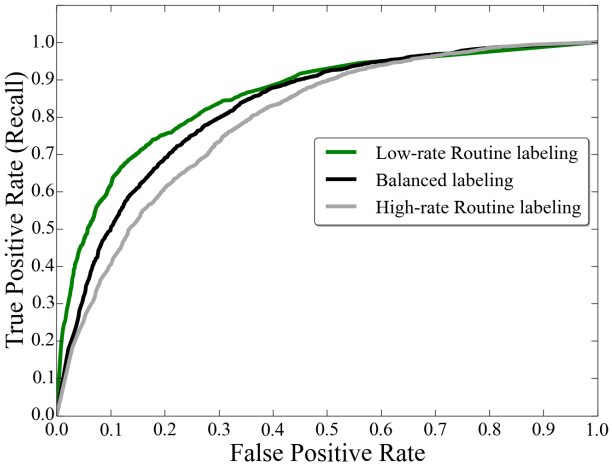

Fig. 7. ROC curves for three variations of R/NR classifier

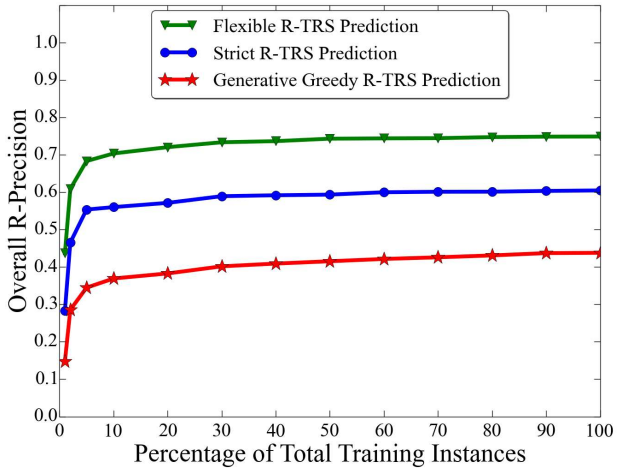

Fig. 8. Overall R-Precision of flexible, strict, and greedy models.

our two-level recommendation framework is evaluated by measuring the proportion of tickets that their $t.rs$ got correctly recommended, to all tickets that got recommended as R. Formally:

$$Overall\ R-Precision = \frac{\#(t.rs\ correctly\ classified)}{\#(tickets\ predicted\ as\ R)} \tag{13}$$

Evaluating the R/NR labeling strategy: As mentioned in Section IV our R/NR labeling strategy was chosen from [9]. This is an unsupervised method that finds a non-trivial optimal cut that bifurcates the ticket set such that the distance between the two content distributions is maximized. The content distribution with the higher average log likelihood is then labeled as R and the other distribution is labeled as NR. A path is labeled as R if and only if majority of the tickets that it has resolved in the history fall within the R content distribution. Otherwise that path is labeled as NR.

The above strategy in our experiments labeled most of the TRSs as NR (77.4%), which favorably conforms to our machine learning goal proposed in Subsection II-D. Thus we called this labeling strategy as *'Low-rate Routine labeling' (LRL)*. To evaluate the optimal bifurcation strategy, we chose two alternative labeling strategies as baselines: (1) *'Balanced labeling' (BL)* where the most frequent paths are labeled as R in such a way that these paths together resolve 50% of all tickets, and the rest are labeled as NR. (2) *'High-rate Routine labeling' (HRL)* where the most frequent paths are labeled as R such that these paths together resolve 75% of all tickets, and the rest are labeled as NR.

Per each labeling strategy we constructed a top level R/NR classifier (using the TWCNB learning algorithm). Our goal here was to find the classifier that consistently outperforms the other two. Figure 7 presents the *Receiver Operating Characteristic (ROC)* curves corresponding to different labeling strategies. The concept of ROC was first introduced in [5] and it generally aims to show performance of a binary classifier as its decision threshold varies. In the context of this study the *true positive rate (TPR)* (i.e. recall) is the fraction of actual-R tickets that also got classified as R. The *false positive rate (FPR)* is the fraction of actual-NR tickets that unfavorably got classified as R. The perfect case is to have TPR at 1 and the FPR at 0. The ROC curves in Figure 7 are drawn as a result of varying classifiers' decision thresholds from 0 to 1. Performance of these classifiers are evaluated by the *area under the ROC curve (AUROC)*. Observably our adapted optimal cut strategy (i.e. LRL) outperforms both of the baselines. To be precise, AUROC for LRL, BL, and HRL are respectively 0.86, 0.83, and 79. Thus we continued to use the optimal cut strategy in construction of our top level classifier.

