Modeling Adversarial Physical Movement in a Railway Station: Classification and Metrics

CARMEN CHEH, Department of Computer Science and Coordinated Science Laboratory, University of Illinois
BINBIN CHEN, Singapore University of Technology and Design (SUTD) and Advanced Digital Sciences Center (ADSC), Singapore
WILLIAM G. TEMPLE, Advanced Digital Sciences Center (ADSC), Singapore
WILLIAM H. SANDERS, Department of Electrical and Computer Engineering and Coordinated Science Laboratory, University of Illinois

Many real-world attacks on cyber-physical systems involve physical intrusions that directly cause damage or facilitate cyber attacks. Hence, in this work, we investigate the security risk of organizations with respect to different adversarial models of physical movement behavior. We study the case in which an intrusion detection mechanism is in place to alert the system administrator when users deviate from their normal movement behavior. We then analyze how different user behaviors may present themselves as different levels of threats in terms of their normal movement behavior within a given building topology. To quantify the differences in movement behavior, we define a *WeightTopo* metric that takes into account the building topology in addition to the movement pattern. We demonstrate our approach on a railway system case study and show how certain user roles, when abused by attackers, are especially vulnerable in terms of the physical intrusion detection probability . We also evaluate quantitatively how the similarity between an attacker's movement behavior and a user's movement behavior affects the detection probability of the evaluated intrusion detection system. Certain individual users are found to pose a higher threat, implying the need for customized monitoring.

CCS Concepts: • **Computer systems organization → Embedded and cyber-physical systems**; • **Security and privacy →** *Intrusion/anomaly detection and malware mitigation*;

Additional Key Words and Phrases: Railway transportation system, adversary model, physical movement

ACM Reference format:
Carmen Cheh, Binbin Chen, William G. Temple, and William H. Sanders. 2019. Modeling Adversarial Physical Movement in a Railway Station: Classification and Metrics. *ACM Transactions on Cyber-Physical Systems* 1, 1, Article 1 (January 2019), 24 pages.

This material is based upon work supported by the Maryland Procurement Office under Contract No. H98230-18-D-0007, and in part by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-31) and administered by the National Cybersecurity R&D Directorate, and supported in part by the research grant for the Human-Centered Cyber-physical Systems Programme at the Advanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR). We also want to thank the experts from SMRT Trains LTD for providing us with data and domain knowledge.
Authors' addresses: C. Cheh, Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin Ave., Urbana, IL 61801; email: cheh2@illinois.edu; B. Chen, Singapore University of Technology and Design, 8 Somapah Road, Singapore; email: binbin_chen@sutd.edu.sg; W. G. Temple, Advanced Digital Sciences Center, 1 Create Way #14-02, Create Tower, Singapore; email: william.t@adsc.com.sg; W. H. Sanders, Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 306 N. Wright St, Urbana, IL 61801; email: whs@illinois.edu.

XXXX-XXXX/2019/1-ART1 $15.00
https://doi.org/10.1145/3349584

https://doi.org/10.1145/3349584

1 INTRODUCTION

Organizations often perform risk assessments of their systems in order to understand the weak points in their architectures and analyze how they can be targeted by various kinds of adversaries [3, 18]. By doing so, they can direct resources into monitoring and responding to threats more effectively. This risk assessment is typically conducted on the cyber aspect of the system architecture by analyzing the networked system and considering cyber attacks such as denial of service (DoS) and advanced persistent threats (APTs).

However, to have a well-rounded defense posture, an organization needs to take into account its physical defense in addition to its cyber defense. This is especially crucial for critical infrastructure systems such as power grids and transportation systems, because a physical attack could have severe real-world impacts. Physical attacks also often serve as the first step to facilitate follow-up cyber attacks, e.g., to help the attacker gain an initial foothold in the system, or to help the attacker gain configuration knowledge of the target. The current front-line defense for physical security is building access controls, which are used to restrict a user's movement to the necessary spaces of a building. They do so by statically assigning a set of permissions to a user's tracking identifier (e.g., RFID tag, access card, biometrics data). The permissions represent locations that a user is allowed to access, and as the user moves between spaces (e.g., swiping a card at a door), the movement is logged. In addition, video surveillance technology is widely used to monitor critical locations.

However, building access control systems can only prevent certain simple attacks from happening, e.g., those attacks that try to gain access to unauthorized areas. They do not provide a mechanism for detecting abnormal physical movement behavior within a building. Video surveillance is also limited in its ability to detect intrusions. Intelligent video systems can detect when a person is in a room and performing some actions, but they are unable to give further context to that information. In particular, they cannot decide whether a person should or shouldn't be in that room or whether his or her actions are normal or anomalous. Thus, the organization is encouraged to employ some intrusion detection mechanism on top of basic building access control monitoring and video surveillance. The intrusion detection method provides context for the information observed by video surveillance and building access controls by building a model of normal user behavior in order to detect deviations from the user's typical movement pattern. Notable work in this area includes [6, 17, 20]. In [6], we model normal user movement behavior by using Markov chains, and, based on information about building topology, we mark as anomalous behavior any accesses that deviate from the typical sequence of visited rooms . In [17, 20], two patents were filed by IBM and Honeywell that focus on presenting a general design for use of physical access data to detect anomalous movement behavior. These new technologies can extend an organization's capability to detect possible physical breaches when a user does not violate the building access control rules.

In this paper, we investigate how physical security can be breached with respect to different adversarial threats. Our adversarial threat model assumes that an attacker has gained access to a valid user's tracking identifier, whether it be through duplicating the device or data, theft, or social engineering. Furthermore, different types of attackers may possess varying degrees of knowledge regarding the accesses granted to the tracking identifier and the behavior patterns associated with the original owner of the tracking identifier. For example, an outsider may know nothing about the tracking identifier's permissions, whereas another staff member may be more familiar with the rooms that can be accessed. Those differences will affect their chances of being detected by the organization's intrusion detection mechanisms.

