Agent-based Modelling for Security Risk Assessment

Stef Janssen and Alexei Sharpanskykh

Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands
{S.A.M.Janssen,O.A.Sharpanskykh}@tudelft.nl

Abstract. Security Risk Assessment is commonly performed by using traditional methods based on linear probabilistic tools and informal expert judgements. These methods lack the capability to take the inherent dynamic and intelligent nature of attackers into account. To partially address the limitations, researchers applied game theory to study security risks. However, these methods still rely on traditional methods to determine essential model parameters, such as payoff values. To overcome the limitations of traditional methods, we propose an approach which combines agent-based modelling with Monte Carlo simulations. Agent-based models allow more realistic representation of essential aspects and processes of socio-technical systems at cognitive, social and organisational levels. Such models can be used to estimate risks and parameters related to them. An application of the approach is illustrated by a case study of an airport security checkpoint.

1 Introduction

Security Risk Management is a field in which one aims to identify, calculate and mitigate security risks of a system by utilizing a finite set of resources. An important step within Security Risk Management is Security Risk Assessment, in which one aims to qualitatively or (semi-)quantitatively define security risks. A commonly used method to do this is the Threat, Vulnerability & Consequence (TVC) methodology [18], of which an adaptation is outlined in Figure 1.

In this method, *Threat Identification* forms the first step, where a set of security scenarios is identified. Then, for each identified security scenario, *Consequence Assessment* is performed, where one aims to quantify losses in case the identified security scenario were to happen. *Threat Likelihood Assessment* is then used to estimate the probability that the security scenario will happen in some time period. *Vulnerability Assessment* is performed to determine the probability that all defense measures in the security scenario fail, and thus, the attackers are successful. *Risk* then forms the product of each of these three aforementioned factors. In Security Risk Management these risks values are then used to setup proper defense measures.

In general, each of the steps is quantified using analytic tools at the disposal of a security expert. This can for instance be linear probabilistic tools like Event trees [5], historical data, intelligence data and the experience of security experts

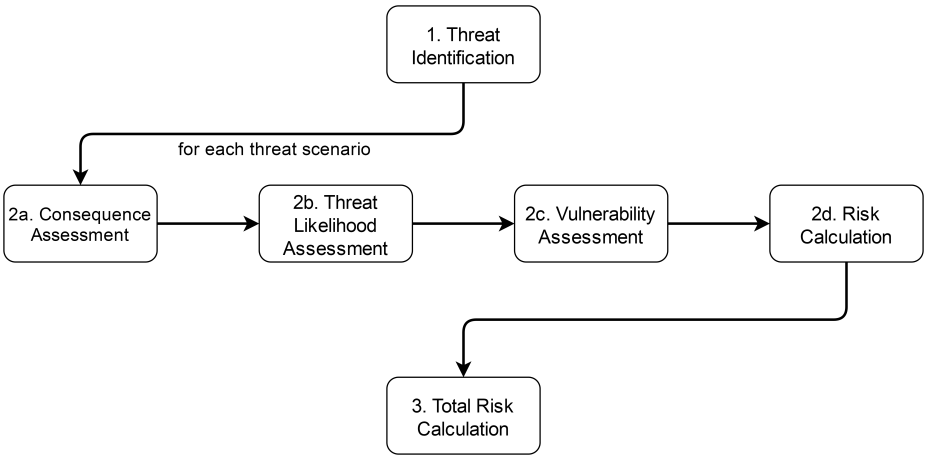

Fig. 1. The Threat, Vulnerability & Consequence (TVC) methodology.

[9, 18]. It is often observed that these methods do not properly take the inherent dynamic and intelligent nature of an adversary into account [4, 11].

To partially overcome this problem, researchers applied game theoretic methods that model a security scenario s_i as a security game [3, 13]. In such a security game, a defender agent and an attacker agent are modelled as the respective row and column players of this game. Columns represent the options an attacker has to attack a target, while rows represent the available actions the defender has to defend the target. Based on the chosen strategies of the attacker and defender a pay-off is determined.

