A stochastic model-based approach to online event prediction and response scheduling

Marco Biagi**, Laura Carnevali, Marco Paolieri,
Fulvio Patara, and Enrico Vicario

Department of Information Engineering, University of Florence, Italy

Abstract. In a variety of contexts, time-stamped and typed event logs enable the construction of a stochastic model capturing the sequencing and timing of observable discrete events. This model can serve various objectives including: diagnosis of the current state; prediction of its evolution over time; scheduling of response actions. We propose a technique that supports online scheduling of actions based on a prediction of the model state evolution: the model is derived automatically by customizing the general structure of a semi-Markov process so as to fit the statistics of observed logs; the prediction is updated whenever any observable event changes the current state estimation; the (continuous) time point of the next scheduled action is decided according to policies based on the estimated distribution of the time to given critical states. Experimental results are reported to characterize the applicability of the approach with respect to general properties of the statistics of observable events and with respect to a specific reference dataset from the context of Ambient Assisted Living.

1 Introduction

The prediction of future system events from past ones is a challenging and general problem that finds several applications (e.g., *autonomic computing* [1,9], *models at runtime* [2]), especially in online settings where predictions can be updated several times after observing new data [15].

Model-based approaches have also been proposed in other application areas such as Ambient Assisted Living (AAL). An important goal in AAL is to recognize human activities from smart home sensor data. Common activities of interest are Activities of Daily Living (ADLs) such as "bathing", "sleeping", "dinner"; home appliances and furniture can generate sensor events indicating, for example, the use of a faucet, the opening of a door, or the use of a light switch. The problem of assessing the current human activity, also known as *diagnosis*, has been investigated in [16,4] through the use of hidden stochastic models.

In this work, we are interested in the problem of *event prediction* and *response scheduling*: given a sequence of past events, each with a type and a timestamp, our goal is to select a time point for the activation of a response action. A

** Corresponding author: `marco.biagi@unifi.it`

response action can, for example, replace a hardware component to prevent failures; in AAL, the response action can be a reminder about the intake of some drug before the beginning of a meal, or surveillance escalation when the user is going to use some dangerous device.

To this end, we model the system under analysis as a Semi-Markov Process (SMP) where states represent activities or *idle times* between activities. For each state, the sojourn time distribution and transition probabilities are estimated from a supervised dataset of event logs. Different parametric models are compared for sojourn time distributions, in order to evaluate the reduction in the prediction error due to more accurate models; in particular, we consider fitting the mean of the sojourn time with an exponentially distributed (EXP) or Erlang random variable, or fitting its mean and variance using a shifted EXP random variable, or sums and mixtures of EXP random variables [17]. After fitting the model parameters from a training dataset, we use a test set of events to assess the prediction error of scheduling policies for the response action. After each event of the test set, the current state and elapsed sojourn time (i.e., a diagnosis) are used as initial conditions to compute first-passage probabilities for critical states (e.g., dinner). Scheduling policies analyze this transient information to select an actuation time for the response action.

Related work. In [4], we proposed a model-driven online approach to the problem of diagnosis: a model of feasible behaviors, enhanced with stochastic parameters derived from a dataset [13], was used to evaluate a measure of likelihood of the current ongoing activity, which is updated after each event. A different approach to diagnosis is considered in [16], where hidden semi-Markov processes are used to detect the current user activity from recent events; in [15], hidden semi-Markov processes are used to predict and prevent imminent failures; in [7], Markov-modulated Poisson processes are proposed to detect anomalous patterns. In [12], a short-term prediction problem is considered: the next state is predicted by combining different information sources through Dempster-Shafer theory. Our work takes a different approach: instead of matching the sequence of recent events to determine the current or next state, we use a state diagnosis as the initial condition to analyze the transient evolution of a semi-Markov process and derive first-passage probabilities of critical states using the ORIS Tool [6,3]. This provides fine-grained information for policies that schedule response actions.

Organization. In Section 2, we formulate the prediction and scheduling problem; in Section 3, we define the structure, fitting, and analysis technique for the SMP model adopted by our solution. In Section 4, we evaluate the prediction error of the approach, both on synthetic datasets and on a real-world dataset of ADLs. Finally, we draw our conclusions in Section 5.

2 Problem formulation

We consider an online setting where the system receives a stream of *events* (e.g., sensor readings), each with a type and a timestamp; after receiving an event, the system has the opportunity to request or update the scheduling of a *response action* (e.g., surveillance escalation or a user reminder), to be activated after a delay if no other event is received. The goal is to intercept a class of *target activities* (e.g., a security attack, a hardware fault, the dinner activity or other ADLs) with the response action. The response action is active for a fixed *response duration* $t_d \geq 0$: target activities are successfully handled by the system if they start while the response action is active. In order to cope with real-world applications, we also introduce a fixed *response actuation time* $t_w \geq 0$: the response action must be issued by the system t_w time units before its activation time; after being issued, response actions cannot be canceled. For example if a hardware failure is predicted in the future, a maintenance operation must be scheduled t_w time units before it happens, so as to prevent the failure of the system.

In order to learn the relationship between input events and target activities, we are given a supervised dataset $\mathcal{D}$ including: *i)* a recorded sequence of events $e_1, e_2, \ldots, e_m$ where each event $e_i = \langle \varepsilon_i, t_i \rangle$ for $i = 1, \ldots, m$ includes an event type ε_i from a finite set $\mathcal{E} = \{\texttt{evt}_1, \ldots, \texttt{evt}_M\}$ and a timestamp $t_i \geq 0$, with $t_1 \leq \cdots \leq t_m$; *ii)* the sequence of activities $a_1, a_2, \ldots, a_n$ performed during the generation of the events; each activity $a_i = \langle \alpha_i, \tau_i, \delta_i \rangle$ for $i = 1, \ldots, n$ includes an activity type α_i from a finite set $\mathcal{A} = \{\texttt{act}_1, \ldots, \texttt{act}_N\}$, a start time $\tau_i \geq 0$ (such that $\tau_1 \leq \cdots \leq \tau_n$) and an activity duration $\delta_i \geq 0$. As in [4,16], the underlying assumption of this problem is that distinct activities take place in the system and a statistical characterization is derived taking into account: *i)* the duration of each activity; *ii)* the time between subsequent events; *iii)* the types of event occurred within each activity; *iv)* the transition probabilities among activities. The sequence of activities thus provides a "ground truth" to learn the relationship between events and activities, and the likelihood of different activity sequences.

