What You Say Versus When You Say It: Efficiently Predicting Service Completions with LLMs and Stochastic Processes

Antonio Castellanos
The Hebrew University Business School, Hebrew University of Jerusalem, antonio.catellanos@mail.huji.ac.il

Andrew Daw
Marshall School of Business, University of Southern California, dawandre@usc.edu

Galit B. Yom-Tov
Faculty of Data and Decision Sciences, Technion—Israel Institute of Technology, gality@technion.ac.il

Accurately determining whether an ongoing service conversation will continue or has effectively ended is central to operational control in chat-based service systems. Premature conversation closures can induce costly customer re-contacts, whereas overly cautious closures waste agent capacity. Using large-scale conversational data from a food delivery service organization, we study two related prediction tasks: short-term conversation continuation and end-of-service completion. We compare three modeling paradigms—stochastic self-exciting point processes, metadata-based neural networks, and large language models (LLMs) operating on full conversation text. Our results reveal a sharp distinction between the tasks. For short-term continuation prediction, lightweight Hawkes process models capture temporal dynamics most effectively and achieve competitive accuracy at minimal computational cost. In contrast, end-of-service prediction benefits substantially from semantic information: a fine-tuned Transformer-based text classifier outperforms all temporal models, while zero-shot prompted LLMs perform poorly despite their scale. A cost–accuracy analysis shows that large models are only justified for specific, high-stakes decisions. Motivated by these findings, we propose a hybrid approach that combines real-time Hawkes-based monitoring with selective LLM queries. This strategy exploits the complementary strengths of temporal and semantic models and achieves a best-of-both-worlds cost–accuracy tradeoff for operational decision-making in service conversations.

Key words: Service Operations, Hawkes Processes, Neural Networks, LLMs, Cost-Accuracy Tradeoff

1. Introduction

Text-based chat platforms have become a primary channel for providing service delivery and customer support. These chats may be handled by a human representative, a chatbot system, or a hybrid of the two. Relative to traditional call centers, a major advantage of chat-based services is the ability to reduce waiting times by allowing resources (either humans or bots) to handle multiple customers concurrently and improve effective capacity (Luo and Zhang 2013, Tezcan and Zhang 2014). However, concurrency also introduces new challenges. In particular, because these parallel conversations are conducted through sporadic texts, it is difficult to monitor progress and determine when a conversation is truly over and whether the service-providing resources can be released (whether this means freeing chatbot-server memory or opening a human agent's concurrency slot). This uncertainty directly affects resource utilization. Relying on customers to explicitly signal that the chat is over is not effective, as many customers leave the chat without providing any closing signal e.g., indicate resolution or closing the application. This phenomenon, known as "silent abandonment," was formally measured in Castellanos et al. (2025), which showed that it occurs at substantial rates (often exceeding 20% of the customers).

To manage this uncertainty, companies typically apply a simple rule of thumb for chat closure. For example, setting a fixed inactivity timeouts (e.g., close a chat after X minutes of no customer activity). These policies have several risks. If the timeout is too short, chats may be closed prematurely, leading to long waits when the customers attempt to reopen them and causing dissatisfaction (especially if the customer needs to re-explain their situation to a new agent). Premature closure also wastes resources, since previously provided information must be reprocessed (e.g., by the next agent). On the other hand, letting chats remain open for too long reduces effective capacity and increase system-wide delays. Using real conversational data from a food delivery service organization, we estimate that approximately 19% of conversations in our dataset were prematurely closed (see Appendix EC.2).

In this work, we develop methods to predict whether and when a service problem was solved in real time. Such prediction may enable automatic closure of chats in a safe and efficient manner (Castellanos et al. 2024). These type of prediction models must operate at scale, be computationally efficient, and avoid overloading the system with excessive requirements: keeping resource consumption lean is critically important. To this end, we compare lightweight prediction models that rely only on metadata capturing conversation dynamics with richer models that leverage on the full text of the conversation. Specifically, the focus of this paper will be comparing leading ML/AI models that leverage the conversation's text (i.e., *what you say*) in comparison to stochastic process models that are temporally driven (i.e., *when you say it.*)

Empirically, prior research has shown that a bivariate marked Hawkes cluster model effectively captures burstiness, side-to-side reciprocity, and speaker-specific patterns in large-scale chat data from a human-based contact center, outperforming memoryless baselines and passing both residual and Monte Carlo goodness-of-fit diagnostics (Daw et al. 2025). Hawkes processes model *when* the messages arrive and yield interpretable intensities with minimal dependence on raw text (Hawkes 1971, Laub et al. 2021). This family of models represents the state of each conversation through a single quantity—the process intensity—and prediction models that rely on this intensity are therefore lightweight. The Hawkes intensity rate function may depend solely on simple metadata or incorporate more elaborate message-level and/or system-level statistics, such as message word count, sentiment score, or concurrent agent load.

On the other end of spectrum are full-text models. Conversation text encodes intent, sentiment, and resolution cues, and large language models (LLMs) can leverage this information to achieve high predictive power. Recent practitioner-facing studies document the rapid adoption of LLM-based tools in the customer service industry, where LLMs can be integrated into the service itself, either by helping agents to provide faster solutions or by replacing human agents in more and more complex tasks (Buesing et al. 2024, Ferraro et al. 2024). Yet, this does not comes freely. There are operational and governance costs: LLM inference is compute- and memory-intensive, and serving many concurrent conversations requires careful throughput/latency engineering and scheduling (Kwon et al. 2023, Mitzenmacher and Shahout 2025). Moreover, using full conversation text raises additional privacy and compliance risks, including potential memorization and disclosure of sensitive information, which has motivated both technical safeguards and regulatory guidance (Carlini et al. 2021, (EDPB) 2025).

In this work, we contrast three modeling paradigms — full-text LLMs, metadata-based neural networks, and stochastic self-exciting models — across two prediction tasks within customer-agent service interactions. The first task concerns short-term conversation surveillance: predicting whether a conversation will be active within the next Δ units of time. The second task targets closure decisions by predicting whether service has been completed. The context for the real-world data we use in this study is given in Section 3. Section 4 describes in detail the specific model families and variants considered. To ensure a fair comparison, we evaluate all methods using a common experimental scaffold, including identical checkpoint definitions, labels, and computational constraints, allowing us to isolate where each approach excels.

Contrasting the three prediction approaches, our results in Section 5 highlight clear complementarities across the three model types. LLMs, which leverage the full conversation text, excel at content understanding and therefore perform best on the end-of-service prediction task. In contrast, lightweight Hawkes-based models capture short-term temporal dynamics well and are better suited for real-time monitoring and control. Section 5.3 quantifies the resulting cost–accuracy tradeoffs for both tasks.

Motivated by these empirical findings, in Section 6 we propose a stylized hybrid approach that combines the strengths of both paradigms in optimal manner. The proposed model tracks Hawkes intensities in real time and selectively queries an LLM only in situations where semantic information is most valuable. We show that this hybrid strategy achieves a superior cost–accuracy tradeoff when deciding whether and when to close service conversations.

2. Literature Review

Our work connects several streams of literature: (i) modeling customer-agent interactions in service systems, (ii) temporal point process and intensity-based models for short-term activity prediction, and (iii) the use of machine learning and LLMs in conversation analysis and operational decision-making. We review each stream in turn and clarify our contribution relative to existing work.

2.1. Customer-Agent Interactions in Service Systems

A substantial literature studies the dynamics of customer-agent interactions in service systems, with particular emphasis on workload and performance outcomes (see surveys Gans et al. 2003, Aksin et al. 2007, and references therein). A growing body of work further demonstrates the operational value of incorporating predictive information into service-system decision making. Even imperfect foresight can meaningfully improve admission, routing, and completion policies when mapped to simple thresholds (Spencer et al. 2014, Bertsimas and Kallus 2020, Castellanos et al. 2024, Daw et al. 2025). The predictive signals considered in this literature span a wide range of information sources. These include metadata related to arrival processes and system state (Xu and Chan 2016), customer characteristics such as expected value (Yom-Tov et al. 2020), and semantic information extracted from interactions, such as customer sentiment (Altman et al. 2021). Collectively, these studies highlight how diverse forms of information can be operationalized to improve system performance.

In this context, determining when a service interaction has effectively ended is a particularly important operational challenge. Closure decisions directly affect routing, workload management, and agent utilization. Recent work suggests that real-time information about service termination can be leveraged to enhance system performance and reduce inefficiencies (Daw and Yom-Tov 2024). Premature closures of service interaction may trigger customer recontacts and additional workload (Armony and Maglaras 2004, Goes et al. 2018), while delayed closures waste scarce agent capacity (Castellanos et al. 2024).

Our paper contributes to this line of work by studying real-time prediction of both short-term conversation continuation and service completion at the conversation level, using detailed interaction data from a large-scale service system.

2.2. Temporal Point Processes and Conversation Dynamic Modeling

Recent work has increasingly emphasized the behavioral nature of service interactions, in which customer behavior and agent actions mutually influence one another and jointly determine outcomes such as service duration, quality, and satisfaction (Ilk and Shang 2022, Ashtar et al. 2021). In this context, Daw et al. (2025) demonstrate the effectiveness of marked and bivariate Hawkes models for capturing such behavioral dynamics when modeling conversations on chat and messaging platforms.

More broadly, temporal point process models, and Hawkes processes (Hawkes 1971) in particular, have been widely used to model bursty and self-exciting event sequences across a range of domains, including financial markets (Embrechts et al. 2011), social media and information diffusion (Rizoiu et al. 2017), epidemiology (Rizoiu et al. 2018, Chiang et al. 2022), and queueing systems (Daw and Pender 2018, Koops et al. 2018). In human communication settings, Hawkes-type models naturally capture event clustering, mutual excitation between participants, and time-varying activity intensity (Blundell et al. 2012, Halpin and De Boeck 2013, Daw and Pender 2022).

These properties make Hawkes models particularly well suited for activity prediction tasks, as they explicitly encode temporal dependence and excitation effects. Building on this literature, our work evaluates several Hawkes-based specifications and benchmarks their performance against neural and text-based models, highlighting their strong cost-effectiveness for short-term conversation continuation prediction. We also offer a new way to use Hawkes models — to control the use of LLMs in operations. To the best of our knowledge, we are the first to offer such use.

2.3. Machine Learning Models and LLMs for Conversation Dynamic Analysis

Neural architectures such as multilayer perceptrons (MLPs) and long short-term memory (LSTM) networks have been widely used to model sequential conversational data, including applications such as next-sentence prediction (Chandwani et al. 2020), response-time estimation (Roddy and Harte 2020), and time-to-derailment prediction (Janiszewski et al. 2021). These models offer substantial flexibility in capturing nonlinear relationships and temporal dependencies. However, they typically rely on handcrafted features, and their operational effectiveness depends critically on the availability, stability, and real-time reliability of such features.

LLMs represent the most recent advances in machine learning for text analysis. Transformer-based models such as BERT and its variants (Devlin et al. 2019, Sanh et al. 2019), as well as causal LLMs including Llama (Grattafiori et al. 2024) and Phi (Abdin et al. 2024), have shown strong performance across a broad range of NLP tasks by capturing rich semantic and contextual structure. In the context of conversational dynamics, Pinto and Belpaeme (2024) study LLMs' ability to predict turn-switching in human-bot interactions, while Jiang et al. (2023) extend TurnGPT to condition end-of-turn prediction on both conversation history and anticipated responses.

Recent work further explores the potential of LLMs for time-series and sequential prediction tasks (Jin et al. 2024b), often requiring architectural adaptations or fine-tuning (Jin et al. 2024a, Gruver et al. 2023). Consistent with this emerging evidence, we find that fine-tuned LLMs substantially outperform zero-shot and few-shot prompt-based LLMs for operational service prediction tasks.

Our work also connects to a growing literature emphasizing the importance of cost-accuracy tradeoffs in deploying predictive models. Despite their impressive representational power, the suitability of LLMs for real-time operational control remains unclear due to latency, privacy, and cost considerations. While large models often deliver significant accuracy gains, they incur substantial computational and energy costs (Strubell et al. 2019). From an operations perspective, models must be evaluated not only by predictive power but also by feasibility under real-time constraints (Mahadevan and Mathioudakis 2024). Our paper contributes to this emerging literature by providing one of the first systematic, end-to-end evaluations of learning models for real-time service prediction, explicitly quantifying the cost–accuracy tradeoff across a wide spectrum of models, from lightweight Hawkes processes to heavyweight LLMs.

Finally, our work relates to recent efforts on hybrid approaches that combine multiple modeling paradigms. For example, Lee and Deng (2024) integrate DistilBERT with GRU networks for real-time end-of-turn prediction, while Shahout et al. (2025) reuse LLM embeddings as inputs to lightweight classifiers. We propose a different hybrid strategy: dynamically scheduling queries to a fine-tuned DistilBERT model using real-time signals from a lightweight Hawkes process, thereby exploiting the complementary strengths of temporal intensity modeling and semantic understanding.

3. Data and Context

We analyze 78,386 human-agent chat conversations from a U.S.-based company in the food delivery industry. The chats were handled by the company's service center, where customers seek assistance with issues such as late deliveries and billing problems.[1] The service platform is AI-augmented: agents receive automatically generated response suggestions from an AI system, which they may use when replying to customers. The conversations were conducted between October and December 2021.

We exclude approximately 8,000 chats in which customers abandoned the queue before being assigned to an agents. These conversations contain only customer messages and were closed by the customer. This final dataset therefore consists of 70,662 chats that include messages from both customers and agents. Of these chats, 27.3% were closed by the system, a phenomenon we refer to as *automatic closure*, which we identify by the presence of a canned (i.e., pre-populated) closing message at the end of the conversation. The median number of messages in a chat is 8, and the median chat duration is 5 minutes. Additional summary statistics are reported the Appendix in Table EC.1.