While security games allow for the modelling of intelligent and dynamic adversaries, they still require the definition of pay-off values. These pay-off values still have to be defined by relying on the above discussed methods to quantify *Vulnerability* and *Consequence*.

We therefore propose an Agent-based modelling and simulation method, which forms a promising alternative method for *Vulnerability* and *Consequence* assessment. It is capable of more realistic modelling of the underlying socio-technical processes, often problematic for the above mentioned methods. It can include rich cognitive, social and organisational models and explicit representation of the environment. As these models form a closer representation of the underlying socio-technical system, this can lead to improved estimates of security risks. It further reduces dependency on security experts and leads to more consistent quantitative results. Further, results of this method can be used as input for both the TVC methodology and game-theoretic method described above.

This paper sets a first step towards the development of this approach. We provide an illustrative case study in the area of an airport security checkpoint, and show the results of some basic experiments.

This paper is structured as follows. Section 2 provides an overview of the Agent-based Security Risk Assessment approach. Then, Section 3 discusses the details of a case study and the associated model that illustrate the workings of the Agent-based Security Risk Assessment approach. Section 4 discusses the experiments that were performed with the model, and finally, Section 5 states the conclusions of this work and the possible directions for future research.

2 An Overview of Agent-based Security Risk Assessment

In this section, we describe our Agent-based Security Risk Assessment method to estimate *Vulnerability* and *Consequence*. The method focuses on outcomes of specific security scenarios, and *Threat Likelihood* is therefore not considered. For generality purposes we do not commit to a specific MAS architecture, but merely describe the set of agents and environment objects present in the underlying Agent-based model. A more concrete example that applies this Agent-based Security Risk Assessment method can be found in Section 3.

Agent-based simulation model m_i replicates and elaborates on some security scenario s_i. It contains the following sets of agents: D_i, A_i and O_i. The set D_i contains defender agents, the agents that are responsible for the defense in s_i. A_i is the set of adversary agents, executing the subversive actions in security scenario s_i. O_i is the set of other agents present in s_i. This can for instance be a set of pedestrians or airport passengers. The set of environment objects E_i, then represent the environment objects present in s_i.

Consequence and *Vulnerability* are estimated using a Consequence function and a Fail function respectively. We define the (real-valued) Consequence function $C(m_i^j)$ determining the *Consequence* value for simulation run j, denoted m_i^j. This Consequence function incorporates estimates of direct losses and indirect losses. Direct losses for instance include fatalities and physical damages of an attack are estimated from m_i^j. Indirect losses like decreased number of future passengers and business disruptions are then based on the estimated direct losses and historical data. A boolean Fail function $F(m_i^j)$ is defined, determining the adversaries' success (and therefore the failure of the defense) in m_i^j. The function is equal to 1 if the defenders failed and 0 otherwise. Monte Carlo simulations are performed to estimate *Consequence* and *Vulnerability* values. This is done by performing N simulations and calculating the following estimates of *Consequence* and *Vulnerability* in s_i respectively.

$$\hat{C}(m_i) = \frac{\sum_{j=1}^{N} C(m_i^j)}{N}$$

$$\hat{F}(m_i) = \frac{\sum_{j=1}^{N} F(m_i^j)}{N}$$

This approach can easily be extended to multiple security scenarios of a system by replacing the set of adversary agents with a new set that executes different actions. The next section will describe a case study to illustrate the workings of this approach.

3 Illustrative Case Study

To illustrate the workings of the agent-based approach for Security Risk Assessment, a case study in the area of airport security is elaborated. In this case study, a terrorist aims to bring an improvised explosive device (IED) past a security checkpoint of an airport in his/her carry on luggage. Employees of the security checkpoint aim to find illegal items of passengers, while being under constant (time) pressure influencing their performance.

An agent-based modelling framework is defined and outlined in Figure 2. In this framework, Human Agents and an Environment are distinguished. These elements will be discussed in the following subsections.

