



Analysis of Spatio-temporal Properties of Stochastic Systems Using TSTL

LUDOVICA LUISA VISSAT, School of Informatics, University of Edinburgh, UK

MICHELE LORETI, School of Science and Technology, University of Camerino, Italy

LAURA NENZI, Department of Mathematics and Geosciences, University of Trieste, Italy

JANE HILLSTON, School of Informatics, University of Edinburgh, UK

GLENN MARION, Biomathematics and Statistics Scotland, Edinburgh, UK

In this article, we present Three-Valued spatio-temporal Logic (TSTL), which enriches the available spatiotemporal analysis of properties expressed in Signal spatio-temporal Logic (SSTL), to give further insight into the dynamic behavior of systems. Our novel analysis starts from the estimation of satisfaction probabilities of given SSTL properties and allows the analysis of their temporal and spatial evolution. Moreover, in our verification procedure, we use a three-valued approach to include the intrinsic and unavoidable uncertainty related to the simulation-based statistical evaluation of the estimates; this can be also used to assess the appropriate number of simulations to use depending on the analysis needs. We present the syntax and three-valued semantics of TSTL and specific extended monitoring algorithms to check the validity of TSTL formulas. We introduce a reliability requirement for TSTL monitoring and an automatic procedure to verify it. Two case studies demonstrate how TSTL broadens the application of spatio-temporal logics in realistic scenarios, enabling analysis of threat monitoring and privacy preservation based on spatial stochastic population models.

CCS Concepts: • **Theory of computation** → **Verification by model checking**;

Additional Key Words and Phrases: Spatio-temporal logics, multi-valued logics, stochastic spatial population models, Statistical Model Checking

ACM Reference format:

Ludovica Luisa Vissat, Michele Loreti, Laura Nenzi, Jane Hillston, and Glenn Marion. 2019. Analysis of Spatio-temporal Properties of Stochastic Systems Using TSTL. *ACM Trans. Model. Comput. Simul.* 29, 4, Article 20 (December 2019), 24 pages.

<https://doi.org/10.1145/3326168>

This work was supported by Microsoft Research Cambridge through its PhD Scholarship Programme, by the EU project QUANTICOL 600708 and by the Austrian Science Fund (FWF) for the project ZK 35. Glenn Marion was funded by the Scottish Government Rural and Environment Science and Analytical Services Division (RESAS).

Authors' addresses: L. Luisa Vissat and J. Hillston, School of Informatics, University of Edinburgh, 10 Crichton St, Edinburgh EH8 9AB, United Kingdom; emails: l.luisa-vissat@sms.ed.ac.uk, Jane.Hillston@ed.ac.uk; M. Loreti, School of Science and Technology, University of Camerino, Piazza dei Costanti 4, 62032 Camerino, Italy; email: michele.loreti@unicam.it; L. Nenzi, Department of Mathematics and Geosciences, University of Trieste, via Valerio 12/1, 34127 Trieste, Italy; email: lnenzi@units.it; G. Marion, Biomathematics and Statistics Scotland, Edinburgh, JCMB, The King's Buildings, Peter Guthrie Tait Road, Edinburgh EH9 3FD, United Kingdom; email: Glenn.Marion@bioss.ac.uk.

Author's current address: L. L. Vissat, Department of Environmental Science, Policy, and Management, UC Berkeley 130 Mulford Hall #3114, Berkeley, CA 94720, United States; email: lluisavissat@berkeley.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2019 Association for Computing Machinery.

1049-3301/2019/12-ART20 \$15.00

<https://doi.org/10.1145/3326168>

1 INTRODUCTION

Stochastic analysis has proved to be effective and useful for many systems that encompass behavior that is not entirely predictable, either through inherent randomness or through abstractions that characterize ranges of behavior as random variables. In this context, temporal logics and their associated model checking algorithms have proved to be very useful for providing rigorous investigation of the possible behavior of the underlying system [3, 22]. In this article, we focus particularly on systems in which, in addition to stochastic behavior, a key characteristic is the spatial arrangement of elements of the system. For example, some interactions may only be possible when entities are co-located or communications may have a restricted range. In particular, we focus on dispersive processes such as spread of disease or information, invasive species, or fire, which all have an intrinsic and fundamental spatial dimension that has to be included in the model.

In many systems, the interaction of stochastic dynamics with spatial structure yields considerable levels of heterogeneity across the system, and this can give rise to interesting global properties absent or less pronounced in analogous non-spatial models. The difference between spatial and non-spatial systems can often be understood in terms of spatial correlations that, for example, induce a shift in mean behavior [34]. Such effects have important real-world consequences. For example, in ecology, spatial dynamics have been associated with increased persistence of interacting species [23], and in epidemiology, selection pressures on pathogens with limited dispersal have been shown to favor reduced virulence in spatial systems [28].

Thus, the dynamics of this class of systems is captured by *spatial* stochastic models. Such models are typically studied through simulations, which can be complemented by logics with both spatial and temporal modalities, providing the ability to describe and verify properties of the spatiotemporal evolution of systems. Thus, a statistical approach is generally taken to spatio-temporal model checking [27]: the satisfaction of given properties, expressed as logical formulas, is estimated based on the evidence gained from randomly generated simulation trajectories, which describe the spatial and temporal evolution. This procedure leverages value from a finite set of spatio-temporal trajectories, which alone are difficult to interpret with respect to the dynamic behavior or to use to compare different systems, while the exhaustive exploration of all possible spatio-temporal trajectories via an explicit state space model is often computationally infeasible.

Current simulation-based approaches provide summary information about the satisfaction of properties over the spatial domain, providing estimated values that include intrinsic uncertainty. In this work, we seek to take into account this uncertainty and to enrich the summary information through the use of a novel logic, the Three-Valued Spatio-temporal Logic (TSTL), which allows us to reason not only about the behavior of the system but also about the evolution of the satisfaction of properties expressed in a spatio-temporal logic. This provides additional insight into the dynamic behavior of the system under study. For example, in the analysis of the efficacy of a control measure for fire spread, we can verify whether the spread in a specific area will happen with probability under a given threshold over time. We can also identify the locations that are at highest risk, because they are surrounded by locations with high probability of burning. The new TSTL atomic propositions are inequalities on the estimated satisfaction probabilities of given spatiotemporal logical formulas. In this article, we use the Signal Spatio-temporal Logic (SSTL) [37], but the framework is general and could be applied to any source of spatio-temporal property satisfaction probabilities. SSTL describes and verifies properties of spatio-temporal trajectories, whose satisfaction probabilities are estimated using Statistical Model Checking [27]. This simulation-based evaluation has an intrinsic and unavoidable uncertainty, but has the advantage of only requiring an executable model.

Using TSTL, we investigate the evolution of these spatio-temporal properties using a three-valued approach. The underlying model checking of SSTL is based on Statistical Model Checking with associated uncertainty. We keep track of this uncertainty in the results of our TSTL model checking, interpreting the inequalities with different degrees of truth, using *true*, *false*, and a third value, *unknown*. Since further simulation trajectories tend to reduce the uncertainty in the atomic propositions, the third value can be taken to be an indication of when more simulation runs are required to make the evaluation of atomic propositions more precise, thus allowing stronger conclusions to be drawn. Conversely, this enables initial explorations with relatively few simulations and assessment of whether they result in sufficient precision. We implemented the monitoring algorithms for the TSTL logical operators to evaluate the satisfaction function of TSTL properties. The operators and the procedures are defined in a similar way to SSTL but on a different domain, dealing with three truth values.

Contribution. The main contribution of this article is to give a formal definition of TSTL and its semantics and demonstrate its use. The logic defined here is an extension of that originally defined in [31]. Specifically, we have added a novel atomic proposition that allows us to compare between two estimated satisfaction probabilities and a novel *comparison* operator, $\langle \psi, v \rangle$, which allows us to verify whether the truth value of the TSTL property ψ is equal to a given truth value v in the three-valued domain. TSTL is also equipped with monitoring algorithms, which are the basis of TSTL model checking. We further develop the idea of the reliability of a TSTL result, allowing the user to specify the degree of acceptable uncertainty, and automating the generation of further spatio-temporal trajectories until this specification is met. This notion was first introduced in the short paper [29] but is fully presented here for the first time. Moreover, we have now implemented the automatic procedure. The capabilities of TSTL, and the reliability specification, are demonstrated on two case studies: the fire spread and evacuation models presented in [31] are more fully explored, and a novel case study on privacy in a communication network is developed.

Article structure. The article is structured as follows: Section 2 introduces notation and background work on SSTL, while Section 3 introduces the process algebra MELA we used to perform stochastic simulations, the monitor jsSTL we used to verify SSTL properties, and how we linked all these aspects together. Section 4 presents the running case-study on fire spread and evacuation routes with examples of SSTL properties. The new logic TSTL is presented in Section 5, together with how TSTL monitoring is linked with the previous framework. Section 6 presents TSTL analysis of the running case study, while Section 7 introduces the reliability requirement and the application of the automatic procedure for TSTL monitoring of properties related to the fire spread model. The novel case study on privacy in a communication network and TSTL applications are introduced in Section 8. Section 9 presents related work, while Section 10 reports the conclusion and future directions for investigation.

2 BACKGROUND

In this section, we introduce some fundamental concepts and notation that we will use in this article aligned with the syntax and semantics of the existing spatio-temporal logic SSTL.

2.1 Notation

We define a *spatial population model*, on a discrete representation of space; it describes a large number of different agents that can perform actions, take different states, interact, and move between different locations. More formally, a *spatial population model* \mathcal{M} is defined as a tuple $\mathcal{M} = (S, G, X, X_0, Tr)$, where:

- $S = \{1, \dots, n\}$ is the set of states that the population agents can take.
- $G = (L, E, w)$ is a *finite weighted undirected graph* that represents the current choice of underlying spatial structure of the spatial population model where:
 - L is the finite set of locations (nodes)
 - $E \subseteq L \times L$ is the set of connections (edges)
 - $w : E \rightarrow \mathbb{R}_{\geq 0}$ is the function *cost* (weights). We extend w to E^* , the transitive closure of E (set containing all the pairs of connected nodes); w gives the sum of costs of the shortest path between two different nodes, where this shortest path is the one that minimizes the sum of the costs.
- $\mathbf{X} : L \rightarrow \mathbb{N}_0^n$, where $\mathbf{X}(l) = (X_1, \dots, X_n) \in \mathbb{N}_0^n$ is the state vector that represents the state of the population in each location. X_i , the entries of the vector $\mathbf{X}(l)$, represent the number of agents in location l in the i^{th} state; therefore, these *counting variables* are $X_i \in \mathbb{N}_0$, where $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.
- $\mathbf{X}_0 : L \rightarrow \mathbb{N}_0^n$, where $\mathbf{X}_0(l)$ is the initial state of the state vector, for each location.
- Tr is the set of transitions, $\tau_i = (\alpha_i, v_i, r_i)$, describing the events that change the global state of the system. Each transition consists of a label α_i , an update vector v_i , and a rate function r_i . The label belongs to the label set \mathcal{L} , which provides information related to the transition, such as the agent performing it and its current location. The vector $v_i : L \rightarrow \mathbb{Z}^n$ records the change to each counting variable in each location due to the transition. With r_i , we indicate the rate function, which may depend on the global state of the system.

