Analysis of spatio-temporal properties of stochastic systems using TSTL

Citation for published version:

Digital Object Identifier (DOI):
10.1145/3326168

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published in:
ACM Transactions on Modeling and Computer Simulation

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
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 paper, we present Three-Valued Spatio-Temporal Logic (TSTL), which enriches the available spatio-temporal analysis of properties expressed in Signal Spatio-Temporal Logic (SSTL), to give further insight into the dynamic behaviour 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:
1 INTRODUCTION

Stochastic analysis has proved to be effective and useful for many systems which encompass behaviour that is not entirely predictable, either through inherent randomness or through abstractions that characterise ranges of behaviour 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 behaviour of the underlying system [3, 22]. In this paper we focus particularly on systems in which, in addition to stochastic behaviour, 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 spread, 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 which, for example, induce a shift in mean behaviour [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 favour 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 spatio-temporal 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 behaviour or to use to compare different systems, whilst 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 behaviour 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 behaviour 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 spatio-temporal logical formulas. In this paper we use the 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 \textit{true}, \textit{false} and a third value \textit{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.

\textbf{Contribution.} The main contribution of this paper 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 \textit{comparison} operator, $\langle \psi, v \rangle$, which allows us to verify whether the truth value of the TSTL property $\psi$ 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.

\textbf{Paper structure.} The paper 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 example 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 conclusion and future directions for investigation.

2 BACKGROUND

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

2.1 Notation

We define a \textit{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 \textit{spatial population model} $M$ is defined as a tuple $M = (S, G, X, X_0, Tr)$ where:

- $S = \{1, \ldots, n\}$ is the set of states that the population agents can take.
We can interpret the dynamical evolution of these models either stochastically as a Markov chain with population counts, given in the spatio-temporal trajectory. The SSTL atomic proposition, the bounded somewhere current one and that at all preceding time instants \( \phi \). 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 paper. We can describe the temporal evolution of our spatial population models using: 

- \( G = (L, E, w) \) 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)
  - \( w : E \to \mathbb{R}_{\geq 0} \) 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.
  - \( X : L \to \mathbb{N}_0^n \), where \( X(l) = (X_1, \ldots, X_n) \in \mathbb{N}_0^n \) is the state vector, that represents the state of the population in each location. \( X_i \), the entries of the vector \( X(l) \), 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 \( \mathbb{N}_0 = \mathbb{N} \cup \{0\} \).
  - \( X_0 : L \to \mathbb{N}_0^n \), where \( X_0(l) \) is the initial state of the state vector, for each location.
  - \( T \) 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 \( \alpha_i \), an update vector \( v_i \) and a rate function \( r_i \). The label belongs to the label set \( L \), which provides information related to the transition, such as the agent performing it and its current location. The vector \( v_i : L \to \mathbb{Z}^n \) 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:

- \( \sigma \), a spatio-temporal trajectory of \( M \); \( \sigma : L \times T_{\sigma} \to \mathbb{N}_0^n \) gives the state of the population vector for each location \( l \in L \) and each time \( t \in T_{\sigma} \), up to the temporal horizon. \( T_{\sigma} = [0, T_D] \subseteq \mathbb{R}_{\geq 0} \), 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 paper.
- \( \Sigma \), 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:

\[
\varphi ::= \mu | \neg \varphi | \varphi \lor \varphi | \varphi \mathcal{U}[t_1,t_2] \varphi | \Phi_{[w_1,w_2]} \varphi | \varphi S_{[w_1,w_2]} \varphi
\]