3.1 Human Agent

A human agent is the representation of a human in the airport environment. Human agents can interact with their environment, other (human) agents and have a (set of) goal(s) that they want to complete. Based on the works of Blumberg [1], Hoogendoorn [8] and Reynolds [14] we distinguish three levels of abstraction in a human agent: the Motivation Layer, the Task Layer and the Motor Layer. The Motivation Layer is responsible for high-level goal planning, (processing of) communication with other agents and the selection of activities. It further is responsible for setting and reaching high level goals. The Task Layer is responsible for the execution of specific activities and navigation. Then, the Motor Layer is responsible for low level interactions with the environment. It is responsible for sensing the environment and determines and executes the next move accordingly.

Three different types of human agents are distinguished: defending agents, passengers and attacker agents.

Defending Agents Defending Agents in this model work at the security checkpoint to detect illegal items from passengers. They form the boundary between the secure and public areas of the airport. Four types of checkpoint employees exists, each having a different task within checkpoint operations: WTMD officer, Bag Checker officer, X-Ray officer and Directions officer.

The X-Ray officer is discussed in detail, while other employees are modelled in a similar fashion. The X-Ray officer has one activity, the *detect illegal items activity*, which is always active. In this activity, the X-Ray officer observes the output of the X-Ray machine he/she controls. An observation of an X-Ray machine is interpreted by the X-Ray officer to determine if the bag under consideration contains an illegal item. If an illegal item was detected, it is communicated to the Bag Checker officer, who then manually checks the bag. Three relevant parameters are distinguished: T_{base} representing the mean processing time of an observation, FN_{base} representing the false negative probability (i.e. the bags that *did* contain an illegal item, but were not observed by the X-Ray officer) and FP_{base} representing the false positive probability (i.e. the bags that *did not* contain an illegal item, but were identified as such).

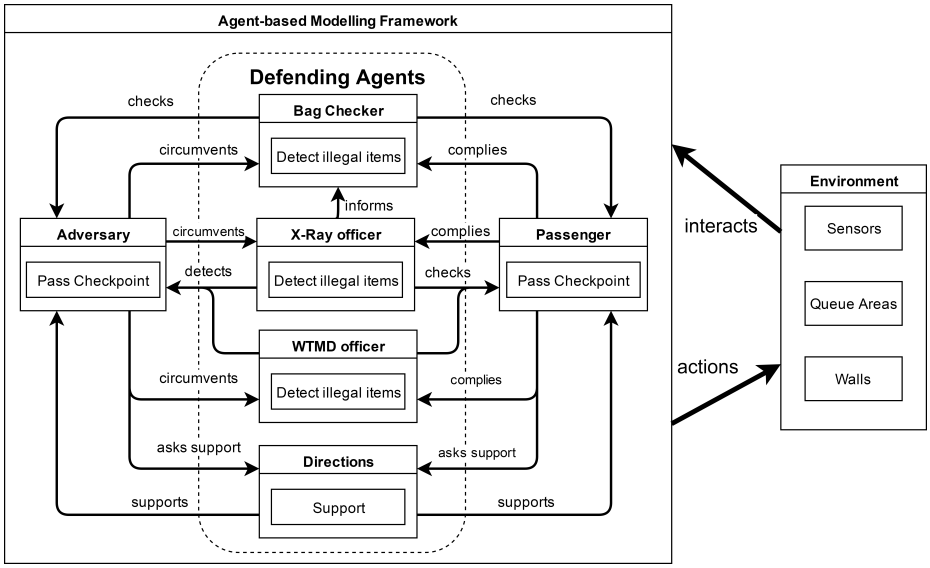

Fig. 2. Overview of the Agent-based Modelling Framework, containing attackers, defenders and passengers. The body of each agent shows a single activity that he/she can execute, represented in the Task Layer of the model. The two other layers are not visualized in this figure.

To incorporate varying performances of checkpoint employees under demanding circumstances, the Function State Model [2] is used. The Function State Model is used to determine the experienced pressure ($EP \in [0, 1]$) and performance quality ($PQ \in [0.4, 1.6]$) of an agent, based on factors like personality profile, cognitive abilities and external task demands ($TL \in [150, 500]$).

