# Using Agent-Based Simulations to Evaluate Bayesian Networks for Criminal Scenarios

Leeuwen, Ludi van; Verheij, Bart; Verbrugge, Rineke; Renooij, Silja

*Published in:*
19th International Conference on Artificial Intelligence and Law, ICAIL 2023 - Proceedings of the Conference

*DOI:*
10.1145/3594536.3595125

# Using Agent-Based Simulations to Evaluate Bayesian Networks for Criminal Scenarios

Ludi van Leeuwen, Bart Verheij
Rineke Verbrugge
Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen

Silja Renooij
Department of Information and Computing Sciences,
Utrecht University

## ABSTRACT

Scenario-based Bayesian networks (BNs) have been proposed as a tool for the rational handling of evidence. The proper evaluation of existing methods requires access to a ground truth that can be used to test the quality and usefulness of a BN model of a crime. However, that would require a full probability distribution over all relevant variables used in the model, which is in practice not available. In this paper, we use an agent-based simulation as a proxy for the ground truth for the evaluation of BN models as tools for the rational handling of evidence. We use fictional crime scenarios as a background. First, we design manually constructed BNs using existing design methods in order to model example crime scenarios. Second, we build an agent-based simulation covering the scenarios of criminal and non-criminal behavior. Third, we algorithmically determine BNs using statistics collected experimentally from the agent-based simulation that represents the ground truth. Finally, we compare the manual, scenario-based BNs to the algorithmic BNs by comparing the posterior probability distribution over outcomes of the network to the ground-truth frequency distribution over those outcomes in the simulation, across all evidence valuations. We find that both manual BNs and algorithmic BNs perform similarly well: they are good reflections of the ground truth in most of the evidence valuations. Using ABMs as a ground truth can be a tool to investigate Bayesian Networks and their design methods, especially under circumstances that are implausible in real-life criminal cases, such as full probabilistic information.

## CCS CONCEPTS

• **Bayesian Networks**; • **Reasoning with Evidence**; • **Agent-based Models**;

## KEYWORDS

Bayesian Networks, evidential reasoning, agent-based simulation, scenarios

**ACM Reference Format:**
Ludi van Leeuwen, Bart Verheij, Rineke Verbrugge and Silja Renooij. 2023. Using Agent-Based Simulations to Evaluate Bayesian Networks for Criminal Scenarios. In *Nineteenth International Conference on Artificial Intelligence and Law (ICAIL 2023), June 19–23, 2023, Braga, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/https://doi.org/10.1145/3594536.3595125

*ICAIL 2023, June 19–23, 2023, Braga, Portugal*
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0197-9/23/06.
https://doi.org/https://doi.org/10.1145/3594536.3595125

## 1 INTRODUCTION

The goal of evidential reasoning in the courtroom is to find reasons for or against believing the factual circumstances that determine the guilt of a defendant. In a criminal trial, evidence is established. Evidence can support unobserved hypotheses that relate to a defendant's possible offence [3]. This inference is traditionally presented as an argument: 'Evidence $x_1, x_2$ supports the claim that the suspect committed act $X$'. However, in such an argument, it is unclear how much a piece of evidence should ultimately change the judge's belief in a defendant's guilt: should it have a large or a small effect? How should we weigh contradictory evidence? How is this process influenced by the prior beliefs of the judge?

Such questions could be answered using a Bayesian Network (BN) model of that criminal case. A Bayesian Network is a joint probability distribution over a set of relevant variables. It is a normative standard for reasoning with evidence: for a given set of evidence, the BN calculates the probability of the hypothesis. However, creating an acceptable Bayesian Network is not trivial. Bayesian Networks require full probabilistic information, yet, the events in criminal cases are typically very specific, hence it is hard to empirically determine all required probabilities. For events such as motive and opportunity, it is unclear how a probability estimate should be established at all. Hence subjective methods for creating BNs have been proposed [6], which cannot guarantee an accurate representation of an actual joint probability distribution over all events represented, and hence also the inferred probability of guilt is based on subjective elements.

The lack of full probabilistic information is visible in existing methods for building Bayesian Networks in law. When proposing a method for building Bayesian Networks on limited data in the criminal domain, the method is often illustrated by creating a BN that is based on a scenario of a (simplified) criminal case or fictional story [11, 24, 27]. These scenarios do not include probabilistic data; they are only descriptions of events over time. This means that modellers have to elicit subjective probability estimates for every event in the scenario in order to create a BN with no access to any ground truth. The resulting BN can hence not be properly evaluated because there is no objective ground truth distribution to compare the BN distribution to.

In this paper, our aim is to test the scenario idiom method as established by [26]. We construct a Bayesian Network according to the scenario idiom as well as as a control network that is learned from data collected in a simulation, without the use of the scenario idiom. Then we evaluate whether the scenario-based BN represents the ground truth and how it compares to the algorithmically generated network. We can evaluate the BNs because we create an artificial ground truth that includes full probabilistic information, in the form of an agent-based model (ABM). In the ABM, we model alternative scenarios of criminal and non-criminal behaviour, and we record the frequency for each event in the simulation. Hence, we can go beyond the current approach of evaluating Bayesian Networks by including the possibility to evaluate whether the networks, or parts of them, correspond to the probabilistic ground truth. In Section 2, we introduce background on reasoning with evidence, Bayesian Networks and agent-based simulations. Section 3 explains the ABM and how we construct Bayesian Networks. Section 4 compares the performance of the Bayesian Networks as compared to the ground truth. Section 5 evaluates the scenario idiom and limitations and Section 6 presents the conclusion with suggestions for future work.

## 2 BACKGROUND

### 2.1 Reasoning with evidence

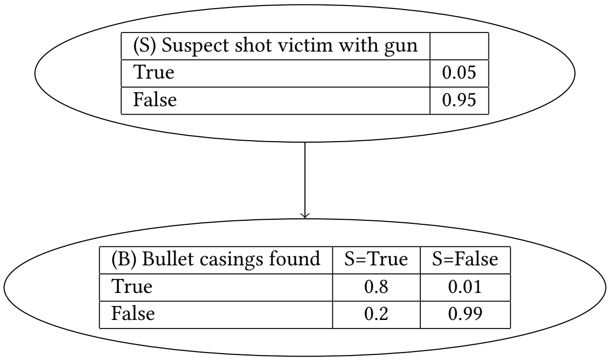
There exist three main approaches to reasoning with evidence within law [7, 25]: argumentation, scenarios and probabilities. In the argumentation approach, hypotheses and evidence are represented as propositions that attack or support each other (going back to [30]). In the scenario approach, coherent hypotheses are combined into stories [22, 28] that are supported by evidence. Evidence needs to be anchored, which means that the evidence needs to be grounded in common-sense rules. The extent to which this is successful determines our belief in the entire story. In the third, probabilistic approach, hypotheses and evidence are assigned probabilities and the relation between hypothesis and evidence, and hypothesis and hypothesis, is represented as conditional probability (e.g., [4]). Bayesian Networks can model criminal cases [11, 12, 17, 24]. A Bayesian Network can also represent specific (forensic) aspects of a case, such as DNA or blood-spatter evidence; for methods see [20]. In this paper, we focus on the design method proposed in [26].