For each chat, the data include the text of each message, an indicator of the message sender, and the timestamp the message. Figure 1 presents a schematic timeline of events within a conversation. We define the following notations for conversation $i \in \{1, \dots, M\}$. Let $T_{i,k}$ denote the timestamps of the k-th messages in conversation i; the timestamps are ordered so that $T_{i,1} < T_{i,2} < \cdots < T_{i,n_i}$, where n_i is the total number of messages in conversation i and $T_{i,1} = 0$. Let $R_{i,k} \in \{\text{Ag}, \text{Cu}\}$ indicate

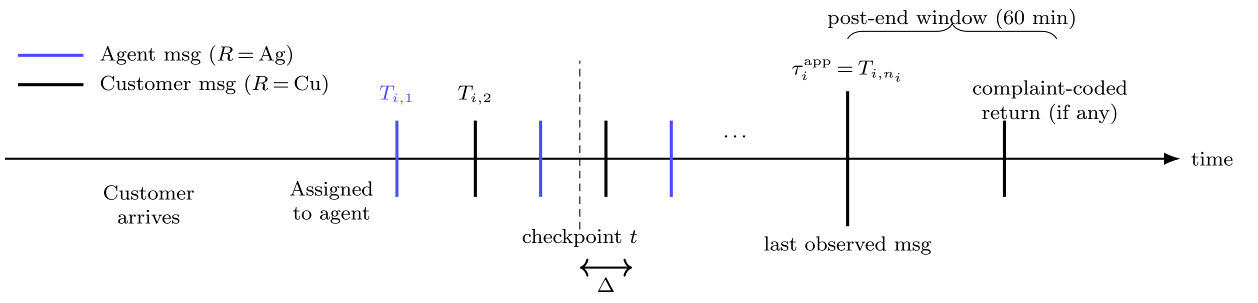

Agent msg ($R = \mathrm{Ag}$)
Customer msg ($R = \mathrm{Cu}$)
$T_{i,1}$
$T_{i,2}$
post-end window (60 min)
$\tau_i^{\mathrm{app}} = T_{i,n_i}$
complaint-coded return (if any)
...
time
Customer arrives
Assigned to agent
checkpoint t
Δ
last observed msg

Figure 1 Timeline of conversation i: message times $T_{i,k}$ (agent/customer). τ_i^{app} is the apparent end (last observed message).

the sender of message k, where 'Ag' indicates an agent and 'Cu' indicates a customer, and let $W_{i,k} \in \mathbb{N}$ denote the number of words in message k within conversation i. Finally, define $\tau_i^{\mathrm{app}} := T_{i,n_i}$ as the *apparent end* of conversation i, corresponding to the time of its last observed message or an explicit closure message.

Define, $N_i(t) := \sum_{k=1}^{n_i} \mathbb{1}\{T_{i,k} \le t\}$ as the counting process of messages up to time t, and $\mathcal{H}_{i,t} := \{(T_{i,k}, R_{i,k}, W_{i,k}) : T_{i,k} \le t\}$ as the observed history of conversation i up to time t. For brevity, when the conversation index is clear from the context, we omit it and write $N(t)$ and $\mathcal{H}_t$.

3.1. Context and Cohort Labeling

A natural quality indicator in a service system is the proportion of conversations that are successfully closed (i.e., where the customer's issue is resolved). However, the system's automatic-closure policy can interfere with successful resolution, as the system may time out a customer before the service is completed, simply because they respond slowly. In a retrospective study, we use positive-unlabeled (PU) learning (Elkan and Noto 2008, Bekker and Davis 2020) to estimate the proportion of chats that were prematurely closed due to these system policies in our data (see Appendix EC.2). The finding suggests that 19% of the chats where prematurely closed by the system. Operationally, however, what interests organizations is to be able to verify if a *specific* chat can be safely closed in real time. To support such decisions, we build models that classify chats into those that were successful closed versus those that were prematurely closed.

A natural indicator of premature closure is customer return. However, our data do not contain unique customer identifiers, so conversations cannot be directly linked to returning customers. Instead, we use text analysis to classify conversations to three groups: class $\mathcal{N}$ for *successful closures*, class $\mathcal{P}$ for *premature closures with verified returns*, and class $\mathcal{U}$ for *uncertain closures.*

A chat is labeled as a *successful closure* if the conversation ends with explicit resolution signals from the customer, such as expressions of gratitude (e.g., "thank you") or a negative response to a follow-up question such as "Do you need any further help today?". By contrast, a conversation is a *premature closure* if the customer returns with a new chat for the same issue soon after the first was closed. Conversations that do not fall into either of these two categories are labeled as *uncertain closures.*

To identify *premature closures with verified returns*, we used the following procedure to overcome the fact that our data does not contain customer ID's. First, we identify conversations that begin with customer messages referring to a prior premature termination of service *by the service platform*, for example by stating that the chat was "cut off" or "timed out." Conversations in which the customer explicitly attributes the return to their own action (e.g., "I closed the conversation by mistake") are excluded from this set. We denote by $C_i \in \{0,1\}$ an indicator equal to one if chat i is identified as a system-blamed returning chat; we refer to such chats as having a *complaint-coded return* indicator. Next, for each conversation with $C_i = 1$, we examine all chats that occurred within the preceding 60-minute interval. We manually match the current conversation to a prior one if both refer to the same customer issue (e.g., topic or location), correspond to the same stage of the service process (e.g., the payment stage), and the earlier conversation does not carry a *successful closure* label. (See Fig. 1 for the post-end window and indicator.) The manual matching step ensures that the identified returns correspond to conversations that were prematurely closed.

To summarize, the three classes are:

- $\mathcal{P}$ (premature closures with verified returns): conversations with a verified complaint-coded return within 60 minutes.
- $\mathcal{N}$ (successful closures): conversations where the conversation ends with explicit resolution signals from the customer and no complaint-coded return within 60 minutes.
- $\mathcal{U}$ (uncertain closure): Conversations that do not fall into either class $\mathcal{P}$ nor $\mathcal{N}$.

We identified 96 conversations in class $\mathcal{P}$ (due to human constraints of the significant effort required to identify these idiosyncratic cases, not for lack of their existence) and 5,145 conversations in class $\mathcal{N}$. Note that the majority of conversations could not be classified into either of these categories. This results in 65,421 conversations in class $\mathcal{U}$, representing 86% of our sample. Hence, we have used PU learning was needed to estimate the proportion of premature closures in our data.

4. Methods

Next, we develop and evaluate methods for predicting, in real time, whether a conversation has ended and can safely be closed. Specifically, we aim to predict the probability that a conversation will remain active over a predefined future interval.

Let $\Delta > 0$ denote a short prediction horizon (e.g., 30 seconds). We define *Chat Continuation* as the probability that at least one message will be sent in the next Δ units of time. Formally, we predict

$$p_i(t;\Delta) := \Pr\left(N_i(t+\Delta) - N_i(t) \geq 1 \mid \mathcal{H}_{i,t}\right). \tag{1}$$

Note that Δ may also be infinite. In this case, $p_i(t;\infty)$ denotes the probability that the conversation has not ended (regardless of when the next message will occur), and $1-p_i(t;\infty)$ is the probability that the conversation ended before time t.

This prediction problem is evaluated both at discrete checkpoints $t \in \mathcal{T}_i$ during the conversation and after an apparent end, which we denote by $t = \tau_i^{\text{app}}$.

4.1. Model Families

We consider three model families, which differ in the type of information they exploit. The most parsimonious family consists of Hawkes models that rely solely on metadata, specifically timestamp information. The second family comprises neural network models that additionally incorporate textual and system features. The final family consists of large language models (LLMs), which operate on the full conversation text. All these models are used to estimate the continuation probability defined in Eq. (1) at a given checkpoint time t for two values of Δ.

4.1.1. Self-Exciting Stochastic Models. Following the modeling approach and empirical application in Daw et al. (2025), the first family of models we test is based on an endogenously driven cluster process, derived from the original self-exciting point process introduced by Hawkes (1971). A Hawkes process is a stochastically time-varying point process whose intensity is defined as an integral over the past history of the process. In general, a Hawkes process may include both an exogenous baseline component (equivalent to a homogeneous Poisson process) and an endogenous self-exciting component. In our setting, as in Daw et al. (2025), we view each conversation as a single cluster with only a self-exciting component. Each process cluster starts when a customer opens a service request.

Let $\lambda_i(t \mid \mathcal{H}_{i,t})$ denote the intensity of messages in conversation i at time t, given its history up to time t, $\mathcal{H}_{i,t}$. The Hawkes process intensity is governed by a kernel function $g(\cdot)$, such that

$$\lambda_i(t \mid \mathcal{H}_{i,t}) = \int_0^t g(t-u)\, dN_i(u).$$

For a point process with conditional intensity $\lambda_i(u \mid \mathcal{H}_{i,t})$ for $u > t$, the probability of observing at least one event in the interval $(t, t+\Delta]$ is

$$\hat{p}_i(t;\Delta) = 1 - \exp\left(-\int_t^{t+\Delta} \lambda_i(u \mid \mathcal{H}_{i,t})\, du\right). \tag{2}$$

Hence, the prediction relies solely on timestamp information. To fit such a model, one must specify a kernel function and estimate its parameters via maximum likelihood. Generally, the quality of the prediction depends on how well the Hawkes model fits the event dynamics in the data.

We test three variants of Hawkes process models for conversations, which differ in their kernel structure.[2]

- **UHP (univariate exponential).** The first variant is a standard univariate Hawkes cluster model with an exponential kernel and independent and identically distributed instantaneous impact functions on the intensity rate.

Specifically, for $t > 0$, the intensity rate function is

$$\lambda_i(t \mid \mathcal{H}_{i,t}) = \sum_{k:\, T_{i,k} < t} \alpha e^{-\beta(t - T_{i,k})},$$

where α denotes the jump in intensity induced by each event, and β is the exponential decay rate.

- **IUHP-Exp (initialized univariate with exponential kernel).** The second variant uses the same exponential kernel but the conversation-start excitation is scaled separately by a constant λ_0, capturing an initial "kick-off" effect of the first automated agent greeting message:

$$\lambda_i(t \mid \mathcal{H}_{i,t}) = \lambda_0 e^{-\beta t} + \sum_{k:\, T_{i,k} < t} \alpha e^{-\beta(t - T_{i,k})}. \tag{3}$$

- **IUHP-Gamma (initialized univariate with Gamma kernel).** The third variant is again an initialized univariate Hawkes model, but with a Gamma kernel rather than an exponential kernel. A Gamma kernel allows the excitation response to peak after a delay and provides additional shape flexibility. Accordingly, the exponential term above is replaced by a Gamma probability intensity function (pdf) kernel, denoted by $g_{\rho,\theta}(\cdot)$:

$$\lambda_i(t \mid \mathcal{H}_{i,t}) = \lambda_0\, g_{\rho,\theta}(t) + \sum_{k:\, T_{i,k} < t} \alpha\, g_{\rho,\theta}(t - T_{i,k}),$$

where, as before, α denotes the jump in intensity induced by a single event and λ_0 is the scale parameter differentiating the start-event excitation from a regular excitation (modeling the kick-off effect), and $g_{\rho,\theta}(u) = \theta^\rho / \Gamma(\rho)\, u^{\rho-1} e^{-u\theta}$.

We also estimated and evaluated a Bivariate Hawkes Model (BHP) with exponential kernels (Appendix EC.3), but its omitted from the main results due to its low performance on this data set.

For all Hawkes models, parameters are estimated by maximum likelihood by maximizing the point-process log-likelihood aggregated across conversations. We treat each conversation as an independent Hawkes cluster, re-center time so the conversation start is at $t = 0$. We evaluate the point-process log-likelihood on the observed event times from conversation initiation until termination, and the overall objective sums these contributions across all conversations. Parameter estimation is optimized numerically, subject to standard positivity and stability constraints. For background on maximum-likelihood estimation for Hawkes processes see Laub et al. (2021, Ch. 5). The estimated parameters for our dataset can be found in Table EC.2 in the Appendix.

4.1.2. Neural Network Models. The second family of models consists of deep neural networks that learn a nonlinear mapping (Goodfellow et al. 2016) from structured conversation features to the chat-continuation probability in (1).

At each checkpoint time t, we define the supervised continuation label

$$Y_i(t;\Delta) \coloneqq \mathbb{1}\{N_i(t+\Delta) - N_i(t) \geq 1\}, \tag{4}$$

and train neural models to approximate $p_i(t;\Delta) = \Pr(Y_i(t;\Delta) = 1 \mid \mathcal{H}_{i,t})$. These models incorporate (i) message-level features, such as message timestamps, message order, word counts, and the corespondent's role (agent or customer), (ii) chat-level features, such as the Hawkes intensity level (see Section 4.1.1); and (iii) system-level features, such as the agent's concurrency level.

We examine two variants that differ in the depth of history they utilize.

• **Multilayer Perceptron (MLP) model with current-event features.** The first neural model is an MLP model, a feed-forward neural network composed of multiple layers of interconnected processing units (neurons). MLPs are among the most common learning architectures and are widely used to approximate complex, nonlinear functions (Goodfellow et al. 2016).

At checkpoint time t, the MLP input is a feature vector $\mathbf{Z}_i(t) \in \mathbb{R}^p$ summarizing the current state of conversation i. The vector $\mathbf{Z}_i(t)$ comprises p features, including message-level features of the most recent message (e.g., last-message word count, and corespondent role), as well as chat-level and system-level features at time t, such as the Hawkes intensity estimate and agent's concurrency level. In our implementation, we train several MLP variants that differ in the subset of features included in $\mathbf{Z}_i(t)$.

Let $f_\gamma : \mathbb{R}^p \to \mathbb{R}$ denote the MLP logit function parameterized by γ. For an input vector $\mathbf{z} \in \mathbb{R}^p$ (in our application, $\mathbf{z} = \mathbf{Z}_i(t)$), an MLP with L affine layers can be written as a composition of affine maps and pointwise nonlinearities:

$$f_\gamma(\mathbf{z}) = A_L\, h_{L-1} + a_L, \qquad h_\ell = \phi(A_\ell\, h_{\ell-1} + a_\ell)\,, \quad \ell = 1, \ldots, L-1, \qquad h_0 = \mathbf{z}.$$

Here, h_ℓ denotes the hidden-layer representation ℓ ($\ell = 0, \ldots, L-1$), A_ℓ and a_ℓ are the weight matrix and bias vector of layer ℓ, respectively, $\phi(\cdot)$ is the hidden-layer activation (ReLU (Nair and Hinton 2010) in our implementation), and L is the total number of affine layers (hidden layers plus the output layer). The output layer applies a logistic sigmoid to map the logit to a probability:

$$\hat{p}_i(t;\Delta) = \sigma(f_\gamma(\mathbf{Z}_i(t)))\,, \qquad \sigma(x) = (1 + e^{-x})^{-1}.$$

Before fitting, we standardize the input features using training-set statistics (z-scoring). We estimate γ by minimizing the binary cross-entropy loss on the continuation labels, using class weights to mitigate class imbalance. Optimization is performed using Adam (Kingma and Ba 2014), and dropout regularization is applied during training (Srivastava et al. 2014).