In order to evaluate the online performance of a scheduling system $f_{\mathcal{D}}$ trained from $\mathcal{D}$, we define the metrics of *precision* and *recall* for a given test sequence of events $e_1, e_2, \ldots, e_h$, an actuation time $t_w \geq 0$, and a response duration $t_d \geq 0$. Let $\tilde{t}$ denote the time point at which the system is currently scheduled to issue the activation of the response action, which will be active in the time window $[\tilde{t} + t_w, \tilde{t} + t_w + t_d]$. Initially, no response activation is scheduled, i.e., $\tilde{t} = \infty$; for each event $e_i = \langle \varepsilon_i, t_i \rangle$ in the sequence $e_1, e_2, \ldots, e_h$: *i)* if $\tilde{t} \leq t_i$, the activation of a response action is issued at time $\tilde{t}$; *ii)* the time point $\tilde{t}$ is updated by $f_{\mathcal{D}}$ according to the new information provided by event e_i, i.e., $\tilde{t} \leftarrow f_{\mathcal{D}}(e_1, e_2, \ldots, e_i; t_w, t_d)$. Let $I_1, I_2, \ldots, I_p$ denote the time windows of executed response actions and let $\overline{\tau}_1, \overline{\tau}_2, \ldots, \overline{\tau}_q$ denote the start times of target activities in the ground truth. We say that the system has produced: *i)* a *true positive*, if an activation window contained the start time of (at least) a target activity, i.e., $\mathrm{TP} = |\{i \leq p \mid \exists j \leq q \text{ such that } \overline{\tau}_j \in I_i\}|$; *ii)* a *false positive*, if an activation window did not contain any target activity, i.e., $\mathrm{FP} = |\{i \leq p \mid \overline{\tau}_j \notin I_i \ \forall j \leq q\}|$ *iii)* a *false negative*, if a target activity was not contained in any activation window, i.e., $\mathrm{FN} = |\{j \leq q \mid \overline{\tau}_j \notin I_i \ \forall i \leq p\}|$ Then, we define (as usual in information retrieval) *precision* $= \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$ and *recall* $= \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$.

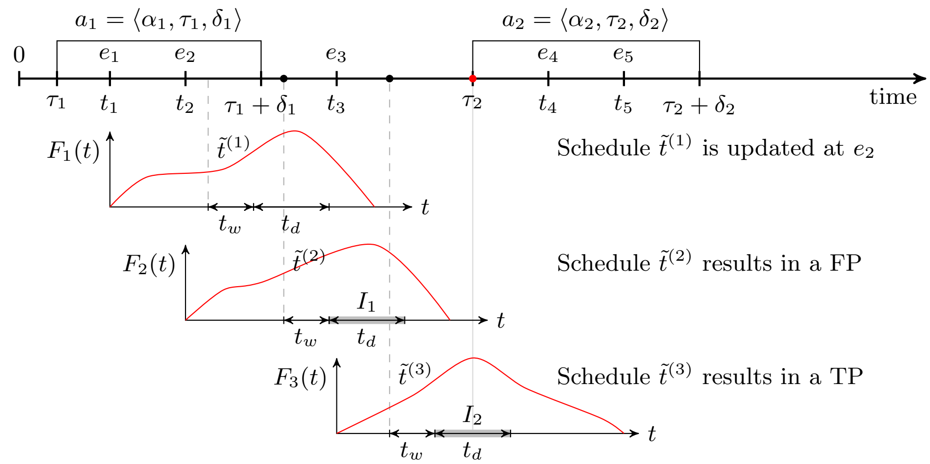


Fig. 1. A sample execution of the online scheduling system: α_2 is the target activity; the start of response actions is issued at $\tilde{t}^{(2)}$ and $\tilde{t}^{(3)}$, producing the activity windows I_1 and I_2 (a false positive and a true positive, respectively).

Fig. 1 illustrates the mechanism of online event prediction and response scheduling on the sequences of events e_1, e_2, e_3, e_4, e_5 and activities a_1, a_2. The type $\alpha_2 \in \mathcal{A}$ of $a_2 = \langle \alpha_2, \tau_2, \delta_2 \rangle$ is the target activity: after event e_1, the system uses a metric $F_1(t)$ to select the best activation window of duration t_d for the response action; the activation command is first scheduled at time $\tilde{t}^{(1)}$. Since event e_2 is received at time $t_2 < \tilde{t}^{(1)}$, a new metric $F_2(t)$ and schedule $\tilde{t}^{(2)}$ is computed; given that no event is received before $\tilde{t}^{(2)}$, the response action is issued, resulting in the activation window $I_1 = [\tilde{t}^{(2)} + t_w, \tilde{t}^{(2)} + t_w + t_d]$. This response is a "false positive", because it does not intercept the start of any target activity α_2. At event e_3, the metric $F_3(t)$ and schedule $\tilde{t}^{(3)}$ are computed. As $t_4 > \tilde{t}^{(3)}$ (i.e., event e_4 is received after $\tilde{t}^{(3)}$), the response action is activated during the time interval $I_2 = [\tilde{t}^{(3)} + t_w, \tilde{t}^{(3)} + t_w + t_d]$. The interval I_2 contains the start time τ_2 of a target activity, resulting in a "true positive" response.

Our formulation is similar to that of [14] for the problem of online failure prediction, as it considers both an actuation time t_w and a response duration t_d, which are constant parameters of the specific application. Nonetheless, in contrast to [14], it requires that the system selects the time point $\tilde{t}$ that corresponds to the best schedule for the activation window $[\tilde{t} + t_w, \tilde{t} + t_w + t_d]$.

3 A model-based solution

3.1 System architecture

We propose a solution based on a stochastic model capturing the statistics of events and activities observed in the training dataset. The model allows us to estimate the current state and predict the occurrence of target activities.

We identify four distinct modules required in this approach.

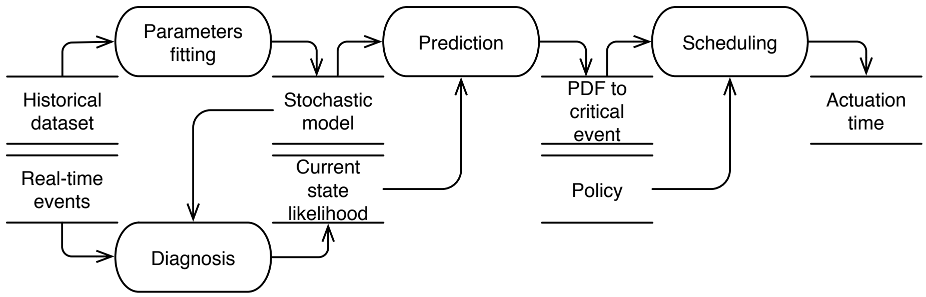


Fig. 2. Data flow diagram of the proposed solution: events are processed in real-time to assess the current state and evaluate first-passage probabilities of a set of critical activities; response actions are then scheduled based on a given policy.