R/NR Classification – tuning the Precision/Recall trade-off: In this application domain, increasing the precision of the R class can significantly improve the SL performance overall. Therefore the goal here is to find an effective decision threshold which favors precision a bit more over recall. Based on equation 12 the decision threshold is used to classify a ticket as R based on: $\hat{P}(C = \text{`}R\text{'} \mid \tau) > \theta$

Here $\hat{P}$ is the inferred probability for a test ticket τ to be classified as R. θ is the decision threshold acting as the minimum acceptable probability value to classify a ticket as R. We used the LRL ROC curve from Figure 7 to pinpoint an effective decision threshold. After examination of the coverage of candidate decision thresholds we arrived at a point on the ROC curve which yeilds a resonable high precision (through a low FPR) with an acceptable recall and coverage. More specifics of this sweet spot are as follows: recall=0.553, FPR=0.073, precision=0.802, and Routine Coverage=0.202. The decision threshold corresponding to this point found to be $\theta = 0.650$. Thus clearly resulting in a more conservative routine calls by the top level classifier.

VI. Experimental Validation

Finally by applying the same enterprise data, we compared two variations of the proposed framework, *Strict model* and *Flexible model*, against an existing sequence recommendation model called *Generative greedy model* taken from [7].

Strict and Flexible models: For the validation of path recommendations we define two different ways of claiming successful classification on a test ticket: (1) *strict* TRS matching: a ticket is called correctly classified if its predicted R-TRS matches exactly with its actual $t.rs$. (2) *flexible* TRS matching: a ticket is called correctly classified if its predicted R-TRS is within the *congruence set* of its actual $t.rs$.

The congruence set of a certain path like P consists other paths that are equally eligible to resolve same tickets that historically got resolved by P. Such replications exist by design among some of the routine paths in order to (1) balance the regular workload over more nodes in the network to improve the network throughput, and (2) make the network more tolerant against unavailability of certain nodes. Here for each of the routine paths in our domain, subject matter experts established a handcrafted congruence set representing corresponding qualified alternative paths, which we used for the flexible matching.

Baseline model - Generative Greedy: The Generative Greedy is considered a robust transfer prediction model [7]. This model is designed to make one-step transfer predictions and select the most probable resolver next. In our experiment Generative Greedy has shown effectiveness in predicting the final group in the sequence for actual-NR tickets with long TRSs. To be able to compare the results, we re-defined the 'Overall R-Precision' for Generative Greedy: for any test ticket predicted as R, we let the Generative Greedy also predict n transfers at once where n is the length of the actual TRS. If the Generative Greedy matches the actual TRS, we consider it as correctly classified. The ratio of correctly classified TRSs divided by total number of predictions is considered as Overall R-Precision for this method.

Figure 8 shows the overall R-precision of the developed sequence models as the size of the training set grows. All three models converge to a stable precision before reaching to 60% of the size the training set. Many of the misclassified TRSs in the strict model are found to be within the congruence set of the actual TRS. Therefore as can be seen we achieved 17% improvement over strict model by allowing misclassification within congruence sets. Also the flexible model outperforms the baseline by 34%. (flexible:77%, strict: 60%, generative: 43%).

SL and Time to Resolve (TTR) for classified tickets: For the fraction of test tickets that R-TRSs are recommended, we developed a simple expectation model to further estimate their TTR and SL compliance (SL Estimation in Figure 6). Let $T_{P,RP}$ be a subset of the training set that includes all tickets with priority P that were resolved by a particular routine path RP. For a test ticket τ with priority P (i.e. $\tau.p = P$) and recommended path RP the *Expected Time to Resolve (ETTR)* is estimated as the mean TTR of all tickets in $T_{P,RP}$. Formally:

$$\tau.ettr = \frac{1}{\mid T_{P,PR} \mid} \sum_{t \in T_{P,PR}} t.ttr \qquad (14)$$