The SSTL atomic proposition \( \mu \) is of the form \( \mu \equiv (f \geq 0) \), \( f : \mathbb{R}^n \to \mathbb{R} \), an inequality on expressions with population counts, given in the spatio-temporal trajectory. Negation \( \neg \) and disjunction \( \lor \) 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 \( \varphi_2 \) 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 \( \varphi_1 \) holds. SSTL introduces two spatial operators: the bounded somewhere operator \( \Phi_{[w_1,w_2]} \) and the bounded surround operator \( S_{[w_1,w_2]} \), with \( w_1, w_2 \) real values, \( w_1 \leq w_2 \). The bounded somewhere operator requires that the property \( \varphi \) holds in a
location reachable from the current one with a cost \( w, w \in [w_1, w_2] \). With the operator \textit{bounded surround} we can describe the property \( \varphi_1 S_{[w_1, w_2]} \varphi_2 \) of being surrounded by a \( \varphi_2 \)-region, while being in a \( \varphi_1 \)-region: the formula \( \varphi_1 S_{[w_1, w_2]} \varphi_2 \) 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 \( \varphi_1 \) holds, such that its external boundary \( B^+(A) \) contains only locations satisfying \( \varphi_2 \). The external boundary of a subset of locations \( A \) is defined as \( B^+(A) := \{ l \in L \mid l \not\in A \land \exists l' \in A \text{ s.t. } (l, l') \in E \} \). 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 paper.

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 paper.

### 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 \( \varphi \) is interpreted over a spatio-temporal trajectory \( \sigma \) of \( M \), for each location \( l \in L \) and at time \( t \in T^\sigma \), given values in the set \( \mathbb{B} \):

\[
\beta(M, \sigma, l, t, \varphi) \in \mathbb{B}
\]

The satisfaction function \( \beta \) is defined as follows:

\[
\beta(M, \sigma, l, t, \mu) = \mu(\sigma(l, t)) \\
\beta(M, \sigma, l, t, \neg \varphi) = \neg \beta(M, \sigma, l, t, \varphi) \\
\beta(M, \sigma, l, t, \varphi_1 \vee \varphi_2) = \beta(M, \sigma, l, t, \varphi_1) \lor \beta(M, \sigma, l, t, \varphi_2)
\]

\[
\beta(M, \sigma, l, t, \varphi_1 U_{[t_1, t_2]} \varphi_2) = \bigvee_{t' \in [t_1 + t_2]} (\beta(M, \sigma, l, t', \varphi_2) \land \bigwedge_{t'' \in [t, t']} \beta(M, \sigma, l, t'', \varphi_1))
\]

\[
\beta(M, \sigma, l, t, \Phi_{[w_1, w_2]} \varphi) = \bigvee_{l' \in L, w(l, l') \in [w_1, w_2]} \beta(M, \sigma, l', t, \varphi)
\]

\[
\beta(M, \sigma, l, t, \varphi_1 S_{[w_1, w_2]} \varphi_2) = \bigvee_{A \in SR^l_{[w_1, w_2]}, l' \in A} (\bigwedge_{l' \in B^+(A)} \beta(M, \sigma, l', t, \varphi_1) \land \bigwedge_{l'' \in B^+(A)} \beta(M, \sigma, l'', t, \varphi_2))
\]

where the surrounding region, \( SR^l_{[w_1, w_2]} \), is defined as:

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

Note that in the definition of the until formula the disjunction/conjunction operators, which are defined on interval 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 paper.

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.

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 a SSTL property \( \varphi \), we shift the analysis from a single trajectory \( \sigma \) to a set of trajectories \( \Sigma \), 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 \( \varphi \), provided with a confidence interval. For the sake of simplicity, in this paper this interval is assumed to be symmetric, of a certain width \( 2\delta \). To calculate the values \( p^* \) and \( \delta \), the function \( P_\beta \) is defined over the set of trajectories \( \Sigma \), in terms of \( \beta \):

\[
P_\beta(M, \Sigma, l, t, \varphi) = (p^*, \delta)
\]

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 \( \delta \) is based on the standard approach of using the binomial confidence interval and approximating the error distribution by a normal distribution.

\[
p^* = \frac{|\Sigma_T|}{|\Sigma|}
\]

and

\[
\delta = z_\alpha \times \sqrt{\frac{p^*(1-p^*)}{|\Sigma|}}
\]

where \( \Sigma_T = \{ \sigma \in \Sigma : \beta(M, \sigma, l, t, \varphi) = T \} \) 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 MODELLING 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 SSTL 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 behaviours 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 SSTL 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 jSSTL [38], a Java library developed to support monitoring of SSTL properties, as shown in Figure 1. Since SSTL 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 SSTL framework, with all the weights equal to 1.