### 2.2 Bayesian Networks

A Bayesian Network $B = \langle V, E, \mathbf{P} \rangle$ is a compact representation of a joint probability distribution $\mathbf{P}$ over a set of variables $V$ [21]. The tuple $\langle V, E \rangle$ is a directed acyclic graph that represents the independence relation of the variables $V$ as nodes and directed edges $E$. Every variable $V_i$ in the network has a conditional probability table (CPT) that captures the probability distributions $\Pr(V_i \mid parents(V_i))$ over $V_i$ conditioned on the combinations of values for the parents $V_j$ of $V_i$ in the graph. $V_j$ is a parent of $V_i$ if the nodes are connected by an edge $E_{j,i}$; $V_i$ is then the child of $V_j$.

We can find the joint probability distribution $\mathbf{P}$ of a network through the chain rule: $\mathbf{P} = \prod_{i=1}^{n} = \Pr(V_i|parents(V_i))$, where $n$ is the number of nodes in the network. We assume that all variables are boolean, with possible outcomes $True$ and $False$.

Figure 1 shows a BN that represents a shooting $S$ leaving evidence in the form of bullet casings on the ground $B$. We might not be able to observe the shooting directly, but we can observe the bullet casings $b$ or $\neg b$. This network is hence an example of an evidence idiom [11]: the observation serves as a piece of evidence for an unobserved hypothesis.



| (S) Suspect shot victim with gun | |
|---|---|
| True | 0.05 |
| False | 0.95 |

| (B) Bullet casings found | S=True | S=False |
|---|---|---|
| True | 0.8 | 0.01 |
| False | 0.2 | 0.99 |

**Figure 1: A Bayesian Network with two nodes, where the evidence is the child node and the hypothesis the parent.**

The evidence idiom reflects a causal relationship between variables in the direction of the arc: the hypothesised event causes the observable evidence. We can specify a prior probability distribution in $S$, and in $B$ we specify conditional probabilities: the probability distribution over $B$ when $S$ is true, and the probability distribution over $B$ when $S$ is false. The edges in a Bayesian Network do not necessarily need to point in a causal direction; they describe probabilistic relationships only [5].

To reason with evidence in our example, let us say that we found $B = True$, or $b$: there are bullet casings on the ground. We want to know what this new evidence will do to our belief in $S = True$, or $s$. This means that we are updating our belief in $S = True$, or finding the posterior of $S = True$, based on the evidence $B = True$ that we found. We can update out belief in $s$ given $b$ using Bayes' Law:

$$P(s|b) = \frac{P(b|s) \cdot P(s)}{P(b)}$$

This requires us to find the probabilities $P(b|s), P(s)$ and $P(b)$. In the case of our network in Figure 1, these are specified in the CPTs.

$$P(b|s) = 0.8, P(s) = 0.05, P(b) = 0.8 \cdot 0.05 + 0.01 \cdot 0.95$$

From this information, we can calculate $P(s|b)$:

$$P(s|b) = \frac{0.8 \cdot 0.05}{0.8 \cdot 0.05 + 0.01 \cdot 0.95} = 0.81$$

We find that our belief in $S = True$ has increased from its prior probability of 0.05 to a posterior probability of 0.808, given evidence $B = True$. On the other hand, if we do not find bullet casings on the ground, using the same calculation but now for $B = False$, our belief in $S = True$ decreases from 0.05 to 0.012.

Hence, we can exactly specify how our belief in a hypothesised event changes, given that we have added some piece of evidence: either $B = True$ or $B = False$. Before we found this evidence, we could not say either $B = True$ or $B = False$, as we did not have evidence either way: we did not even look for bullets. This is the simplest form of a Bayesian update, with one piece of evidence and one hypothesis. In real-life situations, we are reasoning with many pieces of evidence that might be conditioned on more than one hypothesis, resulting in tedious and error-prone calculations if we would do them manually. To construct networks, we can build them manually in (proprietary) software with a GUI like AgenaRisk or Hugin. They can also be built, by hand, or automatically constructed from datasets in PyAgrum [8], a free Python software package, or with the R package bn.learn.

## 2.3 Building Bayesian Networks for crimes

Bayesian Networks can represent evidence in crime cases by integrating aspects of argumentation [1, 29] or scenario theory, through the use of idioms. An idiom is a systematic and consistent way of modelling a pattern that occurs in a criminal case. Examples of idiom based-approaches are the evidence-idiom [11] (Figure 1) and the scenario-idiom [26]. Here we focus on a method developed by Vlek et al. [26] for creating Bayesian Networks using the scenario idiom, which represents mutually exclusive and exhaustive scenarios within one BN. In this idiom, separate hypotheses within a scenario are children of a scenario node, which is mutually exclusive with all other possible scenarios, as set using a constraint node [10]. The hypothesis nodes within one scenario are ordered temporally or causally. Adding positive evidence to one part of a scenario makes the entire scenario more probable and other scenarios less probable. The main focus of this method is on the structural aspects of the BNs, and there is less focus on assigning the probabilities. The resulting networks are evaluated in two ways: either assessing robustness with sensitivity analysis [12], or assessing the change in posteriors for a given set of evidence valuations [26]. However, neither of these methods can answer the question we are interested in: Does the BN reflect the ground truth? Therefore in this paper we create a ground truth using agent-based models.

## 2.4 Agent-Based Models

Agent-based models (ABM) allow researchers to study models in which agents interact with their environment and with other agents [15]. Agents can perceive (part of) the simulation, as well as perform actions that are permitted by their behavioural rules. ABMs can be spatially and temporally specified: the agents can be placed on a grid and exist over a given number of epochs called a run. The modeller has access to an ABM's complete state-space, and has full control over the simulation.

Agent-based models have been used to model crimes. They can be based (partially) on empirical data [32] and model specific crimes in areas, or be based on sociological theories of criminal behaviour, as in [14] or theoretical mechanisms of crimes [2]. For agent-based models to be useful in modelling criminal behaviour, they should adhere to certain standards: the models should be based on empirical data, the model should be replicable and the modeller's choices should be transparent [16].

Agent-based simulations have been combined with Bayesian Networks before [19] for modelling probabilistic events with a spatial aspect, yet not as a ground-truth for design method evaluation and not in the domain of law. Agent-based and Bayesian Network models have a lot to gain from each other: an agent-based model makes spatial and temporal relationships between events explicit and gives meaning to the variables in the BN. On the other hand, a BN is an abstracted summary of the relationships at play in an ABM, that a user can interact with [13].

# 3 METHOD

This section describes the agent-based simulation and the construction of the four networks. [1] Note that the ABM in this work is meant only to test the method for constructing Bayesian Networks, and is not meant as something approaching an empirical ABM of any real crime case.

## 3.1 Scenarios

We start our process with one or more written scenarios. These scenarios can be obtained from abridged or simplified case descriptions of a crime, or they can be wholly fictional (such as in this paper, where the focus is on the evaluation of the BN design method [26], and not on the modeling of an actual crime). The scenarios should contain all and only those hypotheses and evidence that are relevant to the case. Both the prosecution and the defence should be able to select relevant events and their evidence.

### 3.1.1 *Running example.*

We establish three different scenarios of criminal and alternative behaviour. A scenario is true only when all parts of the scenario are true. The central part of the scenario is the theft of, accidental loss of, or nothing happening to, some valuable object. In all scenarios, there are two people walking around the Grote Markt, the main square in Groningen. One person is young, and the other person is old and carries a valuable object.

In scenario 1 (*scn*1), the young person sees the old person carrying an object. They assess whether the object is valuable enough to risk stealing. Then, they consider whether the old person is vulnerable enough to steal from; this establishes motive. If and only if the young person has a motive, they will attempt to sneak up on the old person. If and only if they are able to get close to the old person, they steal from them.

In scenario 2 (*scn*2), the young person might still be doing all of the above (or they might not), however, before they can steal, or if they decide that they are not stealing after all, the old person drops the object accidentally.