1. *Fitting.* The input dataset $\mathcal{D}$ of events and activities is analyzed in order to fit its relevant statistics with a semi-Markov process. The state space $S = \mathcal{A} \cup \mathcal{I}$ of the process includes the activities $\mathcal{A} = \{\mathtt{act}_1, \ldots, \mathtt{act}_N\}$ observed in $\mathcal{D}$ and idle states $\mathcal{I} = \{\mathtt{idle}_{xy} \mid (x, y) \in \mathcal{A} \times \mathcal{A}\}$ between each pair (x, y) of activities. We use parametric continuous-time models to fit mean value and coefficient of variation (CV) of the duration of each activity, of the inter-time between events, and of the idle times between activities; transition probabilities among activities and event selection are modeled with discrete distributions.

2. *Diagnosis.* As presented in [4], the hidden semi-Markov process of event generation can be used to determine a measure of likelihood of *current state estimates* given a sequence of events observed in real-time by the system. Each estimate is a triplet $\langle \pi, x, R(t) \rangle$ where $\pi \in [0, 1]$ is a likelihood measure, $x \in S$ is a state of the semi-Markov process, and $R(t)$ is the PDF of the remaining sojourn time in x (estimated numerically on a grid of time points).

3. *Prediction.* From each estimate of the current state, we analyze the transient evolution of the semi-Markov process, computing first-passage transient probabilities for a set of target activities. First-passage probabilities computed for each current state estimate are then combined according to their likelihoods, in order to obtain the expected probability $F(t)$ of reaching some critical state within time t.

4. *Scheduling.* The first-passage probabilities $F(t)$ are analyzed according to a *policy* that selects a time point where the activation of the response action is scheduled. As input parameters of the problem, the policy considers the response actuation time $t_w \geq 0$, required to activate the response action, and the response duration $t_d \geq 0$.

Fig. 2 presents a data flow diagram illustrating the components of the system and the information exchanged as input and output. By decomposing the system architecture, we are able to isolate the responsibilities of the components and evaluate their individual performance. While [4] focused on the diagnosis of the current state, in this work we study the prediction and scheduling components. In Section 4, we will use an exact state diagnosis (from the ground truth of recorded activities) as input for the prediction phase; in so doing, we will decouple the performance of prediction and scheduling from that of state diagnosis.

3.2 Model definition

We model the system under analysis as a semi-Markov process where states represent activities or *idle times* between activities.

Definition 1 (Markov Renewal Sequence). *A sequence of random variables* $\{(X_n, T_n),\ n \in \mathbb{N}\}$ *such that, for all* $n \in \mathbb{N}$, X_n *takes values in a finite set* S, T_n *takes values in* $\mathbb{R}_{\geqslant 0}$, *and* $0 = T_0 \leq T_1 \leq T_2 \leq \cdots \leq T_n$, *is called* Markov renewal sequence *with state space* S *and* kernel $G_{ij}(t)$ *if and only if*

$$P\{X_{n+1} = j,\, T_{n+1} - T_n \leq t \mid X_n = i, X_{n-1}, \ldots, X_1, X_0, T_n, \ldots, T_1, T_0\} \\ = P\{X_1 = j,\, T_1 \leq t \mid X_0 = i\} := G_{ij}(t)$$

for all $n \in \mathbb{N}$, $i, j \in S$ *and* $t \in \mathbb{R}_{\geqslant 0}$.

The Markov renewal sequence is thus time-homogeneous and memoryless: given the current state $i \in S$, the kernel $G_{ij}(t)$ defines the joint distribution of the next *renewal time* T_1 and *regeneration state* $X_1 \in S$. From a Markov renewal sequence, a semi-Markov process can be constructed as follows.

Definition 2 (Semi-Markov Process). *Let* $\{(X_n, T_n),\ n \in \mathbb{N}\}$ *be a Markov renewal sequence with state space* S*; we define* semi-Markov process *the process* $\{X(t),\ t \geq 0\}$ *such that* $X(t) = X_n$ *for* $t \in [T_n, T_{n+1})$ *and all* $n \in \mathbb{N}$.

Given a training dataset of N activities $\mathcal{A} = \{\mathtt{act}_1, \ldots, \mathtt{act}_N\}$, we construct a semi-Markov process with state space $S = \mathcal{A} \cup \mathcal{I}$, where $\mathcal{I} = \{\mathtt{idle}_{xy} \mid (x, y) \in \mathcal{A} \times \mathcal{A}\}$ is the set of "idle states" between each pair (x, y) of activities. The state of the system evolves as follows: *i)* the system randomly selects a sojourn time in the current activity $\mathtt{act}_x \in \mathcal{A}$ and the successive idle state $\mathtt{idle}_{xy} \in \mathcal{I}$ for some $y \in \mathcal{A}$; *ii)* a sojourn time is selected for the state $\mathtt{idle}_{xy}$; *iii)* from $\mathtt{idle}_{xy}$, the model moves to the state corresponding to $\mathtt{act}_y \in \mathcal{A}$. This structure of the semi-Markov process is reflected in the definition of its kernel

$$G_{ij}(t) := \begin{cases} H_x(t)\, p_{xy} & \text{if } i = \mathtt{act}_x \in \mathcal{A} \text{ and } j = \mathtt{idle}_{xy} \in \mathcal{I} \text{ for some } y \in \mathcal{A}, \\ D_{xy}(t) & \text{if } i = \mathtt{idle}_{xy} \in \mathcal{I} \text{ and } j = \mathtt{act}_y \in \mathcal{A}, \\ 0 & \text{otherwise,} \end{cases}$$

where: $H_x(t)$ is the distribution of the sojourn time in $\mathtt{act}_x$; p_{xy} is the transition probability from activity $\mathtt{act}_x$ to activity $\mathtt{act}_y$, with $\sum_{y \in \mathcal{A}} p_{xy} = 1$ for all $x \in \mathcal{A}$; $D_{xy}(t)$ is the sojourn time distribution in the idle state $\mathtt{idle}_{xy}$ between $\mathtt{act}_x$ and $\mathtt{act}_y$. Figure 3 illustrates the structure of this semi-Markov model. Note that, in our definition, the transition probabilities p_{xy} do not depend on the sojourn