Fig. 1. \( \Sigma \) is the set of spatio-temporal trajectories, \( \Phi \) the set of SSTL formulas
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 paper 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 centre 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 neighbouring 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:

\[ \varphi_{occ} := \text{Occ} > 0 \quad \varphi_{fire} := B > 0 \]

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

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

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 Three-Valued Spatio-Temporal Logic (TSTL) is given by:

\[ \psi ::= e_1 < e_2 | \neg \psi | \psi \wedge \psi | \psi \Upsilon [t_1, t_2] \psi | \Diamond [w_1, w_2] \psi | \psi \Sigma [w_1, w_2] \psi | \langle \psi, \nu \rangle \]

where \( e_i := \mathcal{P}(\varphi), p \in [0, 1] \), \( \varphi \) is a given SSTL formula and \( \mathcal{P}(\varphi) \) indicates its estimated satisfaction probability. The atomic TSTL formula \( e_1 < e_2 \) 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 $\tilde{\sim}$ and disjunction $\tilde{\lor}$ operators, bounded until $\tilde{U}^{[1, t_2]}$, bounded somewhere $\tilde{\top}^{[w_1, w_2]}$ and bounded surround $\tilde{S}^{[w_1, w_2]}$. 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 $(\psi, \nu)$ verifies if the truth value of the TSTL property $\psi$ is equal to a given truth value $\nu$, where $\nu \in \mathcal{T} = \{T, U, F\}$, 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 spatio-temporal logics; we will use the letter $\phi$ for SSTL formulas and $\psi$ for TSTL ones. We will use the compact notations $\mathcal{P}_{<p}(\phi)$ and $\mathcal{P}_{>p}(\phi)$ to indicate the TSTL formulas $\mathcal{P}(\phi) < p$ and $\mathcal{P}(\phi) > p$ respectively.

5.2 Three-valued semantics

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

$$
\begin{array}{c|ccc}
\tilde{\sim} & T & U & F \\
------------&---&---&---\\
T & T & T & T \\
U & T & U & U \\
F & T & U & F \\
\end{array}
\begin{array}{c|ccc}
\tilde{\lor} & T & U & F \\
--------------&---&---&---\\
T & T & T & T \\
U & U & U & U \\
F & F & F & F \\
\end{array}
\begin{array}{c|ccc}
\tilde{\land} & T & U & F \\
----------&---&---&---\\
T & T & T & T \\
U & U & U & U \\
F & F & F & F \\
\end{array}
$$

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

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

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

$$
E(M, \Sigma, l, t, e_i) = \begin{cases} 
\mathcal{P}_\beta(M, \Sigma, l, t, \phi) = (p^*_i, \delta_1) & \text{if } e_i = \mathcal{P}(\phi) \\
\mathcal{P}_\beta(M, \Sigma, l, t, p) = (p, 0) & \text{if } e_i = p 
\end{cases}
$$

Note that, given a model $M$, a set of spatio-temporal trajectories $\Sigma$ and an SSTL property $\phi$, the evaluation of $e_i$ is a function of time $t$ and location $l$. Consequently, the same holds for $p^*_1, p^*_2, \delta_1$ and $\delta_2$. 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 $e_2$:

$$
E(M, \Sigma, l, t, e_1) = (p^*_1, \delta_1) \quad E(M, \Sigma, l, t, e_2) = (p^*_2, \delta_2)
$$
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:

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

The three-valued satisfaction function \(\tau\) for the TSTL operators is defined as follows, in an analogous manner to SSTL, where the standard Boolean operators \(!, \lor\) and \(\land\) have been replaced with the three-valued operators \(\sim, \overline{\lor}\) and \(\sim\):

\[
\tau(M, \Sigma, l, t, \sim \psi) = \sim \tau(M, \Sigma, l, t, \psi)
\]

\[
\tau(M, \Sigma, l, t, \psi_1 \overline{\lor} \psi_2) = \tau(M, \Sigma, l, t, \psi_1) \overline{\lor} \tau(M, \Sigma, l, t, \psi_2)
\]

\[
\tau(M, \Sigma, l, t, \sim [t_1, t_2] \psi_1 \mathcal{U} \psi_2) = \bigvee_{t' \in [t+t_1, t+t_2]} (\tau(M, \Sigma, l, t', \psi_2) \land \bigwedge_{t'' \in [t, t']} \tau(M, \Sigma, l, t'', \psi_1))
\]

\[
\tau(M, \Sigma, l, t, \sim \mathcal{O}_{[w_1, w_2]} \psi) = \bigvee_{l' \in L, w(l', l') \in [w_1, w_2]} \tau(M, \Sigma, l', t, \psi)
\]

\[
\tau(M, \Sigma, l, t, \sim [w_1, w_2] \psi_1 \mathcal{S} \psi_2) = \bigvee_{A \in S\mathcal{R}_{[w_1, w_2]}} (\bigwedge_{l' \in A} \tau(M, \Sigma, l', t, \psi_1) \land \bigwedge_{l'' \in B^+ (A)} \tau(M, \Sigma, l'', t, \psi_2))
\]

The truth value of the novel \textit{comparison} operator will be:

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

where \(v \in T = \{T, U, F\}\).

Note the similarity between the structure of \(\beta\) and \(\tau\), 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 \textit{everywhere} spatial operator \(\sim \mathcal{O}_{[w_1, w_2]} \) can be defined as:

\[
\sim \mathcal{O}_{[w_1, w_2]} \psi := \sim \mathcal{O}_{[w_1, w_2]} \neg \psi
\]

This requires \(\psi\) to hold in all the locations reachable from the current one with a total cost between \(w_1\) and \(w_2\). The \textit{eventually} \(\mathcal{F}\) and the \textit{globally} \(\mathcal{G}\) operators are defined as usual:

\[
\sim [t_1, t_2] \mathcal{F} \psi := T \mathcal{U} \psi \quad \sim [t_1, t_2] \mathcal{G} \psi := \neg \mathcal{F} \neg \psi
\]

The \textit{eventually} formula holds if \(\psi\) becomes true within \(t_1\) and \(t_2\) time units from the current one, while the \textit{globally} formula requires \(\psi\) 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 $\psi_1$ and $\psi_2$:

$$
\psi_1 := \mathcal{P}_{<p}(\phi_1 \land \phi_2) \\
\psi_2 := \mathcal{P}_{<p}(\phi_1) \sim \mathcal{P}_{<p}(\phi_2)
$$

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 $B$:

$$
\phi_1 := B > 5 \\
\phi_2 := B > 10
$$

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

$$
\tau(M, \Sigma, l, t, \mathcal{P}_{<p}(\phi_1)) = F \\
\tau(M, \Sigma, l, t, \mathcal{P}_{<p}(\phi_2)) = T
$$

This can happen if we choose the value of $p$ between the two estimates, outside their respective intervals. Since $\phi_1 \land \phi_2 \equiv \phi_2$ then:

$$
\tau(M, \Sigma, l, t, \mathcal{P}_{<p}(\phi_1 \land \phi_2)) = \tau(M, \Sigma, l, t, \mathcal{P}_{<p}(\phi_2)) = T
$$

while:

$$
\tau(M, \Sigma, l, t, \mathcal{P}_{<p}(\phi_1)) \sim \tau(M, \Sigma, l, t, \mathcal{P}_{<p}(\phi_2)) = F
$$

Moreover, the first could perhaps be derived empirically from observations, but the second is only expressible with the new logical operator $\sim$ 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.