We can interpret the dynamical evolution of these models either stochastically as a Markov chain or deterministically as a system of Ordinary Differential Equations (ODEs); in this work we focus on stochastic spatio-temporal systems. Considering the spatial dimension can increase the system unpredictability, since the population becomes fragmented and the assumption of well-mixed population might not hold. In this case, the use of stochastic models is even more appropriate, since they do not represent only the average system evolution.

We can describe the temporal evolution of our spatial population models using:

- σ , a *spatio-temporal trajectory* of \mathcal{M} ; $\sigma : L \times \mathcal{T}_\sigma \rightarrow \mathbb{N}_0^n$, gives the state of the population vector for each location $l \in L$ and each time $t \in \mathcal{T}_\sigma$, up to the temporal horizon. $\mathcal{T}_\sigma = [0, T_D] \subseteq \mathbb{R}_{\geq 0}$, for some $T_D > 0$. These trajectories are generated by discrete-event simulation: the system changes its state at particular time points and it remains in that state for some time. Therefore, these spatio-temporal trajectories are bounded and piece-wise constant. This aspect is important in the definition of the logic semantics in the following part of the article.
- Σ , a *set of spatio-temporal trajectories*, that will be used in the analysis.

2.2 SSTL Syntax

Signal Spatio-temporal Logic (SSTL) [37] is a spatial extension of Signal Temporal Logic (STL) [33], a temporal logic suitable for describing properties of real-valued signals. The syntax of SSTL is given by

$$\varphi ::= \mu \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \mathcal{U}^{[t_1, t_2]} \varphi \mid \diamond_{[w_1, w_2]} \varphi \mid \varphi \mathcal{S}_{[w_1, w_2]} \varphi.$$

The SSTL *atomic proposition* μ is of the form $\mu \equiv (f \geq 0)$, $f : \mathbb{R}^n \rightarrow \mathbb{R}$, an inequality on expressions with population counts, given in the spatio-temporal trajectory. *Negation* \neg and *disjunction* \vee are the standard Boolean operators and $\mathcal{U}^{[t_1, t_2]}$ is the *bounded until* operator. For example, $\varphi_1 \mathcal{U}^{[t_1, t_2]} \varphi_2$ requires that the property φ_2 will be satisfied at some time instant in the interval $[t_1, t_2]$ from the current one and that at all preceding time instants φ_1 holds. SSTL introduces two spatial operators: the *bounded somewhere* operator $\diamond_{[w_1, w_2]}$ and the *bounded surround* operator $\mathcal{S}_{[w_1, w_2]}$, with w_1, w_2 real values, $w_1 \leq w_2$. The *bounded somewhere* operator requires that the

property φ holds in a location reachable from the current one with a cost w , $w \in [w_1, w_2]$. With the operator *bounded surround* we can describe the property $\varphi_1 \mathcal{S}_{[w_1, w_2]} \varphi_2$ of being surrounded by a φ_2 -region, while being in a φ_1 -region: the formula $\varphi_1 \mathcal{S}_{[w_1, w_2]} \varphi_2$ is true in a location l if l belongs to a set A of locations reachable with a cost less than w_2 from l and where φ_1 holds, such that its external boundary $B^+(A)$ contains only locations satisfying φ_2 . The external boundary of a subset of locations A is defined as $B^+(A) := \{l \in L \mid l \notin A \wedge \exists l' \in A \text{ s.t. } (l, l') \in E\}$. Moreover, the locations in the $B^+(A)$ have to be reached from location l with a cost w , $w \in [w_1, w_2]$. Examples of SSTL formulas will be provided throughout the article.

SSTL is provided with two semantics: a *Boolean* semantics, presented in the next section, and a *quantitative* semantics, used to measure the robustness of the satisfaction or dissatisfaction of a property. This latter semantics is not considered in this article.

2.3 SSTL Boolean Semantics

The Boolean semantics of SSTL returns the value true/false ($\mathbb{B} = \{T, F\}$) depending on whether the observed trajectory satisfies the defined SSTL formula or not. The Boolean semantics of an SSTL formula φ is interpreted over a spatio-temporal trajectory σ of \mathcal{M} , for each location $l \in L$ and at time $t \in \mathcal{T}_\sigma$, given values in the set \mathbb{B} :

$$\beta(\mathcal{M}, \sigma, l, t, \varphi) \in \mathbb{B}.$$

The satisfaction function β is defined as follows:

$$\begin{aligned} \beta(\mathcal{M}, \sigma, l, t, \mu) &= \mu(\sigma(l, t)) \\ \beta(\mathcal{M}, \sigma, l, t, \neg\varphi) &= \neg\beta(\mathcal{M}, \sigma, l, t, \varphi) \\ \beta(\mathcal{M}, \sigma, l, t, \varphi_1 \vee \varphi_2) &= \beta(\mathcal{M}, \sigma, l, t, \varphi_1) \vee \beta(\mathcal{M}, \sigma, l, t, \varphi_2) \\ \beta(\mathcal{M}, \sigma, l, t, \varphi_1 \mathcal{U}^{[t_1, t_2]} \varphi_2) &= \bigvee_{t' \in [t+t_1, t+t_2]} \left(\beta(\mathcal{M}, \sigma, l, t', \varphi_2) \wedge \bigwedge_{t'' \in [t, t']} \beta(\mathcal{M}, \sigma, l, t'', \varphi_1) \right) \\ \beta(\mathcal{M}, \sigma, l, t, \diamond_{[w_1, w_2]} \varphi) &= \bigvee_{l' \in L, w(l, l') \in [w_1, w_2]} \beta(\mathcal{M}, \sigma, l', t, \varphi) \\ \beta(\mathcal{M}, \sigma, l, t, \varphi_1 \mathcal{S}_{[w_1, w_2]} \varphi_2) &= \bigvee_{A \in SR_l^{[w_1, w_2]}} \left(\bigwedge_{l' \in A} \beta(\mathcal{M}, \sigma, l', t, \varphi_1) \wedge \bigwedge_{l'' \in B^+(A)} \beta(\mathcal{M}, \sigma, l'', t, \varphi_2) \right), \end{aligned}$$

where the surrounding region, $SR_l^{[w_1, w_2]}$, is defined as:

$$SR_l^{[w_1, w_2]} = \{A \subseteq L \mid \forall l' \in A : 0 \leq w(l, l') \leq w_2 \wedge \forall l'' \in B^+(A) : w_1 \leq w(l, l'') \leq w_2\}.$$

Note that in the definition of the until formula, the disjunction/conjunction operators, which are defined on intervals of reals, play the same role of the existential/universal quantifiers in the considered intervals. We chose this notation to stress the relation with the semantics of the novel spatio-temporal logic presented later in this article.

Monitoring algorithms have been defined to evaluate the validity of SSTL properties, given a spatio-temporal trajectory, working inductively bottom-up on the parse tree of the formula.

To make the verification procedure tractably computable, the input spatio-temporal trajectory must be piece-wise constant. These trajectories are usually obtained by performing discrete-event simulations, as outlined in Section 2.1, and we assume that we observe a finite number of events for each finite spatio-temporal trajectory. For this reason, in the analysis we talk about a discrete time set.



Fig. 1. Σ is the set of spatio-temporal trajectories, Φ the set of SSSL formulas.

As discussed previously, in the study of stochastic systems, we are generally interested in evaluating the probability that given properties are satisfied; a commonly used approach consists of estimating these values using statistical methods on a set of trajectories. Therefore, given an SSSL property φ , we shift the analysis from a single trajectory σ to a set of trajectories Σ , assigning to each trajectory a truth value, according to the Boolean semantics. After this step, we can estimate the satisfaction probability p^* of the formula φ , provided with a confidence interval. For the sake of simplicity, in this article this interval is assumed to be symmetric, of a certain width 2δ . To calculate the values p^* and δ , the function \mathcal{P}_β is defined over the set of trajectories Σ , in terms of β :

$$\mathcal{P}_\beta(\mathcal{M}, \Sigma, l, t, \varphi) = (p^*, \delta), \quad (1)$$

where $(p^*, \delta) \in [0, 1] \times [0, 1]$ and represents the interval $[p^* - \delta, p^* + \delta]$. Given the Boolean nature of the observations, the calculation of p^* and δ is based on the standard approach of using the binomial confidence interval and approximating the error distribution by a normal distribution:

$$p^* = \frac{|\Sigma_T|}{|\Sigma|} \quad \text{and} \quad \delta = z_\alpha \times \sqrt{\frac{p^*(1-p^*)}{|\Sigma|}}, \quad (2)$$

where $\Sigma_T = \{\sigma \in \Sigma \mid \beta(\mathcal{M}, \sigma, l, t, \varphi) = T\}$. From this point on, all the results of Statistical Model Checking are given at 95% confidence ($\alpha = 0.05$), for which $z_\alpha = 1.96$.

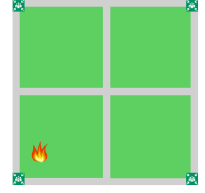
3 MODELING AND MONITORING: MELA AND JSSTL

We used the process algebra MELA [30] to formally describe spatial population models and to perform stochastic simulations, in order to produce spatio-temporal trajectories for the SSSL monitoring. This process algebra MELA has been developed to build spatial population models of ecological systems, since consideration of the spatial aspect has been recognized as of key importance in ecology. MELA allows one to build models on different discrete spatial structures, to define agent behaviors with spatial constraints on their interactions and probability for these interactions to be effective. A variety of different spatial structures are supported in MELA, but here we focus on graph and grid structures. Agents can perform different types of actions that might change their state, their location, or their number in the system. The components in the MELA model generate the states of the underlying stochastic model, a Continuous Time Markov Chain (CTMC), and we perform stochastic simulations using Gillespie's Stochastic Simulation Algorithm (SSA) [18], extracting initial configuration, model structure, and parameter values directly from the MELA model description. We chose to use MELA to facilitate the creation of spatial population models since it presents features that fit perfectly with SSSL monitoring settings, such as discrete representation of space and focus on spatial population models. Accordingly, it has been used in order to produce spatio-temporal trajectories, used as input for jSSL [38], a Java library developed to support monitoring of SSSL properties, as shown in Figure 1. Since SSSL works with a discrete space, in particular with weighted graphs, the grid spatial structures in MELA are mapped to a weighted graph structure, to fit with the SSSL framework, with all the weights equal to 1.

4 CASE STUDY: EMERGENCY EVACUATION ROUTE

We now present a case study related to fire propagation, inspired by [12], which we will use throughout the article as a running example. In this section, we introduce the model and examples of related SSTL properties. We aim to identify the safe evacuation routes from the center of the grid to the assembly points, located in the corners, where there is a fire starting in the location in the lower left corner that can spread to the neighboring locations. We build a MELA model on a 2D grid (25×25) where the fire can spread everywhere, apart from the assembly points (safe zones). The grid cell size is equal to 1 km. In the parts of the grid defined as a route, we have two agents in parallel, one that identifies the presence of people and the other one that identifies the presence of fire.

- Fire spread model (on a 25×25 grid)
- Agents scattered around the center
- Different exit routes (grey lines) to safe zones (located in the corners)
- For each route: (*fire* || *people*)



In the inflammable area, the *fire* agent can be on fire (*B*, burning) or not (*I*, inflammable), while the exit route locations can be empty (*EM*), occupied (*Occ*), or passed (*P*); *P* represents a cell that was occupied but now is empty again. We can evaluate SSTL properties such as

$$\varphi_{occ} := Occ > 0 \quad \varphi_{fire} := B > 0,$$

which identify the occupied location (φ_{occ}) and the burning ones (φ_{fire}). We can also identify the locations in danger, using the property φ_{danger} :

$$\varphi_{danger} := \Diamond_{[d_1, d_2]}(\varphi_{fire}),$$

which evaluates if there are locations on fire, reachable with a bounded cost. The actual MELA model and more details about the spatio-temporal analysis can be found in <https://ludovicalv.github.io/TOMACS/>.