• **Long short-term memory (LSTM) model with full-history features.** The second neural model is an LSTM network, a recurrent architecture designed to learn patterns in sequential data while capturing long-range dependencies (Hochreiter and Schmidhuber 1997). LSTMs are especially designed to handle vanishing and exploding gradients when sequences are long. They maintain and update an internal memory state, allowing the network to selectively store, retrieve, or discard information over time. This makes them particularly effective for tasks where the order and context of past events influence future predictions. Therefore, they are well suited to our setting, where response times may include long silent intervals and messages impact subsequent ones (Daw et al. 2025).

Let $K_i(t)$ denote the index of the most recent message observed by checkpoint time t, i.e., $K_i(t) \coloneqq \max\{k : T_{i,k} \le t\}$. For each message $k \le K_i(t)$, we build a per-message feature vector $\boldsymbol{\psi}_{i,k}$ and form the sequence

$$\boldsymbol{\Psi}_i(t) = \left(\boldsymbol{\psi}_{i,1}, \ldots, \boldsymbol{\psi}_{i,K_i(t)}\right).$$

Each feature vector $\boldsymbol{\psi}_{i,k}$ is measured at the time of message k and includes message index, elapsed time since conversation start, word count, corespondent role, agent concurrency, and Hawkes intensity. As with the MLP, we train several LSTM variants that differ in the subset of features included in $\boldsymbol{\psi}_{i,k}$.

Sequences are right-padded to a fixed maximum length and masked. After processing $\boldsymbol{\Psi}_i(t)$, the LSTM outputs a hidden state $h_i(t)$, which is mapped to continuation probability via

$$\hat{p}_i(t;\Delta) = \sigma\left(u^\top h_i(t) + b\right),$$

where u and b are output-layer parameters and $\sigma(\cdot)$ denotes the logistic sigmoid.

Model parameters are estimated by minimizing binary cross entropy loss on the continuation labels, using the Adam optimizer (Kingma and Ba 2014) with dropout regularization (Srivastava et al. 2014). Early stopping based on validation loss is employed during training.

4.1.3. Large Language Models (LLMs). The final family of models we consider consists of LLMs that operate directly on the raw conversation text, rather than structured features. These models are based on the Transformer architecture (Vaswani et al. 2017) and can exploit semantic cues related to intent, sentiment, and service resolution that are unavailable to timing- or metadata-based models. Because service transcripts may contain sensitive information, and due to our privacy agreement with the company, we evaluate open-weight models locally (offline) and do not transmit text to external services.

Let $x_{i,k}$ denote the text of message k in conversation i, and let $X_i(t)$ denote the concatenated message text observed by checkpoint time t, ordered chronologically. Formally,

$$X_i(t) \coloneqq x_{i,1} \parallel x_{i,2} \parallel \cdots \parallel x_{i,K_i(t)},$$

where $\|$ denotes text concatenation with whitespace separators. Each LLM model maps $X_i(t)$ to a continuation probability $\hat{p}_i(t;\Delta)$ using softmax probability.

We examine the following two paradigms:

- **Fine-Tuned Transformer Encoder (DistilBERT).** DistilBERT is a compact Transformer encoder distilled from BERT (Devlin et al. 2019, Sanh et al. 2019). Unlike prompted LLMs, DistilBERT is trained specifically for the continuation prediction task. We fine-tune `distilbert-base-uncased` for binary classification using checkpoint-level examples. Given the tokenized transcript prefix $X_i(t)$, the model outputs two logits $(\ell_{i,0}(t), \ell_{i,1}(t))$ corresponding to the classes $\{$`no continuation`, `continuation`$\}$. We define the continuation probability as the softmax probability of the continuation class:

$$\hat{p}_i(t;\Delta) = \frac{\exp(\ell_{i,1}(t))}{\exp(\ell_{i,0}(t)) + \exp(\ell_{i,1}(t))}.$$

Model parameters are estimated by minimizing cross-entropy loss on the continuation labels using AdamW optimizer (an Adam variant that applies weight decay regularization) (Loshchilov and Hutter 2019). We employ a linear learning-rate schedule with warmup (controls the step size in gradient updates) and train for a fixed number of epochs.

- **Prompted-Based LLMs (Phi-3, and Llama-3).** We also evaluate instruction-tuned causal LLMs in a zero-shot or few-shot settings, using open-weight models from the Phi-3 and Llama-3 families (Abdin et al. 2024, Grattafiori et al. 2024). In contrast to DistilBERT, these models are not fine-tuned for our task and are instead queried via natural-language prompts.

Given the transcript prefix $X_i(t)$, we append a fixed yes/no question aligned with the continuation target (e.g., "Will there be another message within the next Δ minutes? Answer strictly with 'yes' or 'no'."). The model is then asked to generate the next token following the prompt, and we record each token's logit score.

Specifically, using the model tokenizer, which maps text strings to sequences of token IDs from a fixed vocabulary $\mathcal{V}$, we identify token IDs $v_{\texttt{yes}}$ and $v_{\texttt{no}}$ corresponding to a single-token realization of the answers `yes` and `no`, respectively. We search over common whitespace variants (e.g., `" yes"` and `"␣yes"`).

Let $s_{i,v}(t)$ denote the logit assigned by the LLM to some token v as the next token following the prompt. The next-token probabilities for `yes` and `no` are computed by softmax:

$$p_{i,\texttt{yes}}(t) = \frac{\exp\big(s_{i,v_{\texttt{yes}}}(t)\big)}{\sum_{v\in\mathcal{V}} \exp\big(s_{i,v}(t)\big)}, \qquad p_{i,\texttt{no}}(t) = \frac{\exp\big(s_{i,v_{\texttt{no}}}(t)\big)}{\sum_{v\in\mathcal{V}} \exp\big(s_{i,v}(t)\big)}.$$

If no single-token realization is found for either answer (`yes` and `no`), we assign both answers next-token probability 0.5.

We map these quantities to a scalar continuation probability by renormalizing over the two candidate answers:

$$\hat{p}_i(t;\Delta) = \frac{p_{i,\texttt{yes}}(t)}{p_{i,\texttt{yes}}(t) + p_{\texttt{i,no}}(t)}.$$

Overall, the three model families we consider span a spectrum of information usage and modeling complexity. Structured-feature models (Hawkes-based and neural models) rely on timing, count, and system-level signals that are available in real time and are straightforward to deploy, but they abstract away the semantic content of messages. In contrast, text-based Transformer models operate directly on raw conversation transcripts and can exploit rich linguistic cues related to intent, sentiment, and resolution, at the cost of substantially higher computational requirements and stricter privacy constraints. By evaluating these approaches side by side under a common prediction task and checkpointing framework, we can quantify, in the next section, the incremental predictive value of semantic information beyond event timing and operational features.

4.2. Evaluation Design and Performance Measures

We evaluate the models described in Section 4.1 at predefined checkpoint times and under two prediction horizons, yielding two complementary test cases. The *short-term continuation* test assesses each model's ability to predict whether a conversation will remain active over the next $\Delta = 60$ seconds, that is, to estimate $p_i(t;60)$. The *end-of-service* test evaluates the ability to predict whether a conversation has not yet ended, corresponding to $p_i(t;\infty)$.

We report performance as a function of time relative to the conversation's apparent end time τ_i^{app}. Specifically, we stratify checkpoints by their offset from the apparent end,

$$\mathrm{offset}_i(t) := t - \tau_i^{\mathrm{app}},$$

measured in minutes, and compute evaluation metrics within offset bins. Offsets may be negative (during active service) or positive (after the apparent end of service). The offset variable is used solely for evaluation and reporting. Because it depends on τ_i^{app}, it is not available to the models at inference time. Bins with fewer than 5 positive or negative examples are omitted when reporting accuracy measures for stability.

4.2.1. Short-Term Conversation Continuation.

For the short-term continuation test ($\Delta = 60$ seconds), we train all models using unclassified conversations from the $\mathcal{U}$ cohort (see Section 4.3). At each checkpoint time t, we use the trained models to estimate $p_i(t;60)$. Predictions are evaluated every 30 seconds up to the apparent end time τ_i^{app}; that is, only at negative offset times.

The short-term continuation task can be viewed as a dynamic binary classification problem, in which a conversation is classified as either `continuing` or `not continuing` over a forward-looking window of length $\Delta = 60$ seconds. Accordingly, model performance at each checkpoint time t is assessed using the area under the receiver operating characteristic curve (ROC-AUC) and the average precision (AP) derived from the precision-recall curve. We report both metrics. For completeness, we also report pooled ROC-AUC and pooled AP scores aggregated across all checkpoints.

4.2.2. End-of-Service Evaluation. For the end-of-service test ($\Delta = \infty$), we retrain the models to predict the probability that a conversation has not yet ended at checkpoint time t, which is equivalent to estimating $p_i(t;\infty)$. A high value of $p_i(t;\infty)$ indicates that additional messages are likely to occur and that closing the conversation at time t would be premature.

Model training for this task uses labeled data from the $\mathcal{P}$ and $\mathcal{N}$ cohorts (see Section 4.3). The $\mathcal{P}$ cohort consists of conversations that were verified to have ended prematurely, while the $\mathcal{N}$ cohort contains conversations that were verified to have successfully ended at their apparent end time τ_i^{app}.

We evaluate end-of-service predictions at both negative and positive offset times relative to τ_i^{app}. Because model behavior may change sharply around the apparent end time, we distinguish between two checkpoints in its immediate neighborhood. Specifically, we consider a checkpoint just before the apparent end, at $t = \tau_i^{\text{app}} - \varepsilon$, where the available history is

$$\mathcal{H}_{i,t_{0-}} = \{(T_{i,k}, R_{i,k}, W_{i,k}) : k \leq n_i - 1\},$$

and a checkpoint just after the apparent end, at $t = \tau_i^{\text{app}} + \varepsilon$, where the available history is

$$\mathcal{H}_{i,t_{0+}} = \{(T_{i,k}, R_{i,k}, W_{i,k}) : k \leq n_i\}.$$

Here, $\varepsilon > 0$ denotes a vanishingly small numerical offset that separates the regimes immediately before and after the final message.

In the results presented in Section 5, we report these two checkpoints separately as 0^- and 0^+ to highlight the model's behavior around the apparent end of service.

Because the labels of *all* conversations at negative offset times are deterministically equal to one (there is at least one additional message observed in the transcript), ranking-based metrics using ROC-AUC and AP measurements are not informative in this regime. Instead, we summarize time-varying performance at each offset using (i) mean predicted probability and (ii) the Brier score. Both qualities are reported separately for the $\mathcal{P}$ and $\mathcal{N}$ cohorts, reflecting the probability of correctly classifying conversations in each group.

4.3. Train and Test Cohorts

Cohorts are constructed based on the labeling procedure described in Section 3.1. As noted there, we identified 96 conversations in cohort $\mathcal{P}$, corresponding to premature closures with verified returns; 5,145 conversations in cohort $\mathcal{N}$, corresponding to successful closures; and 65,421 conversations in cohort $\mathcal{U}$, for which closure status is uncertain. From the unlabeled cohort $\mathcal{U}$, we construct disjoint training and test samples. Specifically, $\mathcal{U}_{\text{train}}$ consists of a random subsample of 16,000 conversations, and $\mathcal{U}_{\text{test}}$ consists of a random subsample of 5,000 conversations. Because cohorts $\mathcal{P}$ and $\mathcal{N}$ are substantially smaller, we employ resampling strategies when these cohorts are used for model training.

For the short-term continuation task, all models are trained on $\mathcal{U}_{\text{train}}$, and performance is reported separately on the $\mathcal{P}$, $\mathcal{N}$, and $\mathcal{U}_{\text{test}}$ cohorts. Because each conversation contributes multiple checkpoints, we quantify statistical uncertainty using a *conversation-level* bootstrap. Specifically, we resample test conversations with replacement, recompute the metric of interest on each resampled dataset, and report pointwise 2.5%–97.5% percentile intervals). For time-varying performance curves presented in Section 5, these intervals are shown as shaded confidence bands.

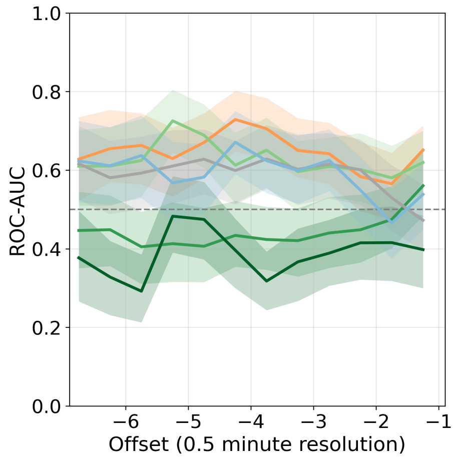
For the end-of-service prediction task, training requires labeled data from the much smaller $\mathcal{P}$ and $\mathcal{N}$ cohorts. Accordingly, Hawkes models are trained on $\mathcal{U}_{\text{train}}$, while neural network models and LLM-based models are trained using a five-fold cross validation procedure. We first matched the sample size between the cohorts, taking only 96 random conversations from the $\mathcal{N}$ cohort. Then, in each fold repetition, the training sets $\mathcal{P}_{\text{train}}$ and $\mathcal{N}_{\text{train}}$ each consist of 77 conversations, respectively. Performance is evaluated on the hold-out fold test sets, $\mathcal{P}_{\text{test}}$ and $\mathcal{N}_{\text{test}}$, each containing 19 conversations in each iteration. Again, we perform the cross-validation on conversation-level data, ensuring that all checkpoints from a given conversation appear exclusively in either the training or the test set within a fold, but never in both. Once again bootstrap is used within each test fold and averaged to gain confidence intervals.

5. Empirical Results

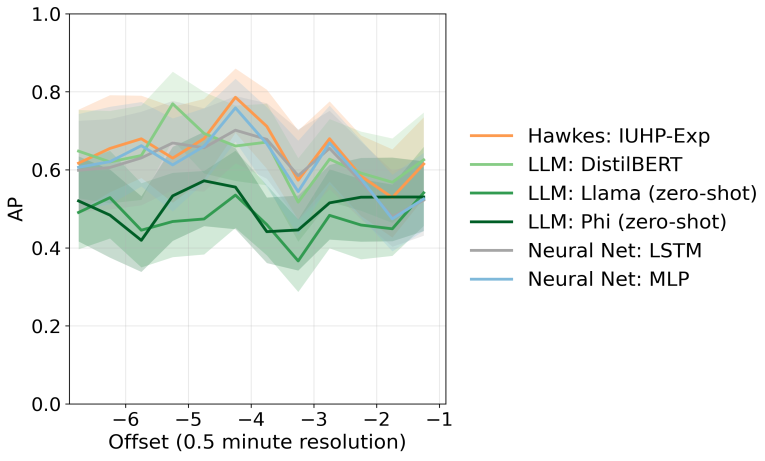
We now present empirical results for the two prediction tasks described in Section 4.2.