In this paper, we consider the intrusion detection system described in [6]. That detection system uses historical physical access logs to build models of users' normal behavior and raises an alarm in real-time when an access that deviates from the owner's normal behavior is detected. In that work, promising results were obtained for a simple attacker model. Therefore, in this paper, we conduct a more in-depth analysis with a variety of attacker models. More precisely, given that detection system, we want to study (1) what factors can help us characterize differences in users' movement behaviors, (2) what types of users are more likely to be targeted by attackers, and (3) how closely an attacker needs to mimic a user's movement behavior to remain undetected. In this work, we provide a classification of a spectrum of different attacker models with respect to adversarial movements. To better compare these models, we define a weighted topological (*WeightTopo*) metric that can quantify the differences between an attacker's movement behavior and a normal system user's movement behavior with respect to the physical architecture of the system. We apply our classification and the *WeightTopo* metric on a real-world railway station case study. Given a dataset that allows us to extract normal user movement behavior, we analyze the detection rates given by the intrusion detection system under different attacker models to answer the questions we pose above.

The outline of the rest of this paper is as follows. We present a railway transit station use case in Section 2 to motivate this study. Then, in Section 3, we classify different attacker models and define a metric to quantify differences in attacker movement behavior and user movement behavior. Section 4 summarizes our results and analyses. Finally, we discuss related work in Section 5 and conclude in Section 6.

2 MOTIVATING USE CASE

Physical security is a top concern for industrial control facilities such as critical infrastructure systems. Through a project partnership, we have gained deep knowledge and understanding of the physical security challenges faced by railway transit system operators. Our study is motivated by that real-world use case.

The railway transit system is an important component of a nation's transportation system. An attack or fault in the system can have severe impact, ranging from loss of service and station blackouts to derailment. For example, in one incident, a Polish teenager rewired a remote control to communicate with wireless switch junctions, causing the injury of twelve people and derailment of a train [2]. In that case, the track was easily accessible to the public, allowing such an attack to occur. The underground railway system in our case study, however, prevents such attacks. Insiders are also a serious threat, as demonstrated by the 2006 case in which two traffic engineers hacked into a Los Angeles signal system, causing major traffic disruption [13].

2.1 System Architecture

A railway station is a single building that houses at least one railway line through it. The general public moves through the station to access the railway lines by entering through fare gates in the concourse area and then moving to the platform that is located either one floor above or below the concourse area.

In Figure 1, we depict a simplified topology of the railway station in our case study . We represent the building topology as a directed graph $G = (S, E)$ in which the set of vertices S represents the spaces in the buildings. A directed edge $e(v_1, v_2)$ represents possible movement from v_1 to v_2.[1] For example, the floor plan in Figure 1b is represented as the graph in Figure 1c. The set of spaces S can

[1]This implies that if $e(v_1, v_2)$ exists, the backward edge $e(v_2, v_1)$ also exists in G.

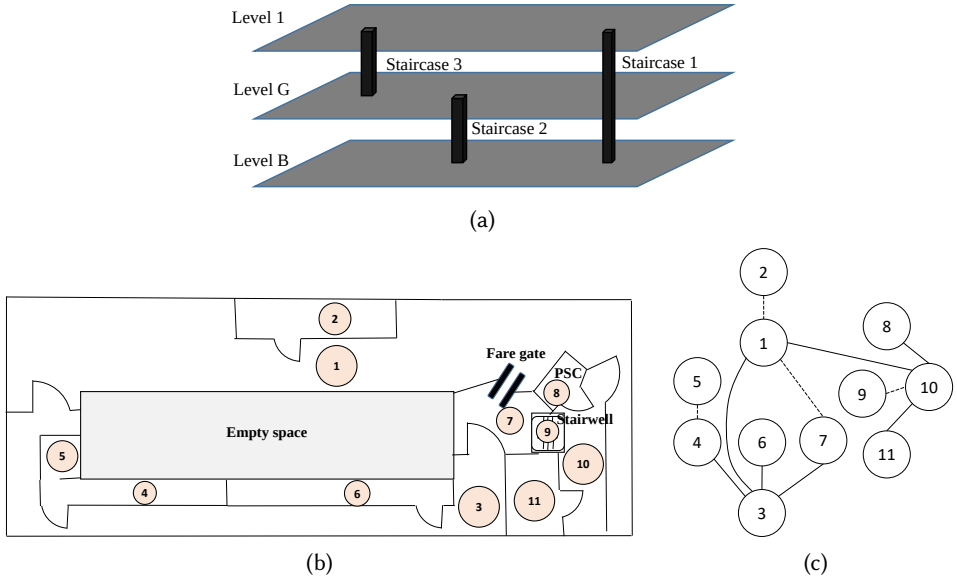


Fig. 1. The railway station building in our case study and its representation.(a) The different levels of a railway station building, with staircases that connect two or more levels. (b) A small sample floor plan of one of the levels. The PSC room represents the Passenger Service Center. (c) Graph representation of (b). Each edge in the graph represents a pair of directed edges between the vertices. Solid edges imply that a card reader exists on the door bordering the spaces (vertices).

be divided into two partitioning subsets: rooms $\mathbf{R}$, and common areas $\mathbf{C}$ (e.g., staircases, corridors), i.e., $S = \mathbf{R} \cup \mathbf{C}$, and $\mathbf{R} \cap \mathbf{C} = \phi$.

In addition to the areas that are accessible to the general public, there are many rooms in the railway station that are hidden away from the public eye. These rooms house the equipment necessary to maintain the running of the station and its portion of the railway track. Each room serves a specific function and is related to a particular subsystem (e.g., power, environment control) of the railway system. The list of subsystems in the railway system is $Subsys = \{OP, POW, EC, TM, ME\}$; a description of each subsystem is given below.

OP (station operation control): e.g., the *Passenger Service Center* (PSC), server room, office, and storeroom.
POW (power control): e.g., traction power room and *Uninterrupted Power Supply* (UPS) room.
EC (environment control): e.g., water chiller room and ventilation fan room.
TM (transport mechanical equipment control): e.g., escalator rooms and platform screen door room.
ME (mechanical equipment): e.g., lighting control room and pump room.