Task level is defined as a combination of two factors: queue length and bag complexity. These factors were shown to be influential on the performance of X-Ray officers in literature [6, 16]. Specifically, it is defined as follows:

$$
\begin{aligned}
TL(t) &= C_{bag} \times TL_{bag}(t) + (1 - C_{bag}) \times TL_{queue}(t) \\
TL_{bag}(t) &= Norm(BC(t)) \\
TL_{queue}(t) &= Norm(QL(t))
\end{aligned}
$$

Where $TL_{bag}(t)$ and $TL_{queue}(t)$ represent the task demand with respect to the baggage and queue at time t respectively. C_{bag} is a weighing parameter ($\in [0, 1]$) and $BC(t)$ is the bag complexity at time t. $QL(t)$ is the queue length at time t and finally, $Norm(x)$ represents a (unity-based) normalizing function. $BC(t)$ is equal to 0 when the X-Ray officer has no bag under consideration.

We relate the performance quality to the base values for both false negative probability and false positive probability of illegal item detection, as shown

below.

$$FN_{x-ray}(t) = FN_{base} \times Norm(PQ(t))$$
$$FP_{x-ray}(t) = FP_{base} \times Norm(PQ(t))$$

Where $FN_{x-ray}(t)$ and $FP_{x-ray}(t)$ represent the current false negative and false positive probability of illegal item detection respectively, and $Norm(x)$ is a normalizing function.

Previous work showed that experienced pressure influences processing time positively, while bag complexity influences the processing negatively [6]. This is modelled as follows.

$$T_{x-ray}(t) = T_{base} \times I(t)$$
$$I(t) = C_{EP} \times Norm(EP(t)) + (1 - C_{EP}) \times Norm(TL(t))$$

Where $T_{x-ray}(t)$ is the current mean processing time, C_{EP} is a weighing parameter ($\in [0, 1]$) and $I(t)$ is the current influence factor. The influence factor is a combination of two contributing factors $EP(t)$ and $TL(t)$. A linear relationship is assumed here, while other types of relationships are possible too.

Passenger & Attacker Agent The Passenger aims to pass the security checkpoint of the airport. It contains a *pass checkpoint activity*, which enables the passenger to move past the checkpoint. The checkpoint activity consists of three sub-activities: baggage drop-off, WTMD passage and baggage collection. Baggage drop-off and baggage collection are parametrized by T_{drop} and $T_{collect}$ respectively. These parameters determine the mean processing time of the associated sub-activities. Passengers are randomly generated in a designated area with interarrival time $T_{arrival}$.

The Motor Layer of Passengers is defined using the Social Force Model [7], which defines movement in terms of interacting particles.

The attacker agent is a special type of passenger, that carries an IED in his/her carry on luggage. He/she shows standard passenger behavior, but aims to pass the security checkpoint without being detected.

3.2 Environment

The Environment of the model consist of sensors and physical objects. Sensors are devices that enable agents to sense using a mechanic object. We distinguish two types of sensors: X-ray machines and Walk Through Metal Detectors (WTMD). X-ray machines produce an observation based on the bag under consideration, which is then interpreted by the X-ray officer. WTMDs also produce an observation based on the passenger under consideration. This observation is then interpreted by the WTMD officer.

Two important physical objects exist: *walls* and *queue separators*. Queue separators specify boundaries of queuing areas, which allow for measurements of the number of people in the queue ($QL(t)$) and average queuing time ($QT(t)$).

4 Experiment & Results

In this section, the implementation of the above described simulation model is discussed. Two experiments performed with this simulation model are discussed and the corresponding results are shown.

4.1 Implementation & Setup

For the implementation, we created an open-source microscopic agent-based simulator specifically built for Agent-based Security Risk Assessment[1]. The simulator is entirely Java-based and can therefore easily be used across different platforms. It allows for simple visualization and is modularly structured. It contains a collection of airport specific structures, like checkpoint functionality and basic passenger behavior. A visualization of the simulator is shown in Figure 3.