Fig. 2. $\Sigma$ is the set of spatio-temporal trajectories, $\Phi$ the set of SSTL formulas and $\Theta$ the set of TSTL formulas

### 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 piecewise constant function $s_{\psi, \hat{l}}$ that maps each time $t$, in the finite set of sample times $\mathcal{T}$, with $\tau(M, \Sigma, \hat{l}, t, \psi)$. 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

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

evaluation of $e_1$ and $e_2$ and the atomic proposition $\psi$. Given a model $M$, a set of spatio-temporal trajectories $\Sigma$ 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 for a given location $\hat{l}$. For each time point in $\mathcal{T}$, the evaluations are compared following the SSTL semantics evaluating the truth value of the SSTL inequality $<\,\llcorner$, assigning a truth value to $s_{\psi, t}$. The output of this procedure is $s_{\psi, l}$, which describes the satisfaction of the property $\psi$ for location $\hat{l}$ over time. Algorithm 2 evaluates the satisfaction of the negation operator $\neg$. In this case the algorithm receives as input a location $\hat{l}$ and the SSTL property $\psi_{\neg \neg \psi}$. This procedure simply applies the negation operator to the truth value of $\psi$ for each time point. This algorithm returns the satisfaction of the property $\psi_{\neg \neg \psi}$ over time for location $\hat{l}$. The last algorithm, Algorithm 3, verifies the SSTL bounded surround formula $\psi = \psi_1 \mathcal{S}_{[w_1, w_2]} \psi_2$. 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 $\psi_2$-region, while being in a region satisfying $\psi_1$. For example, we can extract which are the areas in high danger of being on fire, being surrounded by a region which 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 SSTL subformulas $\psi_1$ and $\psi_2$. 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:

1. $\chi_0(l) = s_{\psi_1, l}(t)$
2. $\chi_{i+1}(l) = \chi_i(l) \land (\bigwedge_{l' : (l, l') \in E} (\chi_i(l') \lor s_{\psi_2, l'}(t)))$

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 modelled 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 ($\psi_{\text{fire}}$) and probability greater than 0.01 of being occupied ($\psi_{\text{occ}}$), given the agent movement in the model:

$$\psi_{\text{occ}} := \text{Occ} > 0 \quad \psi_{\text{fire}} := B > 0$$
Algorithm 1:
Three-Valued Spatio-Temporal Logic: atomic proposition

\[ \text{input: } \hat{l}, E_1(\hat{l}, t) = E(M, \Sigma, \hat{l}, t, e_1), E_2(\hat{l}, t) = E(M, \Sigma, \hat{l}, t, e_2), \psi = e_1 < e_2 \]

\[ \text{for all } t \in \mathcal{T} \text{ do} \]
\[ \begin{array}{l}
[p_1(\hat{l}, t) - \delta_1(\hat{l}, t), p_1(\hat{l}, t) + \delta_1(\hat{l}, t)] = E_1(\hat{l}, t) \\
[p_2(\hat{l}, t) - \delta_2(\hat{l}, t), p_2(\hat{l}, t) + \delta_2(\hat{l}, t)] = E_2(\hat{l}, t) \\
\text{if } p_1(\hat{l}, t) + \delta_1(\hat{l}, t) < p_2(\hat{l}, t) - \delta_2(\hat{l}, t) \text{ then} \\
\quad s_{\psi, i}(t) = T \\
\text{else} \\
\quad \text{if } p_1(\hat{l}, t) - \delta_1(\hat{l}, t) > p_2(\hat{l}, t) + \delta_2(\hat{l}, t) \text{ then} \\
\quad \quad s_{\psi, i}(t) = F \\
\quad \text{else} \\
\quad \quad s_{\psi, i}(t) = U \\
\end{array} \]

\[ \text{return } s_{\psi, i} \]