Multiple rooms may share the same functionality. For each room, we label the corresponding vertex, with the subsystems associated with it, $label : \mathbf{R} \rightarrow Subsys$. The rooms are distributed throughout the station on multiple levels, including the concourse and platform area. Unlike an enterprise system in which the office building has a simple systematic layout across all levels (a single corridor branching out to multiple rooms), a railway station has a more complex, asymmetrical layout. A user can take multiple paths with varying lengths in order to get from one room to

another. This suggests that the topology of a building can impact detection capability, and thus that some users may be imitated by attackers more easily than others because of their different sets of accessible rooms.

Users are assigned to a subset of the subsystems based on their job descriptions, $assign : U \rightarrow 2^{Subsys}$ where 2^{Subsys}, is the power set of *Subsys*, and their tracking identifiers are associated with permissions $Perm : U \rightarrow 2^{\mathbf{R}}$ to access certain rooms. Concretely, $Perm(u_i \in U) = \{s \in \mathbf{R} | label(s) \in assign(u_i)\}$.

2.2 Building Access Control Systems

The building access control system in our case study uses access cards as tracking identifiers. Most of the doors inside the staff-only spaces have card readers installed, although there are some that permit free access.

In recent years, more advanced building access control systems that use biometrics solutions have been deployed. Those building access control systems use the unique physical characteristics of a human as tracking identifiers. Typical physical characteristics used include fingerprints, facial features, and irises.

Compared to traditional access card systems, those advanced biometrics systems can significantly raise the bar for adversaries who wish to obtain physical access to the system. However, those biometrics-based systems have their own sets of security issues [23]. Specifically, in many cases, the adversary can still duplicate (or spoof) the tracking identifier. To make things worse, the loss of a user's biometric features cannot be easily countered by revoking or replacing the identifier. Such a loss will affect all the systems that rely on such biometric features, including the user's personal devices and the services (e.g., banking, government service) used by the user. Thus, it can have a serious impact on a user's privacy.

In particular, it has been shown that adversaries have been able to obtain a user's biometric features (e.g., pictures of their faces or fingerprints). For example, the Office of Personnel Management breach in 2015 compromised 5.6 million people's fingerprints [14]. The next step for an adversary is then to feed that tracking identifier in some form to the system to authenticate him or herself. A great deal of research has been conducted to identify ways to spoof tracking identifiers [4, 15, 16, 29], and it has shown that it is indeed possible to steal and use an authorized user's tracking identifier. Thus, our assumption that adversaries can obtain and use the tracking identifiers of authorized users can potentially be applicable to biometrics access control systems.

Also, a building access control system only prevents an adversary from accessing rooms that are not within the set of permitted spaces for the tracking identifier in the adversary's possession. If an adversary attempts to access a non-permitted space, he or she will be denied access, and the action will be logged as an "Invalid Attempt." However, if the tracking identifier held by the adversary has the necessary permissions, the building access control system will allow the adversary to access the room, even if the movement behavior of the adversary is suspicious. Thus, we envision that additional systems that can detect anomalies based on a user's movement pattern could be deployed to complement the building access control system. In particular, we describe one such detection system in Section 2.3.

2.3 Monitoring Capability

In addition to the building access control system, the railway station can be equipped with a physical movement-based *intrusion detection system* (IDS) that models normal user movement behavior. When the adversary deviates from the prediction of the model, an alarm will be raised. The potentially malicious accesses are marked and presented to a staff member for further action.

In this paper, we take the simple but effective approach of detecting suspicious movement behavior as described in [6]. In that approach, the detection mechanism proceeds in two phases: an offline phase and an online phase. In the offline phase, we construct models of the user's normal movement behavior by using historical access logs. In the online phase, the IDS analyzes each physical access that is made in real-time and compares it to the constructed user movement models produced by the offline phase. If the physical access deviates from the user's movement model, then the IDS raises an alarm.

In more detail, we represent a user's movement behavior by using a first-order Markov model. The states in the Markov model are the rooms in the station, and a transition from state i to state j means that a user visits room R_j after R_i. In the offline phase, we use historical access logs to learn the initial and transition probabilities of the Markov model of a user.

Then, in the online phase, we keep track of the last room that the user visited. Every time a user swipes a card, we use the model's transition probabilities to predict the set of next possible rooms. Based on the building topology, we can determine whether the user is likely to move to that set of rooms based on whether the user's attempted access to a space is on the shortest path to those rooms. Then, we can calculate the probability that the attempted access is normal by summing up the transition probabilities associated with the rooms that the user is likely to visit next. If that probability is below a certain set threshold, then the IDS raises an alarm. After each user swipe, the IDS takes 0.07 ms on average to decide whether or not to raise an alarm [6].

3 CLASSIFICATION OF ATTACKER MODELS

The threat model considered postulates that an attacker obtains possession of a valid access card that gives him or her the ability to access any spaces to which the owner of the card has permission. The attacker may have obtained the card through social engineering, or by duplicating or forging a card. In this paper, we want to analyze the physical security of a system with respect to different attacker models. Therefore, we consider a range of attackers who have varying levels of information about the building topology and the movements of the original owner of the access card. Attackers will move through the building in different ways depending on the information they possess and the permissions associated with the card. Thus, certain attackers will be more difficult than others to detect. In this section, we define a classification for a few different attacker models that can represent realistic adversarial threats. We also define a metric to quantify how much an attacker's movement behavior can emulate a legitimate user's movement behavior. The important symbols used in this section are summarized in Appendix A for ease of reference.