The following is specified in our experiments. Defending agents, $D = \{d^1_{x-ray}, d^2_{x-ray}, d^1_{bag}, d^2_{bag}, d_{wtmd}, d_{directions}\}$, consists of two X-Ray officers, two Bag Checker officers, a WTMD officer and a Directions officer. The set of attackers is defined to be $A = \{a_{IED}\}$, a single attacker agent carrying an IED. $O = \{o_1, ..., o_q\}$ is a set of q passengers, randomly generated over time. The environment, $E = \{e_{wall}, e_{queue}, e_{wtmd}, e^1_{x-ray}, e^2_{x-ray}\}$ is specified, which consists out of walls, a single queuing area, a Walk Through Metal Detector and two X-ray machines. A visualization of the experimental setup is shown in Figure 3. Finally, the Fail function is defined as follows.

$$F(m^j_i) = \begin{cases} 1 & a_{IED} \text{ passed the checkpoint undetected.} \\ 0 & \text{otherwise.} \end{cases}$$

We do not define the Consequence function $C(m^j_i)$ as this is outside the scope of this experiment. Further, two types of personality profiles based on the work of Bosse et al. [2] are specified, denoted as *Type I* and *Type II*. *Type I* has the capability to cope well with high stress levels, while *Type II* does not cope with stress well. For simpler comparison, we adapt personality *Type I* such that it has the same *optimal experienced pressure* level as *Type II*. Some important parameters were set using values provided in literature and are shown in Table 1. If relevant data is unavailable in literature, experts can be consulted to estimate a range for each parameter. Here, we show results of two experiments that were performed with this model. In one experiment we study the influence of interarrival time $T_{arrival}$ on estimated vulnerability, while in the other experiment we we study the influence of bag complexity BC_μ on estimated vulnerability.

4.2 Interarrival Time Experiment

We set C_{bag} to be 0, meaning that the task level $TL(t)$ of an X-Ray officer is only influenced by the queue length $QL(t)$. C_{EP} is set to 0.5, meaning that

[1] The simulator can be found at: https://github.com/StefJanssen/SeRiMa-ABM

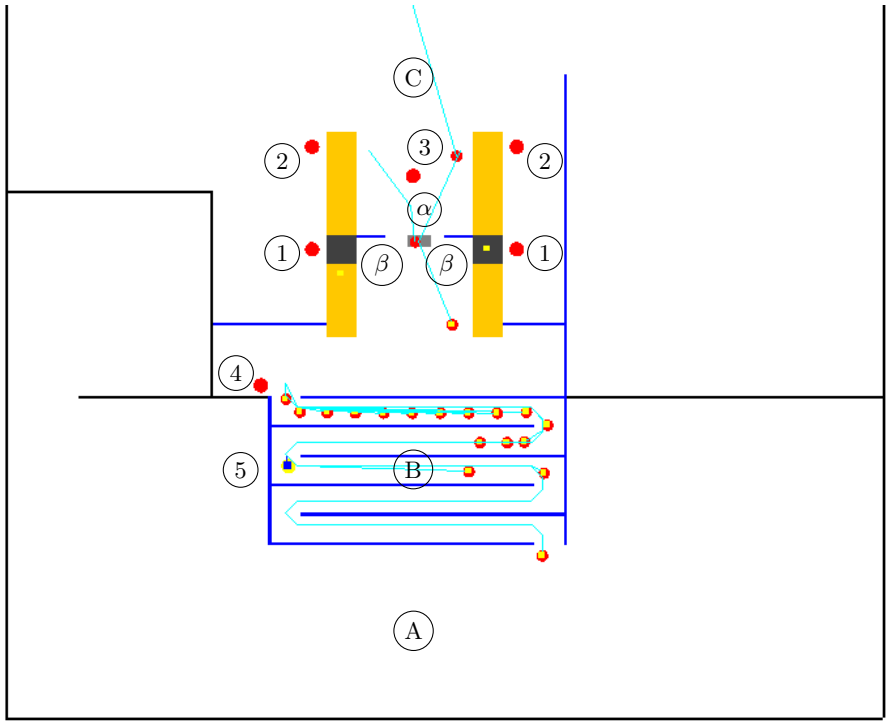

Fig. 3. A visualization of the experimental setup in the simulation tool. The following agents are shown in this figure. 1: X-Ray officers d_{x-ray}, 2: Bag Checker officers d_{bag}, 3: WTMD officer d_{wtmd}, 4: Directions officer $d_{directions}$, 5: attacker agent a_{IED}. All unlabelled agents are passengers o_i. The area in which A is located represents the agent-generation area, area B represents the queuing area and area C is the secure area. Passengers o_i and the attacker agent a_{IED} are generated in area A and go to area C. Walk Through Metal Detector e_{wtmd} is indicated by α and the X-Ray machines are indicated by β.