5.1. Short-Term Conversation Continuation Prediction

Figures 2, 3, and 4 report time-varying performance for the short-term continuation prediction task ($\Delta = 60$ seconds) for the $\mathcal{P}$, $\mathcal{N}$, and $\mathcal{U}_{\text{test}}$ cohorts, respectively. Evaluation is conducted at 30-second checkpoints, spanning offsets from -7 minutes to -1.5 minutes relative to the apparent end time. In each figure, the left panel displays ROC-AUC scores, while the right panel displays AP scores. Each figure includes results for the best-performing Hawkes model variant, which is the IUHP–Exp model; the best-performing neural network model of each type (MLP and LSTM); and the three LLM-based models. For the MLP, the best-performing feature set includes (i) message-level features—elapsed time since conversation start, word count, and correspondent role—and (ii) a chat-level feature, namely the UHP intensity evaluated at the checkpoint. For the LSTM, the best-performing feature set includes (i) a message-level feature, elapsed time, and (ii) a conversation-level feature, the UHP intensity evaluated at the checkpoint.



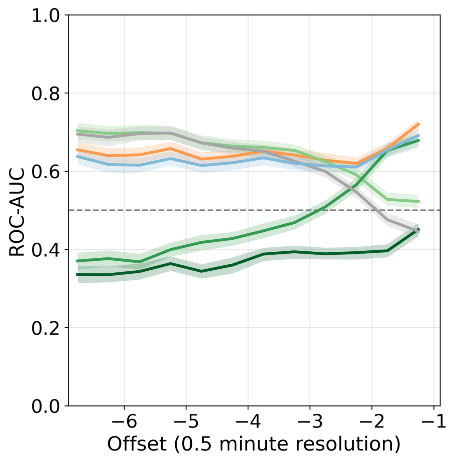
(a) ROC-AUC



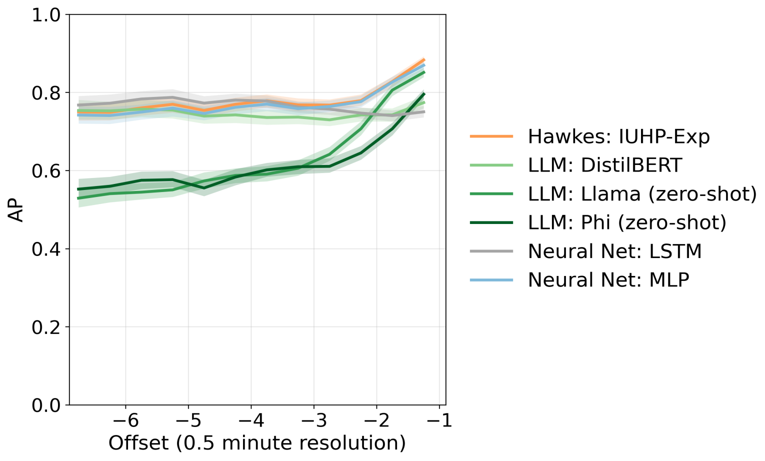
(b) AP

Figure 2 Short-Term Continuation Test (Δ=60s): Prediction Accuracy by Offset Time, Cohort $\mathcal{P}$ (Premature Closures with Verified Returns), 96 Conversations.

We make the following observations. Across all cohorts, models based on conversational metadata—namely the Hawkes processes and the neural networks—exhibit the strongest and most stable performance, reliably identifying whether a conversation will continue in the near term. In particular, the IUHP variants and the MLP typically achieve ROC-AUC values in the range of



(a) ROC-AUC



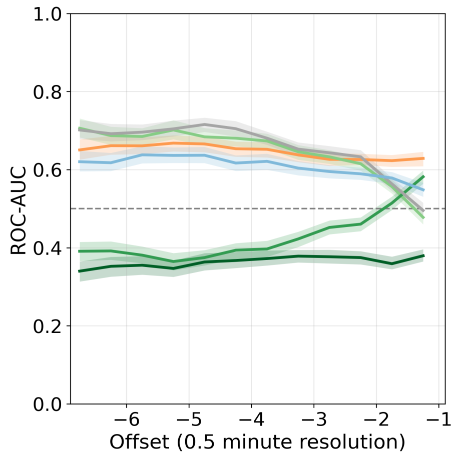
(b) AP

Figure 3 Short-Term Continuation Test (Δ=60s): Prediction Accuracy by Offset Time, Cohort $\mathcal{N}$ (Successful Closures), 5,145 Conversations.

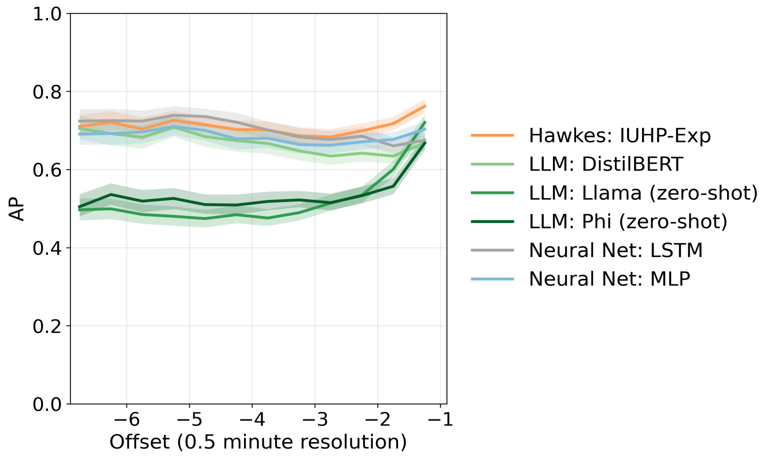
approximately 0.60–0.75 and AP values in the range of approximately 0.65–0.80, with performance varying by cohort and offset bin.

In contrast, zero-shot prompted LLMs (Phi-3 and Llama 3) perform substantially worse in this task. Although Llama 3 shows some improvement as the conversation approaches its apparent end, both zero-shot models remain inferior to the Hawkes and neural network models across most offsets. However, when trained explicitly for the task, text-based models improve markedly. The fine-tuned DistilBERT model performs competitively across cohorts, though its performance degrades near the apparent end of the conversation (offsets between −2.5 and −1 minutes), where the Hawkes and MLP models retain an advantage.

Overall, these results suggest that short-term continuation prediction within an ongoing chat can be accurately identified by temporal dynamics and operational state. Such signals are well captured



(a) ROC-AUC



(b) AP

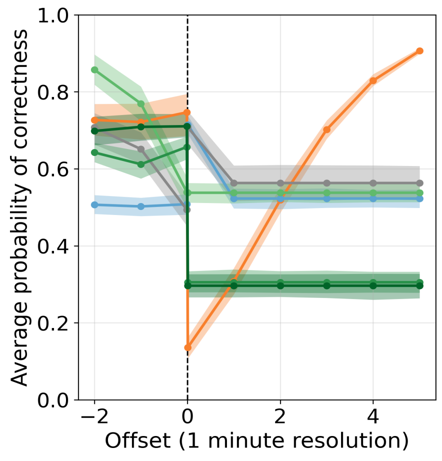
Figure 4 Short-Term Continuation Test (Δ=60s): Prediction Accuracy by Offset Time, Cohort $\mathcal{U}_{\text{test}}$ (Uncertain Closure), 5,000 Conversations.

by the Hawkes intensities and lightweight structured features. Textual content provides only limited incremental predictive value for this task, and mainly at earlier stages of the conversation.

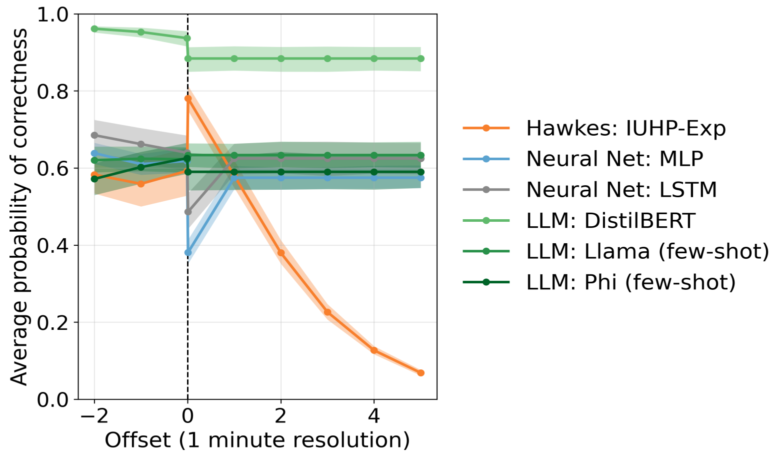
5.2. End-of-Service Prediction

Figures 5(a) and 5(b) report time-varying performance for the end-of-service prediction task ($\Delta = \infty$) for the $\mathcal{N}$ and $\mathcal{P}$ cohorts, respectively. Evaluation is conducted at 60-second checkpoints, spanning offsets from -2 minutes to 5 minutes relative to the apparent end time, and includes the boundary checkpoints 0^- and 0^+, corresponding to instants immediately before and after the last observed message.

In each figure, we plot the average predicted probability assigned to the correct label, which we refer to as the *probability of correctness*. For checkpoints with ground-truth label 1 (the conversation



(a) 5-fold $\mathcal{N}_{\text{test}}$ cohorts (Successful Closures)



(b) 5-fold $\mathcal{P}_{\text{test}}$ cohorts (Premature Closures)

Figure 5 **End-of-service Test ($\Delta = \infty$): Average Probability of Correctness as a Function of Offset Time.**

has not yet ended), this quantity equals the model prediction $\hat{p}_i(t;\infty)$. For checkpoints with ground-truth label 0 (the conversation has ended), it equals $1-\hat{p}_i(t;\infty)$. At negative offsets (corresponding to checkpoints before the last message) the ground-truth label is always 1, since at least one additional message is observed in the transcript. At positive offsets (corresponding to checkpoints after the last message) the ground-truth label depends on the cohort: conversations in cohort $\mathcal{N}$ have ended, whereas conversations in cohort $\mathcal{P}$ eventually resume. Accordingly, we report separate curves for cohorts $\mathcal{N}$ and $\mathcal{P}$. In both cases, higher values indicate better performance.

Figure 5 shows that, for this task, the DistilBERT model achieves the strongest overall performance across most timestamps. This advantage is particularly pronounced for the P cohort, where its probability of correctness is close to 90% over a wide range of offsets, making it the best-performing model for end-of-service prediction. In contrast, the Hawkes-based models exhibit substantially more volatile behavior and comparatively weaker performance for this task.

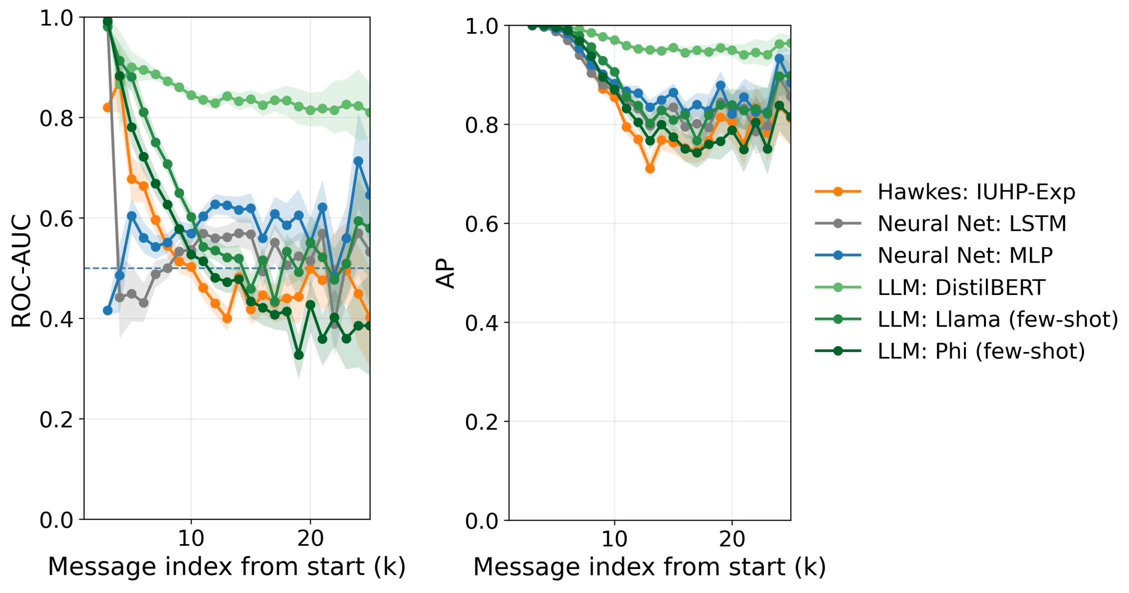
Delving deeper, we observe that the LLM-based models exhibit constant performance levels at positive offsets. This behavior is expected: because predictions at each checkpoint may only use information available up to that time, once the conversation reaches τ_i^{app} the observed message history becomes fixed. Consequently, the textual input to the LLMs does not change for offsets $t > \tau_i^{\text{app}}$, and their predictions remain constant over time. In contrast, models that incorporate time-dependent covariates, most notably Hawkes-based intensities, continue to update their predictions as the elapsed time since the last message increases, resulting in time-varying behavior at both negative and positive offsets. However, time adaptivity does not necessarily translate into better performance. The key challenge in this task is the ability to distinguish between the post-end dynamics of the $\mathcal{N}$ and $\mathcal{P}$ cohorts. Here, we identify a fundamental limitation of the Hawkes models, which is most clearly illustrated by examining their behavior around offset time 0.

Naturally, we observe a discontinuity at the boundary between 0^- and 0^+ for most models, reflecting the arrival of highly informative new data at time 0, namely the last observed message of the conversation. This discontinuity is particularly pronounced for the Hawkes models. By construction, the occurrence of a new event immediately increases the predicted probability of future events in the Hawkes model, regardless of the underlying cohort (see Equation 3). As a result, Hawkes models experience an immediate drop in correctness for the $\mathcal{N}$ cohort—where no further messages will arrive—and a corresponding increase for the $\mathcal{P}$ cohort—where the conversation will eventually resume. Over time, however, as no additional messages are observed, the Hawkes intensity decays. This leads the model to become increasingly pessimistic about continuation, which is appropriate for the $\mathcal{N}$ cohort but incorrect for the $\mathcal{P}$ cohort, ultimately reversing the initial correctness evaluation.