5 THREE-VALUED SPATIO-TEMPORAL LOGIC

In this section, we present the novelty of our research, introducing the syntax and three-valued semantics of TSTL, providing also derived operators and specific monitoring algorithms.

5.1 TSTL Syntax

As presented in Section 2, with the existing SSTL we are able to verify spatio-temporal properties of stochastic systems and estimate the satisfaction probabilities of given formulas. After this initial analysis, we use our proposed extension to perform spatio-temporal analysis of these estimated values. The syntax of TSTL is given by

$$\psi ::= e_1 \tilde{<} e_2 \mid \neg\psi \mid \psi \tilde{\vee} \psi \mid \psi \tilde{\mathcal{U}}^{[t_1, t_2]} \psi \mid \tilde{\Diamond}_{[w_1, w_2]} \psi \mid \psi \tilde{\mathcal{S}}_{[w_1, w_2]} \psi \mid \langle \psi, v \rangle,$$

where $e_i ::= p \mid \mathcal{P}(\varphi)$, $p \in [0, 1]$, φ is a given SSTL formula and $\mathcal{P}(\varphi)$ indicates its estimated satisfaction probability. The atomic TSTL formula $e_1 \tilde{<} e_2$ expresses an inequality on estimated values. With this atomic proposition we are able to check if an estimated satisfaction probability of a given property is either below a given threshold p or below another estimated satisfaction probability. The logical TSTL operators link the TSTL propositions in a similar way to the SSTL ones, but working with estimated values and on a three-valued domain, as explained in the next section. We have

negation $\tilde{\neg}$ and disjunction $\tilde{\vee}$ operators, bounded until $\tilde{\mathcal{U}}^{[t_1, t_2]}$, bounded somewhere $\tilde{\diamond}_{[w_1, w_2]}$, and bounded surround $\tilde{\mathcal{S}}_{[w_1, w_2]}$. Conceptually all these operators are identical to the SSTL operators, but they operate on a different domain, reasoning about estimated satisfaction probabilities and not population counts. The novel *comparison* operator $\langle \psi, v \rangle$ verifies if the truth value of the TSTL property ψ is equal to a given truth value v , where $v \in \mathbb{T} = \{T, U, F\}$, the set of three truth values *true*, *unknown*, and *false*. This operator was introduced to allow the definition of properties that reflect on the truth values themselves, as shown in Section 8 in the case study on communication networks. In the remainder, we will show examples and differences between the two spatiotemporal logics; we will use the letter φ for SSTL formulas and ψ for TSTL ones. We will use the compact notations $\mathcal{P}_{<p}(\varphi)$ and $\mathcal{P}_{>p}(\varphi)$ to indicate the TSTL formulas $\mathcal{P}(\varphi) < p$ and $\mathcal{P}(\varphi) > p$, respectively.

5.2 Three-Valued Semantics

TSTL presents a three-valued semantics that returns a truth value in $\mathbb{T} = \{T, U, F\}$. The truth tables for TSTL negation $\tilde{\neg}$, disjunction $\tilde{\vee}$, and conjunction $\tilde{\wedge}$ (that can be defined in terms of negation and disjunction) are given by:

$\tilde{\neg}$	T	U	F
	F	U	T

$\tilde{\vee}$	ψ_2		
	T	U	F
ψ_1	T	T	T
	U	T	U
	F	T	F

$\tilde{\wedge}$	ψ_2		
	T	U	F
ψ_1	T	T	U
	U	U	U
	F	F	F

as for Kleene's logic of indeterminacy K3 [26]. The three-valued satisfaction function τ for the atomic TSTL proposition $e_1 < e_2$ will return a value in \mathbb{T} :

$$\tau(\mathcal{M}, \Sigma, l, t, e_1 < e_2) \in \mathbb{T}.$$

The evaluation of e_i , $\mathcal{E}(\mathcal{M}, \Sigma, l, t, e_i)$, will return an interval that can represent either $\mathcal{P}_\beta(\mathcal{M}, \Sigma, l, t, \varphi)$, the estimated satisfaction probability of a given SSTL property φ , or a constant p , represented as an interval where $\delta = 0$. Note that we present p as an argument of the function \mathcal{P}_β . In this case, no computation is needed and this value will be constant for each l and each t .

$$\mathcal{E}(\mathcal{M}, \Sigma, l, t, e_i) = \begin{cases} \mathcal{P}_\beta(\mathcal{M}, \Sigma, l, t, \varphi) = (p_i^*, \delta_i) & \text{if } e_i = \mathcal{P}(\varphi) \\ \mathcal{P}_\beta(\mathcal{M}, \Sigma, l, t, p) = (p, 0) & \text{if } e_i = p \end{cases}$$

Note that, given a model \mathcal{M} , a set of spatio-temporal trajectories Σ , and an SSTL property φ , the evaluation of e_i is a function of time t and location l . Consequently, the same holds for p_1^* , p_2^* , δ_1 , and δ_2 . For readability purposes, we omit the time and spatial variables.

We now present the three-valued semantics of the TSTL atomic proposition. Let us consider the evaluation of e_1 and e_2 :

$$\mathcal{E}(\mathcal{M}, \Sigma, l, t, e_1) = (p_1^*, \delta_1) \quad \mathcal{E}(\mathcal{M}, \Sigma, l, t, e_2) = (p_2^*, \delta_2),$$

where $(p_1^*, \delta_1), (p_2^*, \delta_2) \in [0, 1] \times [0, 1]$ and they represent the intervals $[p_1^* - \delta_1, p_1^* + \delta_1]$ and $[p_2^* - \delta_2, p_2^* + \delta_2]$. The associated truth value of the TSTL atomic formula will be

$$\tau(\mathcal{M}, \Sigma, l, t, e_1 < e_2) = \begin{cases} T & \text{if } p_1^* + \delta_1 < p_2^* - \delta_2 \\ U & \text{if } (p_1^*, \delta_1) \cap (p_2^*, \delta_2) \neq \emptyset \\ F & \text{if } p_1^* - \delta_1 > p_2^* + \delta_2. \end{cases}$$

The three-valued satisfaction function τ for the TSTL operators is defined as follows, in an analogous manner to SSTL, where the standard Boolean operators \neg , \vee , and \wedge have been replaced with the three-valued operators $\tilde{\neg}$, $\tilde{\vee}$, and $\tilde{\wedge}$:

$$\begin{aligned}
\tau(\mathcal{M}, \Sigma, l, t, \tilde{\neg}\psi) &= \tilde{\neg}\tau(\mathcal{M}, \Sigma, l, t, \psi) \\
\tau(\mathcal{M}, \Sigma, l, t, \psi_1 \tilde{\vee} \psi_2) &= \tau(\mathcal{M}, \Sigma, l, t, \psi_1) \tilde{\vee} \tau(\mathcal{M}, \Sigma, l, t, \psi_2) \\
\tau\left(\mathcal{M}, \Sigma, l, t, \psi_1 \tilde{\mathcal{U}}^{[t_1, t_2]} \psi_2\right) &= \bigvee_{t' \in [t+t_1, t+t_2]} \left(\tau(\mathcal{M}, \Sigma, l, t', \psi_2) \tilde{\wedge} \bigwedge_{t'' \in [t, t']} \tau(\mathcal{M}, \Sigma, l, t'', \psi_1) \right) \\
\tau\left(\mathcal{M}, \Sigma, l, t, \bigotimes_{[w_1, w_2]} \psi\right) &= \bigvee_{l' \in L, w(l, l') \in [w_1, w_2]} \tau(\mathcal{M}, \Sigma, l', t, \psi) \\
\tau\left(\mathcal{M}, \Sigma, l, t, \psi_1 \tilde{\mathcal{S}}_{[w_1, w_2]} \psi_2\right) &= \bigvee_{A \in SR_l^{[w_1, w_2]}} \left(\bigwedge_{l' \in A} \tau(\mathcal{M}, \Sigma, l', t, \psi_1) \right. \\
&\quad \left. \tilde{\wedge} \bigwedge_{l'' \in B^+(A)} \tau(\mathcal{M}, \Sigma, l'', t, \psi_2) \right).
\end{aligned}$$

The truth value of the novel *comparison* operator will be

$$\tau(\mathcal{M}, \Sigma, l, t, \langle \psi, v \rangle) = \begin{cases} T & \text{if } \tau(\mathcal{M}, \Sigma, l, t, \psi) = v \\ F & \text{if } \tau(\mathcal{M}, \Sigma, l, t, \psi) \neq v, \end{cases}$$

where $v \in \mathbb{T} = \{T, U, F\}$.

Note the similarity between the structure of β and τ , with operators that refer to SSTL and TSTL, respectively. We want to clarify that SSTL results are provided performing Statistical Model Checking with a given confidence level. Therefore, we are not talking about confidence level of TSTL results, but about TSTL results, given the confidence level for the Statistical Model Checking procedure. With the current definition of TSTL, we can derive more operators. The *everywhere* spatial operator $\tilde{\square}_{[w_1, w_2]}$ can be defined as

$$\tilde{\square}_{[w_1, w_2]} \psi := \tilde{\neg} \bigotimes_{[w_1, w_2]} \tilde{\neg}\psi.$$

This requires ψ to hold in all the locations reachable from the current one with a total cost between w_1 and w_2 . The *eventually* $\tilde{\mathcal{F}}^{[t_1, t_2]}$ and the *globally* $\tilde{\mathcal{G}}^{[t_1, t_2]}$ operators are defined as usual:

$$\tilde{\mathcal{F}}^{[t_1, t_2]} \psi := T \mathcal{U}^{[t_1, t_2]} \psi \quad \tilde{\mathcal{G}}^{[t_1, t_2]} \psi := \tilde{\neg} \mathcal{F}^{[t_1, t_2]} \tilde{\neg}\psi.$$

The *eventually* formula holds if ψ becomes true within t_1 and t_2 time units from the current one, while the *globally* formula requires ψ to be satisfied for each time unit in the relative interval $[t_1, t_2]$. As we already presented, TSTL provides an additional level of analysis for evaluation of spatio-temporal properties of estimated satisfaction probabilities of SSTL properties. Hence, there is a crucial difference between both the analysis and the logical operators used in SSTL and TSTL. For example, the following two TSTL properties ψ_1 and ψ_2 :

$$\psi_1 := \mathcal{P}_{\sim p}(\varphi_1 \wedge \varphi_2) \quad \psi_2 := \mathcal{P}_{\sim p}(\varphi_1) \tilde{\wedge} \mathcal{P}_{\sim p}(\varphi_2),$$

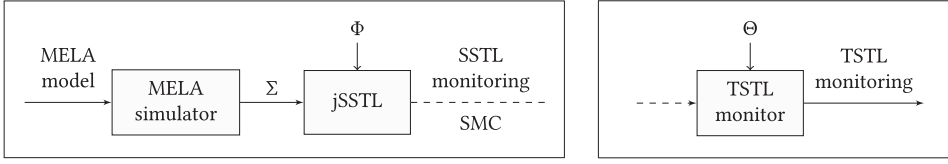


Fig. 2. Σ is the set of spatio-temporal trajectories, Φ the set of SSTL formulas, and Θ the set of TSTL formulas.

are intrinsically different and therefore they can take on different truth values. To give a concise example, let us assume that we are working with a fire spread model with multiple inflammable agents for each location and we have the following SSTL properties on the number of burning agents B :

$$\varphi_1 := B > 5 \quad \varphi_2 := B > 10.$$

Let assume that, for a given burning probability threshold p :

$$\tau \left(\mathcal{M}, \Sigma, l, t, \mathcal{P}_{<p}(\varphi_1) \right) = F \quad \tau \left(\mathcal{M}, \Sigma, l, t, \mathcal{P}_{<p}(\varphi_2) \right) = T.$$

This can happen if we choose the value of p between the two estimates, outside their respective intervals. Since $\varphi_1 \wedge \varphi_2 \equiv \varphi_2$, then:

$$\tau \left(\mathcal{M}, \Sigma, l, t, \mathcal{P}_{<p}(\varphi_1 \wedge \varphi_2) \right) = \tau \left(\mathcal{M}, \Sigma, l, t, \mathcal{P}_{<p}(\varphi_2) \right) = T,$$

while:

$$\tau \left(\mathcal{M}, \Sigma, l, t, \mathcal{P}_{<p}(\varphi_1) \right) \tilde{\wedge} \tau \left(\mathcal{M}, \Sigma, l, t, \mathcal{P}_{<p}(\varphi_2) \right) = F.$$

Moreover, the first could perhaps be derived empirically from observations, but the second is only expressible with the new logical operator $\tilde{\wedge}$ and the domain \mathbb{T} .