3.1 Attacker Models

We define three different types of attacker models: Amateur Outsider, Informed Outsider, and Insider. In Table 1, we classify attacker profiles into the three attacker models and identify the possible ways in which the attacker may obtain access to the building. An attacker may have information about the building layout G, the permissions associated with the obtained access card $Perm(i)$, and the movement behavior of a user u_i, $Mov(i)$. We represent the extent of information possessed by the attacker as $Kwg \in 2^{\{G, Perm(i), Mov(i)\}}$, where $2^{\{G, Perm(i), Mov(i)\}}$ denotes the power set of $\{G, Perm(i), Mov(i)\}$. We describe each of the attacker models starting with the one with the least information and moving up to the model with the most information. We represent the physical movement behavior of a user as a transition matrix $\mathbf{M}$, i.e., $Mov : U \rightarrow \mathbf{M}$, where the entry m_{ij} in the matrix indicates the probability that a person will move from room r_i to r_j. Each user i is associated with a transition matrix $Mov(i) = H_i \in \mathbf{M}$ that represents his or her movement behavior.

The attacker movement behavior is similarly represented as a transition matrix $K \in \mathbf{M}$. We then define our attacker models by specifying the entries in K.

Amateur Outsider. The attacker has no information about the building layout, the permissions associated with the access card, or the user's normal behavior, i.e., $Kwg = \phi$. This can model an outsider who is unable to obtain the necessary information and has to resort to manual reconnaissance. Therefore, the attacker moves through the building at random to explore its layout. We model the attacker's transition matrix K by setting each entry to a uniformly distributed value, i.e., $k_{ij} = \frac{1}{|\mathbf{R}|}$, where $|\mathbf{R}|$ denotes the number of rooms in the building.

Informed Outsider. In addition to the building topology, the attacker has information about the permissions associated with the access card, i.e., $Kwg = \{G, Perm(i)\}$. This can model an attacker who has already done reconnaissance and knows the layout of the station and which rooms can be accessed by the card. The attacker has no information about the user's movement behavior and so may randomly enter rooms outside of the user's normally visited rooms. We model the attacker's transition matrix K by setting each entry to a uniformly distributed value over the permitted rooms in $Perm(i)$ associated with the access card, i.e., $k_{ij} = \frac{1}{|Perm(i)|}$.

Insider. The attacker is a legitimate user u_j in the system and thus is already aware of station layout and the mapping of rooms to subsystems. In addition, the attacker has some information about the movement behavior of another user u_i. Thus, $Kwg = \{G, Perm(i), Mov(i)\}$. Although the attacker possesses her own access card, she uses the access card of user u_i to carry out her malicious behavior so as to shift the blame and evidence to that user. However, it is non-trivial to completely emulate the movement behavior of user u_i because that requires the attacker to follow and observe that user for an extended period of time. So in this case, $K = H_j$. If the attacker is in the same subsystem as the owner, their access patterns will be more similar than they would be if the attacker belonged to a different subsystem.

For each of the general attacker models above, we can extend the definition of the entries in the transition matrix K to include a noise vector $\mathbf{e}$, i.e., $K = norm(K' + \mathbf{e})$, where K' is given by the definitions above, and the *norm* function renormalizes the matrix such that the sum of each row remains as 1. The noise vector $\mathbf{e}$ models the detailed differences between one attacker and another, like attacker preference or bias towards entering certain rooms. For example, an attacker belonging to the *Insider* model may decide to stick completely to his or her normal behavior. Then, the noise $\mathbf{e} = \mathbf{0}$. Another attacker in the same *Insider* model may decide to be more stealthy and randomly add Gaussian noise to his or her normal behavior in order to disassociate the accesses from his or her movement profile. So these three attacker models can model a continuous range of possible attackers.

3.2 Quantifying Attacker Information

In order to gauge the effectiveness of different IDSes under various threat scenarios, we need to quantitatively differentiate among the different attacker models. We propose a metric that succinctly measures the difference between an attacker's movement behavior and a legitimate user's movement behavior. We can then use this metric to objectively compare the performance of an IDS against different attacker models. Following along the lines of the previous section, we represent a user's movement behavior as the transition matrix H_i or H for short. So we can quantify the attacker information as the difference between transition matrices K and H.

$$\textbf{WeightTopo} = \textbf{diff(H,K)} \tag{1}$$

Table 1. Classifying attacker profiles into the three attacker models

	Amateur Outsider	**Informed Outsider**	**Insider**
Attacker Information	Building Layout (G)	Building Layout (G) Card Permissions ($Perm(i)$)	Building Layout (G) Card Permissions ($Perm(i)$) Existing Job ($Mov(i)$)
Profiles	General Public	Contractor Past Employee	Cleaner Security Guard Maintenance Staff Administrators
Attack Vectors	Stealing Card, Forging Card, RFID Attack Social Engineering Tailgating Collusion		
Transition Matrix	$\begin{array}{c} \\ r_1 \\ \\ \vdots \\ \\ r_R \end{array} \begin{array}{c} \begin{array}{ccccc} r_1 & & \cdots & & r_R \end{array} \\ \begin{pmatrix} 0 & \frac{1}{\lvert R\rvert} & \cdots & \cdots & \frac{1}{\lvert R\rvert} \\ \frac{1}{\lvert R\rvert} & 0 & \ddots & & \vdots \\ \vdots & \ddots & 0 & \ddots & \vdots \\ \vdots & & \ddots & 0 & \frac{1}{\lvert R\rvert} \\ \frac{1}{\lvert R\rvert} & \cdots & \cdots & \frac{1}{\lvert R\rvert} & 0 \end{pmatrix} \end{array}$	$\begin{array}{c} \\ \vdots \\ r_i \\ \\ r_j \\ r_k \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} \cdots & r_i & r_j & r_k & \cdots \end{array} \\ \begin{pmatrix} 0 & & \cdots & & 0 \\ & 0 & \frac{1}{\lvert Perm(i)\rvert} & \frac{1}{\lvert Perm(i)\rvert} & \\ \vdots & \frac{1}{\lvert Perm(i)\rvert} & 0 & \frac{1}{\lvert Perm(i)\rvert} & \vdots \\ & \frac{1}{\lvert Perm(i)\rvert} & \frac{1}{\lvert Perm(i)\rvert} & 0 & \\ 0 & & \cdots & & 0 \end{pmatrix} \end{array}$	$\begin{array}{c} \\ \vdots \\ r_i \\ \\ r_j \\ r_k \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccc} \cdots & r_i & r_j & r_k & \cdots \end{array} \\ \begin{pmatrix} 0 & & \cdots & & 0 \\ & 0 & h_{ij} & h_{ik} & \\ \vdots & h_{ji} & 0 & h_{jk} & \vdots \\ & h_{ki} & h_{kj} & 0 & \\ 0 & & \cdots & & 0 \end{pmatrix} \end{array}$

One way of defining the function *diff* is to use the square of the Frobenius norm function, which takes the squares of the differences in the values of the two transition matrices, $\sum_{ij}(h_{ij} - k_{ij})^2$. The Frobenius norm is a typical metric to quantify the distance between matrices and so, we define that function as the metric *Values*, which will serve as a baseline for comparison with our *WeightTopo* metric.