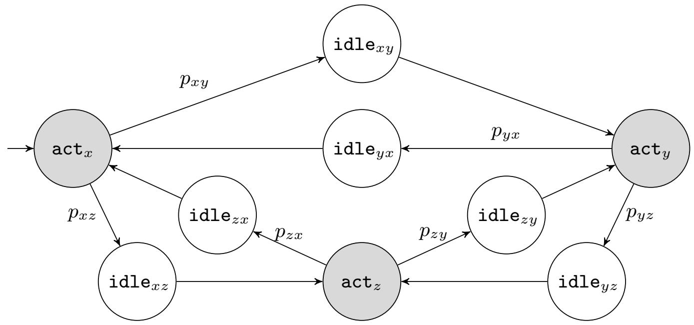


Fig. 3. Structure of the semi-Markov process for the activities $\mathcal{A} = \{\texttt{act}_x, \texttt{act}_y, \texttt{act}_z\}$.

time in the current activity. Nonetheless, the definition of the semi-Markov kernel $G_{ij}(t)$ and the prediction and scheduling approach described in the following can model this dependency; we adopt time-homogeneous transition probabilities p_{xy} due to the limited amount of data available for fitting in real-world applications.

3.3 Parameter fitting from a training dataset

The stochastic parameters of the model are automatically derived from the statistics of activities observed in the dataset. Specifically, we are interested in characterizing three aspects: (1) the sojourn time distribution $H_x(t)$ in each activity $\texttt{act}_x \in \mathcal{A}$; (2) the sojourn time distribution $D_{xy}(t)$ in each idle state $\texttt{idle}_{xy} \in \mathcal{I}$; and, (3) the transition probability p_{xy} from each activity $\texttt{act}_x$ to any other reachable activity $\texttt{act}_y$.

Let $a_i = \langle \alpha_i, \tau_i, \delta_i \rangle$ for $i = 1, \ldots, n$ be the sequence of activities in the training dataset, each including an activity type $\alpha_i \in \mathcal{A}$, a start time $\tau_i \geq 0$, and an activity duration $\delta_i \geq 0$.

For each activity $\texttt{act}_x \in \mathcal{A}$, we consider its observed durations $\delta(\texttt{act}_x) = \{\delta_i \mid 1 \leq i \leq n \text{ and } \alpha_i = \texttt{act}_x\}$ and estimate the sojourn time distribution $H_x(t)$ using one of three different strategies:

1. *Exp strategy.* Fitting the mean value μ of $\delta(\texttt{act}_x)$ with an exponentially distributed random variable with rate $\lambda = 1/\mu$.
2. *Erlang strategy.* Fitting the mean value μ of $\delta(\texttt{act}_x)$ with an Erlang random variable with shape parameter $k = 2$ and rate $\lambda = k/\mu$.
3. *Whitt strategy.* Fitting the mean value μ and coefficient of variation CV of $\delta(\texttt{act}_x)$ with the approach presented in [17], which requires: *i)* if $\mathrm{CV} \leq 1/\sqrt{2}$, a shifted exponential random variable with PDF $f(t) = \lambda e^{-\lambda(t-d)}$ over $[d, \infty)$, where $\lambda = 1/(\mu\,\mathrm{CV})$ and $d = \mu(1 - \mathrm{CV})$; *ii)* if $1/\sqrt{2} < \mathrm{CV} < 1$, a hypo-exponential random variable (sum of two exponential random variables) with PDF $f(t) = \lambda_1\lambda_2(e^{-\lambda_2 t} - e^{-\lambda_1 t})/(\lambda_1 - \lambda_2)$ with $\lambda_i = 1/[(\mu/2)(1 \pm \sqrt{2\,\mathrm{CV}^2 - 1})]$ for $i = 1, 2$; *iii)* if $\mathrm{CV} > 1$, a hyper-exponential random variable with PDF $f(t) = p_1\lambda_1 e^{-\lambda_1 t} + p_2\lambda_2 e^{-\lambda_2 t}$, where $p_i = [1 \pm \sqrt{\frac{CV^2-1}{CV^2+1}}]/2$ and $\lambda_i = 2p_i/\mu$ for $i = 1, 2$.

We adopt the same strategies for the estimation of the sojourn time distribution $D_{xy}(t)$ of each idle state $\texttt{idle}_{xy} \in \mathcal{I}$; in this case, the mean value and coefficient of variation of the sojourn in $\texttt{idle}_{xy}$ are computed from the observed durations $\{\tau_{i+1} - (\tau_i + \delta_i) \mid 1 \leq i < n,\ \alpha_i = \texttt{act}_x \text{ and } \alpha_{i+1} = \texttt{act}_y\}$.

Finally, the transition probability p_{xy} that activity $\texttt{act}_y$ will follow $\texttt{act}_x$ is estimated as

$$p_{xy} = \frac{|\{1 \leq i < n \mid \alpha_i = \texttt{act}_x \text{ and } \alpha_{i+1} = \texttt{act}_y\}|}{|\{1 \leq i < n \mid \alpha_i = \texttt{act}_x\}|}$$

for each pair of activities $(\texttt{act}_x, \texttt{act}_y)$ in the training dataset.

3.4 Prediction

After receiving an event, a new set of state estimates $\langle \pi_u, x_u, R_u(t) \rangle$, $u = 1, \ldots, h$ is produced by the diagnosis component, each with a likelihood measure $\pi_u \in [0, 1]$, a candidate state $x_u \in S$, and a PDF $R_u(t)$ of the remaining sojourn time in x_u. The prediction component uses this information on the current state of the system in order to compute first-passage probabilities for a set of target activities using the semi-Markov model.