In scenario 3, neither scenario 1 *Steal* nor that of scenario 2 *Drop* happens. Both people walk around at Grote Markt and then go home and the object is neither dropped nor stolen. This is *scn*3 and is equivalent to 'neither scenario 1 nor scenario 2'.

For our evidence, we have a psychological evaluation of the young person, which assesses whether they are psychologically capable of stealing from the old person. We also have cameras at Grote Markt that show whether the young person was seen at all, or was seen stealing. Additionally, we know whether the object is gone from the potential victim.

## 3.2 Simulation

We can think of the agent-based model as having two parts: 1) a model of the scenarios, by first identifying and operationalising all relevant agents, objects and environments; and 2) a way of observing the model by identifying and operationalising all relevant events. In this paper, the model of the scenarios is an agent-based simulation and the observation procedures are random variables that map specified in-simulation events to truth values.

[1] Code available at https://github.com/aludi/evaluatingScenarioBNs2023

*3.2.1 Agent-based model.* The agent-based model is a simulation that includes all events for all the scenarios as outlined in step 1 as well as being spatially and temporally defined. It contains agents, objects and an environment. The agents traverse the environment, interact with each other and use objects to perform interactions (like stealing an object).

*3.2.2 Running example.* The simulation was run 10,000 times. One run took 100 epochs, or until both agents were in their goal location at the edge of the simulation (with or without theft).

The environment of the simulation is a discrete grid of size $x = 75$, $y = 50$ that represents the geography of the Grote Markt in Groningen. In real life, the area of interest is approximately 425m x 280m. This means that one cell in the simulation is equivalent to a square of 5.6m x 5.6m in real life. An agent can move 1 cell per epoch, which means that, given an average human walking speed of 1.4m/s, [2] one epoch is equivalent to 4 seconds in real life. For the purposes of this simulation, this spatial and time resolution is high enough.

To simulate which parts of the Grote Markt were accessible to the agents, we converted a map image[3] of the Grote Markt into an agent-readable world by overlaying a grid on the map image. Cells that were filled with solid structures such as buildings were coded as inaccessible, while other cells representing roads or open spaces were coded as accessible. The resulting map was shared by all agents and constrained their possible movements as well as their vision, additionally it was used to calculate sight lines for both cameras and agents. An agent or a camera can only see another agent if there are no inaccessible grid cell on the sight line between the two and the other agent is within their visual range. The environment is shown in Figure 2.

We populate this environment with agents. The agents in this model are created from the base MESA agent class [18], with additional features that are relevant for representing the three scenarios. This means that there are always exactly two agents in the simulation, one older victim-agent and one younger potential thief. These agents have the following **attributes**:

- **role** Either thief or victim.
- **id** The thief has ID 1, the victim has ID 0.
- **object** The thief has an object with a value of 0, the victim's object has a value drawn from a uniform distribution between 500 and 1000.
- **goal location** Once the thief has a motive, the thief's goal location is the current location of the victim. For the victim and thief without motive, it is a random accessible location at the edge of the map.
- **age** The thief's age is 25, the victim's age is drawn from a uniform distribution between 60 and 90.
- **risk threshold** The risk threshold for the thief is randomly drawn from a uniform distribution between 800 and 1200.
- **age threshold** The age threshold signifies at what age an agent considers another agent vulnerable: older agents are more vulnerable than younger agents. The thief's age threshold is randomly drawn from a uniform distribution between 50 and 100. The victim's age threshold is set to 100, older than an agent can be, so the victim never steals.
- **steal state** The possible steal states that the agents could be in, are described as follows:
  - **N** Initial state, failed to steal
  - **MOTIVE** Selected a target (vulnerable and valuable)
  - **SNEAK** Moves towards target's position
  - **STEALING** Attempts to steal
  - **SUCCESS** Successfully stolen object
  - **DONE** Reached initial goal location
  - **LOSER** Stolen from

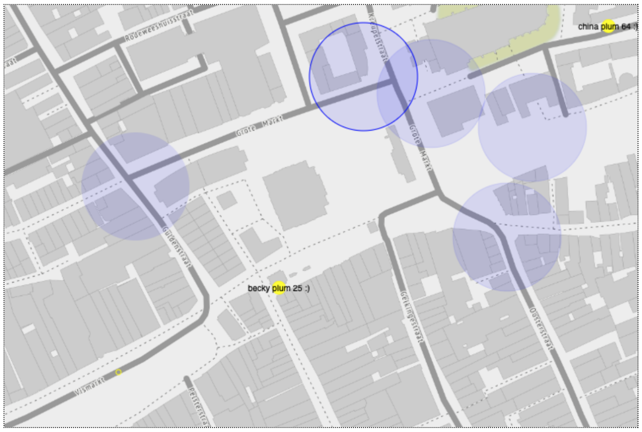


**Figure 2: Map of the Grote Markt, Groningen, as spatial environment in the model. Dark grey represents roads, light grey represents open space; agents can traverse both. Midtone grey represents buildings; agents cannot move through them. Agents are yellow circles, with names attached. Camera locations are randomly initialised. Blue circles represent the camera vision radius.**

All agents can perform the following **actions**:

- **hang-around** agents move around randomly.
- **walk** agents move 1 cell in Moore neighbourhood towards goal location by minimising the distance between goal and self, as measured by Euclidean distance (this does not take buildings into account).
- **escape** agents can get trapped in tight corners of the spatial environment when they are moving towards their goal. There is an epoch tracker that counts when agents get stuck, then moves the agent into the hang-around state of random movement, for a given time. This means that eventually the agent will randomly move away from being stuck.
- **see** agents detect all objects within a radius of 10 cells, as calculated based on line-of-sight. Bresenham's line algorithm was used to select a list of the relevant cells that lie on the grid between the agent and the object that it is seeing. For every cell on the straight path between the two objects, we check whether it is accessible or inaccessible. Agents and light (vision) cannot pass through inaccessible cells. If there is at least one inaccessible cell in the list, then the agent is not able to see the object.

[2] https://en.wikipedia.org/wiki/Preferred_walking_speed
[3] http://maps.stamen.com/terrain/#18/53.21618/6.57225

**decide valuable** once the thief sees the victim, the thief knows the value of the object of the victim. If the value of the object is larger than the risk threshold of the agent, then the object is deemed valuable.

**decide vulnerable** once the thief sees the victim, the thief knows the age of the victim. If the age of the victim is older than the thief's age threshold, then the thief considers the victim vulnerable.

**motive** if an agent decides that the victim is vulnerable and carries a valuable item, the agent has a motive.

**sneak** once an agent has a motive, agent moves to the position of the victim. It can move with a radius of 2 instead of 1, which simulates a higher speed.

**steal** whenever the thief is in the same cell as the victim, the victim still has their object, and they are both present within the model (so the victim has not reached its goal yet), the thief steals successfully.

**drop** agent drops the object accidentally.

There are two types of objects in this model, namely, the valuable object and the cameras. In the scenario, an object is described as being in the possession of an agent, or accidentally dropped. In this model, the object was operationalised as being a feature of the agent, and not as an individual thing by itself. There are 5 cameras in the simulation. They are placed randomly on the accessible cells on the map. Every camera has a visual radius of 5, corresponding to a range of 28m, which is about equivalent to the visual range of real-life security cameras.[4]

*3.2.3 Observation Procedures.* An observation procedure is a random variable that reports the outcome of a relevant event in the simulation. The procedures are embedded in the code. If an event happens (or does not happen) during a run, the observation procedure reports that the event is true (or false). In essence, the observation procedure ($R$) is a random variable (RV) that maps an event ($e$) to a truth value: $R : e \longrightarrow \{0, 1\}$. For every run of the simulation, every random variable $R$ that is defined, is given a value. The random variables that are defined are at the modeller's discretion. The random variables and their outcomes are stored for every run of the simulation.