However, the metric *Values* disregards the semantics of the transition matrices since the entries represent geographical locations (rooms in the station). For example, if an attacker believes that the user normally moves to room R_i instead of R_j, the Frobenius norm function would return the same value regardless of where R_i is located. But intuitively, we know that if room R_i is located near R_j, the difference value should be smaller than it would be if room R_i were located far away from R_j, because the attacker's guess is close to R_j.

So we include the topological layout of the building in our calculations of the difference function. In particular, we consider in our metric definition the spaces that are visited when a person moves from one room to another. For a room (or row) r_s in a transition matrix $M \in \mathbf{M}$, we map the set of possible locations that will be visited next onto the building topology $L^M_{r_s} = \{r_i | m_{r_s,r_i} > 0, r_i \in \mathbf{R}\}$. Then, we examine the vertices situated on the shortest paths between the room r_s and the set of next locations, $P^M_{r_s}$.

$$P^M_{r_s} = \{v | v \in path(r_s, r_i), r_i \in L^M_s\} \tag{2}$$

where $path(v, w) = \{(v, s_i, \ldots s_{i+n}, w) | \forall (v, s_k, \ldots s_{k+m}, w), n \leq m\}$ and $(v, s_i \ldots w)$ refers to the sequence of vertices on the path from v to w. One way of defining the difference function would then be to ignore the probabilities and consider only the topological distance. We can define such a function as another baseline metric *Topo*, which is defined much like our *WeightTopo* metric. More precisely, the difference function *diff* for *Topo* is defined as:

$$diff(h_{r_s}, k_{r_s}) = \sum_{v \in (P^H_{r_s}) \cup (P^K_{r_s})} \begin{cases} 1 & \text{if } v \in (P^K_{r_s}) - (P^H_{r_s}) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

However, not all spaces in the building are visited equally. So we associate a score Sc_M with each vertex $v \in P^M_{r_s}$. This score $Sc_M(v)$ represents the probability of visiting that location. By taking the differences between these scores for the rows of the two transition matrices H and K, we are quantifying the difference in topological distance in addition to the differences in probabilities of visiting rooms. We define $Sc_M(v)$ below.

$$Sc_M(v) = \begin{cases} m_{r_s,v} & \text{if } v \in L^M_{r_s} \\ \sum_{w \in R_P} m_{r_s,w} & \text{if } v \in P^M_{r_s} - L^M_{r_s} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $R_P = \{w | v \in path(r_s, w), w \in L^M_{r_s}\}$ is the set of rooms that are reachable from v. In other words, we propagate the probabilities associated with the rooms (i.e., the vertices at the end of the paths) backwards, as shown in Figure 2.

Now, we can compare the scores from transition matrices H and K. For each room (or row) r_s in K, we calculate the difference $diff(h_s, k_s)$ as follows.

$$diff(h_s, k_s) = \sum_{v \in (P^H_{r_s}) \cup (P^K_{r_s})} \begin{cases} (Sc_H(v) - Sc_K(v))^2 & \text{if } v \in (P^K_{r_s}) \cap (P^H_{r_s}) \\ Sc_K(v)^2 & \text{if } v \in (P^K_{r_s}) - (P^H_{r_s}) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

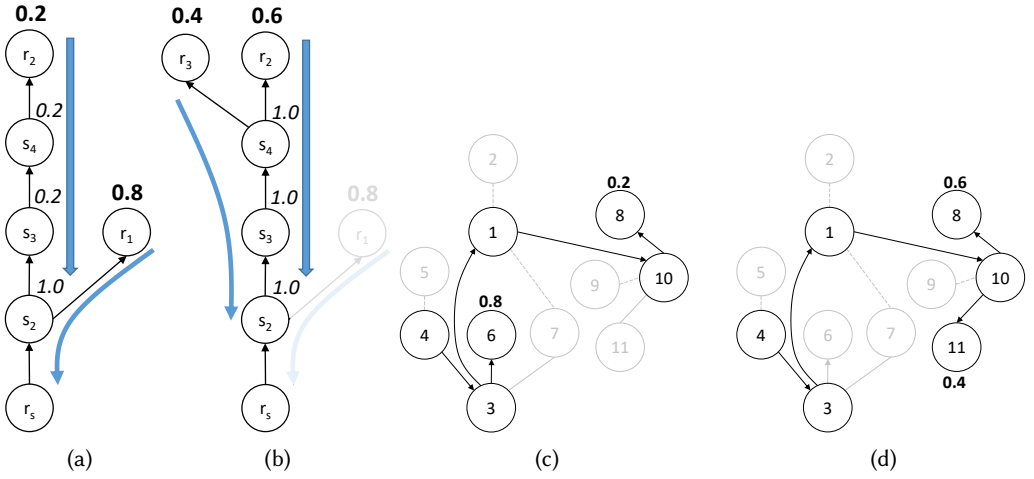


Fig. 2. Back propagation of probabilities associated with the rooms $r_i, i \in \{1, 2, 3\}$ down to the source r_s. (a) Example of normal user movement behavior. (b) Example of attacker movement behavior. (c) The user movement in (a) superimposed on the graph of the building topology. (d) The attacker movement in (b) superimposed on the graph of the building topology. The $diff_{r_s}$ metric returns 1.6.