In contrast, text-based models, including DistilBERT and the prompt-based LLMs, while flat across positive offsets for the reasons discussed above, nevertheless exhibit distinct post-end prediction levels for the $\mathcal{N}$ and $\mathcal{P}$ cohorts. These differences are driven solely by the content of the observed transcript—most notably the final message—indicating that semantic cues can provide meaningful information about restart risk at τ_i^{app}.

We also report in Figure 6(a) ROC-AUC performance evaluated at sequential message timestamps, where each model is applied after each message to predict whether the conversation has reached end-of-service. This evaluation can only be conducted on the $\mathcal{N}$ cohort, for which we verified the successful closure message of each conversation.

We observe that, here as well, the fine-tuned DistilBERT model achieves consistently high classification performance, with ROC-AUC values in the range of approximately 0.82–0.90. This indicates



(a) ROC-AUC (b) AP

Figure 6 **End-of-Service Test ($\Delta = \infty$): Prediction Accuracy by Message Index, Cohort $\mathcal{N}$ (Successful Closures), 5,145 Conversations.**

strong reliability in distinguishing between conversations that are still ongoing and those that have already completed. Notably, DistilBERT substantially outperforms both few-shot prompt-based LLMs and the non-text-based models in this message-wise evaluation.

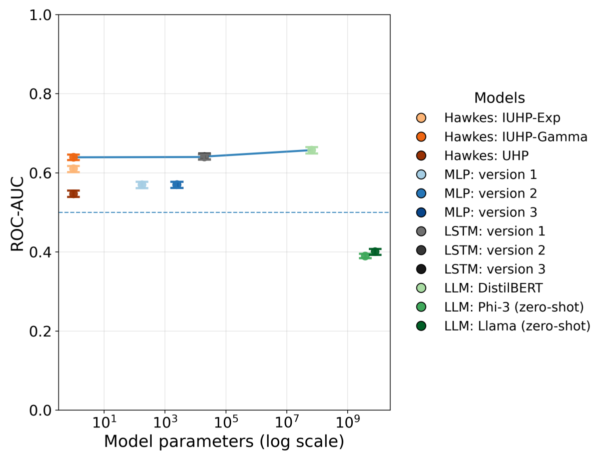
In contrast, the few-shot prompted LLMs perform well only at early stages of the conversation, but their accuracy degrades markedly as the number of messages increases. Specifically, both Llama and Phi exhibit a decline in ROC-AUC from roughly 0.85–0.90 at the beginning of the conversation to approximately 0.40–0.55 at later stages. The widening confidence intervals at higher message indices (toward the right side of the figure) are expected, as fewer conversations are sufficiently long to contribute observations at those indices, resulting in increased statistical uncertainty.

Taken together, Figures 5 and 6(a) highlight that end-of-service prediction hinges more on semantic cues in the final message than on temporal dynamics, making fine-tuned text-based models such as DistilBERT substantially more effective than intensity-based approaches once the conversation reaches its apparent end.

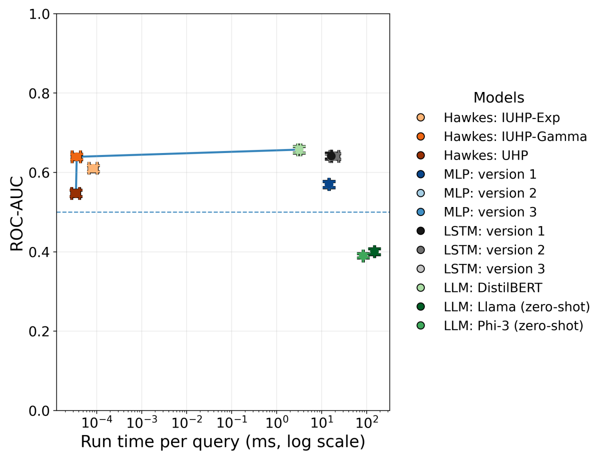
The analysis also reveal a potential collaboration between the models—if no-activity occurs in a conversation for a long time and the Hawkes give high recommendation to close, the LLM can verify that decision. Such collaboration is also motivated by the cost-accuracy analysis we perform in the next section.

5.3. Cost-Accuracy Tradeoff

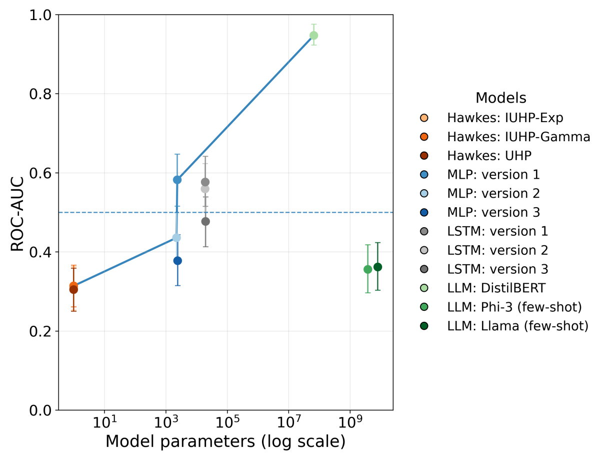
Figure 7 summarizes the cost-accuracy landscape across model families for both prediction tasks, using two complementary notions of computational cost. The horizontal axis reports model cost measured either by the number of features (left panels) or by query run time (right panels), both for inference time queries and shown on a logarithmic scale. The vertical axis reports predictive accuracy, measured by pooled ROC-AUC aggregated across all offset times. Models closer to the upper-left corner achieve higher accuracy with lower cost and are therefore preferable. The blue curve depicts the efficiency frontier, consisting of the models that are not dominated in terms of the cost-accuracy tradeoff.



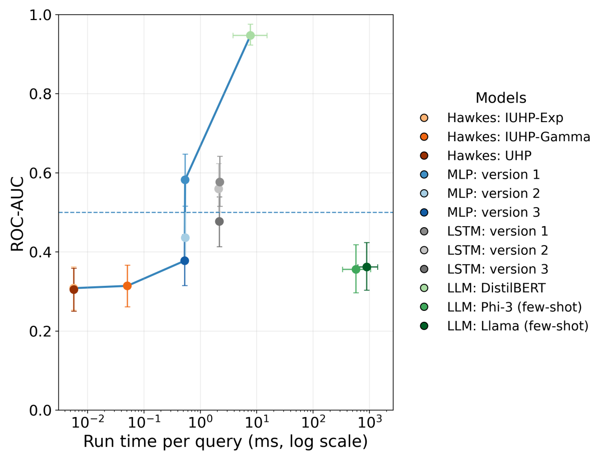
(a) Short-Term Continuation, Num. of Parameters (Cohort $\mathcal{U}_{\text{test}}$)



(b) Short-Term Continuation, Run Time (Cohort $\mathcal{U}_{\text{test}}$)



(c) End-of-Service, Number of Parameters (Cohorts $\mathcal{P}$ and $\mathcal{N}$)



(d) End-of-Service, Run Time (Cohorts $\mathcal{P}$ and $\mathcal{N}$)

Figure 7 Cost-Accuracy Tradeoff Across Model Families for Different Prediction Tasks.

In the short-term continuation task ($\Delta = 60$s; upper panels), performance clusters within a relatively narrow horizontal band, with the best models achieving pooled ROC-AUC approximately 0.65. Extending the comparison in Section 5.1, we report the best three variants within each model family. A key operational takeaway is that the initialized Hawkes models (IUHP-Gamma and IUHP-Exp) lie on, or very close to, the efficient frontier: despite having only a handful of fitted parameters, they achieve pooled ROC-AUC comparable to substantially larger neural architectures. Their run time is also at least 4 orders of magnitude shorter than the other models. Moving from Hawkes to learned neural models (MLP and LSTM) yields, at best, modest accuracy improvements, but at a substantially higher computational cost (more parameters and much longer run time). These accuracy gains are also sensitive to feature design: the strongest neural variants combine message-level structure (e.g., word count and corespondent role) with chat-level temporal state (Hawkes intensity). Among text-based models, as was seen in Section 5.2, the fine-tuned DistilBERT is the only LLM that reaches the top tier of ROC-AUC in this task, but it does so at several orders-of-magnitude higher parameter count than Hawkes or neural networks and at several orders-of-magnitude higher run time than the Hawkes. (The run time of neural networks and LLMs are approximately of the same magnitude.) In contrast, zero-shot prompted-based causal LLMs (Phi-3 and Llama 3) are clearly dominated in this short-horizon task: they require billions of parameters yet deliver substantially lower ROC-AUC and the longest run time.

The picture changes sharply for the end-of-service task ($\Delta = \infty$; lower panels), evaluated *at the* 0^+ *checkpoint* for both $\mathcal{P}$ and $\mathcal{N}$ cohorts. Again, we compare models by the number of parameters (left panel) and their inference-time latency per checkpoint (right panel). The efficiency frontier is formed by some representatives of the three model families: (i) the Hawkes model, which achieves very low latency, require low complexity, but provide overall low accuracy, (ii) the MLP neural network with with medium range complexity and run time, and (iii) the fine-tuned DistilBERT classifier, which delivers excellent accuracy at the cost of substantially higher complexity and latency. Notably, the fine-tuned LLM is still two orders of magnitude faster than the few-shot prompt-based LLMs. The prompted Phi-3 and Llama-3 models are not competitive in this setting: they incur very high inference times (hundreds to thousands of milliseconds per query) while achieving substantially lower accuracy than fine-tuned DistilBERT. The LSTM models also lie off the efficiency frontier, as they attain only moderate ROC-AUC while requiring inference times comparable to that of the DistilBERT classifier. The best MLP offers moderate performance with mid-range run time, balancing cost and accuracy.

To summarize, the long inference latency of LLMs may render them unsuitable for continuous end-of-conversation surveillance, as activity prediction must be performed in real time and at high frequency to support operational concurrency control. In the next section, we propose a new approach that addresses this challenge by leveraging the complementary strengths of the lightweight models and cost–accuracy tradeoffs identified in our analysis.

6. Combining Prediction Methods Optimally: A Stylized Model and a Data-Driven Experiment of Hybrid Hawkes-LLM Prediction Policies

As the experiments on real-world data shows, both the strength and the weakness of the Hawkes model lie in the fact that it is inherently cautious. Because each new jump in the intensity increases the likelihood of future activity, the Hawkes will hedge its bets in predictions. As we have seen in the data, this is a highlight of the model for short-term predictions within services, but it is also a drawback for long-term predictions, especially towards the close of service, where the temporal hesitance of the model limits its ability to make meaningful insights. By contrast, those same settings are precisely where the ML/AI models shine, with their predictions can leverage the actual text to make sense of long-term and end-of-service questions. However, the performance of the ML/AI models suffers on the intra-service settings, where the contexts may not be clear enough to make reliable predictions.

To illustrate how these different types of predictions can be best combined, let us walk through a stylized example based around the Hawkes model. First, notice that, in some sense, the Hawkes is overly cautious *even when it is the true data generation model.* That is, because the Hawkes intensity jumps with every message, it will always increase its predicted probability that there are more messages to be sent, even after the last message itself. So, when possible, we would want to try to augment the purely temporal predictions from the Hawkes with some additional context.

In this stylized example, let us suppose that the data is being generated according to an initialized univariate Hawkes process as defined in (3). Then, let us additionally assume that we have access to another predictor for the (in)existence of future service activity that is highly accurate but expensive to query (such as the fine-tuned DistilBERT). By comparison to the self-excitation-driven predictions directly from the Hawkes process, a natural tradeoff arises: if we query the expensive predictor often, we regularly gain valuable information on whether the service is ongoing, but we pay a high cost. On the other hand, if we query the expensive predictor infrequently, we save on costs but don't have many opportunities to leverage potential improvements in predictions.

Specifically, let us consider policies of the following form: let $\eta \in \mathbb{R}_+$ with $\eta \leq \lambda_0$ be some level such that, whenever the Hawkes intensity $\lambda(t)$ reaches η, then we query the expensive predictor. Immediately, we have that the probability of future activity in the Hawkes process at this level will be $e^{-\eta/\beta}$. Hence, the total number of times that the Hawkes intensity reaches (meaning, decays down to, not jumps over) the level η is geometrically distributed with probability parameter $e^{-\eta/\beta}$, meaning that we expect to run the expensive predictor for an $e^{\eta/\beta}$ number of queries. On the other hand, if η is near 0, there is already a high probability that no messages remain, and thus the information gain from the expensive predictor is low. By comparison, if η is high (e.g., near the initial level λ_0, which itself can be larger than the jump size α), then the potential information gain should be nontrivial. Accordingly, we can measure the weighted entropy of the event that no messages remain, $-e^{-\eta/\beta}\log(e^{-\eta/\beta}) = (\eta/\beta)e^{-\eta/\beta}$.

Together, this yields a cost-accuracy tradeoff objective for this prediction combination problem of

$$\mathcal{C}e^{\frac{\eta}{\beta}} - \frac{\eta}{\beta}e^{-\frac{\eta}{\beta}}, \tag{5}$$

where $\mathcal{C} > 0$ reflects the weighting of the expensive query costs relative to the benefits of the information gain. In Proposition 1, we show that, so long as the cost of queries are not too expensive, this objective has a unique optimal level at which to combine the semantic model with the temporal.[3]

Proposition 1. *Suppose $\mathcal{C} < 1$. Then, the optimal Hawkes intensity level at which to query the expensive predictor with minimal expected cost is*

$$\eta^* = \frac{\beta}{2}\left(2 - \mathsf{W}_0\left(2\mathcal{C}e^2\right)\right) > 0, \tag{6}$$

where W_0 is the principal branch of the Lambert-W function. For $\mathcal{C} \geq 1$, the expensive predictor is not cost effective.

Notice that, though the Lambert-W may be less familiar than many other named functions, its basic properties can quickly show that η^* provides an intuitive policy. First, because $\mathsf{W}_0(\cdot)$ is non-decreasing, we have that as the per-query cost increases, then the optimal level decreases, meaning we call the expensive predictor less often. At the extreme, as $\mathcal{C} \to 1$, meaning that the cost of running the expensive predictor is as large as the value of gaining the information for this binary event, then $\eta^* \to 0$, meaning we will never query the expensive predictor despite its perfect accuracy. On the other hand, if $\mathcal{C}$ decreases, then η^* increases, and we query the expensive predictor more often. Interestingly, though, as the predictor becomes effectively cost-free, this stylized model still suggests to only query it a handful of times: $\eta^* \to \beta$ as $\mathcal{C} \to 0$, meaning that the expected number of queries is just $e^{\eta^*/\beta} \to e$.

6.1. Evaluating the Hybrid Hawkes-LLM Prediction Policy on the Service Data

Following the insights of the stylized model in which the conversation is truly driven by a Hawkes process and the expensive predictor has perfect accuracy, let us now actually evaluate these ideas on the real service data, where we have already seen that no predictor is perfect and no model exactly captures the true dynamics. Specifically, let us now present the accuracy results of a hybrid model for the end-of-service real-time prediction task.