*3.2.4 Running example.* We define observation procedures $R$ for events that happen in the simulation. There are two types of $R$: *evidence (E)* or *hypothesis (H)*. Every random variable $R$ assigns either a 1 or a 0 to the event that it represents, hence all variables are binary variables. We have $R_1$, ..., $R_8$, described below:

$R_1$ **(H) motive_1_0** agent 1's steal state is MOTIVE, targeting agent 0.

$R_2$ **(H) sneak_1_0** agent 1's steal state is SNEAK, targeting agent 0.

$R_3$**(H) stealing_1_0** agent 1's steal state is STEALING, targeting agent 0.

$R_4$ **(H) object_dropped_accidentally_0** agent 0 drops the object. At every epoch, there is a 1/500 probability that agent 0 drops the object.

$R_5$ **(E) E_psych_report_1_0** agent 1 has the capacity to steal from agent 0. If the thief does not have a motive, no psych report is established: the observation procedure results in 0. If the thief has a motive, there is a 0.9 probability that the psych report indicated that the thief is capable of stealing and a 0.1 probability that the thief could not have stolen the object from agent 0. In this second case, the psych report is incorrect. This observation procedure is not represented spatially in the simulation.

$R_6$ **(E) E_camera_1** agent 1 is seen on the same camera as agent 0.

$R_7$ **(E) E_camera_seen_stealing_1_0** agent 1 is seen on the same camera as agent 0 when agent 1's steal state is STEALING, targeting agent 0.

$R_8$ **(E) E_object_gone_0** agent 0 does not have the object anymore (hence, object is dropped accidentally, or if the object has been stolen).

A run of the simulation in which the thief stole the object and all the evidence pointed in this direction would be represented as $r = (1, 1, 1, 0, 1, 1, 1, 1)$. If we only consider the evidence-events, we would represent this run as (1, 1, 1, 1). We run the simulation 10,000 times, hence we have 10,000 entries: $r_1, \ldots, r_{10000}$.

There are 16 evidence valuations in total, $v_1 \ldots v_{16}$. Some evidence valuations cannot occur in the simulation due to its internal rules. These are the 6 states $v_{11} \ldots v_{16}$: (1, 1, 1, 0), (1, 0, 1, 1), (1, 0, 1, 0), (0, 1, 1, 0), (0, 0, 1, 1) and (0, 0, 1, 0). There are two reasons why these valuations are impossible. First, it is not possible that the thief was not seen on the camera, yet the camera saw the thief stealing (as in the combination (·,0, 1, ·) for $v_{12}, v_{13}, v_{15}, v_{16}$). Second, it is not possible that the camera saw the thief stealing, yet the object was not gone ((·, ·, 1,0), for $v_{11}, v_{13}, v_{14}, v_{16}$). Hence, these valuations do not occur in the simulation and have a frequency of 0.

## 3.3 Construction

We create four Bayesian Networks by combining 2 ways of drawing edges (manual construction $E_{hum}$ and algorithmic construction $E_{alg}$) with 2 ways of assigning probabilities (probabilities drawn from the entire unit interval $P_{unit}$ and drawn from Table 1's finite constrained set $P_{cons}$). The networks are shown

*3.3.1 Structure: The manual method ($E_{hum}$).* For the manual method, we follow the 5 step process as described by Vlek et al. [26]. The network structure was created using Hugin.

(1) **Represent** We add three parent nodes: *scn*1, *scn*2, *scn*3 that are not explicitly represented in the simulation. For each scenario, we select a scenario scheme idiom that the scenario fits. For all scenarios, this would be a temporal ordering of the nodes. For scenario 1: [*motive_1_0* ⟶ *sneak_1_0* ⟶ *stealing_1_0*], as in the simulation, a motive comes before sneaking, which comes before stealing. For scenario 2: [*dropped_0*]. For scenario 3, no events are included. We connect every event as a child to its parent scenario. All scenarios are complete and consistent, which we know due to these being the only events possible in the simulation.

(2) **Unfold** For each scenario, unfold for more detailed subscenarios. Since no event needs to be further unfolded, this step is completed.

(3) **Merge** We use the merged scenarios idiom to combine *scn*1, *scn*2, and *scn*3 by using the constraint node. This means that