Let $S = \mathcal{A} \cup \mathcal{I}$ be the set of states of the SMP (i.e., activities and idle states) and let $G_{ij}(t)$ be its kernel: for all $i, j \in S$ and $t \geq 0$, $G_{ij}(t)$ gives the probability that the next state j is reached from i in a time lower or equal to t. We compute first-passage probabilities of a set of target activities $A \subset \mathcal{A}$ by solving a system of Markov renewal equations [11]

$$P_{ij}(t) = \Big(1 - \sum_{j \in S} G^A_{ij}(t)\Big)\delta_{ij} + \sum_{k \in S} \int_0^t dG^A_{ik}(x) P_{kj}(t - x)$$

for all $i, j \in S$ and $0 \leq t \leq t_{max}$, where t_{max} is the time bound of the analysis, $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ otherwise, and

$$G^A_{ij}(t) = \begin{cases} 0 & \text{if } i \in A, \\ G_{ij}(t) & \text{if } i \notin A, \end{cases}$$

effectively making any state in A an *absorbing state*. A numerical solution based on the trapezoidal rule [6] requires $O((\frac{t_{max}}{\Delta})^2)$ multiplications of $|S| \times |S|$ matrices, where Δ is the step size used in the discretization. The first-passage probabilities of states in A are then given, for each initial state $i \in S$, by $F_i(t) = \sum_{j \in A} P_{ij}(t)$.

We numerically compute $F_i(t)$ only once for a fixed time bound t_{max} and each $i \in S$. After a new event, the updated diagnosis $\langle \pi_u, x_u, R_u(t) \rangle$ for $u = 1, \ldots, h$ is used to compute

$$F(t) = \sum_{u=1}^{h} \pi_u \int_0^{t_{max}} R_u(x) \left(\sum_{j \in S} G_{x_u, j}(\infty)\, F_j(t - x) \right) dx$$

which accounts for each state x_u according to its likelihood π_u, remaining sojourn time PDF $R_u(x)$, transition probabilities $G_{x_u,j}(\infty)$ to the next state j, and first-passage probability $F_j(t)$ from j to a state in A.

3.5 Response scheduling

Given $F(t)$ and a policy, the scheduling component (Fig. 2) selects a response actuation time $\tilde{t}$. In this work, we consider a *maximum probability interval* policy. The policy considers a discrete grid of points $\mathcal{X}$ equispaced in $[t_w, t_{max} - t_d]$: for each $x \in \mathcal{X}$, the probability

$$v(x) = F(x + t_d) - F(x)$$

of reaching a target activity in the interval $[x, x + t_d]$ is evaluated. Let $x^* = \arg\max_{x \in \mathcal{X}} v(x)$. If $v(x^*) > \epsilon\, t_d$, then the activation of the response action is scheduled at time $\tilde{t} = x^* - t_w$; otherwise, no response action is scheduled.

4 Experimental evaluation

An experimentation was carried out to evaluate the system performance in terms of precision and recall metrics. We experimented both with synthetic datasets constructed so as to make evident how the type of the distribution can affect the prediction and scheduling performance, and with a real dataset [16] so as to evaluate the applicability of the proposed approach in a problem of Activity Recognition (AR) from the context of Ambient Assisted Living (AAL).

In both cases, an exact state diagnosis was assumed as input for the prediction phase, so as to focus on the evaluation of performance achieved by predictor and scheduler components.

Each dataset has been split into a test and a training set using a Leave One Day Out (LOO) approach, where one full day event log is used for testing and remaining days are exploited for training the stochastic model. All experiments were performed with a time bound $t_{max} = 3\,600$ s and a step size $\Delta = 1$ s.

4.1 Evaluation on synthetic datasets

In order to evaluate the predictor accuracy in a controlled manner, three synthetic datasets are generated through the simulation of a stochastic model of activities. As depicted in Fig. 4, the model is shaped as a semi-Markov process with 4 activities, i.e., $\mathcal{A} = \{\texttt{act}_0, \texttt{act}_1, \texttt{act}_2, \texttt{act}_3\}$, where $\texttt{act}_3$ represents the target critical activity.

On completion of any activity, a random switch $p_{\mathrm{x,y}}$ makes a selection to establish if the activity sequence must move back to the initial state $\texttt{act}_0$ or continue with the next activity. The random switch $p_{\mathrm{x,y}}$ is set equal to 0.5 for all choices, which represents a kind of worst case for predictability. In so doing, the set of idle states is derived as follow: $\mathcal{I} = \{\texttt{idle}_{0,0}, \texttt{idle}_{0,1}, \texttt{idle}_{1,0}, \texttt{idle}_{1,2}, \texttt{idle}_{2,0}, \texttt{idle}_{2,3}, \texttt{idle}_{3,0}\}$. In each state $S = \mathcal{A} \cup \mathcal{I}$, events occur with exponentially distributed inter-times with PDF $f_\lambda(t) = \lambda e^{-\lambda t}$, $\lambda = 1$.

For this model, three different configurations are considered, preserving a mean value $\mu = 2$, but with different values of the coefficient of variation of the sojourn time distribution, so as to evaluate the impact of the dispersion measure

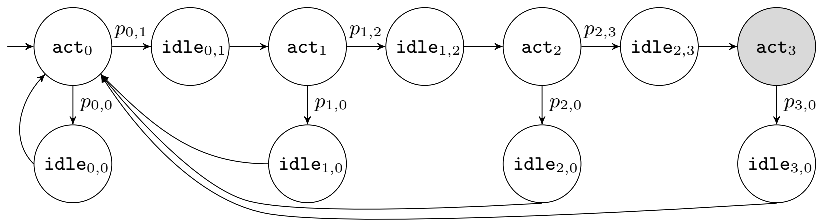


Fig. 4. The stochastic model of activities used for generating synthetic datasets.

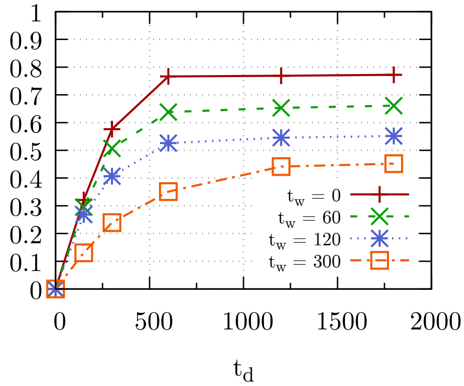
on the predictor accuracy. In the first configuration, named *8-phase Erlang*, sojourn times are generated as Erlang distributed random variables with shape $k = 8$ and rate $\lambda = 4$, resulting in a coefficient of variation of $1/\sqrt{8}$. The second configuration, named *2-phase Erlang*, preserves the same Erlang distribution type but varying the shape to $k = 2$, the rate to $\lambda = 1$, and the coefficient of variation to $\text{CV} = 1/\sqrt{2}$. In the last configuration, named *2-phase Hyper-exp*, a 2-phase hyper-exponential distribution with $\text{CV} \approx 1.92$ and parameters $\lambda_1 = 1/6$, $\lambda_2 = 3/2$, $p_1 = 1/4$, and $p_2 = 3/4$ is applied. For each configuration, a synthetic dataset characterized by a sequence of events and a sequence of activities is generated simulating 4 000 completions of sojourn times. In the simulation, the time unit of temporal parameters is 180 s.