If the vertex (or space) exists in both vertex sets $P^H_{r_s}$ and $P^K_{r_s}$, the square of the Frobenius norm of the difference between the vertices' scores is taken. However, if the vertex exists in $P^H_{r_s}$ and not in $P^K_{r_s}$, then we set a difference of 0 because the attacker has no knowledge of that vertex and will not move to the space it represents. If the vertex exists in $P^K_{r_s}$ and not $P^H_{r_s}$, we take the squared value of the vertex's score. This is the reverse of the previous situation because the attacker moves to the space represented by the vertex even though the user does not normally move to that vertex at all.[2]

Finally, the difference function $diff(H, K)$ is the sum of the $diff(h_i, k_i)$ for all nonzero rows i in K. The algorithm for calculating the *diff* function is given in Algorithm 1.

$$diff(H, K) = \sum_i \begin{cases} diff(h_i, k_i) & \text{if } h_i \neq \vec{0}^T, k_i \neq \vec{0}^T \\ diff^Z(H, k_i) & \text{if } k_i \neq \vec{0}^T, h_i = \vec{0}^T \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where diff^Z is defined as follows.

$$diff^Z(H, k_s) = \sum_{v \in (P^K_{r_s}) - (\cup_i P^H_i)} Sc_K(v)^2 \tag{7}$$

[2]This implies that the difference metric *diff* is not symmetric.

Algorithm 1 WeightTopo Metric

Require: $G(S, E)$
  function DIFFERENCE(H, K, G)
    $sum \leftarrow 0$
    for all i **where** $k_i \neq \vec{0}^T$ **do**
      $Sc_K = backProp(i, k_i, G)$
      if $h_i \neq \vec{0}^T$ **then**
        $Sc_H = backProp(i, h_i, G)$
      else
        $Sc_H = backProp(i, h_j, G)\ \forall j, h_j \neq \vec{0}^T$
      end if
      for all v **where** $v \in Sc_K - Sc_H$ **do**
        $sum \leftarrow sum + (Sc_K(v))^2$
      end for
      if $h_i \neq \vec{0}^T$ **then**
        for all v **where** $v \in Sc_H \cap Sc_K$ **do**
          $sum \leftarrow sum + (Sc_H(v) - Sc_K(v))^2$
        end for
      end if
    end for
    return sum
  end function

Algorithm 2 BackPropagate

  function BACKPROP($start, [\{r_i, p_i\}], G$)
    $Sc \leftarrow \vec{0}$
    for all $r_i \in [\{r_i, p_i\}]$ **do**
      $path \leftarrow ShortestPath(G, start \rightarrow r_i)$
      for all $v \in path$ **do**
        $Sc(v) \leftarrow Sc(v) + p_i$
      end for
    end for
    return Sc
  end function

Normalizing WeightTopo metric. The range of values assumed by the *WeightTopo* metric changes depending on the building topology G. The reason is the definition of the *WeightTopo* metric given above, which sums up the difference values along the paths in the building topology G. So when we compare *WeightTopo* values between two different building topologies, the same value returned for one building topology, say, a high number like 400, may not convey the same meaning that it would in the context of another building's topology whose maximum *WeightTopo* value may be 800. Since the scales of comparison have different ranges of values, comparing the values does not provide useful insights. Thus, we need to normalize the *WeightTopo* metric to provide a standard unit of comparison.

In order to normalize the metric, we need to determine the range of metric values for a given building topology. The minimum possible value is 0, when the attacker's movement mimics a

portion of the full user movement behavior. The maximum possible value, on the other hand, depends on G. So for a given building topology, we determine the maximum metric value by construction, i.e., we choose the user's transition matrix H_{max} and the attacker's transition matrix K_{max} such that $diff(H_{max}, K_{max}) > diff(H, K)$ for all possible matrices $H, K \in \mathbf{M}$. What follows is a proof by construction of H_{max} and K_{max}.

Proof by construction. For a nonzero row k_i in the attacker's transition matrix K_{max}, the largest possible difference occurs when the corresponding row h_i in the user's transition matrix H_{max} is zero, i.e., $k_i \neq \vec{0}, h_i = \vec{0}$. However, the user's transition matrix cannot be completely zero. So the minimum number of nonzero rows is 1. Thus, the number of nonzero rows in H_{max} is 1, while the number of zero rows in K_{max} is 0. Let the nonzero row in H_{max} be r_s.

We first assume that $P_{r_s}^{H_{max}}$ is an empty set, for simplicity's sake. That does not affect the theorem we will define or the construction method of K_{max}. Later, we will show how constructing H_{max} affects the construction of K_{max} and final *diff* value.

For each row k_i in K_{max} that represents room $r_i \in \mathbf{R}$, the *diff* function returns the sum of the squared values of each vertex along the back-propagated path, $\sum_{v \in P_i^{K_{max}}} Sc_{K_{max}}(v)^2$. We therefore need to choose the set of possible rooms that will be visited next, $L_i^{K_{max}}$, and assign probabilities to each room in the set such that the *diff* function is maximized. We claim that the maximum possible value is achieved when we assign a probability of 1 to the room that is farthest away from r_i, say r_F. This claim is stated and proven in Theorem 1.

Theorem 1. *The maximum diff value for a row k_i is achieved when $k_{ij} = 0$ and $k_{iF} = 1$ $\forall j \in \{1, 2, \ldots |\mathbf{R}|\}, j \neq F$, where r_F represents the room that is located the farthest away from r_i. In other words, $d(r_F, r_i) \geq d(r_j, r_i)$, $\forall r_j \in \mathbf{R}$, where $d(u, v)$ is the graph distance or geodesic distance defined as the number of edges in a shortest path between vertices u and v.*

Proof. By definition, the vertices in $P_i^{K_{max}}$ form a tree in which the root vertex is r_i and the leaf vertices are the vertices in $L_i^{K_{max}}$. *Base case:* The smallest possible subtree that we consider is a starlike tree with a root vertex r_S and two path subgraphs leading to two leaf vertices, r_1 and r_2, as shown in Figure 3. Our aim is to distribute a probability sum of T over the two rooms (or vertices) r_1 and r_2 such that the *diff* value $\sum_{v \in P_S^{K_{max}}} Sc_{K_{max}}(v)^2$ is maximized, i.e., $k_{S1} + k_{S2} = T$.