For ETTR evaluation, a held-out test set was used from

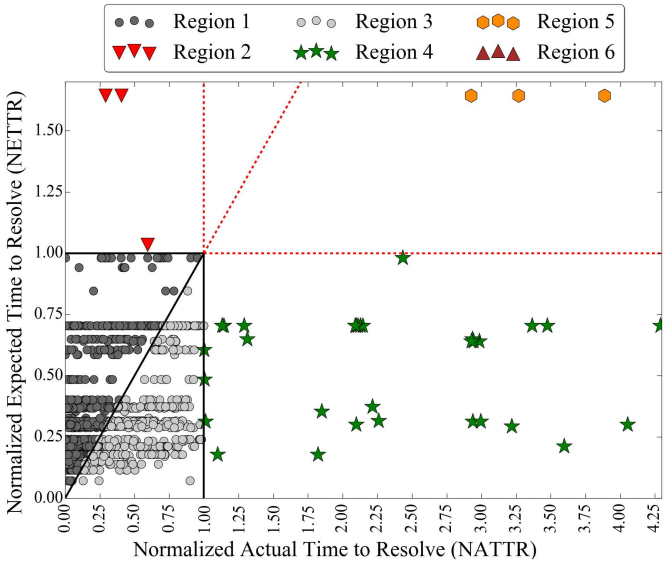

Fig. 9. ETTR vs ATTR for tickets with a recommended R-TRS (refer to Table II for description of the regions)

TABLE II
EVALUATION OF EXPECTED TIME TO RESOLVE

Region#	ETTR	ATTR	ETTR>ATTR?	% of test tickets
1	Met	Met	TRUE	65.9% [1078]
2	Breached	Met	TRUE	0.2% [3]
3	Met	Met	FALSE	31.8% [521]
4	Met	Breached	FALSE	1.9% [31]
5	Breached	Breached	FALSE	0.2% [3]
6	Breached	Breached	TRUE	0.0% [0]

which 1636 routine tickets eventually received recommended R-TRSs from the two-level classifier. Figure 9 illustrates a scatter plot of these tickets which compares ETTR of tickets against their actual time to resolve (ATTR). In order to present different ticket priorities within a unified scale we normalized all ETTR and ATTR values by their service time, thus generating NETTR and NATTR values. As a result of normalization any NETTR or NATTR value greater than 1 signals a SL breach. Also the diagonal line represents the *identity relation* between ETTR and ATTR. Tickets above the diagonal line imply ETTR $>$ ATTR, and ticket below it imply ETTR $<$ ATTR. Therefore, there will be six regions on the scatter plot subject to further analysis.

In Table II the common SL and TTR properties of tickets in each region is presented. Also the last column reports the probability (and frequency) distribution of tickets over different regions.

The key insights reported in Figure 9 and Table II are as follows: (1) Almost all routine tickets that actually met their SL were also estimated to meet their SL based on their recommended R-TRS with an exception of tickets in region 2 (SL Recall = 0.998). (2) Most of the routine tickets that were estimated to meet their SL were also found to actually meet their SL with an exception of tickets in region 4

(SL Precision = 0.980). This confirms the fact that estimated SL compliance is a true indicator of the actual SL compliance. (3) Most of the tickets that actually breached their SL were estimated to *meet* their SL with an exception of tickets in region 5 (SL false positive rate = 0.911). Despite the common intuition that FPR is an error measure and has to be minimized, *here a high FPR is a point of* ***strength*** *for our estimation model*. The reason for high FPR is that in the absence of recommendations, human decision anomalies cause a fraction of routine tickets to take NR-TRSs. Our data has shown that 87% of all routine breached tickets were actually routed through NR-TRSs. However, nearly all of these tickets could have met their SL had they taken their correct R-TRSs through recommendations. That is why ETTR is significantly lower than ATTR for most of the tickets in regions 4 and 5. *This clearly portrays the contribution of our statistical learning approach in reducing the negative impact of human decision anomalies.* (4) Based on ETTRs calculated above, recommendations significantly reduced the MTTR of the routine tickets by 34%. Viewing the system as a whole, the two-level classification method reduces the MTTR of *all* tickets by an average of 7%.