With the hyperparameter η, the hybrid predictor operates as follows: At every message timestamp T_i, the intensity level of the IUHP-Exp model is updated to $\lambda(T_i)$. Then, we define an LLM checkpoint between message i and $i+1$ (i.e, during the time interval $[T_i, T_{i+1})$) if the value of the intensity $\lambda(t)$ decreases below η from above during that time interval ($\lambda(T_i)\exp\{-\beta(T_{i+1}-T_i)\} \leq \eta$). The prediction at this checkpoint is computed via the best-performing end-of-service LLM, fine-tuned DistilBERT.

In order to ensure that there are both positive and negative values on every sample path so that we can compute the ROC-AUC of the experiment, we evaluate this hybrid prediction policy on the $\mathcal{N}$ cohort. To evaluate the nature of the combination, we consider a range of η values. Recall that, in the theoretical setting of the stylized model, we can let the hyperparameter take values up to the initial value of the IUHP, $\eta \leq \lambda_0$, to ensure that the threshold is reached at least once. However, in this data-driven experiment, we allow η to actually exceed $\lambda_0 = 112.72$ up to a slightly higher value, $\eta \leq 148$, at which the average number of total LLM queries in the hybrid predictor in the experiment is the same as the average number of messages per conversation in the data.[4] We search over values of η with a unit step size.

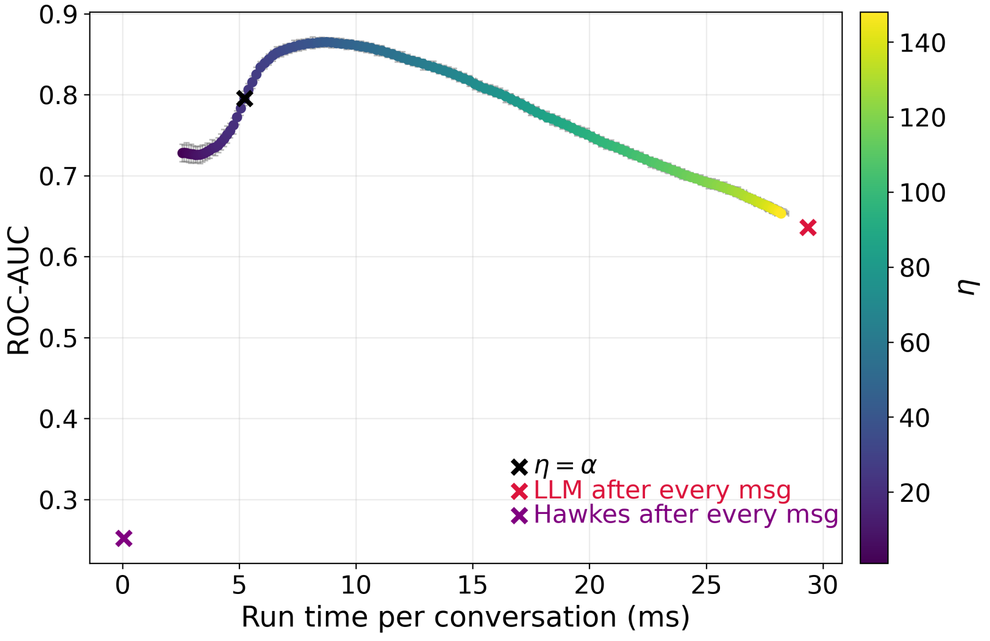


Figure 8 Cost-Accuracy Tradeoff of the Hybrid Hawkes-LLM Policy Across η Values: Real-Time End-of-Service Prediction, Cohort $\mathcal{N}$ (Successful Closures), 5,145 Conversations.

Figure 8 illustrates the resulting cost–accuracy tradeoff as a function of η, together with two benchmarks. The first benchmark corresponds to a Hawkes-only predictor, in which the LLM is never queried; is shown by the purple $\times$ (with run time on the order of 10^{-5} ms). The second benchmark corresponds to an LLM-only predictor, in which the DistilBERT model is queried after every message; this policy is marked by a red $\times$ in the figure. For reference, we also mark the point $\eta = \alpha$ using a black $\times$. All ROC-AUC values in this figure use a conversation-level construction to report an overall accuracy measurement. For each conversation i, we calculate the continuation probabilities $\hat{p}_i(T_{i,k};\infty)$ at each message checkpoint and form two scores: $s_i^{\text{pre}} = \min_{k=1,\dots,n_i-1} \hat{p}_i(T_{i,k};\infty)$ (labeled 1, since the conversation continues after every pre-last checkpoint) and $s_i^{\text{end}} = \hat{p}_i(T_{i,n_i};\infty)$ (labeled 0, since no messages remain after the final checkpoint). We then compute pooled ROC–AUC over the resulting $2|\mathcal{N}|$ labeled points $\{(s_i^{\text{pre}},1),(s_i^{\text{end}},0)\}_{i\in\mathcal{N}}$.) This construction evaluates a model's ability to avoid premature closure by penalizing any overly low continuation probability before the last message (via the minimum), while still requiring the probability to drop at the true close of service. In this sense, this ROC-AUC calculation summarizes the within-service and end-of-service accuracies into one metric.

We observe that even for very small values of η, the hybrid policy substantially improves prediction accuracy relative to the Hawkes-only baseline without incurring high run times, approaching and even exceeding the accuracy of the always-on LLM policy while reducing computational cost by an order of magnitude. Moreover, we observe that that accuracy does not increase monotonically with η. As η increases, the LLM is invoked more frequently, and the policy gradually improves, until reaching maximal performance at $\eta = 43$, which is slightly above β but below 2α. Then, we see that it is possible to have too much of a good thing: the accuracy values of η beyond this range are not beneficial, and the performance begins to deteriorate.

Notice that, for $\eta \gg \alpha$, the LLM will be invoked mostly at times when the conversation has a burst of activity (i.e., sequences of closely timed messages). That is, after a period of inactivity, it takes several jumps, and not merely one jump, for the intensity to cross η and trigger an LLM query. This may be inherently capitalizing on natural cadences in customer-agent communication patterns, where quick exchanges of messages are mostly focused on the same subject, and thus repeated LLM calls do not add much value. In this way, it seems that a little bit of such screening is beneficial for performance whereas too much screening can hamper performance. Thus, in this setting, more frequent LLM usage can be counterproductive, leading to both higher cost and lower accuracy. Consequently, intermediate values of η achieve the most favorable cost–accuracy tradeoff, underscoring the importance of carefully tuning η in practice, with the insight above suggesting $\alpha \le \eta \le \lambda_0$ as the most promising range of values, yet any choice of $\eta > 0$ offers a hybrid model performance that achieves better accuracy than either predictor alone.

Interestingly, this data-driven experiment matches the observation from the stylized model that, in terms of accuracy but regardless of cost, it is most effective to query the LLM just a handful of times. That is, for the observed peak of the ROC-AUC at $\eta = 43$, the average number of LLM queries per conversation was 3.61. By contrast, the average number of messages per conversations in the $\mathcal{N}$ cohort is 12.57. Hence, across all values of η and the corresponding range of total computational costs, the most accurate performance is achieved through only a few LLM queries.

Much like the stylized model suggests, our overarching takeaway from this experiment is that the best way to combine these two types of models is to let each focus on their respective strengths. The Hawkes process, as a temporal-history-driven self-exciting stochastic model, is inherently cautious and expects that present activity will beget future activity. That concept captures the spirit of the back-and-forth interaction between the customer and agent, but eventually the service will cease. It would take the Hawkes some time to recognize that inactivity, and this is precisely where the text-driven LLM can step in. By selectively calling the fine-tuned DistilBERT only at the handful of times at which the Hawkes process intensity starts to drop low, we both save on computational effort and set the AI model up to handle tasks at which it is best suited. By comparison to the simple and analytically tractable setting of the stylized model, though, this experiment now provides evidence that this stochastic-model-and-AI combination holds promise even when the stylized model's assumption clearly do not apply: the data used here is authentic data, rather than Hawkes-generated, and the AI model is good but not perfect. Nevertheless, we have seen here that the hybrid predictor can both improve performance and save cost relative to simply deploying the AI model on its own.

7. Conclusion

In this paper we have tested three family of learning methods to predict conversation continuation. Our first conclusion is that both predicting the short-term conversation continuation and determining whether a conversation has truly ended are challenging tasks, even when the full conversation transcript is available. Most models exhibit limited ability to capture customer behavior in the short run, with the maximal pooled ROC-AUC for short-term continuation prediction reaching only about 0.65. For this short-horizon task, no model clearly dominates the Hawkes approach: it provides the most reliable performance at the lowest computational cost. This is unsurprising, as Hawkes models are specifically designed to capture short-term temporal dynamics and excitation effects, which are central to near-future activity prediction.

In contrast, long-term end-of-service prediction appears to be a more tractable problem, with the best-performing models achieving substantially higher ROC-AUC. For this task, temporal dynamics alone are insufficient. Instead, language models that exploit the full transcript and extract semantic cues related to customer intent offer a significant advantage. Importantly, however, zero-shot LLMs perform poorly in this setting; effective performance requires task-specific fine-tuning.

Both prediction tasks are operationally important but serve different purposes. Short-term continuation prediction supports real-time routing and staffing decisions, where immediate activity intensity informs agent workload and availability. End-of-service prediction, in contrast, is critical for closure decisions and the prevention of premature termination. Furthermore, as our analysis and experiments of prediction combination shows, good short-term predictors actually have an additional use: determining when to run the end-of-service predictions.

The cost-accuracy analysis, together with the distinct strengths of temporal and text-based models, suggests clear complementarities between these approaches. Hawkes models excel at capturing short-term dynamics efficiently, while fine-tuned LLMs are better suited for interpreting long-term intent from textual content. Combining these strengths offers the potential to improve accuracy without prohibitive cost, yielding a system whose performance exceeds that of any individual component. Our stylized theoretical model suggested a practical method for achieving such integration. We posit that by monitoring conversations using the Hawkes prediction model, one can optimally control the frequency of running the LLM and scheduled it to the appropriate times. We show the effectiveness of such approach in a final data-driven experiment. Here, we found that it is possible to actually improve upon the performance of the best ML/AI model, the fine-tuned DistilBERT LLM, while only incurring half the computational cost. This best-of-both-worlds scenario is achieved by selectively querying the LLM at checkpoints strategically determined by a lightweight Hawkes model, which also handles the prediction at all other time points.

Our analysis paves the way for a new approach to combining multiple learning models. While traditional ensemble methods (e.g., bagging, boosting, and stacking) aggregate predictions across models, our approach combines models operationally, by synchronizing when each model is invoked over time. This enables improved predictive performance while making more efficient use of computational resources.

Although we apply this paradigm to chat-based service interactions, the idea extends naturally to other settings, such as AI assistant systems (e.g., chatbots). These systems also manage ongoing, chat-like interactions and must decide when a conversation has effectively ended so that working memory and computational resources can be released.

Notes

[1]We use the terms "*chat*" and "*conversation*" interchangeably throughout the paper.

[2]Some model parameters, i.e. α, are repeated across the model variants for simplicity of expression, but they need not be the same exact value.

[3]The proof appears in Appendix EC.6.

[4]As given in Table EC.2, the parameters for the IUHP-Exp are $\alpha = 24.31$, $\beta = 39.23$, and $\lambda_0 = 112.72$.

Acknowledgments

We are grateful to the company for providing the data used in this study. The authors are grateful for the generous support of this work by the National Science Foundation, Division of Civil, Mechanical and Manufacturing Innovation [Award #2441387] (A. Daw) and the United States-Israel Binational Science Foundation [Award #2022095] (A. Daw and G. B. Yom-Tov). We also thank Amy R. Ward for her early involvement in the project and subsequent feedback, which shaped several key parts of the paper. Part of this work was conducted while A. Castellanos was a postdoctoral researcher at the University of Chicago Booth School of Business, whose financial support is gratefully acknowledged.

References

Abdin M, Jacobs S, Awan AA, et al. (2024) Phi-3 technical report: A highly capable language model locally on your phone. arXiv preprint arXiv:2404.14219.

Aksin Z, Armony M, Mehrotra V (2007) The modern call center: A multi-disciplinary perspective on operations management research. *Production and Operations Management* 16(6):665–688.

Altman D, Yom-Tov GB, Olivares M, Ashtar S, Rafaeli A (2021) Do customer emotions affect agent speed? An empirical study of emotional load in online customer contact centers. *Manufacturing & Service Operations Management* 23(4):854–875.

Armony M, Maglaras C (2004) On customer contact centers with a call-back option: Customer decisions, routing rules and system design. *Operations Research* 52(2):271–292.

Ashtar S, Rafaeli A, Yom-Tov GB, Akiva N (2021) When do service employees smile? response-dependent emotion regulation in emotional labor. *Journal of Organizational Behavior* 42:1202–1227.

Bekker J, Davis J (2020) Learning from positive and unlabeled data: A survey. *Machine Learning* 109(4):719–760.

Bertsimas D, Kallus N (2020) From predictive to prescriptive analytics. *Management Science* 66(3):1025–1044.

Blundell C, Beck J, Heller KA (2012) Modelling reciprocating relationships with Hawkes processes. Pereira F, Burges C, Bottou L, Weinberger K, eds., *Advances in Neural Information Processing Systems*, volume 25 of *NIPS'12*, 2600–2608 (Lake Tahoe, Nevada: Curran Associates, Inc.).

Buesing E, Haug M, Hurst P, Lai V, Mukhopadhyay S, Raabe J (2024) Where is customer care in 2024? McKinsey & Company, URL https://www.mckinsey.com/capabilities/operations/our-insights/where-is-customer-care-in-2024, accessed: 2025-12-25.

Carlini N, Tramèr F, Wallace E, Jagielski M, Herbert-Voss A, Lee K, Roberts A, Brown T, Song D, Erlingsson Ú, Oprea A, Raffel C (2021) Extracting training data from large language models. *30th USENIX Security Symposium*, 2633–2650, USENIX Security 21 (Online: USENIX Association).

Castellanos A, Daw A, Ward AR, Yom-Tov GB (2024) Closing the service: Contrasting activity-based and time-based systematic closure policies. *Proceedings of the 2024 Winter Simulation Conference*, 2440–2451, WSC '24 (Orlando, FL, USA: IEEE Press).

Castellanos A, Yom-Tov GB, Goldberg Y, Park J (2025) Silent abandonment in text-based contact centers: Identifying, quantifying, and mitigating its operational impacts. *arXiv preprint arXiv:2501.08869* .