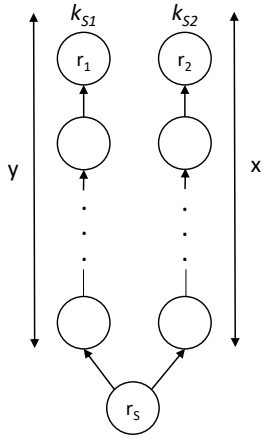


Fig. 3. Base case of tree with r_S as the root vertex.

Let the distance (or the depth) from r_S to r_1 and r_2 be x and y respectively, $d(r_S, r_1) = y, d(r_S, r_2) = x$. W.l.o.g. we assume that $y \geq x$. Now, we prove that choosing $k_{S2} = 0, k_{S1} = T$ produces the maximum *diff* value. We prove by contradiction. Then, choosing a different probability distribution produces a larger value than it would have when $k_{S1} = T$. So we have the following:

$$\sum_x k_{S2}^2 + \sum_y k_{S1}^2 \geq \sum_x 0^2 \sum_y T^2$$
$$x \times k_{S2}^2 + y \times k_{S1}^2 \geq y \times T^2$$
$$x(T - k_{S1}^2)^2 + y k_{S1}^2 \geq y T^2$$
$$x(T^2 - 2T k_{S1} + k_{S1}^2) + y k_{S1}^2 \geq y T^2$$
$$x T^2 - 2T x k_{S1} + x k_{S1}^2 + y k_{S1}^2 \geq y T^2$$
$$x(T^2 - k_{S1}^2) - 2x k_{S1}(T - k_{S1}) \geq y(T^2 - k_{S1}^2)$$
$$(T^2 - k_{S1}^2) - 2k_{S1}(T - k_{S1}) \geq \frac{y}{x}(T^2 - k_{S1}^2)$$
$$1 - \frac{2k_{S1}(T - k_{S1})}{T^2 - k_{S1}^2} \geq \frac{y}{x} \geq 1$$
$$2k_{S1}(T - k_{S1}) = 0$$

Then, either $k_{S1} = T$ or $k_{S1} = 0$. If $k_{S1} = 0$, then $k_{S2} = T$ and thus, $x \times k_{S2}^2 + y \times k_{S1}^2 = x T^2 \geq y \times T^2$ which implies that $x \geq y$. Since we had $y \geq x$, this implies $y = x$. In other words, the maximum value is still $y \times T^2$; thus, there is a contradiction.

Induction: The base case can easily be extended to any tree structure of $P_i^{K_{max}}$. An example is shown in Figure 4. We call vertices of degree greater than two *branch points*. The deepness of a branch point w is quantified by the number of branch points that are ancestors of w, i.e., the number of branch points on the path from the root vertex to w.

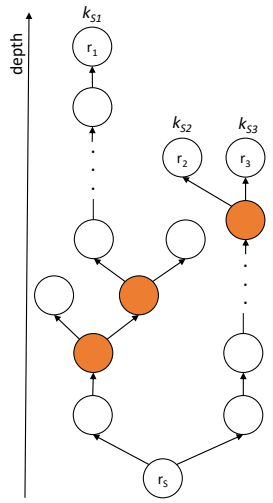


Fig. 4. General tree structure of $P_i^{K_{max}}$. Branch points are shaded.

Given a full tree starting from a root vertex r_S, we start from the deepest branch points, i.e., branch points that are not the ancestors of any other branch points. Assume we want to distribute

a total probability of T_v over the M rooms (or leaves) of the branch point v. We start off with two rooms, say r_j and r_{j+1}, as per the base case and assign the probability T_v to the room with the largest depth from the branch point, i.e., $k_{Sj} = T_v$ w.l.o.g. Then, we iterate the same procedure for all the $M-1$ rooms, resulting in the full probability of T_v being assigned to the room, say r_F located farthest from the branch point v.

Then, we move on to the next deepest branch point that has L descendant branch points. As per the previous exercise, we can remove all the $M-1$ different paths for each of the $v \in L$ branch points, leaving a score of T_v associated with each of the vertices along the path to r_F. We can then repeat the same procedure of distributing $\sum_{v \in L} T_v$ over the L branch points. Thus, we again get the result that the full probability of $\sum_{v \in L} T_v = 1$ will be assigned to the leaf vertex with the longest depth away from the branch point.

We have thus proved by induction that assigning a probability of 1 to the deepest leaf vertex (or associated room) produces the largest *diff* value. □

Now, we describe how to construct the user's transition matrix H_{max}. More specifically, we have to construct a nonzero row r_s such that the path taken by the user has the least intersection with that of the attacker, i.e., $\sum_{j=1}^{n} |\{w | w \in P_{r_s}^{H_{max}} \cap P_j^{K_{max}}\}|$ is minimized. The reason is that all the rows in the constructed matrix K_{max} now have only one nonzero entry and thus $(Sc_{H_{max}}(v) - Sc_{K_{max}}(v))^2$ is either 0 if the vertex v is not in the attacker's path or 1 if it is.

The minimum number of nonzero entries in r_s is one. The greater the number of nonzero entries, the greater the overlap with the attacker's path, so the difference value is maximized if there is only one nonzero entry. To choose the nonzero entry h_{sj}, we iterate through all the possible source-destination room pairs and choose the pair whose shortest path between the two rooms has vertices that appear the least frequently in $P^{K_{max}}$. When a possible source-destination pair is chosen, we update the construction of the attacker's matrix K_{max} by recalculating the distance to leaf vertices. In particular, we do not count vertices that appear in $P_{r_s}^{H_{max}}$ in the distance calculation because we want to minimize the overlap of vertices with the new H_{max}. Then, the final chosen entry is assigned a probability of 1, i.e., $h_{sj} = 1$.