The structure of our spatio-temporal analysis is shown in Figure 2. As shown, the initial framework is extended, with the results of Statistical Model Checking (SMC) based on jSSTL monitoring used as input to verify TSTL properties.

5.3 Monitoring the Three-Valued Semantics of the Atomic Proposition, the Negation, and the Bounded Surround

To evaluate the validity of TSTL formulas, we implemented monitoring algorithms for each logical operator, structured in a similar way to SSTL monitoring [37]. We illustrate now the monitoring algorithms for the TSTL *atomic proposition*, *negation*, and *bounded surround* operator. For each location \hat{l} , the monitoring algorithms return the piece-wise constant function $s_{\psi, \hat{l}}$ that maps each time t , in the finite set of sample times \mathcal{T} , with $\tau(\mathcal{M}, \Sigma, \hat{l}, t, \psi)$. The cardinality of this set \mathcal{T} depends on the given SSTL and TSTL formulas; it is the shortest finite sequence of time points for which we have the values of the satisfaction function of all the formulas involved.¹ Algorithm 1 evaluates the satisfaction of the TSTL atomic proposition. The algorithm receives as input a given location \hat{l} , the evaluation of e_1 and e_2 , and the atomic proposition ψ . Given a model \mathcal{M} , a set of spatio-temporal trajectories Σ , and the respective SSTL properties, the evaluations are functions of location l and time t , in the discrete set \mathcal{T} of time points. In this case, we perform the analysis

¹We need to take into account that a temporal formula looks T_f time units into the future, and hence the domain $[0, T_D]$ becomes $[0, T_D - T_f]$, as we assume we only have access to trajectories up to time T_D .

ALGORITHM 1: Three-Valued Spatio-temporal Logic: Atomic Proposition

input: $\hat{l}, \mathcal{E}_1(\hat{l}, t) = \mathcal{E}(\mathcal{M}, \Sigma, \hat{l}, t, e_1), \mathcal{E}_2(\hat{l}, t) = \mathcal{E}(\mathcal{M}, \Sigma, \hat{l}, t, e_2), \psi = e_1 \tilde{<} e_2$

for all $t \in \mathcal{T}$ **do**

$[p_1^*(\hat{l}, t) - \delta_1(\hat{l}, t), p_1^*(\hat{l}, t) + \delta_1(\hat{l}, t)] = \mathcal{E}_1(\hat{l}, t)$

$[p_2^*(\hat{l}, t) - \delta_2(\hat{l}, t), p_2^*(\hat{l}, t) + \delta_2(\hat{l}, t)] = \mathcal{E}_2(\hat{l}, t)$

if $p_1^*(\hat{l}, t) + \delta_1(\hat{l}, t) < p_2^*(\hat{l}, t) - \delta_2(\hat{l}, t)$ **then**

$s_{\psi, \hat{l}}(t) = T$

else

if $p_1^*(\hat{l}, t) - \delta_1(\hat{l}, t) > p_2^*(\hat{l}, t) + \delta_2(\hat{l}, t)$ **then**

$s_{\psi, \hat{l}}(t) = F$

else

$s_{\psi, \hat{l}}(t) = U$

return $s_{\psi, \hat{l}}$

ALGORITHM 2: Three-Valued Spatio-temporal Logic: Negation

input: $\hat{l}, \psi_{neg} = \tilde{\neg}\psi$

for all $t \in \mathcal{T}$ **do**

$s_{\psi_{neg}, \hat{l}}(t) = \tilde{\neg}s_{\psi, \hat{l}}(t)$

return $s_{\psi_{neg}, \hat{l}}$

for a given location \hat{l} . For each time point in \mathcal{T} , the evaluations are compared following the TSTL semantics evaluating the truth value of the TSTL inequality $\tilde{<}$, assigning a truth value to $s_{\psi, \hat{l}}$. The output of this procedure is $s_{\psi, \hat{l}}$, which describes the satisfaction of the property ψ for location \hat{l} over time. Algorithm 2 evaluates the satisfaction of the negation operator $\tilde{\neg}$. In this case the algorithm receives as input a location \hat{l} and the TSTL property ψ_{neg} . This procedure simply applies the negation operator to the truth value of ψ for each time point. This algorithm returns the satisfaction of the property ψ_{neg} over time for location \hat{l} . The last algorithm, Algorithm 3, verifies the TSTL bounded surround formula $\psi = \psi_1 \tilde{S}_{[w_1, w_2]} \psi_2$. The monitoring of this operator is more elaborate than the other procedures. We recall that this operator is used to verify spatial properties of being surrounded by a ψ_2 region, while being in a region satisfying ψ_1 . For example, we can extract which are the areas in high danger of being on fire, being surrounded by a region that will burn with high probability and that satisfies the distance constraints.

As shown in Algorithm 3, as the first step of the algorithm, we compute the value $s_{\psi_1, l}$ for all the locations $l : 0 \leq w(\hat{l}, l) \leq w_2$ and the value $s_{\psi_2, l}$ for all the locations $l : w_1 \leq w(\hat{l}, l) \leq w_2$. These values are obtained by recursive invocation of the monitoring algorithm on the TSTL subformulas ψ_1 and ψ_2 . We set these values for the other locations to be F , $\forall t \in \mathcal{T}$. After this initial step, we iteratively compute a fixed-point function, on the set of locations satisfying the cost bounds, to get the value of the bounded surround formula, for each time point in the discrete time set \mathcal{T} . This fixed point coincides with the limit of the sequence $(\chi_i)_{i \in \mathbb{N}}$, $\chi_i : L \rightarrow \mathbb{T}$, defined as follows:

ALGORITHM 3: Three-Valued Spatio-temporal Logic: Bounded Surround Operator

```

input:  $\hat{l}, \psi = \psi_1 \tilde{S}_{[w_1, w_2]} \psi_2, \mathcal{T}$ 
for all  $l \in L$  do
  if  $0 \leq w(\hat{l}, l) \leq w_2$  then
    compute  $s_{\psi_1, l}$ ;
    if  $w(\hat{l}, l) \geq w_1$  then
      compute  $s_{\psi_2, l}$ ;
    else
       $s_{\psi_2, l} = F$ 
  else
     $s_{\psi_1, l} = F; s_{\psi_2, l} = F$ 
for all  $t \in \mathcal{T}$  do
  for all  $l \in L$  do
     $\chi_{prec}(l) = T$ 
     $\chi(l) = s_{\psi_1, l}$ 
    while  $\exists l \in L : \chi_{prec}(l) \neq \chi(l)$  do
       $\chi_{prec} = \chi$ 
      for all  $l \in L$  do
         $\chi(l) = \chi_{prec}(l) \tilde{\wedge} (\bigwedge_{l': (l, l') \in E} (\chi_{prec}(l') \tilde{\vee} s_{\psi_2, l'}))$ 
       $s_{\psi, l}(t) = \chi(\hat{l})$ 
return  $s_{\psi, \hat{l}}$ 

```

- (1) $\chi_0(l) = s_{\psi_1, l}(t)$
- (2) $\chi_{i+1}(l) = \chi_i(l) \tilde{\wedge} (\bigwedge_{l': (l, l') \in E} (\chi_i(l') \tilde{\vee} s_{\psi_2, l'}(t)))$

where i indicates the iteration and E is the set of edges of G . The upper bound on the number of iterations of the algorithm is given by the diameter d_G of the graph; given $\chi(l)$, the fixed point of $\chi_i(l)$, then $\chi(l) = \chi_{d_G+1}(l)$, $\forall l \in L$. The proof of the correctness of the method follows that of the SSTL monitoring. The cost of this computation for each location is $O(d_G |L| |\mathcal{T}|)$; therefore, the cost for all locations is $O(d_G |L|^2 |\mathcal{T}|)$. For more details, see [37], where a similar approach is used.

6 CASE STUDY: TSTL ANALYSIS

We now present the TSTL analysis we have performed to identify the most appropriate fire exit in a situation where the evacuation routes are already defined. In this example, the movement of people and fire are modeled separately; they do not influence each other in the model. We have gathered both types of information in the study of TSTL properties. For each property we run 30 simulations to perform TSTL verification. With these relatively few runs we have an overall insight into the dynamics and the differences between distinct models, with a simple representation of complex systems and properties. Also in this case, more details about the spatio-temporal analysis can be found at <https://ludovicalv.github.io/TOMACS/>. To identify the safe evacuation routes, we use TSTL to identify grid cells that have low probability of being on fire (ψ_{fire}) and probability greater than 0.01 of being occupied (ψ_{occ}), given the agent movement in the model:

$$\varphi_{occ} := Occ > 0 \quad \varphi_{fire} := B > 0$$

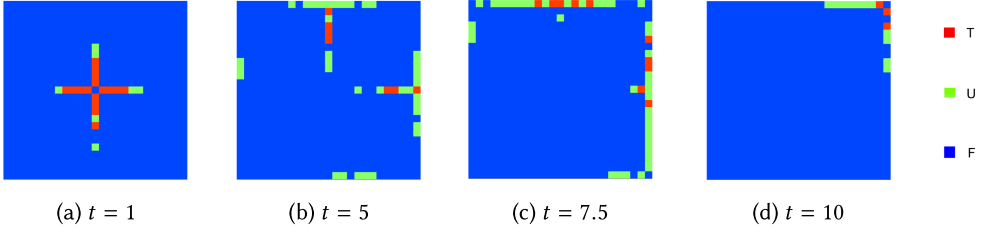


Fig. 3. Temporal evolution of safe evacuation routes: TSTL property ψ_{safe} , with movement rate of the agents equal to 2.0. The color scheme is specified on the right: red (true), green (unknown), and blue (false).