Finally, different settings of the predictor component, obtained with the different fitting strategies *Exp*, *Erlang*, and *Whitt* described in Sect. 3 are experimented, so as to evaluate how different fitting distributions impact on prediction and scheduling performance.

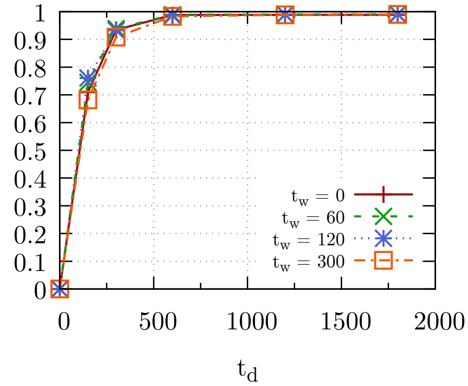
Note that, in so doing, the trained model used for prediction is different than that used in the generation of datasets: the *Whitt strategy* will fit expected value and coefficient of variation, while *Exp* and *Erlang strategies* will fit only the expected value; incidentally, for the *2-phase Erlang* dataset, the *Erlang strategy* fits both the coefficient of variation and the overall shape, and so this benevolent case was not reported in the experimentation.

Fig. 5 compares precision and recall metrics on the *8-phase Erlang* dataset, for different classes of approximants, for t_d equal to 0, 150, 300, 1 200, and 1 800 seconds, and t_w equal to 0, 60, 120, 300 seconds.

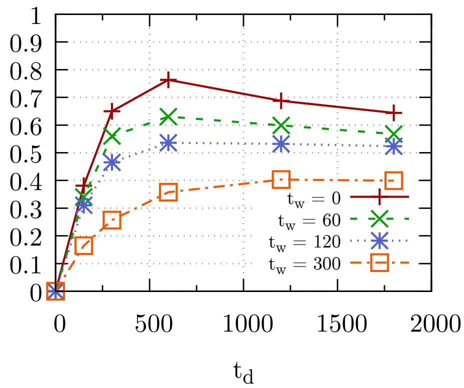
As depicted in Figs. 5b, 5d, 5f, increasing the response duration t_d improves the recall for all fitting strategies. Whereas, the qualitative trend of the precision metric varies with the fitting strategy: in the *Whitt strategy*, the highest precision score is reached around 600 seconds and then steadily maintained until 1 800 seconds, for all t_w except for $t_w = 300$ s (see Fig. 5a); conversely, the *Erlang strategy* showed in Fig. 5c outperforms the precision of the *Whitt strategy* for $t_d \leq 300$ s, and decays for $t_d \geq 600$ s, resulting in a highest precision score lower than the *Whitt strategy*; finally, the *EXP strategy* (Fig. 5e) follows the same qualitative pattern of the *Erlang strategy*, but with an overall performance score reduced with respect to other cases.



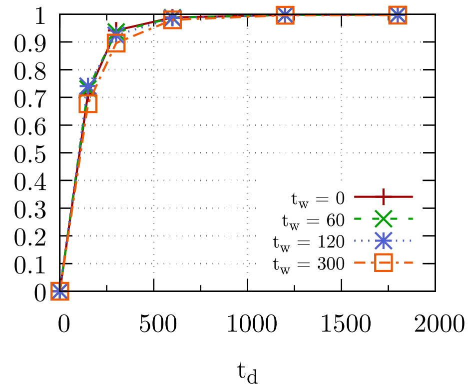
(a) Precision of *Whitt strategy*.



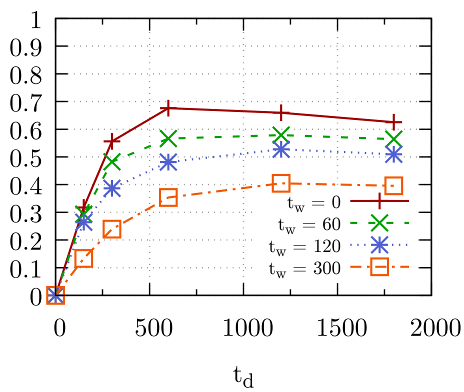
(b) Recall of *Whitt strategy*.



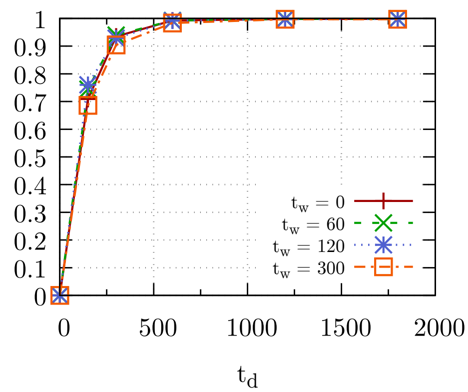
(c) Precision of *Erlang strategy*.



(d) Recall of *Erlang strategy*.

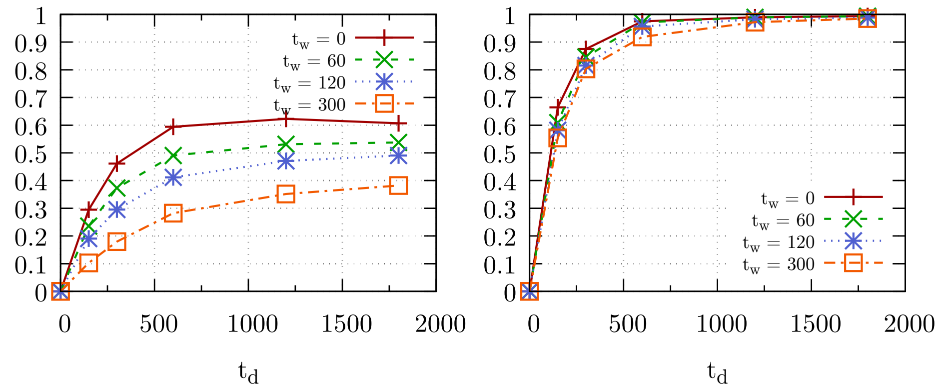


(e) Precision of *Exp strategy*.