[4] https://securitycamcenter.com/how-far-can-security-cameras-see/

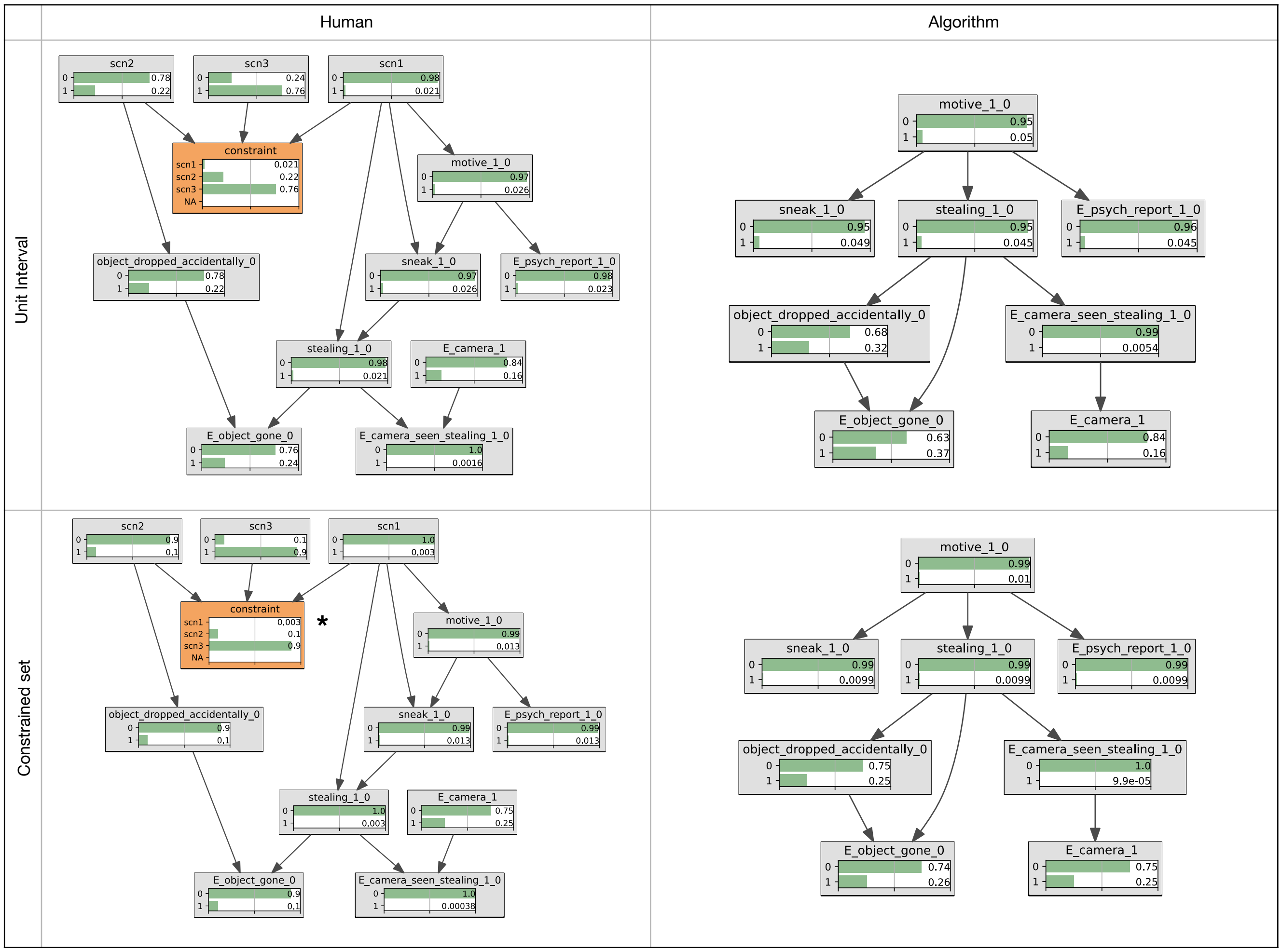


**Figure 3: The four networks that are created based on the simulation. From left to right, top to bottom: the BN with manual structure with full probabilities ($BN_{hum-unit}$); BN with manual structure with constrained probabilities ($BN_{hum-cons}$). the BN with algorithmic structure with full probabilities ($BN_{alg-unit}$); the BN with algorithmic structure with constrained probabilities ($BN_{alg-cons}$). The manual networks have soft evidence entered on the constraint node. (*): The posterior for $BN_{hum-cons}$ does not add up to 1 due to limitation in imaging software.**

we have an additional four nodes in the manual networks as opposed to the algorithmic networks: three scenario nodes and one constraint node.

(4) **Include evidence nodes** We use the evidence idiom from Fenton et al. [10]. For each event, we consider one or more sensible causes that could be the reason for finding the evidence. *E_psych_report_1_0* relates to *motive_1_0* only, *E_object_gone_0* relates to both *stealing_1_0* and *dropped_0*. Finally, the node *E_camera_seen_stealing_1_0* has as parents both *stealing_1_0* and *E_camera_1*.

(5) **Specify probability** We specify the probabilities in the next step. In this paper, we can extract probabilities exactly from the simulation. In cases without a ground truth, probabilities need to be elicited or estimated.

### 3.3.2 *Structure: The algorithmic method ($E_{alg}$).*

We use a structure learning algorithm on the 10,000 run output of the simulation over variables $R_1 \ldots R_8$. We generate a network structure by using a score-based hillclimbing method, using the Bayesian Information Criteria (BIC), as implemented in the R package BN.learn. This results in a set of edges over the 8 variables $E_{alg}$.

### 3.3.3 *Assigning probabilities ($P_{unit}$ and $P_{cons}$).*

We use two ways of assigning probabilities.

The first way is to fill the CPTs with generated probabilities. As the manual network has four more nodes than the algorithmically generated network, this requires extra information. Hence, for every run in the simulation, it was calculated whether scenario 1, scenario 2 or scenario 3 occurred. These are the mutually exclusive and exhaustive options. The scenarios are defined as follows:

| Probability value | Explanation |
|---|---|
| 0 | impossible |
| 0.01 | near impossible |
| 0.25 | uncertain |
| 0.5 | fifty-fifty |
| 0.75 | expected |
| 0.99 | near certain |
| 1 | certain |

**Table 1: The constrained set of probabilities with their explanation. We follow [23, 31], but have added 0.99 and 0.01, which are introduced to avoid rounding to 1 or 0.**

$scn_1 = T \leftrightarrow motive\_0\_1 = T \land sneak\_0\_1 = T \land stealing\_0\_1 = T$, $scn_2 = T \leftrightarrow object\_dropped\_accidentally\_0 = T$, $scn_3 = T \leftrightarrow scn_1 = F \land scn_2 = F$. The constraint node is defined as to result in mutual exclusivity and exhaustivity by setting the probability of combinations of any two or more scenarios, or of no scenario, to 0. For all other nodes, the probabilities were generated by the BN.FIT method with as argument the given network structure and all 10,000 runs of the simulation. The resulting numbers are in the interval $\mathbf{P_{unit}} = [\mathbf{0}, \mathbf{1}]$.

The second way is to use a constrained set of probabilities $\mathbf{P_{cons}} = [\mathbf{0}, \mathbf{0.01}, \mathbf{0.25}, \mathbf{0.5}, \mathbf{0.75}, \mathbf{0.99}, \mathbf{1}]$ that correspond to a natural language interpretation of a degree of certainty (Table 1). The use of this set corresponds to the lower precision of a human expert who has to estimate probabilities subjectively and cannot distinguish between small differences. Finding the probabilities for $P_{cons}$ was done by rounding the values for $P_{unit}$ to the nearest value that is allowed in $P_{cons}$. For example, a value of 0.57 in $P_{unit}$ would become 0.5 in $P_{cons}$.

*3.3.4 The four networks.* In sum, we created 4 Bayesian Networks (Figure 3) with different assumptions about structure and precision in probability assignment:
$BN_{alg-unit} = \langle G_{alg}, P_{unit} \rangle$ with algorithmic graph structure and full precision;
$BN_{hum-unit} = \langle G_{hum}, P_{unit} \rangle$ with manual graph structure and full precision;
$BN_{alg-cons} = \langle G_{alg}, P_{cons} \rangle$ with algorithmic graph structure and constrained precision;
$BN_{hum-cons} = \langle G_{hum}, P_{cons} \rangle$ with manual graph structure and constrained precision.

## 4 RESULTS

We determine to what extent the joint posterior distribution over the three scenarios $scn1, scn2, scn3$ as predicted by each network corresponds to the frequencies of these scenarios in the agent-based simulation. In the manual networks, we can read the posterior probabilities from each scenario from the value in the constraint node. In the algorithmically generated networks and in the ground truth, we need to calculate the posterior probability or frequency of each conjunction. We calculate the joint posterior distribution using the exact inference algorithm LazyPropagation in PyAgrum. We enter soft evidence on the constraint node to ensure that the *NA* option cannot occur, as the scenarios are mutually exclusive and exhaustive, according to Vlek's method [26].

| structure | CPTs | $v_1 \dots v_{16}$ distance | $v_1 \dots v_{10}$ distance |
|---|---|---|---|
| $G_{hum}$ | $P_{unit}$ | 0.141 | 0.025 |
| $G_{hum}$ | $P_{cons}$ | 0.046 | 0.074 |
| $G_{alg}$ | $P_{unit}$ | 0.137 | 0.021 |
| $G_{alg}$ | $P_{cons}$ | 0.140 | 0.026 |

**Table 2: For all Bayesian Networks, the absolute Euclidean distance over all evidence valuations ($v_1 \dots v_{16}$ distance) and over only possible evidence valuations ($v_1 \dots v_{10}$ distance).**

Figure 4 presents the frequency distribution over the three scenarios in the simulation as well as the probability distribution over scenarios as predicted by every BN for all evidence valuations for $v_1 \dots v_{16}$. The first bar, *frequency*, is the ground frequency of scenarios in the simulation for the given evidence valuation. The other bars correspond to the joint posterior probability of each scenario, given the evidence valuation, as predicted by the four Bayesian Networks $BN_{alg-unit}$, $BN_{alg-cons}$, $BN_{hum-unit}$, $BN_{hum-cons}$. Qualitatively, a network predicts the posterior probability of a scenario well when its posterior distribution looks like the frequency distribution in the left column. In general we see in Figure 4 that the performance of the networks is quite high for possible evidence valuations $v_1 \dots v_{10}$. Only for evidence valuations $v_9 = (0, 1, 1, 1), v_6 = (1, 1, 0, 1), v_5 = (1, 0, 0, 1)$ there is a visible difference between $F$ and a given network. In contrast, in the impossible valuation state $v_{15} = (0, 0, 1, 1)$, both manually constructed BNs correctly predict that this evidence valuation is impossible, yet both algorithmic BNs predict a high probability of the stealing scenario. For $v_{11}$ and $v_{14}$, $BN_{hum-unit}$ predicts a drop, yet these evidence states should be impossible.