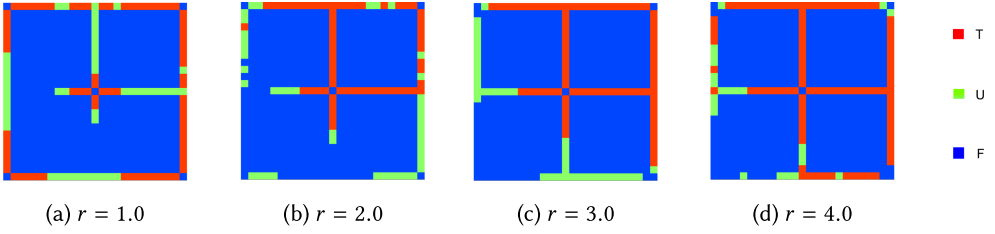


Fig. 4. Safe evacuation routes: TSTL property $\psi_{safeRoute}$, with different movement rates of the agents, $t = 0$.

$$\psi_{occ} := \mathcal{P}_{\sim_{0.01}}(\varphi_{occ}) \quad \psi_{fire} := \mathcal{P}_{\sim_{0.2}}(\varphi_{fire}) \quad \psi_{safe} := \psi_{occ} \tilde{\wedge} \psi_{fire}.$$

The verification output of TSTL property ψ_{safe} shows the routes that will lead to the assembly point safely, as shown in Figure 3. To be able to identify the safe evacuation routes from the beginning, instead of observing their temporal evolution, we can check the TSTL property $\psi_{safeRoute}$ at $t = 0$. We want to identify the route that, with probability higher than 0.8, will not be on fire if occupied, in this case in the first 10 time units:

$$\begin{aligned} \varphi_{route} &:= (EM > 0) \vee (Occ > 0) \vee (P > 0) \\ \varphi_{notFire} &:= \neg((Occ > 0) \wedge (B > 0)) \equiv \neg(Occ > 0) \vee \neg(B > 0) \\ \varphi_{GSafe} &:= \mathcal{G}^{[0,10]}(\varphi_{route} \wedge \varphi_{notFire}) \quad \psi_{safeRoute} := \mathcal{P}_{\sim_{0.8}}(\varphi_{GSafe}). \end{aligned}$$

We will check this TSTL property, changing the rate of agent movement in the MELA model, as shown in Figure 4. We can observe that if the rate of movement is not high enough, there are not safe options to reach the assembly points. As further analysis, we examine the number of unknown values over time, given TSTL properties and changing the quantity of spatio-temporal trajectories to analyze. We study the percentage of locations having unknown as truth value for different formulas and for different numbers of simulation runs. We observed that the percentage decreases with the increase in the number of runs. Since the width of the confidence intervals depends to a large extent on this value, with an increase in the number of runs we tend to give a more precise estimation of the satisfaction probability. Therefore, we have narrower confidence intervals as input for TSTL monitoring and a consequent smaller percentage of unknown values. The three-valued approach is useful to discriminate among TSTL properties in the process of acquiring spatio-temporal trajectories, until the satisfaction set is large enough. In decision-making situations, such as at the beginning of a forest fire, we want to quickly compare different control strategies by evaluating properties of the system with enough accuracy to be able to draw appropriate conclusions.

7 RELIABILITY REQUIREMENT

For the purpose of evaluating TSTL properties rapidly but accurately, we introduce a reliability requirement \mathcal{R} and an automatic procedure to assess if more spatio-temporal trajectories are needed to match the requirement. The evaluation of the reliability requirement \mathcal{R} is a Boolean value that expresses if \mathcal{R} is matched or not, and therefore the need to acquire more spatio-temporal trajectories. This specification \mathcal{R} is defined as the proportion of *unknown* values that we can tolerate during the exploration of the TSTL property satisfaction. As shown in Algorithm 4, given a TSTL formula ψ , a set of spatio-temporal trajectories Σ , and the reliability requirement \mathcal{R} , we perform TSTL monitoring for each location l of the spatial domain L , at each time step of the discrete time set \mathcal{T} , according to the given logical formulas. After this step, our algorithm checks if the reliability specification is matched, identifying if more simulations are needed. If \mathcal{R} is matched for each time point in the given time horizon, we return *reliable* as output, which expresses that the TSTL verification satisfies the requirement. If the requirement \mathcal{R} is not matched, we need to carry out more simulations, performing the monitoring procedure of the formulas and checking the reliability again.

ALGORITHM 4: Automatic Verification of the Reliability Specification

```

input:  $\psi, \Sigma, \mathcal{R}, k$ 
counter = 0
while  $|\Sigma| \leq \max$  do
    Generate  $\sigma$ 
     $\Sigma = \Sigma \cup \{\sigma\}$ 
    for all  $t \in \mathcal{T}$  do
        for all  $l \in L$  do
            TSTL monitoring of  $\psi$ , location  $l$ , time  $t$ , with  $\sigma$ 
        evaluate  $\mathcal{R}$ 
        if  $\mathcal{R}$  true then
            if counter <  $k$  then
                counter++
            else
                return reliable
        else
            counter = 0
return unreliable

```

We observe that the proportion of *unknown* values decreases with the number of analyzed spatiotemporal trajectories. Given that this value largely influences the width of the confidence intervals, by acquiring more spatio-temporal trajectories we tend to give a more precise estimation of the satisfaction probability. Therefore, we provide narrower confidence intervals as input for TSTL monitoring and we have a consequent smaller proportion of unknown values. However, the relationship between the number of spatio-temporal trajectories and the confidence interval width is not guaranteed to be strictly monotonic. Thus, in some circumstances the reliability check may not be stable with respect to a single spatio-temporal trajectory, meaning that one trajectory might change the reliability condition to being matched while the next could negate it. To guard against this, we define as input the value k , $k \in \mathbb{N}$, and we only consider the reliability specification satisfied when a positive outcome is achieved for k consecutive spatio-temporal trajectories. This is monitored through the variable *counter*, as shown in Algorithm 4. In our examples, we perform the automatic procedure for $k = 10$ and we show the results for a single time point.

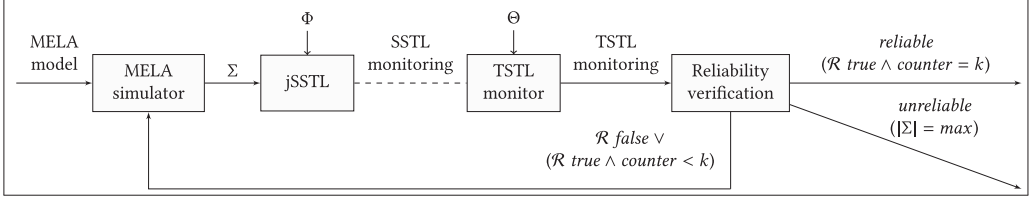


Fig. 5. The figure illustrates the framework with the automatic procedure for the reliability requirement verification. Σ is the set of spatio-temporal trajectories, Φ the set of SSTL formulas, Θ the set of TSTL formulas, and \mathcal{R} the reliability requirement.

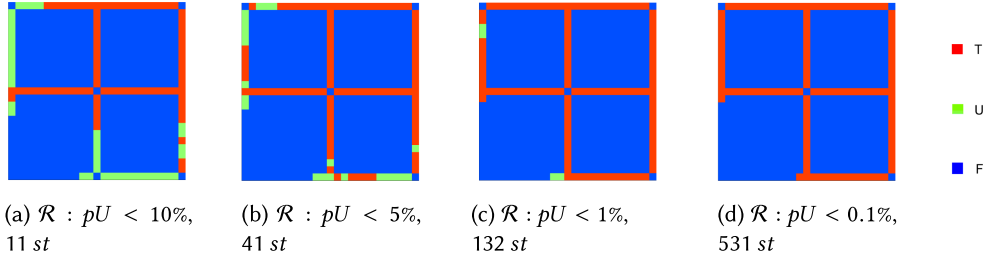


Fig. 6. Automatic procedure for $\psi_{\text{safeRoute}}$, time = 0, $r = 4.0$, $k = 10$.

To ensure the termination of our procedure, we specify an upper bound on the number of spatiotemporal trajectories that we can acquire to satisfy the requirement. As shown in Figure 5, if the upper bound is reached without k consecutive positive outcomes, the procedure will return *unreliable*, since it was not able to verify if the reliability specification was satisfied or not. The complexity of this procedure depends on the confidence level chosen for the SSTL analysis, the cardinality of the discrete time set \mathcal{T} , the number of locations $|L|$, and parameters of the formulas. We have implemented the automatic procedure and applied it to evaluate properties related with our running example. We verify the TSTL properties $\psi_{\text{safeRoute}}$, which was presented in the previous section, and

$$\psi_{\text{fire}} := \mathcal{P}_{>0.8}(\varphi_{\text{fire}})$$

for time 0 and time 5, respectively. The results of the automatic procedure are shown in Figures 6 and 7. For each plot, we provide the given reliability requirement \mathcal{R} , describing with pU the chosen upper bound proportion, and the number of spatio-temporal trajectories (*st*) needed to satisfy it.

8 CASE STUDY: ANONYMITY NETWORK

In this second case study, we analyze a model of a communication network, focusing on properties related to digital privacy and anonymity. We model the communication network using a complete graph composed by 100 nodes, where each node represents either a *sender*, a *forwarder*, or an *exit*. The sender introduces a message in the network and forwards it to a forwarder or exit. Through the action of the forwarders, the message moves, performing a random walk on the graph, until it reaches an exit and leaves the network. The forwarders can be either *honest* (*H*) or *corrupt* (*C*). Corrupt forwarders aim to determine the sender of the messages they obtain. As initial condition, we have five senders S_i , each of them with the same number of messages that can be sent. We use the index i to distinguish among the senders and their generated messages. We represent a generated message with M_i . We have five exits E and five corrupted nodes C . All the remaining nodes are honest forwarders. The complete MELA model can be found at <https://ludovicalv.github.io>.

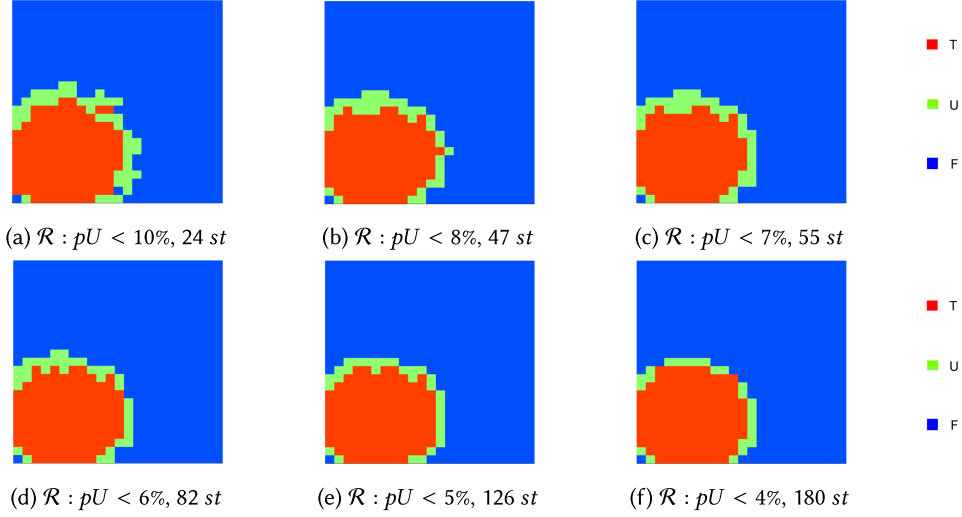


Fig. 7. Automatic procedure for ψ_{fire} , time = 5, k = 10.

io/TOMACS/. We use TSTL to evaluate the privacy requirement for the network, verifying that *each corrupt node has similar probability to see the messages for each sender*. This requirement can be assessed by verifying that the following property:

$$\psi_{ij} := \mathcal{P}(\varphi_{corruptM_i}) \tilde{<} \mathcal{P}(\varphi_{corruptM_j})$$

evaluates U for each pair of senders (M_i, M_j) . As an initial step, we start performing this analysis for messages M_0 and M_1 (messages from sender S_0 and S_1) in one of the corrupt nodes. We evaluate the truth value of the TSTL property ψ_{01} :

$$\psi_{01} := \mathcal{P}(\varphi_{corruptM_0}) \tilde{<} \mathcal{P}(\varphi_{corruptM_1}),$$

where

$$\varphi_{corruptM_i} = (M_i > 0) \wedge (C > 0) \quad i = 0, 1$$

represents the presence of messages M_i in a corrupt node (identified by the presence of the agent C). In this model, we assume that each sender has a fixed number of messages available. We analyzed 500 spatio-temporal trajectories. In Figure 8, we present the results of this analysis, observing that the truth value of property ψ_{01} evaluates as U , implying that the estimated satisfaction probabilities of formulas $\varphi_{corruptM_0}$ and $\varphi_{corruptM_1}$ are close. We evaluate the same properties under different scenarios, which we can easily describe by using the high-level modeling language MELA. We can consider scenarios in which:

- Messages M_0 move five times faster than the other messages
- Messages M_0 move 10 times faster than the other messages
- Sender S_0 has more messages available to send (five times more).