Chandwani V, Kumar S, Singh PK (2020) Long short-term memory based conversation modelling. *2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE)*, 105–109 (Jaipur, India: IEEE).

Chiang WH, Liu X, Mohler G (2022) Hawkes process modeling of covid-19 with mobility leading indicators and spatial covariates. *International journal of forecasting* 38(2):505–520.

Daw A, Castellanos A, Yom-Tov GB, Pender J, Gruendlinger L (2025) The co-production of service: Modeling services in contact centers using Hawkes processes. *Management Science* 71(3):2635–2656.

Daw A, Pender J (2018) Queues driven by Hawkes processes. *Stochastic Systems* 8(3):192–229.

Daw A, Pender J (2022) An ephemerally self-exciting point process. *Advances in Applied Probability* 54(2):340–403.

Daw A, Yom-Tov GB (2024) Asymmetries of service: Interdependence and synchronicity. *arXiv preprint arXiv:2402.15533* .

Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 4171–4186 (Minneapolis, MN, USA: Association for Computational Linguistics).

(EDPB) EDPB (2025) AI privacy risks & mitigations – large language models (LLMs). EDPB report, URL https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf, accessed: 2025-12-25.

Elkan C, Noto K (2008) Learning classifiers from only positive and unlabeled data. *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 213–220, KDD '08 (New York, NY, USA: Association for Computing Machinery).

Embrechts P, Liniger T, Lin L (2011) Multivariate Hawkes processes: An application to financial data. *Journal of Applied Probability* 48(A):367–368.

Ferraro C, Demsar V, Sands S, Restrepo M, Campbell C (2024) The paradoxes of generative AI-enabled customer service: A guide for managers. *Business Horizons* 67(5):549–559.

Gans N, Koole G, Mandelbaum A (2003) Telephone call centers: Tutorial, review, and research prospects. *Manufacturing and Service Operations Management* 5(2):79–141.

Goes PB, Ilk N, Lin M, Zhao JL (2018) When more is less: Field evidence on unintended consequences of multitasking. *Management Science* 64(7):2973–3468.

Goodfellow I, Bengio Y, Courville A (2016) *Deep Learning* (MIT Press).

Grattafiori A, Dubey A, Jauhri A, et al. (2024) The Llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Gruver N, Finzi M, Qiu S, Wilson AG (2023) Large language models are zero-shot time series forecasters. *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23 (Red Hook, NY, USA: Curran Associates Inc.).

Halpin PF, De Boeck P (2013) Modelling dyadic interaction with Hawkes processes. *Psychometrika* 78(4):793–814.

Hawkes AG (1971) Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58(1):83–90.

Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Computation* 9(8):1735–1780.

Ilk N, Shang G (2022) The impact of waiting on customer-instigated service time: Field evidence from a live-chat contact center. *Journal of Operations Management* 68(5):487–514.

Janiszewski P, Lango M, Stefanowski J (2021) Time aspect in making an actionable prediction of a conversation breakdown. *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part V*, 351–364 (Berlin, Heidelberg: Springer-Verlag).

Jiang B, Ekstedt E, Skantze G (2023) Response-conditioned turn-taking prediction. *Findings of the Association for Computational Linguistics: ACL*, 12241–12248 (Toronto, Canada: Association for Computational Linguistics).

Jin M, Wang S, Ma L, Chu Z, Zhang JY, Shi X, Chen PY, Liang Y, Li YF, Pan S, Wen Q (2024a) Time-LLM: Time series forecasting by reprogramming large language models. *The 12th International Conference on Learning Representations*, ICLR'24 (Vienna, Austria: Curran Associates, Inc.).

Jin M, Zhang Y, Chen W, Zhang K, Liang Y, Yang B, Wang J, Pan S, Wen Q (2024b) Position: What can large language models tell us about time series analysis. *Proceedings of the 41st International Conference on Machine Learning*, ICML'24 (Vienna, Austria: JMLR.org).

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations*, ICLR'14 (Banff, Canada: Curran Associates, Inc.).

Koops D, Saxena M, Boxma O, Mandjes M (2018) Infinite-server queues with Hawkes input. *Journal of Applied Probability* 55(3):920–943.

Kwon W, Li Z, Zhuang S, Sheng Y, Zheng L, Yu CH, Gonzalez J, Zhang H, Stoica I (2023) Efficient memory management for large language model serving with PagedAttention. *Proceedings of the 29th Symposium on Operating Systems Principles*, 611–626, SOSP '23 (New York, NY, USA: Association for Computing Machinery).

Laub PJ, Lee Y, Taimre T (2021) *The Elements of Hawkes Processes* (Cham, Switzerland: Springer).

Lee MC, Deng Z (2024) Online multimodal end-of-turn prediction for three-party conversations. *Proceedings of the 26th International Conference on Multimodal Interaction*, 57–65, ICMI '24 (San Jose, Costa Rica: Association for Computing Machinery).

Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. *The 7th International Conference on Learning Representations*, ICLR'19 (New Orleans, USA: Curran Associates, Inc.).

Luo J, Zhang J (2013) Staffing and control of instant messaging contact centers. *Operations Research* 61(2):328–343.

Mahadevan A, Mathioudakis M (2024) Cost-aware retraining for machine learning. *Knowledge-Based Systems* 293:111610.

Mitzenmacher M, Shahout R (2025) Queueing, predictions, and large language models: Challenges and open problems. *Stochastic Systems* 15(3):195–219.

Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 807–814, ICML'10 (Madison, WI, USA: Omnipress).

Pinto MJ, Belpaeme T (2024) Predictive turn-taking: Leveraging language models to anticipate turn transitions in human-robot dialogue. *33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, 1733–1738 (Pasadena, CA, USA: IEEE).

Rizoiu MA, Lee Y, Mishra S, Xie L (2017) Hawkes processes for events in social media. *Frontiers of Multimedia Research*, 191–218 (Association for Computing Machinery and Morgan & Claypool).

Rizoiu MA, Mishra S, Kong Q, Carman M, Xie L (2018) SIR-Hawkes: Linking epidemic models and Hawkes processes to model diffusions in finite populations. *Proceedings of the 2018 World Wide Web Conference*, 419–428, WWW '18 (Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee).

Roddy M, Harte N (2020) Neural generation of dialogue response timings. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2442–2452 (online: Association for Computational Linguistics).

Sanh V, Debut L, Chaumond J, Wolf T (2019) DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Shahout R, Malach E, Liu C, Jiang W, Yu M, Mitzenmacher M (2025) Don't stop me now: Embedding based scheduling for LLMs. *The 13th International Conference on Learning Representations*, ICLR'25 (Singapore: Curran Associates, Inc.).

Spencer J, Sudan M, Xu K (2014) Queueing with future information. *ACM SIGMETRICS Performance Evaluation Review* 41(3):40–42.

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

Strubell E, Ganesh A, McCallum A (2019) Energy and policy considerations for deep learning in NLP. Korhonen A, Traum D, Màrquez L, eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–3650 (Florence, Italy: Association for Computational Linguistics).

Tezcan T, Zhang J (2014) Routing and staffing in customer service chat systems with impatient customers. *Operations Research* 62(4):943–956.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. *Proceedings of the 31st Conference on Neural Information Processing Systems*, 6000–6010, NIPS'17 (Long Beach, CA, USA: Curran Associates Inc.).

Xu K, Chan CW (2016) Using future information to reduce waiting times in the emergency department via diversion. *Manufacturing & Service Operations Management* 18(3):314–331.

Yom-Tov GB, Yedidsion L, Xie Y (2020) An invitation control policy for proactive service systems: Balancing efficiency, value and service level. *Manufacturing & Service Operations Management* 23(5):1077–1095.

EC.1. Summary Statistics of Data and Hawkes Models Estimated Parameters

Table EC.1 provides summary statistics of our data.

Table EC.1 Summary Statistics at the Message and Conversation Levels.

Variable	Mean	SD	50pct	75pct	Max
Conversation-level					
Total messages per conversation	8.41	5.72	8	11	176
Customer messages per conversation	3.77	3.51	3	5	158
Agent messages per conversation	4.65	2.67	4	6	36
Total conversation duration (seconds)	414.2	314.5	333.46	528.97	5,456.10
Inter-message gap (seconds)[a]	48.02	68.86	23.87	54.46	2,907.46
Customer response time (seconds)[b]	36.62	45.11	23.21	43.61	1,422.68
Agent response time (seconds)[c]	56.09	74.72	29.08	67.51	1,468.50
Message-level					
Words per message	133.9	79.8	120	167	1,892
System-level					
Mean concurrency per hour of the day[d]	2.20	0.76	1.96	2.43	4.14

Notes. (a) Time between consecutive messages within the same conversation. (b) Time from an agent message to the next customer message. (c) Time from a customer message to the next agent message. (d) Concurrency is the number of simultaneous conversations handled by an agent per hour of the day.

Table EC.2 provides the Hawkes models parameters estimated with $\mathcal{U}_{\text{train}}$.

Table EC.2 Estimated Hawkes-family parameters.

Model	Parameter	Estimate
UHP	α	14.33
	β	16.04
IUHP-Exp	α	24.31
	β	39.23
	λ_0	112.72
IUHP-Gamma	α	4.56
	ρ	0.62
	θ	0.88
	λ_0	34.52
BHP	α_{11}	4.22
	β_{11}	55.48
	α_{12}	35.25
	β_{12}	3.81
	α_{21}	28.25
	β_{21}	80.47
	α_{22}	46.30
	β_{22}	22.71

EC.2. Estimating the Percentage of Premature Closure using Positive Uncertain (PU) learning

Our analysis relies on real conversational service data from a food-delivery service organization. A central challenge is that data logs do not directly indicate whether a conversation ended successfully or was closed prematurely. While we carefully construct labeled cohorts using text analytics with manual verification (see Section 3), only a small fraction of true premature closures are explicitly observed in the data—namely, those followed by a verified customer return with a complaint within 60 minutes. As a result, a potentially large share of premature closures remains unlabeled.

Conversations that reappear as complaint-returns provide a conservative lower bound on the prevalence of premature closures, but do not reveal their full magnitude. The data therefore consist of positively labeled instances $\mathcal{P}$ (premature closures with verified complaint-return), a limited set of negatively labeled instances $\mathcal{N}$ (successful closures), and a large pool of unlabeled conversations $\mathcal{U}$. To estimate the prevalence of premature closures, we turn to PU learning, which is designed for precisely such settings.

PU learning enables the estimation of both the class prior (the true prevalence of premature closure) and the labeling probability under mild assumptions, such as the Selected Completely At Random (SCAR) (Elkan and Noto 2008, Bekker and Davis 2020). This allows us to quantify the extent of unobserved "silent" premature closures.

Formally, let $V_i \in \{0,1\}$ indicate whether conversation i was prematurely closed (latent ground truth), and let $\bar{V} = \sum_{i=1}^{M} V_i$ denote the (unobserved) total number of premature closures. Let $S_i \in \{0,1\}$ denote a complaint indicator, where $S_i = 1$ if conversation i is followed within 60 minutes by a complaint-return conversation j with $C_j = 1$. Let $\bar{S} = \sum_{i=1}^{M} S_i$ denote the number of *observed positives* (certified premature closures), and let $\bar{C} = \sum_{i=1}^{M} C_i$ denote the total number of *complaint-return conversations* (the returns chats themselves).

The ideal case of having unique customer identifier. When unique customer identifiers are available and complaint-return conversations are de-duplicated to the first qualifying return per predecessor, each complaint-return certifies exactly one predecessor conversation. In this ideal setting, the number of complaint-return conversations equals the number of observed positives, and thus $\bar{C} = \bar{S}$.

Class prior. Let $\pi = \Pr(V = 1) = \bar{V}/M$ denote the (unknown) prevalence of premature closures in the population.

Labeling probability. Let $\delta = \Pr(S = 1 \mid V = 1)$ denote the probability that a prematurely closed conversation is observed as such via a complaint-return. Then,

$$\mathrm{E}[\bar{S}] = \delta \bar{V} \quad \Rightarrow \quad \mathrm{E}\left[\frac{\bar{S}}{M}\right] = \delta\,\pi.$$

Lower bound. Every observed positive, S_i, corresponds to a true premature closure, V_i, whereas some premature closures remain unlabeled. Hence $\bar{S} \leq \bar{V}$, implying $\bar{S}/M \leq \bar{V}/M = \pi$. Therefore, $\bar{S}/M$ provides a conservative lower bound on true prevalence π. Moreover, if a lower bound $\delta_{\min}$ on the labeling probability is available, then

$$\frac{\bar{S}}{M} \leq \pi \leq \frac{\bar{S}}{\delta_{\min} M}.$$

Calibration set. We construct a manually verified calibration set P, consisting of $|P| = 96$ premature-closure conversations with verified complaint.

PU prevalence. In the data, unique customer identifiers are not available; consequently, neither the true number of premature closures $\bar{V}$ nor the indicator S_i is directly observable. What *is* observable is the number of complaint-return conversations, denoted by $\widetilde{C}$, which can serve as an estimator for $\bar{C}$. We found $\widetilde{C} = 436$ such conversations. Under the assumption that (i) the prematurely closed predecessor of each complaint-return lies within the analysis window and (ii) each complaint-return certifies at least one premature closure, we have $\widetilde{C} \leq \bar{V}$, implying $\widetilde{C}/M = 0.0062 \leq \pi$. This bound implies that the true prevalence of premature closures is substantially larger than what is directly observed (96).

To obtain verified positives, we manually constructed a set $\mathcal{P}$ of premature closures by identifying complaint-return conversations that explicitly reference specific items or locations, enabling reliable linkage to a predecessor conversation within a 60-minute window. This procedure yields $|P| = 96$ conversations. Since $P \subseteq \{i : V_i = 1\}$, we obtain the conservative bound $\max\Big(|P|/M, \widetilde{C}/M\Big) \leq \pi$.

Let $\mathcal{U}$ denote the set of unlabeled conversations. Under the PU framework, $|P| = \delta \pi M$. Rearranging this equation so that $\delta = |P|/\pi M$, and inserting the lower bound $\pi \geq \widetilde{C}/M$, we obtain an upper bound $\delta^{max} \leq 96/436 \approx 0.22$.

To estimate δ and π, we follow standard PU-learning practice. We train a probabilistic classifier $g(x) \approx \Pr(S = 1 \mid x)$ to distinguish conversations in cohort $\mathcal{P}$ from ones in cohort $\mathcal{U}$ using covariates x. Under the Selected Completely At Random (SCAR) assumption, in which it is assumed that every true positive is labeled with the same probability,

$$g(x) = \Pr(S = 1 \mid x) = \delta \Pr(V = 1 \mid x).$$

We estimate δ as the average of $g(x)$ over held-out observations from cohort $\mathcal{P}$, and estimate π as the average of $g(x)/\delta$ over $\mathcal{U}$, using 6-fold cross-validation.