VII. Conclusions

We have introduced a new framework that improves collective performance by the Collective Expert Network in applications like the service desk within the enterprise. If a 'routine' path on the CEN has historically achieved the SL by resolving the tickets within service time then it has met the time and customer satisfaction goals. Using this and other principles exhibited by the CEN in its digital trace, we developed the two-level framework suited for enterprise deployment. The path recommendation results are promising as they indicate 77% R-precision for the end-to-end model. The recommended R-TRSs are more than 96% likely to meet the SL goals. The overall two-level classification model has also shown 10% reduction in average SL violation rate mainly by preventing frequent content from getting misrouted by the CEN. More research needs to be conducted in improving the non-routine cases by engaging the human-in-the-loop in production environments.

Acknowledgment

This research is supported by the National Science Foundation under Grant No. 0753710, the CERCS I/UCRC and industry sponsors.

References

[1] Federico Cabitza and Carla Simone. Computational coordination mechanisms: A tale of a struggle for flexibility. *Computer Supported Cooperative Work (CSCW)*, 22(4-6):475–529, 2013.

[2] Yi Chen, Shu Tao, Xifeng Yan, Nikos Anerousis, and Qihong Shao. Assessing expertise awareness in resolution networks. In *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pages 128–135. IEEE, 2010.

[3] Reuven Cohen and Shlomo Havlin. *Complex networks: structure, robustness and function*. Cambridge University Press, 2010.

[4] R Jordan Crouser, Alvitta Ottley, and Remco Chang. Balancing human and machine contributions in human computation systems. In *Handbook of Human Computation*, pages 615–623. Springer, 2013.

[5] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[6] Victor Kaptelinin and Bonnie Nardi. Affordances in hci: toward a mediated action perspective. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 967–976. ACM, 2012.

[7] Gengxin Miao, Louise E Moser, Xifeng Yan, Shu Tao, Yi Chen, and Nikos Anerousis. Generative models for ticket resolution in expert networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 733–742. ACM, 2010.

[8] Gengxin Miao, Louise E Moser, Xifeng Yan, Shu Tao, Yi Chen, and Nikos Anerousis. Reliable ticket routing in expert networks. In *Reliable Knowledge Discovery*, pages 127–147. Springer, 2012.

[9] Kayhan Moharreri, Jayashree Ramanathan, and Rajiv Ramnath. Recommendations for achieving service levels within large-scale resolution service networks. In *Proceedings of the 8th Annual ACM India Conference*, pages 37–46. ACM, 2015.

[10] Mark EJ Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[11] Daniel Oppenheim, Saeed Bagheri, Krishna Ratakonda, and Yi-Min Che. Agility of enterprise operations across distributed organizations: A model of cross enterprise collaboration. In *SRII Global Conference (SRII), 2011 Annual*, pages 154–162. IEEE, 2011.

[12] Brady Orand and Julie Villareal. Foundations of it service management with itil 2011: Itil foundation course in a book. *c. August*, 2011.

[13] Jason D Rennie, Lawrence Shih, Jaime Teevan, David R Karger, et al. Tackling the poor assumptions of naive bayes text classifiers. In *ICML*, volume 3, pages 616–623. Washington DC), 2003.

[14] Qihong Shao, Yi Chen, Shu Tao, Xifeng Yan, and Nikos Anerousis. Efficient ticket routing by resolution sequence mining. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 605–613. ACM, 2008.

[15] Shashank Srikant and Varun Aggarwal. A system to grade computer programming skills using machine learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1887–1896. ACM, 2014.

[16] Michael PH Stumpf and Mason A Porter. Critical truths about power laws. *Science*, 335(6069):665–666, 2012.

[17] Huan Sun, Mudhakar Srivatsa, Shulong Tan, Yang Li, Lance M Kaplan, Shu Tao, and Xifeng Yan. Analyzing expert behaviors in collaborative networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1486–1495. ACM, 2014.

[18] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.

[19] Wil Van Der Aalst. *Process mining: discovery, conformance and enhancement of business processes*. Springer Science & Business Media, 2011.

[20] Wil MP Van der Aalst. Using process mining to bridge the gap between bi and bpm. *IEEE Computer*, 44(12):77–80, 2011.

[21] Lean Yu, Shouyang Wang, and Kin Keung Lai. An intelligent agent-based fuzzy group decision making model for financial multicriteria decision support: The case of credit scoring. *European Journal of Operational Research*, 195(3):942–959, 2009.