The results of these analyses are shown in Figures 9, 10, and 11. If the messages move faster, they tend to leave the network earlier, quickly finding an exit node. We can observe this in the results, where the probability of finding M_0 decreases faster than the other probabilities, and therefore ψ_{01} is evaluated as T after some initial time. With more messages M_0 in the network, we observe that

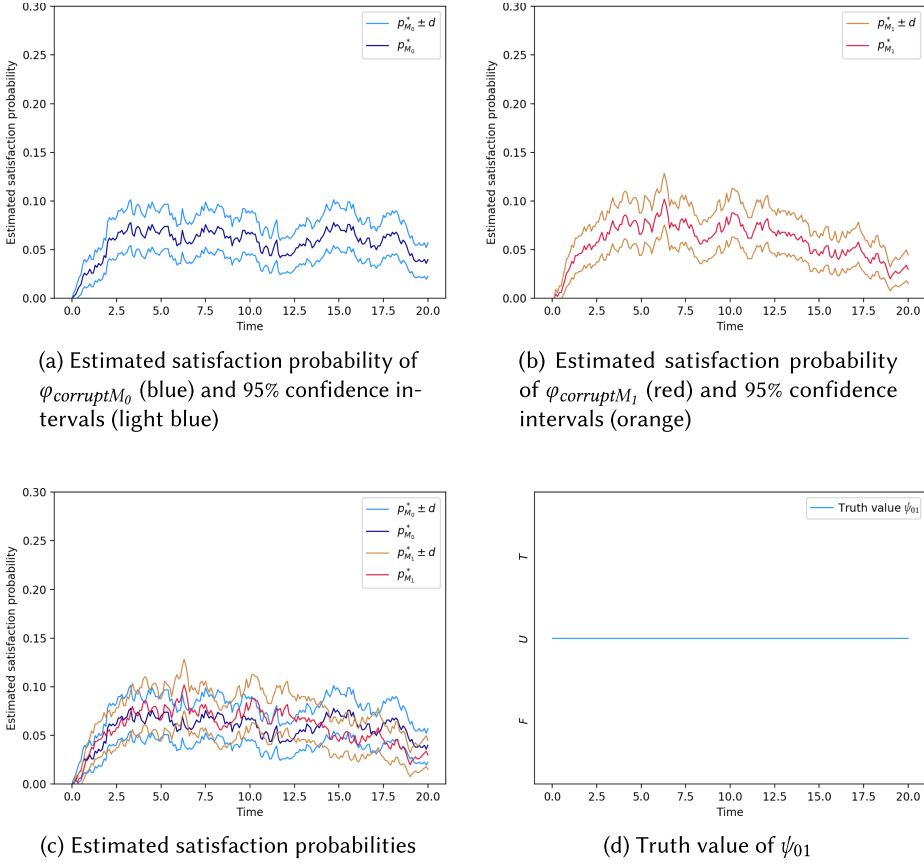


Fig. 8. Basic scenario.

the probability of finding messages M_0 is higher, as expected, and therefore the truth value of ψ_{01} evaluates as F .

Complete analysis. We have gone on to analyze the privacy property for all the possible senders in an arbitrary corrupted node. We have verified the TSTL property ψ_{total} :

$$\psi_{total} := \bigwedge_{i,j} \langle \psi_{ij}, U \rangle$$

using the novel *comparison* operator. To verify that the privacy requirement is satisfied, we want the truth value of this formula to be equal to T . As shown in Figure 12, the condition is mainly satisfied in the basic scenario, while with faster messages it is satisfied just in the first part of the temporal horizon, since afterward the number of messages M_0 in the network will be lower than the rest. In the case of more messages for sender S_0 , the condition is not satisfied, since we have higher probability to find messages M_0 in the network, and therefore the truth value of ψ_{total} is evaluated as false.

9 RELATED WORK

In this section, we give an overview of existing logics dealing with spatial aspects of systems and estimation of satisfaction probability values.

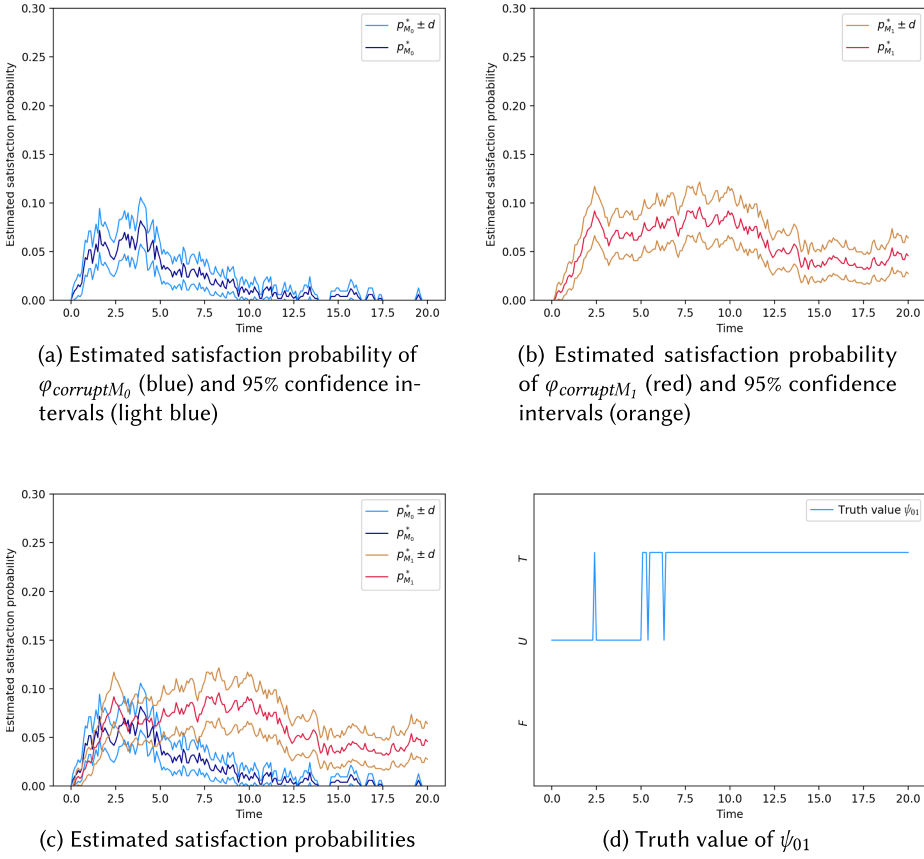


Fig. 9. Messages M_0 move faster in the network ($rate_{M_0} = 5.0$, other $rate_{M_i} = 1.0$).

The literature of applications of temporal logics to stochastic systems has been mainly focused on well-mixed systems, where the spatial nature of interactions is not considered for the sake of simplicity. However, the spatial information can be encoded in the model (e.g., adding spatial information to the agent names), but this choice will result in a cumbersome and less efficient analysis. In many cases, the lack of a notion of spatial structure heavily impacts on the descriptive modeling power and in the type of analysis that can be performed.

There are existing logics for expressing properties on probabilities, such as Probabilistic Computation Tree Logic (PCTL) [22] and Continuous Stochastic Logic (CSL) [3]. Three-valued logics, such as ours, with just one additional truth value, are a simple case in the field of multi-valued logics [20]. The initial concept was created by Łukasiewicz [32] and developed further by different logicians, such as Kleene [26], introducing the concept of “undefined” dealing with partial recursive functions. A similar concept has been later used by Jones et al. in [4, 24]. The three-valued approach is used in [25] for the definition of a new abstraction method for fully probabilistic systems and in [40] for model checking of Discrete-Time Markov Chains. We are not aware of any current use of a three-valued logic approach in the field of spatio-temporal analysis of stochastic systems.

In the area of spatial logics, much work focuses on theoretical investigation [1], expressivity, and decidability, looking at properties of subsets of points of topological spaces. Less attention has

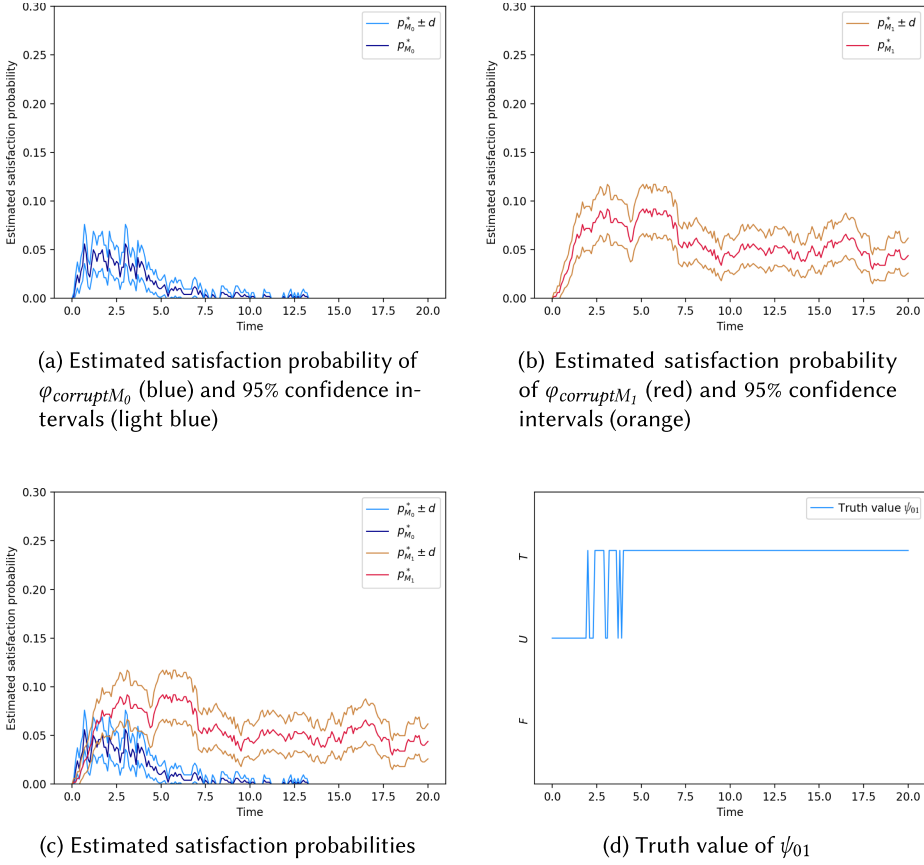


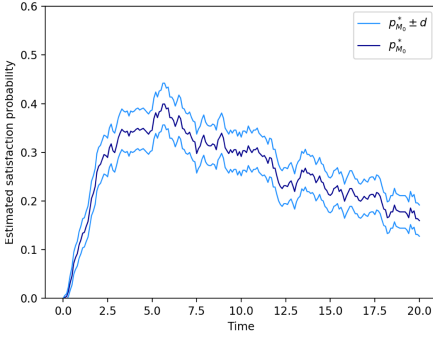
Fig. 10. Messages M_0 move faster in the network ($rate_{M_0} = 10.0$, other $rate_{M_i} = 1.0$).

been placed on more concrete representation of space and on more practical aspects, especially in the validation procedure. In particular, model checking and monitoring routines have a much more recent history.