This procedure yields $\hat{\delta} \approx 0.007$ and $\hat{\pi} \approx 0.19$ ($SE \approx 0.056$), which is within the range we identified. These results indicate that a substantial fraction of premature closures remain silent and don't explicitly complain when reconnecting.

EC.3. Bivariate Hawkes Process

The bivariate Hawkes process (BHP) cluster model with two event types and exponential self- and cross-excitation kernels is defined as follows. For conversation i, let $\{T_{i,k}^{(1)}\}_{k\geq 1}$ and $\{T_{i,k}^{(2)}\}_{k\geq 1}$ denote the timestamps of agent and customer messages, respectively, and let $\mathcal{H}_{i,t}$ denote the joint history of both types up to time t.

Specifically, for $t > 0$, the intensity rate functions are

$$\lambda_{i,1}(t \mid \mathcal{H}_{i,t}) = \sum_{k:\, T_{i,k}^{(1)} < t} \alpha_{11}\, e^{-\beta_{11}(t - T_{i,k}^{(1)})} + \sum_{k:\, T_{i,k}^{(2)} < t} \alpha_{12}\, e^{-\beta_{12}(t - T_{i,k}^{(2)})},$$

$$\lambda_{i,2}(t \mid \mathcal{H}_{i,t}) = \sum_{k:\, T_{i,k}^{(1)} < t} \alpha_{21}\, e^{-\beta_{21}(t - T_{i,k}^{(1)})} + \sum_{k:\, T_{i,k}^{(2)} < t} \alpha_{22}\, e^{-\beta_{22}(t - T_{i,k}^{(2)})}.$$

Here $\alpha_{rs} \geq 0$ denotes the jump in the type-r intensity induced by a type-s event, and $\beta_{rs} > 0$ is the exponential decay rate of that influence (self-excitation when $r = s$ and cross-excitation when $r \neq s$).

EC.4. Metrics Pooled Across Bins

Table EC.3–EC.4 provides the pooled performance measures for the short-term conversation continuation prediction task shown in Figures 2–4 and 7 for Hawkes, MLP, LSTM, and LLM, respectively.

Table EC.3 **Pooled performance (Short-Term Conversation Continuation, $\Delta = 60$s) by evaluation cohort: Hawkes models.**

Model	Cohort	Prevalence	ROC–AUC [95% CI]	PR–AUC [95% CI]
UHP	$\mathcal{P}$	0.569	0.561 [0.531, 0.588]	0.626 [0.593, 0.657]
	$\mathcal{N}$	0.673	0.543 [0.536, 0.549]	0.719 [0.713, 0.726]
	$\mathcal{U}_{\text{test}}$	0.618	0.547 [0.538, 0.554]	0.661 [0.652, 0.669]
IUHP-Exp	$\mathcal{P}$	0.569	0.630 [0.604, 0.654]	0.682 [0.650, 0.711]
	$\mathcal{N}$	0.673	0.638 [0.632, 0.643]	0.770 [0.764, 0.775]
	$\mathcal{U}_{\text{test}}$	0.618	0.639 [0.632, 0.645]	0.728 [0.721, 0.734]
IUHP-Gamma	$\mathcal{P}$	0.569	0.629 [0.599, 0.651]	0.680 [0.648, 0.707]
	$\mathcal{N}$	0.673	0.638 [0.633, 0.644]	0.770 [0.765, 0.775]
	$\mathcal{U}_{\text{test}}$	0.618	0.639 [0.632, 0.646]	0.728 [0.722, 0.735]

EC.5. Brier Score for End-of-Service Evaluation

In Figure 1(a), we report the Brier score for the end-of-service prediction task ($\Delta = \infty$) as a function of offset time, shown separately for $\mathcal{N}$ cohort (left panel) and $\mathcal{P}$ cohort (right panel). Lower Brier scores indicate better probabilistic accuracy. Checkpoints with negative offsets correspond to evaluations conducted before the last message, where the arrival of at least one additional message is guaranteed, whereas checkpoints with positive offsets correspond to evaluations conducted after the last message.

Table EC.4 Pooled performance (Short-Term Conversation Continuation, $\Delta = 60$s) by evaluation cohort: LLMs.

Model	Cohort	Prevalence	ROC–AUC [95% CI]	PR–AUC [95% CI]
DistilBERT	$\mathcal{P}$	0.569	0.630 [0.599, 0.658]	0.680 [0.644, 0.715]
	$\mathcal{N}$	0.673	0.665 [0.659, 0.672]	0.780 [0.772, 0.788]
	$\mathcal{U}_{\text{test}}$	0.618	0.657 [0.649, 0.665]	0.727 [0.718, 0.737]
Phi-3	$\mathcal{P}$	0.565	0.404 [0.383, 0.424]	0.559 [0.528, 0.587]
	$\mathcal{N}$	0.669	0.386 [0.381, 0.391]	0.639 [0.633, 0.646]
	$\mathcal{U}_{\text{test}}$	0.614	0.389 [0.384, 0.395]	0.591 [0.583, 0.599]
Llama-3	$\mathcal{P}$	0.569	0.443 [0.413, 0.469]	0.529 [0.494, 0.570]
	$\mathcal{N}$	0.673	0.399 [0.392, 0.406]	0.610 [0.602, 0.617]
	U_{test}	0.618	0.400 [0.392, 0.407]	0.555 [0.546, 0.564]

Table EC.5 Pooled performance (Short-Term Conversation Continuation, $\Delta = 60$s) by evaluation cohort: MLP models.

Model	Cohort	Prevalence	ROC–AUC [95% CI]	PR–AUC [95% CI]
Version 1: Time+Words+Sender+UHP intensity (λ)	$\mathcal{P}$	0.569	0.572 [0.546, 0.600]	0.645 [0.617, 0.676]
	$\mathcal{N}$	0.673	0.553 [0.547, 0.560]	0.735 [0.729, 0.741]
	$\mathcal{U}_{\text{test}}$	0.618	0.570 [0.561, 0.577]	0.691 [0.684, 0.699]
Version 2: Time+Words+Sender	$\mathcal{P}$	0.569	0.549 [0.521, 0.580]	0.626 [0.595, 0.659]
	$\mathcal{N}$	0.673	0.539 [0.532, 0.545]	0.724 [0.718, 0.730]
	$\mathcal{U}_{\text{test}}$	0.618	0.557 [0.549, 0.565]	0.679 [0.672, 0.687]
Version 3: Time+Position	$\mathcal{P}$	0.569	0.562 [0.533, 0.591]	0.639 [0.611, 0.672]
	$\mathcal{N}$	0.673	0.546 [0.539, 0.553]	0.735 [0.729, 0.741]
	$\mathcal{U}_{\text{test}}$	0.618	0.565 [0.557, 0.572]	0.692 [0.684, 0.700]
Version 4: Time+Words	$\mathcal{P}$	0.569	0.451 [0.418, 0.481]	0.543 [0.508, 0.579]
	$\mathcal{N}$	0.673	0.467 [0.461, 0.473]	0.668 [0.659, 0.676]
	$\mathcal{U}_{\text{test}}$	0.618	0.450 [0.442, 0.458]	0.604 [0.595, 0.616]
Version 5: Time+Words+Concurency	$\mathcal{P}$	0.569	0.565 [0.538, 0.595]	0.639 [0.610, 0.672]
	$\mathcal{N}$	0.673	0.549 [0.542, 0.556]	0.734 [0.728, 0.740]
	$\mathcal{U}_{\text{test}}$	0.618	0.570 [0.561, 0.577]	0.693 [0.686, 0.701]
Version 6: Position only	$\mathcal{P}$	0.569	0.540 [0.507, 0.574]	0.610 [0.579, 0.648]
	$\mathcal{N}$	0.673	0.502 [0.495, 0.509]	0.684 [0.676, 0.691]
	$\mathcal{U}_{\text{test}}$	0.618	0.506 [0.498, 0.514]	0.631 [0.623, 0.639]
Version 7: Time only	$\mathcal{P}$	0.569	0.555 [0.528, 0.583]	0.636 [0.610, 0.667]
	$\mathcal{N}$	0.673	0.548 [0.541, 0.555]	0.737 [0.731, 0.743]
	$\mathcal{U}_{\text{test}}$	0.618	0.570 [0.562, 0.577]	0.698 [0.691, 0.706]

Note. Best seven models.

EC.6. Proof of Proposition 1

Proof. After taking the derivative with respect to η and simplifying, the first order condition (FOC) of the prediction combination objective in (5) is given by

$$\mathcal{C}e^{\frac{2\eta}{\beta}} + \frac{\eta}{\beta} = 1.$$

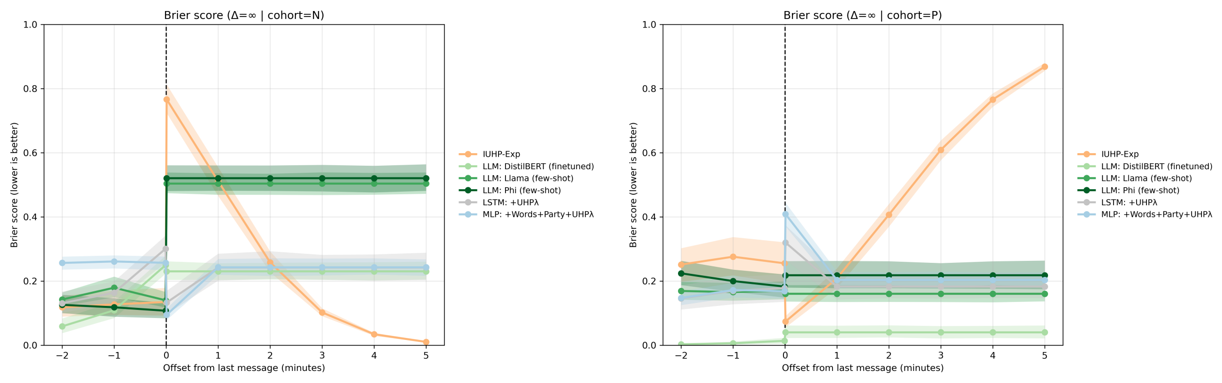
Clearly, the left-hand side of the FOC is strictly increasing in η whereas the right-hand side is constant, and thus there is at most one solution to this equation. Letting $x = \eta/\beta$ for simplicity in notation, this FOC can be re-arranged to

$$2\mathcal{C}e^2 = 2(1-x)e^{2(1-x)}, \tag{EC.1}$$

Table EC.6 Pooled performance (Short-Term Conversation Continuation, $\Delta = 60$s) by evaluation cohort: LSTM models.

Model	Group	Prevalence	ROC–AUC [95% CI]	PR–AUC [95% CI]
Version 1: Position only	$\mathcal{P}$	0.569	0.559 [0.533, 0.585]	0.634 [0.609, 0.663]
	$\mathcal{N}$	0.673	0.603 [0.597, 0.609]	0.755 [0.749, 0.760]
	$\mathcal{U}_{\text{test}}$	0.618	0.597 [0.591, 0.604]	0.699 [0.691, 0.708]
Version 2: Time only	$\mathcal{P}$	0.569	0.572 [0.544, 0.600]	0.644 [0.618, 0.674]
	$\mathcal{N}$	0.673	0.611 [0.605, 0.616]	0.765 [0.759, 0.772]
	$\mathcal{U}_{\text{test}}$	0.618	0.605 [0.599, 0.613]	0.710 [0.702, 0.719]
Version 3: Time+Position	$\mathcal{P}$	0.569	0.572 [0.544, 0.601]	0.646 [0.617, 0.675]
	$\mathcal{N}$	0.673	0.608 [0.601, 0.613]	0.763 [0.756, 0.769]
	$\mathcal{U}_{\text{test}}$	0.618	0.603 [0.597, 0.610]	0.709 [0.701, 0.717]
Version 4: Time+Words+Sender	$\mathcal{P}$	0.569	0.602 [0.576, 0.627]	0.658 [0.629, 0.687]
	$\mathcal{N}$	0.673	0.638 [0.632, 0.644]	0.775 [0.770, 0.782]
	$\mathcal{U}_{\text{test}}$	0.618	0.640 [0.633, 0.646]	0.727 [0.719, 0.734]
Version 5: Time+Words+Sender+Concurrency	$\mathcal{P}$	0.569	0.600 [0.576, 0.624]	0.660 [0.632, 0.689]
	$\mathcal{N}$	0.673	0.638 [0.632, 0.644]	0.776 [0.770, 0.782]
	$\mathcal{U}_{\text{test}}$	0.618	0.641 [0.634, 0.649]	0.729 [0.721, 0.738]
Version 6: Time+UHP intensity (λ)	$\mathcal{P}$	0.569	0.599 [0.572, 0.625]	0.661 [0.632, 0.689]
	$\mathcal{N}$	0.673	0.635 [0.629, 0.641]	0.776 [0.770, 0.783]
	$\mathcal{U}_{\text{test}}$	0.618	0.640 [0.633, 0.647]	0.727 [0.718, 0.735]
Version 7: Time+IUHP-Gamma intensity (λ)	$\mathcal{P}$	0.569	0.594 [0.568, 0.620]	0.657 [0.627, 0.686]
	$\mathcal{N}$	0.673	0.631 [0.625, 0.636]	0.773 [0.767, 0.779]
	$\mathcal{U}_{\text{test}}$	0.618	0.639 [0.631, 0.645]	0.727 [0.719, 0.735]

Note. Best seven models.



(a) Cohort $\mathcal{N}$ (b) Cohort $\mathcal{P}$

Figure EC.1 End-of-Service Task ($\Delta = \infty$): Brier Score as a Function of Offset Time.

which follows by first subtracting x from both sides and then multiplying both sides by $2e^{2-2x}$. At this point we can recognize that there will be no $x > 0$ which solves the FOC if $\mathcal{C} \geq 1$, as the right-hand side of (EC.1) is strictly decreasing in x and thus $2(1-x)e^{2(1-x)} \leq 2e^2$.

Now, taking $C < 1$. the right hand side is ready for us to apply the Lambert-W, and, by the hallmark identity of that function, we can further simplify the expression to

$$\mathsf{W}_0\left(2\mathcal{C}e^2\right) = \mathsf{W}_0\left(2(1-x)e^{2(1-x)}\right) = 2(1-x).$$

Finally, solving for x, we achieve the stated η^* via $\eta^* = \beta x$. $\square$