We compare the difference between the network performance and the ground truth frequency with the Euclidean distance for all 16 evidence valuation $v_1 ... v_{16}$. The scenarios are mutually exclusive, so we can read $\mathbf{P}(scn1|v_i)$ as $\mathbf{P}(scn1 \land \neg scn2 \land \neg scn3|v_i)$:

$$\sum_{i=1}^{16} \frac{1}{16} \sqrt{\begin{array}{c}(\mathbf{F}(scn1|v_i) - \mathbf{P}(scn1|v_i))^2+ \\ (\mathbf{F}(scn2|v_i) - \mathbf{P}(scn2|v_i))^2+ \\ (\mathbf{F}(scn3|v_i) - \mathbf{P}(scn3|v_i))^2\end{array}}$$

A low distance means that the posterior probability of the scenario as reported by the network is close to the observed frequency of the event in the ground truth (Table 2). The average prediction performance of the interval manual network (0.141) is similar to the average performance of the algorithmically constructed networks (0.137, 0.140). The manual network with constrained probabilities even has the best performance in the table, in particular because it is the only one that correctly responds to the impossible evidence valuations in all such cases. Once we take the performance over all possible evidence valuations, we see that constraining the probabilities results in a higher distance from the ground truth, compared to the algorithmically generated networks: 0.074 compared to 0.025 for the other networks.

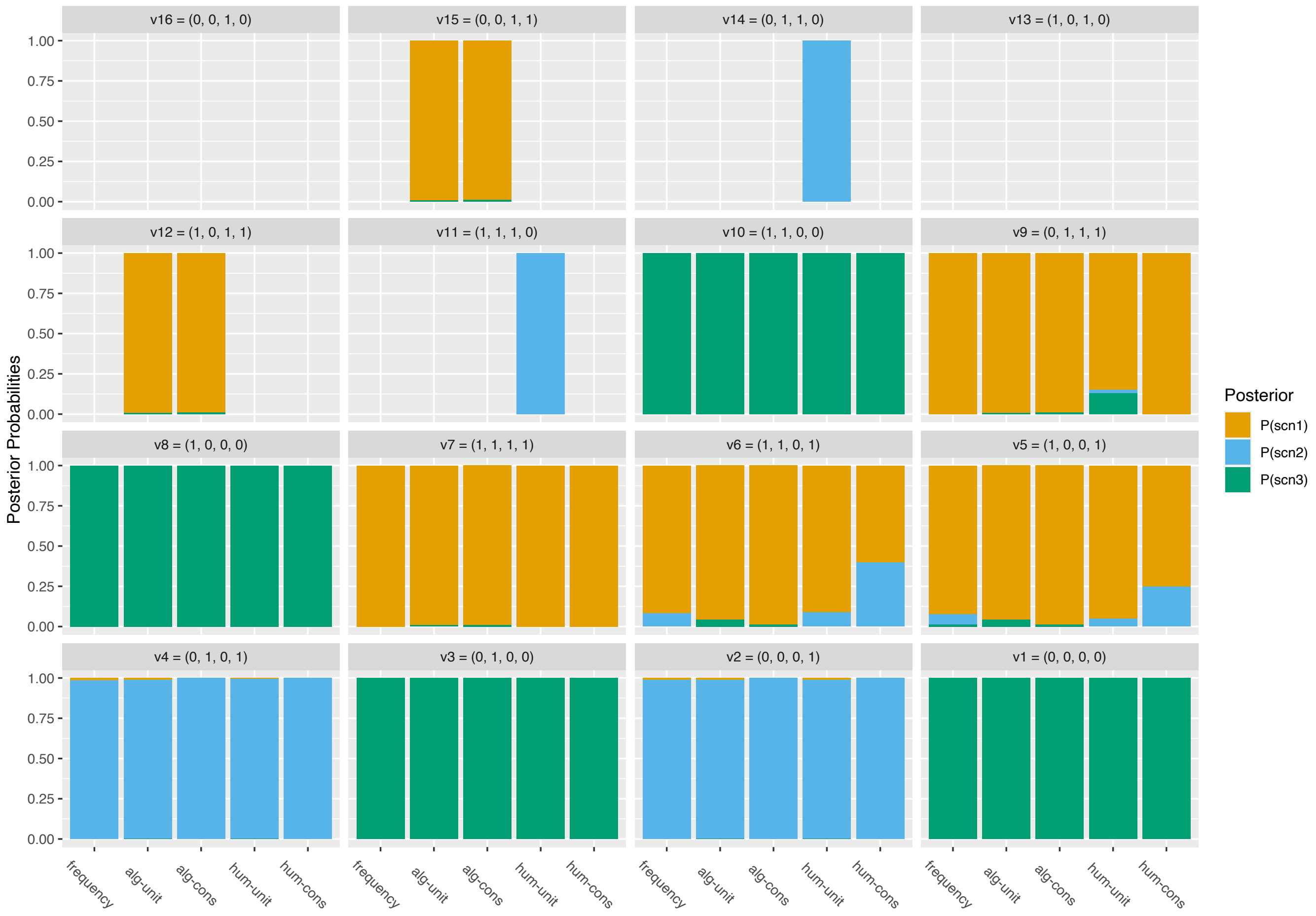


**Figure 4: For all 16 evidence valuations, we show 5 distributions over the three scenarios scn1, scn2 and scn3: the frequency distribution *frequency* in the ground truth and the posterior probability distributions in $BN_{alg-unit}$, $BN_{alg-cons}$, $BN_{hum-unit}$ and $BN_{hum-cons}$. When no distribution is defined, the evidence state is inconsistent.**

## 5 DISCUSSION

We compare the manual and algorithmic BNs, the effect of constrained and unconstrained probabilities, and the use of ABMs to evaluate BNs.

### 5.1 Evaluating the Manual Network

We find that Vlek's method for creating Bayesian Networks with scenario idioms is successful. The performance of the manual networks is similar to the performance of the algorithmic networks, which means that we can use the scenario idiom to structure the BNs without it resulting in a worse performance.

We see that the manual network represents several useful aspects of reasoning explicitly, in contrast to the algorithmic networks: The scenario idiom organizes scenarios and represents them explicitly in the network; mutual exclusivity and exhaustiveness of scenarios are ensured through the constraint node, and we can read the posterior probabilities for every scenario in one glance from the constraint node. Another advantage is that we do not need any data to construct the structure of the manual BNs. This is not possible for the algorithmic BNs.

However, the use of the scenario idiom requires extra nodes beyond the evidence and hypothesis nodes that are used in the algorithmic network. This means, in turn, that there are more parameters to be specified. Specifically, we need to enter the prior probabilities of all three scenarios explicitly into the network as well as parameters for the arcs between the scenario node and the hypotheses that make up the scenario: the manual network has 14 arcs, while the algorithmic network only has 8 arcs. Hence, the manual networks have a higher complexity and require more probabilities to be elicited.