Algorithm 2:
Three-Valued Spatio-Temporal Logic: negation

\[ \text{input: } \hat{l}, \psi_{neg} = \tilde{\neg} \psi \]

\[ \text{for all } t \in \mathcal{T} \text{ do} \]
\[ s_{\psi_{neg}, i}(t) = \tilde{\neg} s_{\psi, i}(t) \]

\[ \text{return } s_{\psi_{neg}, i} \]

(a) \( t = 1 \)  
(b) \( t = 5 \)  
(c) \( t = 7.5 \)  
(d) \( t = 10 \)

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

\[ \psi_{occ} := \mathcal{P}_{>0.01}(\varphi_{occ}) \quad \psi_{fire} := \mathcal{P}_{<0.2}(\varphi_{fire}) \quad \psi_{safe} := \psi_{occ} \land \psi_{fire} \]

The verification output of TSTL property \( \psi_{safe} \) 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:

\[ \varphi_{route} := (EM > 0) \lor (Occ > 0) \lor (P > 0) \]
Algorithm 3:
Three-Valued Spatio-Temporal Logic: bounded surround operator

\textbf{input:} \(\hat{l}, \psi = \psi_1 S_{[w_1, w_2]} \psi_2, T\)

\begin{algorithmic}
\FORALL {\(l \in L\)}
\IF {0 \leq w(\hat{l}, l) \leq w_2}
compute \(s_{\psi_1, l} ;\)
\IF {w(\hat{l}, l) \geq w_1}
compute \(s_{\psi_2, l} ;\)
\ELSE
\(s_{\psi_1, l} = F ;\)
\ENDIF
\ELSE
\(s_{\psi_1, l} = F ; s_{\psi_2, l} = F ;\)
\ENDIF
\ENDFOR
\FORALL {\(t \in T\)}
\FORALL {\(l \in L\)}
\(\chi_{\text{prec}}(l) = T\)
\(\chi(l) = s_{\psi_1, l} ;\)
\WHILE {\(\exists l \in L : \chi_{\text{prec}}(l) \neq \chi(l)\)}
\(\chi_{\text{prec}} = \chi\)
\FORALL {\(l \in L\)}
\(\chi(l) = \chi_{\text{prec}}(l) \lor (\lor_{l' : (l, l') \in E} (\chi_{\text{prec}}(l') \lor s_{\psi_2, l'}))\)
\ENDFOR
\(s_{\psi, l}(t) = \chi(\hat{l})\)
\ENDWHILE
\RETURN \(s_{\psi, l}\)
\end{algorithmic}

\begin{figure}[h]
\centering
\begin{tabular}{cccc}
\includegraphics[width=0.24\textwidth]{figa} & \includegraphics[width=0.24\textwidth]{figb} & \includegraphics[width=0.24\textwidth]{figc} & \includegraphics[width=0.24\textwidth]{figd}
\end{tabular}
\caption{Safe evacuation routes: TSTL property \(\psi_{\text{safeRoute}}\), with different movement rates of the agents, \(t = 0\)}
\end{figure}

\(\phi_{\text{notFire}} := \neg((\text{Occ} > 0) \land (B > 0)) \equiv \neg(\text{Occ} > 0) \lor \neg(B > 0)\)

\(\phi_{\text{Gsafe}} := G_{[0,10]} (\phi_{\text{route}} \land \phi_{\text{notFire}})\)

\(\psi_{\text{safeRoute}} := P_{>0.8} (\phi_{\text{Gsafe}})\)

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 analyse. 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 $R$ and an automatic procedure to assess if more spatio-temporal trajectories are needed to match the requirement. The evaluation of the reliability requirement $R$ is a Boolean value which expresses if $R$ is matched or not, and therefore the need to acquire more spatio-temporal trajectories. This specification $R$ is defined as the proportion of unknown values which we can tolerate during the exploration of the TSTL property satisfaction. As shown in Algorithm 4, given a TSTL formula $\psi$, a set of spatio-temporal trajectories $\Sigma$ and the reliability requirement $R$, we perform TSTL monitoring for each location $l$ of the spatial domain $L$, at each time step of the discrete time set $T$, 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 $R$ 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 $R$ is not matched, we need to carry out more simulations, performing the monitoring procedure of the formulas and checking the reliability again.