experienced pressure $EP(t)$ and $TL(t)$ equally influence the processing time of the X-Ray officer. We generate a_{IED} after 20 minutes of simulation time, while we vary the interarrival time $T_{arrival}$. We perform $N = 10000$ simulation runs and for each run record both the queue length $QL(t)$ at the time that the attacker passes the checkpoint and if the defenders failed to detect the attacker ($F(m_i^j)$).

Results of the experiment are shown in Figure 4. This figure shows $\hat{F}(m_i)$, the estimated *Vulnerability* and the average queue length $QL(t)$ at the time that the attacker passes the checkpoint for each of the interarrival times $T_{arrival}$.

The results show that both personality types perform best with an interarrival time of 17.5 seconds, corresponding to a queue length $QL(t)$ of around 20 passengers. The corresponding *Vulnerability* is 0.116 for Type I and 0.126 for personality type II. This can be explained from the definition of the Functional State Model, with the definition of *optimal experienced pressure*. We also find, as

Table 1. Basic parameters for the experimental setup. It shows the parameter name, description and standard value. It also refers to the work which was used to determine the standard value. In some cases this is an estimate based on related parameters.

Parameter	Description	Standard Value	Source
P_{bag}	The probability that the bag checker agent randomly checks a bag.	0.1	[10]
T_{drop}	The mean time a passenger takes to drop its belongings at the x-ray system.	12.5s	[10]
$T_{collect}$	The mean time a passenger takes to collect its belongings at the x-ray system.	12.5s	[10]
T_{wtmd}	The mean time the WTMD officer takes to check a passenger.	10.0s	[17]
P_{wtmd}	The probability that the WTMD officer randomly checks a passenger.	0.1	[17]
FN_{base}	The base False Negative probability of an X-Ray officer.	0.1	[15]
FP_{base}	The base False Positive probability of an X-Ray officer.	0.2	[15]
T_{base}	The mean time an X-Ray officer takes to check a bag.	6.0s	[12]

expected, that X-Ray officers with personality *Type I* generally produce a lower *Vulnerability*, implying a higher performance quality $PQ(t)$ at the moment attacker agent a_{IED} passes.

4.3 Bag Complexity Experiment

In this experiment we investigate the influence of bag complexity on the performance of the defense agents. We use the same two personality profiles as used in the previous experiment. We set C_{bag} to be 0.75, meaning that the task level

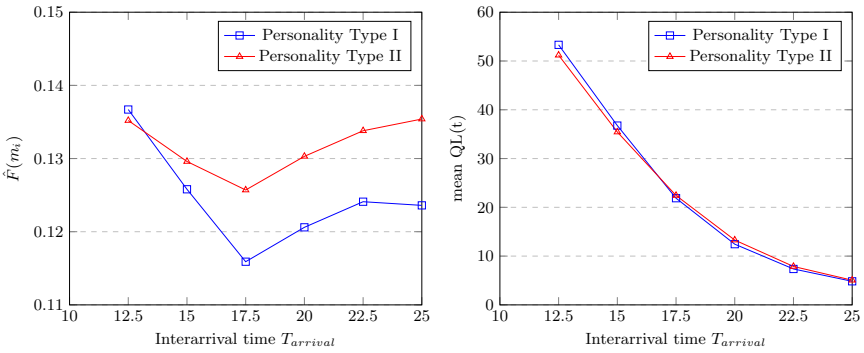

Fig. 4. The left plot shows the estimated *Vulnerability* $\hat{F}(m_i)$ of the system for varying interarrival times $T_{arrival}$, calculated using the defined Fail function. The right plot shows the mean queue length $QL(t)$ at the time a_{IED} was processed.