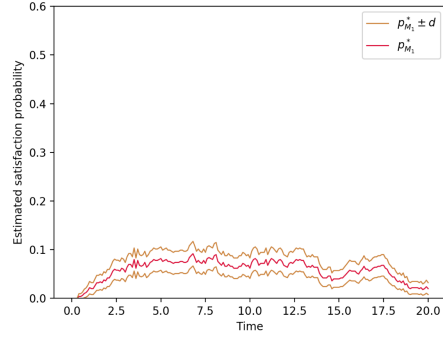
Relevant works are those on spatial logics for process algebra with locations; these logics are specifically defined to describe properties of process algebra models. They are mainly designed to express properties of concurrent systems, in particular for mobile processes and mobile ambients. One example is the Ambient logic [10], where the space is represented as a tree and locations are nested. [16] introduced MoSL for formulating properties of models specified in StoKlaim, a Linda-like communication model with shared memory through tuple spaces.

Other important logic-based formalisms to mention are those that have been proposed for reasoning about the topological [6] or directional [7] aspects of the interacting entities; for networks of processes [39], which can express spatio-temporal properties in discrete time; and for (graph) rewrite theories [2], [36], bigraphs [15], and data structure as graphs [9] and heaps [8]. Model checking of these logics usually has high computational complexity [7] or is even undecidable [35].

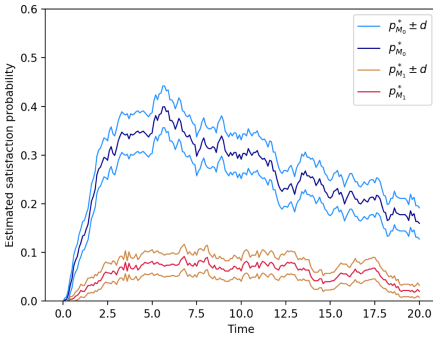
In [13], *Spatial Logic for Closure Spaces* (SLCS) is proposed for a discrete and topological notion of space, based on closure spaces [17]. An extension of the SLCS with temporal aspects, as “snapshot” models, can be found in [14]. It extends the logic with the temporal modalities of the branching logic *Computation Tree Logic* (CTL). However, the algorithms to check snapshot



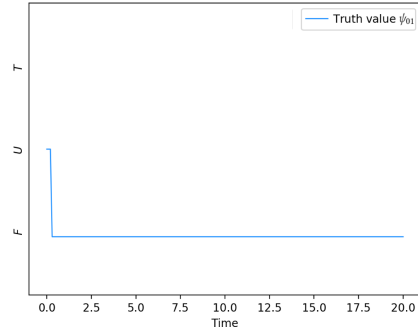
(a) Estimated satisfaction probability of $\varphi_{corrupt}M_0$ (blue) and 95% confidence intervals (light blue)



(b) Estimated satisfaction probability of $\varphi_{corrupt}M_1$ (red) and 95% confidence intervals (orange)



(c) Estimated satisfaction probabilities



(d) Truth value of ψ_{01}

Fig. 11. Fifty initial messages M_0 for sender S_0 , 10 for the other senders S_i .

models have high computational cost and are susceptible to state-space explosion problems because the spatial formulas have to be recomputed at every state. Furthermore, the logic does not have a stochastic semantics.

The only linear-time spatio-temporal logics with efficient monitoring procedures that we are aware of are SSTL [37], described in Section 2; SpaTeL [21]; and STREL [5]. The *Spatial-Temporal Logic* (SpaTeL) is the unification of *Signal Temporal Logic* [33] (STL) and *Tree Spatial Superposition Logic* (TSSL) introduced in [19] to classify and detect spatial patterns. TSSL specifies properties over quad trees, spatial data structures that are designed by recursively partitioning the space into uniform quadrants. The logic permits the capture of very complex spatial structures, but at the price of a complex formulation of spatial properties. These latter, indeed, can be in practice only learned from some template image. The *Spatio-Temporal Reach and Escape Logic* (STREL) has been designed to specify and verify properties of mobile and spatially distributed Cyber Physical Systems. The unique feature of this logic with respect to the two previous ones is that the monitoring procedure in each location depends only on the monitoring of its neighbors. This is the first step in the design of distributed online monitoring algorithms. The logic is relatively new and we believe that, for the flexibility of how we designed TSTL, we will be easily able to extend our logic to express also STREL properties. However, since we are currently using TSTL with a

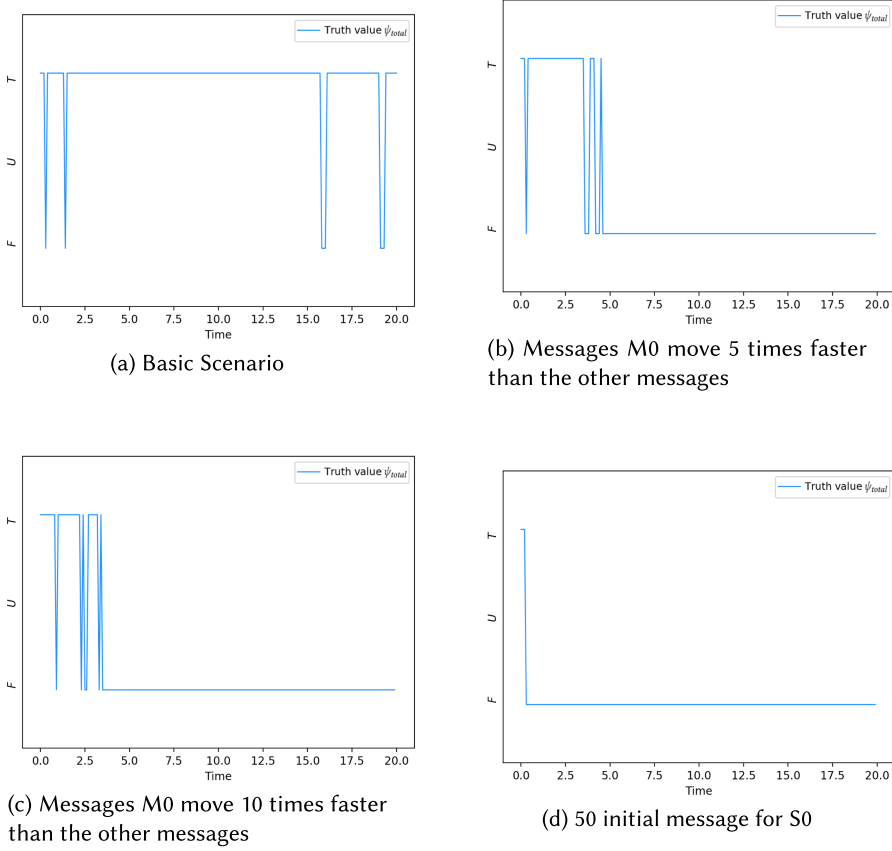


Fig. 12. Complete analysis, TSTL property ψ_{total} in an arbitrary corrupt node.

simulation-based statistical evaluation approach, the online monitoring procedure is less important in our methodology.

10 CONCLUDING REMARKS

In this article, we presented TSTL, an extension of SSTL that allows us to widen the analysis of spatiotemporal properties of stochastic systems. We have shown how this extension is used to study the spatio-temporal evolution of the estimated satisfaction probabilities of given SSTL formulas. We implemented the monitoring algorithms for each TSTL operator and used them in the case studies to perform the novel analysis, checking the validity of different TSTL formulas. We used TSTL to identify the safest evacuation routes during a fire spread and to evaluate the satisfaction of privacy requirements on communication networks. We provide the novel spatio-temporal logic with a three-valued semantics to handle the intrinsic uncertainty related to the statistical methods used to estimate the satisfaction probabilities. The three-valued approach allows us to perform initial analysis with a relatively small set of spatio-temporal trajectories, taking into account the uncertainty; on the other hand, it also provides a decision tool for the number of simulations needed for drawing stronger conclusions, as used in our automatic procedure.

There are many possible directions for future work. In the current framework, we use verification of SSTL formulas as input for TSTL monitoring and the starting point for the spatio-temporal

analysis. However, we highlight that TSTL can be used to predicate on estimated satisfaction probabilities of formulas specified with other logics and also on more general uncertain values with an estimated confidence interval, as long as the required format is maintained (estimated value for each location at each time point).

As future case studies we will apply our framework to model the spread of invasive species, in particular giant hogweed [11]: we will analyze the effectiveness of different control measures to protect areas of interest, such as regional parks, taking into account also the suitability of the different locations for plant colonisation. In particular, we will analyze the difference between prevention (control outside the boundaries of the area) and direct action (eradication when the invasive species are detected inside the area), considering also the expense associated with the different measures.

As a future extension for TSTL we will define and implement the operator *bounded reachable* $\tilde{\mathcal{R}}$. This operator can be seen as a spatial until associated with a path. We will be able to verify properties related with locations reachable within a given cost range and satisfying defined TSTL properties, and the existence of a connecting path formed only by locations satisfying a given set of TSTL properties. In the case study we presented, the use of this new operator would have allowed us to identify safe paths without having to mimic and constrain the actual movement, detecting different possible solutions. Using this new TSTL operator, we could verify if *there is a safe location (assembly point S) that we can reach passing only through locations with low probability of burning, with a cost w , $w \in [w_1, w_2]$* :

$$\psi := (\mathcal{P}_{<0.2}(B > 0)) \tilde{\mathcal{R}}_{[w_1, w_2]} (\mathcal{P}_{>0.01}(S > 0)).$$

As future work for the automatic procedure, we want to investigate an adaptive value of the number of additional spatio-temporal trajectories required. In the current implementation, we acquire one additional spatio-temporal trajectory at a time when the reliability requirement is not yet satisfied and we have not reached the upper bound for $|\Sigma|$. The adaptive value will depend on the current results of the Statistical Model Checking and their evolution during the procedure. Moreover, in a future version of our algorithm, we will introduce a spatio-temporal specialization of this initial version, where we will allow the definition of a specific area of interest and specific time interval for which we want the reliability requirement to be satisfied. This refinement process has several motivations. For example, in the case of a spread of a disease, we will have specific areas of interest, e.g., hospitals, schools, for which we want a quick but accurate evaluation of the risk assessment.