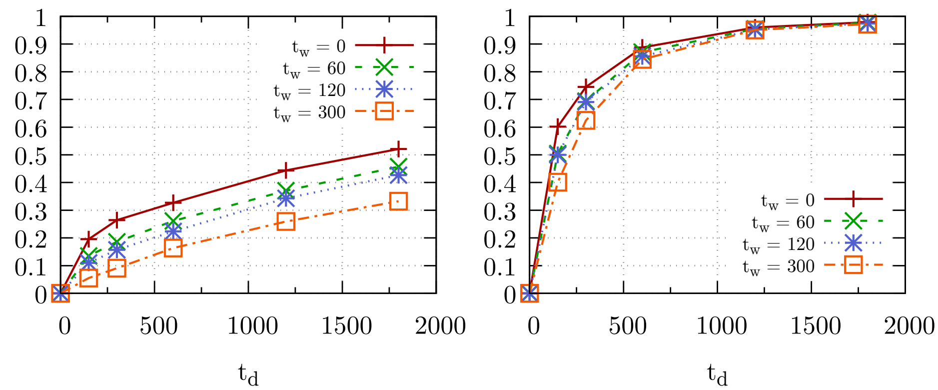
(f) Recall of *Exp strategy*.

Fig. 5. Precision and recall metrics on the *8-phase Erlang* dataset ($\epsilon = 0.0001$).



(a) Precision of *Whitt strategy*. (b) Recall of *Whitt strategy*.

Fig. 6. Precision and recall metrics on the *2-phase Erlang* dataset ($\epsilon = 0.0001$).



(a) Precision of *Whitt strategy*. (b) Recall of *Whitt strategy*.

Fig. 7. Precision and recall metrics on the *2-phase Hyper-exp* dataset ($\epsilon = 0.0001$).

Note that, by increasing the response actuation time t_w (i.e., requiring more time units before the activation of the response action), the overall performance is inevitably reduced, mainly in terms of precision. All metrics are obtained using an ϵ threshold equals to 0.0001. Doubling the ϵ value does not affect precision and recall, while increasing it by a factor of five produces a performance breakdown for $t_d > 600$ s, as the minimum probability required by the scheduling policy is never reached and, consequently, no activation window that contains the start time of the target activity is scheduled.

Since the *Whitt strategy* has been proven to outperform the other fitting strategies on the *8-phase Erlang* dataset, we analyzed how this approximate distribution operates on the *2-phase Erlang* dataset in terms of precision and recall. As a result of a coefficient of variation for sojourn time distributions greater than the *8-phase Erlang* dataset, both metrics perform worse (Fig. 6).

Finally, the same behaviour is emphasized on the *2-phase Hyper-exp* dataset, where a higher coefficient of variation implies the overall lowest performance score, as depicted in Fig. 7.

4.2 Evaluation on a dataset from Ambient Assisted Living

We experiment the proposed approach also using a publicly available dataset [16] for Activity Recognition (AR) [10,5] of Activities of Daily Living (ADLs) [8].

The dataset contains 1 319 time-stamped and typed events collected by 14 state-change sensors and classified in 14 distinct event types (e.g., *hall-bedroom door*, *plates cupboard*, *toilet flush*), placed on various objects (e.g., doors, cupboards, household appliances), and deployed at different locations (e.g. kitchen, toilet, bedroom) inside a 3-room apartment.

These events refer to a period of 28 days, during which a subject was performing 7 distinct activity types $\mathcal{A} = \{$*Leaving house*, *Preparing a beverage*, *Preparing breakfast*, *Preparing dinner*, *Sleeping*, *Taking shower*, *Toileting*$\}$, for a total of 245 activity instances, plus 219 occurrences of *Idle* instances.

Activities were performed in a sequential way (i.e., one activity at a time) with only some limited exceptions (i.e., sometimes *Toileting* occurs at the same time as *Sleeping* or *Preparing dinner*), opportunely removed in order to be cast into the shape of the semi-Markov process. Activities were annotated by the subject himself using a Bluetooth headset, resulting in a stream of activities $a_1, a_2, \ldots, a_n$, each characterized, as described in Sect. 2, by a tuple $a_i = \langle \alpha_i, \tau_i, \delta_i \rangle$.

As described in [4], events were converted from a *raw* sensor representation (which holds a high signal when the sensor is firing, and a low signal otherwise) into a *dual change-point* sensor representation, which emits a time-point signal when the sensor starts to fire and when it switches off, distinguishing activation/deactivation actions and, consequently, doubling the number of event types and instances, from 14 to 28 and from 1 319 to 2 638, respectively. To avoid some inconsistencies in the characterization of *Leave house*, during which all event types were improperly recorded, we have removed from each training set all events not consistent with the activity.

We consider the case in which the system is required to schedule a response action (e.g., an alarm or a reminder) about the intake of some drug before the beginning of a meal. In this scenario, we perform two distinct experiments where the critical activity is either *Preparing breakfast* or *Preparing dinner*. In both cases, we adopt the *Whitt* strategy for the fitting of sojourn times, which outperformed the other two strategies on synthetic datasets. Results are reported in the first two rows of Table 1, showing precision and recall metrics for different values of t_d and t_w. The prediction of the start of the *Preparing dinner* activity achieves $\mathit{precision} = 1, 0.68, 0.68, 0.48$ and $\mathit{recall} = 0.806, 0.708, 0.708, 0.6$ for $t_w = 0, 60, 120, 300$ s and $t_d \geq 1200$ s, respectively. For the *Preparing breakfast* activity, prediction performance is considerably worse, resulting in $\mathit{precision} = 0.929, 0.462, 0.214, 0.2$ and $\mathit{recall} = 0.448, 0.25, 0.136, 0.136$ for $t_w = 0, 60, 120, 300$ s and $t_d \geq 1200$ s, respectively.