Hence, by following Vlek's method, we find that we can represent scenarios in a more organized and understandable way compared to an algorithmic network, with a similar performance on predicting the ground truth. However, this comes at a cost of a greater complexity in the form of a higher number of arcs, and the necessity of specifying prior probabilities over entire scenarios.

## 5.2 Evaluating the Algorithmic Network

The algorithmic networks perform similarly to the $BN_{hum-unit}$ network and they take advantage of data. However, as we see in Figure 4, these networks were unable to correctly predict the joint posterior distribution over outcomes for $v_{12} = (0, 0, 1, 1), v_{15} = (1, 0, 1, 1)$. It could be that 10,000 runs is an insufficient number for the algorithm to be able to distinguish impossible evidence valuations from implausible evidence valuations. The algorithm does allow for 0 and 1 in the CPTs and this shows in other evidence valuations, yet the algorithm misrepresents these two valuations. However, since the algorithmic network was based on a greedy hill-climbing algorithm, it could be that a correct network was possible, yet this network was excluded early: there might be a network structure that results in a better performance.

## 5.3 Evaluating Constrained and Unit Probabilities

We see something unexpected: the manually constructed network $BN_{hum-cons}$, with the constrained values in the CPT, outperforms all other networks when we consider all evidence valuations. A key reason that this network corresponds to the ground truth so well, is because it responds correctly to all inconsistent evidence valuations: it then does not predict anything, in the sense that the posterior probability, conditioned on an impossible evidence combination, is undefined. It is unexpected that the $BN_{hum-cons}$ has a high performance, because the probabilities in its CPTs are constrained. This means that in theory, it would not be able to fit the frequency data exactly. Hence, we would expect a lower performance and a higher distance between $BN_{hum-cons}$ and *frequency*. However, the constrained CPTs are the reason that this network performs so well. Since this network has more 0's in its CPTs than the unconstrained network, it is able to exclude inconsistent evidence valuations.

This is not the case for all other networks. For the $BN_{hum-unit}$ network, these are $v_{11}, v_{14}$; for the algorithmic networks, these are $v_{12}, v_{15}$. The constrained manual network is able to correctly respond to valuations $v_{11}, v_{14}$. It is able to do this because, in the node *stealing_1_0*, the network $BN_{hum-cons}$ has 0 and 1 in its CPTs where the $BN_{hum-unit}$ network has 0.0004 and 0.9995. The 0 in the CPT of the constrained network results in an inconsistent evidence valuation, and the network breaks down. For the algorithmic, constrained network, this does not occur: In $BN_{alg-cons}$, the CPT contains 0.99 and 0.01, instead of 0 and 1. This allows for the propagation of evidence that should be inconsistent, namely, evidence that is inconsistent in the ground truth in the ABM.

Hence, BNs with constrained probabilities are sometimes better when we set evidence in the network that is inconsistent or mutually exclusive, given the domain model. This suggests that when modelling evidence in a crime, we need to consider when certain pieces of evidence are, or should be, mutually exclusive. This qualitative information can then inform the parameters that a modeller enters into a BN. This could guide elicitation.

## 5.4 Evaluating Bayesian Networks with ABMs

Our ABM approach allows us to investigate methods for creating Bayesian Networks, such as the scenario idiom. We can investigate this idiom based not just on features of structure as in [26], but instead test its probabilistic performance over all evidence states. Hence, we find that, given that we have the data available, the network is able to correctly represent the ground truth for most evidence valuations.

Since we have the ground truth, we find that one of the limiting factors for reasoning with evidence with Bayesian Networks is the availability of conclusive evidence. For example, for the evidence valuations $v_5 = (1, 0, 0, 1), v_6 = (1, 1, 0, 1)$ in the ground truth, there is room for reasonable doubt: $F(scn1) = 0.9$, so the suspect is probably guilty, yet $F(scn2) = 0.1$. If we consider the threshold for reasonable doubt to be at 0.99, we should not convict. Hence, this Bayesian Network, with these 4 evidence nodes, is in this case not able to conclude whether a suspect is guilty: Uncertainty remains, as it would in real life. When we consider this fact in a real-life context, we should consider gathering more evidence, so that, given a certain valuation over this evidence, we would be able to, on the new evidence set, convict the suspect beyond a reasonable doubt. The 0.99 threshold as a guideline could help us to know when we have to look for new evidence, and when we have to stop looking.

One advantage of using ABMs in particular to ground the Bayesian Network, compared to other methods of combining scenarios and probabilities, is that ABMs require explicit and specific definitions of the events in simulation that correspond to given variables in the BN. In the real world, some proposition $p$ can have many different interpretations, hence the meaning of the random variable $RV(p)$ is difficult to define: what events are we measuring and when do they count? However, in the ABM, we know exactly what proposition $p$ means, because it is whatever is triggered in the code, which means that events are always explicitly defined [13]. Operationalising some of the variables in the BN in the legal domain is difficult.

The ABM rests on many assumptions that do not reflect reality and on features that cannot be empirically established. Hence, the ABM is only suitable for creating a ground truth on which to test modelling methods, not an accurate reflection of human behaviour. Whether we can implement realistic criminal patterns of behaviour in an ABM is an open question. For one, this depends on the scale of the behaviour: Modelling complex and more realistic agent behaviour means modelling at a higher resolution in order to include more events. This would, in turn, result in a Bayesian Network with many more nodes. When we only look at the evaluation method, we know that this method has a complexity of $2^e$, where $e$ is the number of evidence nodes; it is implausible that we would be able to use it on very large networks that model granular behaviour. In our setting, $e$ equals 4, resulting in 16 combinations, just about manageable (cf. Figure 4). Hence it is unlikely that we could use this approach to evaluate entire Bayesian Networks that are created for real criminal cases, because these networks would become too complex.

# 6 CONCLUSION AND FUTURE RESEARCH

We have shown how ABMs can provide a ground truth for the evaluation of BN methods for modelling of evidence in crime cases, over all evidence states. We saw that the manual networks constructed following [26] perform similarly to, and in some respects better than, Bayesian Networks that are constructed from structure-learning algorithms on an artificial ground truth as represented by an agent-based simulation.

However, our findings do not directly generalise to real-life crime investigation settings, since our setting is relevantly different from a realistic crime investigation. In particular: In real situations, we do not know whether the scenarios we are considering are actually mutually exclusive and exhaustive, which is a requirement of the scenario idiom. We do not know whether all relevant evidence is included in the network. We do not have full probabilistic information. Even if we can construct the scenario-based BN, we might not be able to get a good performance, because we do not know the relevant probabilities for an evidence node, a hypothesis node, or a scenario node. This final problem is especially the case for elicitation of open-ended events that can be modeled in various ways, such as motive and opportunity.

We have shown that BNs constructed with the scenario idiom can model a ground truth about crime scenarios, given sufficient access to such a ground truth. We have used ABMs to evaluate the scenario idiom in this work. In future work, we could use the ABM approach to evaluate other methods for structuring networks or establishing probabilities, such as the Opportunity Prior [9]. The Opportunity Prior is one proposed method for establishing the prior probability of the guilt of a defendant, as based on their proximity to the crime scene. Since proximity to a crime scene can be modelled with ABMs, we can investigate whether the Opportunity Prior provides a sensible, responsible approach to establishing probabilities in a BN with our method. Additional further research would be needed to investigate the relation of posterior probabilities to reasonable doubt and the stopping criterion, i.e., when to stop looking for further evidence.

## ACKNOWLEDGMENTS

This research was funded by the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research, https://hybrid-intelligence-centre.nl.

## REFERENCES

[1] F. Bex and S. Renooij. 2016. From arguments to constraints on a Bayesian Network. In *Computational Models of Argument. Proceedings of COMMA 2016.* IOS Press, Amsterdam, 95–106.