REFERENCES

- [1] Marco Aiello, Ian Pratt-Hartmann, and Johan Van Benthem. 2007. *Handbook of Spatial Logics*. Springer Netherlands.
- [2] Kyungmin Bae and José Meseguer. 2012. A rewriting-based model checker for the linear temporal logic of rewriting. *Electronic Notes in Theoretical Computer Science* 290 (Dec. 2012), 19–36. DOI : <https://doi.org/10.1016/j.entcs.2012.11.009>
- [3] Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. 2003. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering* 29, 6 (June 2003), 524–541. DOI : <https://doi.org/10.1109/TSE.2003.1205180>
- [4] Howard Barringer, Jen Huan Cheng, and Cliff B. Jones. 1984. A logic covering undefinedness in program proofs. *Acta Informatica* 21, 3 (1984), 251–269.
- [5] Ezio Bartocci, Luca Bortolussi, Michele Loreti, and Laura Nenzi. 2017. Monitoring mobile and spatially distributed cyber-physical systems. In *Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE'17)*. ACM, New York, 146–155. DOI : <https://doi.org/10.1145/3127041.3127050>
- [6] Brandon Bennett, Anthony G. Cohn, Frank Wolter, and Michael Zakharyashev. 2002. Multi-dimensional modal logic as a framework for spatio-temporal reasoning. *Applied Intelligence* 17, 3 (Sept. 2002), 239–251. DOI : <https://doi.org/10.1023/A:1020083231504>

- [7] Davide Bresolin, Pietro Sala, Dario Della Monica, Angelo Montanari, and Guido Sciavicco. 2010. A decidable spatial generalization of metric interval temporal logic. In *2010 17th International Symposium on Temporal Representation and Reasoning*. 95–102. DOI: <https://doi.org/10.1109/TIME.2010.22>
- [8] Rémi Brochenin, Stéphane Demri, and Etienne Lozes. 2012. On the almighty wand. *Inf. Comput.* 211 (2012), 106–137. <http://dblp.uni-trier.de/db/journals/iandc/iandc211.html#BrocheninDL12>
- [9] Luca Cardelli, Philippa Gardner, and Giorgio Ghelli. 2002. A spatial logic for querying graphs. In *ICALP (Lecture Notes in Computer Science)*, Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo (Eds.), Vol. 2380. Springer, 597–610. <http://dblp.uni-trier.de/db/conf/icalp/icalp2002.html#CardelliGG02>
- [10] Luca Cardelli and Andrew D. Gordon. 2000. Anytime, anywhere: Modal logics for mobile ambients. In *Proceedings of the 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'00)*. ACM, New York, 365–377. DOI: <https://doi.org/10.1145/325694.325742>
- [11] Stephen Catterall, Alex R. Cook, Glenn Marion, Adam Butler, and Philip E. Hulme. 2012. Accounting for uncertainty in colonisation times: A novel approach to modelling the spatio-temporal dynamics of alien invasions using distribution data. *Ecography* 35, 10 (2012), 901–911. DOI: <https://doi.org/10.1111/j.1600-0587.2011.07190.x>
- [12] Davide Cerotti, Marco Gribaudo, Andrea Bobbio, Carlos T. Calafate, and Pietro Manzoni. 2010. A Markovian agent model for fire propagation in outdoor environments. In *Computer Performance Engineering: 7th European Performance Engineering Workshop EPEW'10. Proceedings*. Springer Berlin Heidelberg, 131–146. DOI: https://doi.org/10.1007/978-3-642-15784-4_9
- [13] Vincenzo Ciancia, Diego Latella, Michele Loreti, and Mieke Massink. 2014. Specifying and verifying properties of space. In *Theoretical Computer Science*, Josep Diaz, Ivan Lanese, and Davide Sangiorgi (Eds.). Springer, Berlin, 222–235.
- [14] Vincenzo Ciancia, Diego Latella, Michele Loreti, and Mieke Massink. 2016. Spatial logic and spatial model checking for closure spaces. In *Formal Methods for the Quantitative Evaluation of Collective Adaptive Systems (Lecture Notes in Computer Science)*. Springer, Cham, 156–201. DOI: https://doi.org/10.1007/978-3-319-34096-8_6
- [15] Giovanni Conforti, Damiano Macedonio, and Vladimiro Sassone. 2007. Static BiLog: A unifying language for spatial structures. *Fundamenta Informaticae* 80, 1–3 (2007), 91–110. <http://dblp.uni-trier.de/db/journals/fuin/fuin80.html#ConfortiMS07>
- [16] Rocco De Nicola, Joost-Pieter Katoen, Diego Latella, Michele Loreti, and Mieke Massink. 2007. Model checking mobile stochastic logic. *Theoretical Computer Science* 382 (2007), 42–70. <http://dblp.uni-trier.de/db/journals/tcs/tcs382.html#NicolaKLLM07>
- [17] Antony Galton. 1999. The mereotopology of discrete space. In *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science (Lecture Notes in Computer Science)*, Christian Freksa and David M. Mark (Eds.). Springer, Berlin.
- [18] Daniel T. Gillespie. 1977. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry* 81, 25 (1977), 2340–2361. DOI: <https://doi.org/10.1021/j100540a008>
- [19] Ebru Gol, Ezio Bartocci, and Calin Belta. 2014. A formal methods approach to pattern synthesis in reaction diffusion systems. In *Proceedings of the IEEE Conference on Decision and Control*, Vol. 2015. DOI: <https://doi.org/10.1109/CDC.2014.7039367>
- [20] Siegfried Gottwald. 2008. Many-valued logic. In *Stanford Encyclopedia of Philosophy*.
- [21] Iman Haghighi, Austin Jones, Zhaodan Kong, Ezio Bartocci, Radu Grosu, and Calin Belta. 2015. SpaTeL: A novel spatial-temporal logic and its applications to networked systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control (HSCC'15)*. 189–198. DOI: <https://doi.org/10.1145/2728606.2728633>
- [22] Hans Hansson and Bengt Jonsson. 1994. A logic for reasoning about time and reliability. *Formal Aspects of Computing* 6, 5 (1994), 512–535. DOI: <https://doi.org/10.1007/BF01211866>
- [23] Carl Huffaker. 1958. Experimental studies on predation: Dispersion factors and predator-prey oscillations. *Hilgardia* 27, 14 (1958), 343–383. DOI: <https://doi.org/10.3733/hilg.v27n14p343>
- [24] Cliff B. Jones, Matthew J. Loeft, and L. Jason Steggle. 2013. Revising basic theorem proving algorithms to cope with the logic of partial functions. *Science of Computer Programming* 94 (2013), 238–252.
- [25] Joost-Pieter Katoen, Daniel Klink, Martin Leucker, and Verena Wolf. 2012. Three-valued abstraction for probabilistic systems. *Journal of Logic and Algebraic Programming* 81, 4 (2012), 356–389. DOI: <https://doi.org/10.1016/j.jlap.2012.03.007>
- [26] Stephen Cole Kleene. 1938. On notation for ordinal numbers. *Journal of Symbolic Logic* 3, 4 (1938), 150–155. <http://www.jstor.org/stable/2267778>
- [27] Axel Legay, Benoît Delahaye, and Saddek Bensalem. 2010. Statistical model checking: An overview. In *Runtime Verification: First International Conference (RV'10). Proceedings*. Springer, Berlin, 122–135. DOI: https://doi.org/10.1007/978-3-642-16612-9_11

- [28] Sébastien Lion and Sylvain Gandon. 2016. Spatial evolutionary epidemiology of spreading epidemics. *Proceedings of the Royal Society of London B: Biological Sciences* 283, 1841 (2016). DOI : <https://doi.org/10.1098/rspb.2016.1170>
- [29] Ludovica Luisa Vissat, Jane Hillston, Michele Loreti, and Laura Nenzi. 2017. Automatic verification of reliability requirements of spatio-temporal analysis using three-valued spatio-temporal logic. In *11th EAI International Conference on Performance Evaluation Methodologies and Tools*.
- [30] Ludovica Luisa Vissat, Jane Hillston, Glenn Marion, and Matthew J. Smith. 2016. MELA: Modelling in ecology with location attributes. In *Proceedings 14th International Workshop on Quantitative Aspects of Programming Languages and Systems (Electronic Proceedings in Theoretical Computer Science)*, Vol. 227. 82–97. DOI : <https://doi.org/10.4204/EPTCS.227.6>
- [31] Ludovica Luisa Vissat, Michele Loreti, Laura Nenzi, Jane Hillston, and Glenn Marion. 2017. Three-valued spatio-temporal logic: A further analysis on spatio-temporal properties of stochastic systems. In *Quantitative Evaluation of Systems (QEST'17)*, Nathalie Bertrand and Luca Bortolussi (Eds.). Springer International Publishing, 317–332. DOI : https://doi.org/10.1007/978-3-319-66335-7_22
- [32] Jan Łukasiewicz. 1970. *Selected Works*. Amsterdam: North-Holland Pub. Co.
- [33] Oded Maler and Dejan Nickovic. 2004. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems: Joint International Conferences on Formal Modeling and Analysis of Timed Systems (FORMATS'04), and Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT'04)*. *Proceedings*. Springer, Berlin, 152–166. DOI : https://doi.org/10.1007/978-3-540-30206-3_12
- [34] Glenn Marion, Xuerong Mao, Eric Renshaw, and Junli Liu. 2002. Spatial heterogeneity and the stability of reaction states in autocatalysis. *Physical Review E* 66, 5 (2002), 051915.
- [35] Maarten Marx and Mark Reynolds. 1999. Undecidability of compass logic. *Journal of Logic Computation* 9, 6 (1999), 897–914. DOI : <https://doi.org/10.1093/logcom/9.6.897>
- [36] José Meseguer. 2008. The temporal logic of rewriting: A gentle introduction. In *Concurrency, Graphs and Models (2008-06-16) (Lecture Notes in Computer Science)*, Vol. 5065. Springer, 354–382. <http://dblp.uni-trier.de/db/conf/birthday/montanari2008.html#Meseguer08>
- [37] Laura Nenzi, Luca Bortolussi, Vincenzo Ciancia, Michele Loreti, and Mieke Massink. 2015. Qualitative and quantitative monitoring of spatio-temporal properties. In *Runtime Verification - 6th International Conference (RV'15)*. 21–37. DOI : https://doi.org/10.1007/978-3-319-23820-3_2
- [38] Laura Nenzi, Luca Bortolussi, and Michele Loreti. 2016. jSSTL - A tool to monitor spatio-temporal properties. In *10th EAI International Conference on Performance Evaluation Methodologies and Tools*.
- [39] John Reif and A. Prasad Sistla. 1985. A multiprocess network logic with temporal and spatial modalities. *Journal of Computer and System Science* 30, 1 (1985), 41–53. DOI : [https://doi.org/10.1016/0022-0000\(85\)90003-0](https://doi.org/10.1016/0022-0000(85)90003-0)
- [40] Koushik Sen, Mahesh Viswanathan, and Gul Agha. 2006. Model-checking Markov chains in the presence of uncertainties. In *Tools and Algorithms for the Construction and Analysis of Systems: 12th International Conference (TACAS'06), Held as Part of the Joint European Conferences on Theory and Practice of Software (ETAPS'06)*. *Proceedings*. Springer, Berlin, 394–410. DOI : https://doi.org/10.1007/11691372_26

Received January 2018; revised March 2019; accepted March 2019