			$t_d(s)$					
			0	**150**	**300**	**600**	**1200**	**1800**
Preparing dinner	$t_w(s)$	**0**	0/0	0.163/0.533	0.265/0.684	0.52/0.684	1/0.806	1/0.806
		60	0/0	0.115/0.25	0.2/0.385	0.2/0.385	0.68/0.708	0.68/0.708
		120	0/0	0.2/0.385	0.2/0.385	0.2/0.385	0.68/0.708	0.68/0.708
		300	0/0	0/0	0/0	0/0	0.48/0.6	0.48/0.6
Preparing breakfast	$t_w(s)$	**0**	0/0	0.093/0.385	0.714/0.385	0.714/0.385	0.929/0.448	0.929/0.448
		60	0/0	0.214/0.136	0.214/0.136	0.214/0.136	0.462/0.25	0.462/0.25
		120	0/0	0/0	0/0	0/0	0.214/0.136	0.214/0.136
		300	0/0	0/0	0/0	0/0	0.2/0.136	0.2/0.136
Preparing breakfast (filtered)	$t_w(s)$	**0**	0/0	0.465/0.741	0.489/0.759	0.557/0.818	0.661/0.871	0.672/0.913
		60	0/0	0.222/0.556	0.319/0.667	0.33/0.735	0.408/0.828	0.476/0.896
		120	0/0	0.063/0.261	0.128/0.467	0.225/0.69	0.288/0.778	0.338/0.839
		300	0/0	0.053/0.28	0.088/0.429	0.178/0.643	0.277/0.811	0.329/0.836

Table 1. Precision/recall metrics obtained on the real dataset by the *Whitt strategy* for *Preparing dinner* ($\epsilon = 0.0001$), filtered and not-filtered *Preparing breakfast* ($\epsilon = 0.0003$) as critical activities.

An investigation on the structural reasons of reduced performance for *Preparing breakfast* activity provided insight about the existing mismatch between the real process of AAL and the semi-Markov process used in our online prediction. In the real phenomenon captured by the dataset, some activities such as *Toileting* and *Preparing a beverage* occur as a kind of "shuffle" and noisy events between any other two activities, and the subject carries some memory of the previous activity. Whereas, due to the 1-order memory of the SMP abstraction, the process loses any memory of the past history at each start of a new activity.

In principle, this limit could be overcome by adopting a 2-order memory SMP, as already implemented for modelling and distinguishing the *Idle* states on the basis of the previous activity and of the subsequent one. To confirm this conjecture, an additional experiment was carried out by removing all activities of type *Toileting* or *Preparing a beverage* from the dataset. The third row of Table 1 reports $\textit{precision} = 0.672, 0.476, 0.338, 0.329$ and $\textit{recall} = 0.913, 0.896, 0.839, 0.836$ for $t_w = 0, 60, 120, 300$ s and $t_d = 1800$ s, respectively. Precision is thus marginally improved (except for $t_w = 0$ s), while recall greatly improves for all values of t_w.

5 Conclusions

We presented a model-based approach for the online scheduling of response actions, and evaluated its performance both on synthetic and real-world datasets. The results highlighted that sojourn-time statistics provide important information to predict the future evolution, and accurate models based on generally-distributed sojourn times can improve the precision of scheduling policies. When sojourn times are highly variable, or the sequence of activities does not follow regular patterns, scheduling performance suffers from inaccurate predictions. The experimentation on a real-world dataset of AAL confirmed these results and showed that, for AR, future evolution often depends on the history of previous activities, breaking the hypothesis of semi-Markov models. Finally, keeping an idle state between each pair of activities might lead to overfitting the waiting times as the sample size may be small in some case. Performance achieved by different model structures will be investigated in the future.

References

1. O. Babaoglu, M. Jelasity, A. Montresor, C. Fetzer, S. Leonardi, A. Moorsel, and M. Steen. *Self-star Properties in Complex Information Systems: Conceptual and Practical Foundations*, chapter The Self-Star Vision, pages 1–20. Springer, 2005.
2. N. Bencomo, R. France, B. H. C. Cheng, and U. Aßmann, editors. *Models@run.time: Foundations, Applications, and Roadmaps*. Springer, 2014.
3. G. Bucci, L. Carnevali, L. Ridi, and E. Vicario. Oris: a tool for modeling, verification and evaluation of real-time systems. *Int. Journal of SW Tools for Technology Transfer*, 12(5):391 – 403, 2010.
4. L. Carnevali, C. Nugent, F. Patara, and E. Vicario. A Continuous-Time Model-Based Approach to Activity Recognition for Ambient Assisted Living. In *Proceedings of QEST'15*, LNCS, pages 38–53. Springer, 2015.
5. L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu. Sensor-based activity recognition. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):790–808, 2012.
6. A. Horváth, M. Paolieri, L. Ridi, and E. Vicario. Transient analysis of non-Markovian models using stochastic state classes. *Perform. Eval.*, 69(7-8):315–335, July 2012.
7. A. Ihler, J. Hutchins, and P. Smyth. Learning to detect events with Markov-modulated Poisson processes. *Knowledge Discovery from Data, ACM Transactions on*, 1(3):13, 2007.
8. S. Katz, T. D. Downs, H. R. Cash, and R. C. Grotz. Progress in development of the index of ADL. *The gerontologist*, 10(1 Part 1):20–30, 1970.
9. J. Kephart and D. Chess. The vision of autonomic computing. *Computer*, 36(1):41–50, Jan 2003.
10. E. Kim, S. Helal, and D. Cook. Human activity recognition and pattern discovery. *Pervasive Computing, IEEE*, 9(1):48–53, 2010.
11. V. Kulkarni. *Modeling and analysis of stochastic systems*. Chapman & Hall, 1995.
12. K. Rasch. An unsupervised recommender system for smart homes. *J. Ambient Intell. Smart Environ.*, 6(1):21–37, Jan. 2014.
13. A. Rogge-Solti, W. M. P. van der Aalst, and M. Weske. Discovering stochastic petri nets with arbitrary delay distributions from event logs. In *Int. Business Process Management Workshops, BPM 2013*, pages 15–27, 2013.
14. F. Salfner, M. Lenk, and M. Malek. A survey of online failure prediction methods. *ACM Comput. Surv.*, 42(3):10:1–10:42, Mar. 2010.
15. F. Salfner and M. Malek. Using hidden semi-markov models for effective online failure prediction. In *Reliable Distributed Systems, 2007. SRDS 2007. 26th IEEE International Symposium on*, pages 161–174, Oct 2007.
16. T. van Kasteren, A. Noulas, G. Englebienne, and B. Kröse. Accurate activity recognition in a home setting. In *Proc. Int. Conf. on Ubiquitous Computing*, UbiComp '08, pages 1–9, New York, NY, USA, 2008. ACM.
17. W. Whitt. Approximating a point process by a renewal process, I: Two basic methods. *Operations Research*, 30(1):125–147, Jan.-Feb. 1982.