$TL(t)$ of an X-Ray officer is influenced by the queue length $QL(t)$ for 25% and the bag complexity $BC(t)$ for 75%. C_{EP} is set to 0.5, meaning equally influence importance for $EP(t)$ and $TL(t)$ processing time. We set the interarrival time $T_{arrival}$ to be 15 seconds and generate a_{IED} after 20 minutes of simulation time. We vary the bag complexity of each agent by drawing a number from a normal distribution with mean BC_μ and standard deviation BC_σ. We perform $N = 10000$ simulation runs and for each run record the performance quality $PQ(t)$ of the responsible d^k_{x-ray} at the time that a_{IED} passes the checkpoint and the outcome of Fail function $F(m^j_i)$.

Results of the experiment are shown in Figure 5. The figure shows $\hat{F}(m_i)$, the estimated *Vulnerability* and the mean performance quality $PQ(t)$ at the time that the attacker passes the checkpoint for each of the bag complexities BC_μ.

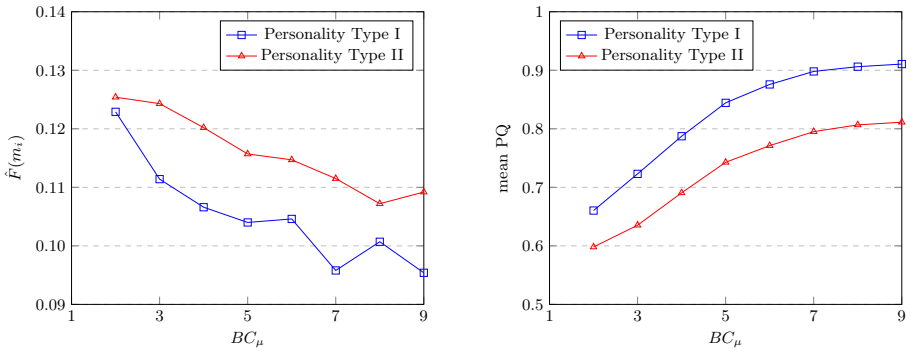

Fig. 5. The left plot shows the estimated *Vulnerability* $\hat{F}(m_i)$ of the system for different mean bag complexities BC_μ, calculated using the defined Fail function. The right plot shows the mean performance quality $PQ(t)$ of the responsible d^k_{X-ray} at the time the attacker agent was processed.

The graphs show that the estimated *Vulnerability* decreases while bag complexity increases. While this sounds counter intuitive, it can be understood from the specification of the Functional State Model. In the FSM a so-called *recovery effort* is defined, allowing an agent to decrease exhaustion in the absence of (large) tasks. Task demand with respect to the baggage $TL_{bag}(t)$ is defined to be 0 in the absence of baggage. This allows for timely decrease of exhaustion and therefore, high performance quality in case a new bag arrives. Higher task demand can, in the short term, result in higher performance qualities due to a direct link between the task level and *current contribution*. This is reflected in the increasing performance quality $PQ(t)$ and the resulting estimated *Vulnerabilities*.

5 Conclusions & Discussion

This paper introduced a novel Security Risk Assessment approach which is based on Agent-based modelling and simulation. It uses Monte Carlo simulations to estimate both *Vulnerability* and *Consequence*, which are important parameters in Security Risk Assessment. It defines an Agent-based model with both defender agents and attacker agents. An attacker agent aims to execute subversive actions within some security scenario identified by security experts, while defender agents are modelled to perform their security tasks.

This approach enables modelling of essential aspects and processes of socio-technical systems at cognitive, social and organisational levels. This is problematic for traditional and game theory based approaches. *Vulnerability* and *Consequence* produced by this method can be used to improve both traditional and game theory based Security Risk Assessment methods. Outputs of this method can be used as estimates for each of the payoffs in a game theoretic approach.

An illustrative case study in the area of airport security has been performed to demonstrate the use of this approach. Using the Functional State Model, it is shown that different *Vulnerabilities* arise for a variety of circumstances at the security checkpoint. It for instance shows preferred stress levels for X-ray officers, resulting in higher performance.