We observe that the proportion of unknown values decreases with the number of analysed spatio-temporal 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 whilst 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.

To ensure the termination of our procedure, we specify an upper bound on the number of spatio-temporal 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 $T$, 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
\( \psi_{\text{fire}} := P_{0.8}(\varphi_{\text{fire}}) \)

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 \( R \), describing with \( pU \) the chosen upper bound proportion, and the number of spatio-temporal trajectories \( (st) \) needed to satisfy it.

\[ \text{Algorithm 4:} \]
Automatic verification of the reliability specification

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

Fig. 5. The figure illustrates the framework with the automatic procedure for the reliability requirement verification. \( \Sigma \) is the set of spatio-temporal trajectories, \( \Phi \) the set of SSTL formulas, \( \Theta \) the set of TSTL formulas, \( R \) the reliability requirement.

8 CASE STUDY: ANONYMITY NETWORK

In this second case study we analyse a model of a communication network, focussing 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 5 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 5 exits $E$ and 5 corrupted nodes $C$. All the remaining nodes are honest forwarders. The complete MELA model can be found at https://ludovicalv.github.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:

$$\psi_{ij} := P(\varphi_{\text{corrupt}M_i}) < P(\varphi_{\text{corrupt}M_j})$$

evaluates $U$ for each pair of senders $(M_i, M_j)$. As 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 $\psi_{01}$:

$$\psi_{01} := P(\varphi_{\text{corrupt}M_0}) < P(\varphi_{\text{corrupt}M_1})$$

where:

$$\varphi_{\text{corrupt}M_i} = (M_i > 0) \land (C > 0) \quad i = 0, 1$$
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 analysed 500 spatio-temporal trajectories. In Figure 8 we present the results of this analysis, observing that the truth value of property $\psi_{01}$ evaluates as $U$, implying the estimated satisfaction probabilities of formulas $\phi_{\text{corrupt}M_0}$ and $\phi_{\text{corrupt}M_1}$ are close. We evaluate the same properties under different scenarios, which we can easily describe by using the high-level modelling language MELA. We can consider scenarios in which:

- Messages $M_0$ move 5 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 (5 times more).

The results of these analysis 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 $\psi_{01}$ is evaluated as $T$ after some initial time. With more messages $M_0$ in the network we observe that the probability of finding messages $M_0$ is higher, as expected, and therefore the truth value of $\psi_{01}$...
evaluates as \( F \).

**Complete analysis** We have gone on to analyse the privacy property for all the possible senders, in an arbitrary corrupted node. We have verified the TSTL property \( \psi_{\text{total}} \):

\[
\psi_{\text{total}} := \bigwedge_{i,j} \langle \psi_{ij}, U \rangle
\]

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 afterwards 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 \( \psi_{\text{total}} \) is evaluated as false.
Analysis of spatio-temporal properties of stochastic systems using TSTL

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.

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 modelling 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
Fig. 11. 50 initial messages $M_0$ for sender $S_0$, 10 for the other senders $S_i$.

[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 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, for (graph) rewrite
Analysis of spatio-temporal properties of stochastic systems using TSTL

(a) Basic Scenario

(b) Messages M0 move 5 times faster than the other messages

(c) Messages M0 move 10 times faster than the other messages

(d) 50 initial message for S0

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

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

10 CONCLUDING REMARKS

In this paper we presented Three-Valued Spatio-Temporal Logic (TSTL), an extension of Signal Spatio-Temporal Logic (SSTL) that allows us to widen the analysis of spatio-temporal 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 intervals, 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 analyse 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 analyse 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 $\tilde{R}$. This operator can be seen as a spatial until with direction and 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 $w$, $w \in [w_1, w_2]$:

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

ACKNOWLEDGEMENT

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

REFERENCES


[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 2015. 21–37. https://doi.org/10.1007/978-3-319-23820-3_2