[2] D. Birks, M. Townsley, and A. Stewart. 2012. Generative explanations of crime: Using simulation to test criminological theory. *Criminology* 50 (02 2012), 221–254. https://doi.org/10.1111/j.1745-9125.2011.00258.x

[3] R. Cook, I. W. Evett, G. Jackson, P. J. Jones, and J. A. Lambert. 1998. A hierarchy of propositions: Deciding which level to address in casework. *Science and Justice* 38, 4 (1998), 231–239.

[4] C. Dahlman. 2020. De-biasing legal fact-finders with Bayesian thinking. *Topics in Cognitive Science* 12, 4 (2020), 1115–1131.

[5] A. P. Dawid. 2010. Beware of the DAG! In *JMLR Workshop and Conference Proceedings: Volume 6. Causality: Objectives and Assessment (NIPS 2008 Workshop)*, I. Guyon, D. Janzing, and B. Schölkopf (Eds.). jmlr.org, 59–86.

[6] J. A. de Koeijer, M. J Sjerps, P. Vergeer, and C. EH Berger. 2020. Combining evidence in complex cases - A practical approach to interdisciplinary casework. *Science & Justice* 60, 1 (2020), 20–29.

[7] M. Di Bello and B. Verheij. 2018. Evidential reasoning. In *Handbook of Legal Reasoning and Argumentation*, G. Bongiovanni, G. Postema, A. Rotolo, G. Sartor, C. Valentini, and D. N. Walton (Eds.). Springer, Dordrecht, 447–493.

[8] G. Ducamp, C. Gonzales, and P. Wuillemin. 2020. aGrUM/pyAgrum : A toolbox to build models and algorithms for probabilistic graphical models in Python. In *10th International Conference on Probabilistic Graphical Models (Proceedings of Machine Learning Research)*, Vol. 138. Skørping, Denmark, 609–612. https://hal.archives-ouvertes.fr/hal-03135721

[9] N. Fenton, D. Lagnado, C. Dahlman, and M. Neil. 2017. The opportunity prior: A simple and practical solution to the prior probability problem for legal cases. In *The 16th International Conference on Artificial Intelligence and Law (ICAIL 2017). Proceedings of the Conference.* ICAIL, ACM, New York (New York).

[10] N. Fenton, M. Neil, and D. A. Lagnado. 2011. Modelling mutually exclusive causes in Bayesian Networks. (2011). Available online: http://www.eecs.qmul.ac.uk/~norman/papers/mutual_IEEE_format_version.pdf.

[11] N. Fenton, M. Neil, and D. A. Lagnado. 2012. A general structure for legal arguments about evidence using Bayesian Networks. *Cognitive Science* 37, 1 (Oct 2012), 61–102. https://doi.org/10.1111/cogs.12004

[12] N. Fenton, M. Neil, B. Yet, and D. Lagnado. 2019. Analyzing the Simonshaven case using Bayesian Networks. *Topics in Cognitive Science* 12, 4 (Mar 2019), 1092–1114. https://doi.org/10.1111/tops.12417

[13] A. Gebharter and D. Koch. 2021. Combining causal Bayes Nets and Cellular Automata: A hybrid modelling approach to mechanisms. *The British Journal for the Philosophy of Science* (2021).

[14] C. Gerritsen. 2015. Agent-based modelling as a research tool for criminological research. *Crime Science* 4, 1 (2015). https://doi.org/10.1186/s40163-014-0014-1

[15] N. Gilbert and P. Terna. 2000. How to build and use agent-based models in social science. *Mind & Society* 1, 1 (2000), 57–72.

[16] E. R. Groff, S. D. Johnson, and A. Thornton. 2019. State of the art in agent-based modeling of urban crime: An overview. *Journal of Quantitative Criminology* 35, 1 (2019), 155–193. https://doi.org/10.1007/s10940-018-9376-y

[17] J. B. Kadane and D. A. Schum. 1996. *A Probabilistic Analysis of the Sacco and Vanzetti Evidence.* Wiley, Chichester.

[18] J. Kazil, D. Masad, and A. Crooks. 2020. Utilizing Python for agent-based modeling: The Mesa framework. In *Social, Cultural, and Behavioral Modeling*, R. Thomson, H. Bisgin, C. Dancy, A. Hyder, and M. Hussain (Eds.). Springer International Publishing, Cham, 308–317.

[19] V. Kocabas and S. Dragicevic. 2013. Bayesian Networks and agent-based modeling approach for urban land-use and population density change: A BNAS Model. *Journal of Geographical Systems* 15, 4 (October 2013), 403–426. https://doi.org/10.1007/s10109-012-0171-2

[20] R. Meester and K. Slooten. 2021. *Probability and Forensic Evidence.* Cambridge University Press, Cambridge.

[21] J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Francisco (California).

[22] N. Pennington and R. Hastie. 1993. *Inside the Juror.* Cambridge University Press, Cambridge, Chapter The Story Model for Juror Decision Making, 192–221.

[23] S. Renooij and C. Witteman. 1999. Talking probabilities: Communicating probabilistic information with words and numbers. *International Journal of Approximate Reasoning* 22, 3 (1999), 169–194.

[24] L. van Leeuwen and B. Verheij. 2019. A comparison of two hybrid methods for analysing evidential reasoning. In *Legal Knowledge and Information Systems: JURIX 2019: The Thirty-second Annual Conference*, M. Araszkiewicz and V. Rodríguez-Doncel (Eds.). IOS Press, Amsterdam, 53–62.

[25] B. Verheij, F. Bex, S. T. Timmer, C. S. Vlek, J.J Ch. Meyer, S. Renooij, and H. Prakken. 2015. Arguments, scenarios and probabilities: Connections between three normative frameworks for evidential reasoning. *Law, Probability and Risk* 15, 1 (Sep 2015), 35–70. https://doi.org/10.1093/lpr/mgv013

[26] C. S. Vlek, H. Prakken, S. Renooij, and B. Verheij. 2016. A method for explaining Bayesian networks for legal evidence with scenarios. *Artificial Intelligence and Law* 24, 3 (2016), 285–324. https://doi.org/10.1007/s10506-016-9183-4

[27] C. S. Vlek. 2016. *When Stories and Numbers Meet in Court. Constructing and Explaining Bayesian Networks for Criminal Cases with Scenarios. Dissertation.* University of Groningen, Groningen.

[28] W. A. Wagenaar, P. J. van Koppen, and H. F. M. Crombag. 1993. *Anchored Narratives. The Psychology of Criminal Evidence.* Harvester Wheatsheaf, London.

[29] R. Wieten, F. Bex, S. Renooij, and H. Prakken. 2019. Constructing Bayesian Network graphs from labeled arguments. In *European Conference on Symbolic and Quantitative Approaches with Uncertainty.* Springer, Cham, 99–110.

[30] J. H. Wigmore. 1931. *The Principles of Judicial Proof or the Process of Proof as Given by Logic, Psychology, and General Experience, and Illustrated in Judicial Trials, 2nd ed.* Little, Brown and Company, Boston (Massachusetts).

[31] C. Witteman and S. Renooij. 2003. Evaluation of a verbal–numerical probability scale. *International Journal of Approximate Reasoning* 33, 2 (2003), 117–131. https://doi.org/10.1016/S0888-613X(02)00151-2

[32] H. Zhu and F. Wang. 2021. An agent-based model for simulating urban crime with improved daily routines. *Computers, Environment and Urban Systems* (2021).