In the future, we will perform case studies in which we estimate *Consequence* as well. This will be done by defining a Consequence function that estimates consequence in a given simulation run. This Consequence function incorporates estimates of direct losses and indirect losses. Direct losses, including fatalities and physical damages of an attack, can be estimated from a simulated security scenario. Indirect losses like decreased number of future passengers and business disruptions are then based on the estimated direct losses and historical data. This work will be extended with a theoretical analysis, more elaborate experiments and different underlying models to investigate the theoretical and practical strengths and weaknesses of this approach.

References

1. Bruce M Blumberg and Tinsley A Galyean. Multi-level direction of autonomous creatures for real-time virtual environments. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 47–54. ACM, 1995.
2. Tibor Bosse, Fiemke Both, Rianne Van Lambalgen, and Jan Treur. An agent model for a human's functional state and performance. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, pages 302–307. IEEE Computer Society, 2008.
3. Matthew Brown, Arunesh Sinha, Aaron Schlenker, and Milind Tambe. One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats. In *AAAI conference on Artificial Intelligence (AAAI)*, 2016.
4. Louis Anthony Tony Cox Jr. Some limitations of risk= threat vulnerability consequence for risk analysis of terrorist attacks. *Risk Analysis*, 28(6):1749–1761, 2008.

5. Barry Charles Ezell, Steven P Bennett, Detlof Von Winterfeldt, John Sokolowski, and Andrew J Collins. Probabilistic risk analysis and terrorism risk. *Risk Analysis*, 30(4):575–589, 2010.
6. Ian Graves, Marcus Butavicius, Veneta MacLeod, Rebecca Heyer, Kathryn Parsons, Natalie Kuester, Agata McCormac, Philip Jacques, and Raymond Johnson. The role of the human operator in image-based airport security technologies. In *Innovations in Defence Support Systems-2*, pages 147–181. Springer, 2011.
7. Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.
8. Serge P Hoogendoorn and Piet HL Bovy. Pedestrian route-choice and activity scheduling theory and models. *Transportation Research Part B: Methodological*, 38(2):169–190, 2004.
9. ICAO. Aviation security manual (doc 8973 restricted). http://www.icao.int/Security/SFP/Pages/SecurityManual.aspx, 2014. [Online; accessed 27-May-2016].
10. Alan Avi Kirschenbaum. The cost of airport security: The passenger dilemma. *Journal of Air Transport Management*, 30:39–45, 2013.
11. Aron Laszka, Mark Felegyhazi, and Levente Buttyan. A survey of interdependent information security games. *ACM Computing Surveys (CSUR)*, 47(2):23, 2015.
12. Kelly Leone and Rongfang Rachel Liu. Improving airport security screening checkpoint operations in the us via paced system design. *Journal of Air Transport Management*, 17(2):62–67, 2011.
13. James Pita, Manish Jain, Janusz Marecki, Fernando Ordez, Christopher Portway, Milind Tambe, Craig Western, Praveen Paruchuri, and Sarit Kraus. Deployed armor protection: the application of a game theoretic model for security at the los angeles international airport. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*, pages 125–132. International Foundation for Autonomous Agents and Multiagent Systems, 2008.
14. Craig W Reynolds. Steering behaviors for autonomous characters. In *Game developers conference*, volume 1999, pages 763–782, 1999.
15. Elena Rusconi, Francesca Ferri, Essi Viding, and Timothy Mitchener-Nissen. Xrindex: a brief screening tool for individual differences in security threat detection in x-ray images. *Frontiers in human neuroscience*, 9:439, 2015.
16. Livia Thomas, Adrian Schwaninger, Nadja Heimgartner, Patrik Hedinger, Franziska Hofer, Ulrike Ehlert, and Petra H Wirtz. Stress-induced cortisol secretion impairs detection performance in x-ray baggage screening for hidden weapons by screening novices. *Psychophysiology*, 51(9):912–920, 2014.
17. Josephus van Boekhold, Ardeshir Faghri, and Mingxin Li. Evaluating security screening checkpoints for domestic flights using a general microscopic simulation model. *Journal of Transportation Security*, 7(1):45–67, 2014.
18. ASME Washington. *All-Hazards risk and resilience: prioritizing critical infrastructures using the RAMCAP Plus SM approach*. ASME, 2009.