We calculate the maximum value of the *WeightTopo* metric by using the two constructed matrices. Then, we normalize the metric by dividing the result of a *WeightTopo* calculation by that maximum value. In the next section, we will use this normalized *WeightTopo* metric as a basis for our calculations.

4 EVALUATION

In this section, we study how the physical movement-based IDS described in Section 2.3 performs under different attacker models and user movement behaviors. We use real-world data traces to extract user movement behavior through a railway transit station. Then, we generate malicious accesses for each of the attacker models that we describe in Section 3.1 and run those accesses through the physical movement-based IDS. First, we evaluate the *WeightTopo* metric against baseline metrics to judge its capability of measuring the similarity of an attacker's movement behavior to a user's normal movement behavior. Then, we identify which users in the system are more likely to be targeted by different attacker models. Finally, we investigate the amount of similarity between an attacker's movement behavior and a user's movement behavior that is needed to avoid detection by the physical movement-based IDS.

4.1 Experiment Setup

We used a real-world data set containing physical card accesses to a railway station in a city. The station has 57 rooms. In Table 2, we show the distribution of the different subsystem rooms in the

station and the number of users whose jobs belong to those subsystems. There are a total of 314 users in the data set. We discount users who do not visit any rooms (82 in total) because they would not pose a security risk to the system according to our threat model.

Table 2. General statistics about subsystem mappings.

	EC	ME	TM	OP	POW
Num. Rooms	15	9	6	14	13
Num. Users	32	2	33	129	16

To extract the normal user movement behavior and train the detection system, we collected a total of 32,100 accesses made by the 314 users over the period from June to October 2016. The data set contains the following information regarding physical accesses: (1) date and time, (2) door code, (3) user identification, and (4) result of access (success or failure). When the access is a failure, it implies either that the user's card had expired or that the user did not have permission to access the room. The successful accesses were used as training data for the detection algorithm, and the failed accesses served as ground truth on known abnormal accesses.

We simulated the attacker's malicious movement based on a given transition matrix K. In this paper, we evaluate the three attacker models in their purest form, excluding any noise vector (i.e., $\mathbf{e} = \vec{0}$). We varied the number of rooms that are visited by the attacker from 1 to 10. For a given number of rooms N, we performed the following steps recursively for $r_i, i \in \{1, 2, \ldots N\}$. To choose a target room r_i, we randomly sampled a room from the non-negative entries in row $k_{r_{i-1}}$. We calculated the shortest path from the previous room r_{i-1} to r_i as $r_{i-1}e_1S_2 \ldots S_n e_n r_i$. For each edge $e_i, i \in \{1, 2, \ldots n\}$ that has a door code, we injected access $A_i = S_i \rightarrow S_{i+1}$. For the initial case r_1, we randomly sampled a room s (row k_s) and used the concourse public area as r_0. (If there had been more entrances to the station, we could have randomly chosen r_0 from those spaces.) Then, we repeated the process 50 times.

The IDS took in a user's trained normal behavior and, for each of the attacker's injected accesses, decided whether the attacker's access was suspicious or not. Then we determined the probability that the attacker's malicious movement would be detected before he or she entered a room in the station. We repeated this process for all the users in the dataset.

We conducted the experiments on a Windows 7 Home Premium machine with a 2.7 GHz CPU core and 4 GB of RAM. The results are presented in the following subsections.

4.2 Metric Choice

To evaluate how well the *WeightTopo* metric represents the similarity of an attacker's movement behavior to a user's movement behavior, we compared the metric against two baselines: a purely Frobenius norm function (*Values*) and a purely topology-based function (*Topo*).

As shown in Table 3, we calculated the Pearson correlation between each of the three difference metrics and the detection probability for the *Amateur Outsider* (AO), *Informed Outsider* (IO), and *Insider* attacker models.[3] A stronger correlation implies that the metric is more closely tied to detection probability. We see that the correlation for our *WeightTopo* metric is moderately positive, indicating that a higher difference is tied to a higher detection probability. This is true because the attacker has less information about the user's behavior, and thus the attacker moves in a very different manner. Thus, the attacker is more easily detected. In comparison to the other two baseline metrics, our metric has a higher correlation in general, with the exception of the AO and TM insider

[3]We do not include the Pearson correlation for ME insiders because there are only 2 users that belong to the ME subsystem.

Table 3. Pearson correlation of the three difference metrics: *Values*, *Topo*, and our own *WeightTopo* metric.

		Values	Topo	WeightTopo
Amateur Outsider (AO)		-0.38	0.50	0.49
Informed Outsider (IO)		0.04	0.37	0.44
Insider	Station Operation Control (OP)	0.13	0.32	0.56
	Environment Control (EC)	-0.006	0.41	0.65
	Transport Mechanical Equipment Control (TM)	0.06	0.69	0.60
	Power Control (POW)	-0.41	0.36	0.52

models, for which the correlation is slightly lower than for the *Topo* metric. Thus, in general, our metric provides a better measure of the similarity between an attacker's movement behavior and a user's movement behavior.

4.3 Attacker Model

We identify the user roles that are more likely to be targeted by an attacker. We also investigate the amount of similarity to a user's movement behavior that an attacker needs to achieve in order to remain relatively undetected.

4.3.1 Amateur Outsider. As described in Section 3.1, the attacker in this model has no information about the user behavior or subsystem-room mapping. The *WeightTopo* difference metric between the users in the dataset and this attacker ranges from 0.1 to 0.34. We generated the attacker's accesses by using the transition matrix defined in Section 3.1.

Figure 5a shows the detection probability of the attacker for all users. As the malicious path length increases, the detection probability increases, since the attacker deviates further and further from the user's normal behavior. The minimum detection probability shown here is 71.4% when the path length is 1.

Thus, this attacker model is easily detectable by the defenses in place. In other words, the system is able to detect with high confidence any attacker whose behavior differs from the true users' behaviors by a metric value of more than 0.1. Thus, this attacker will fail no matter which user he or she targets.

4.3.2 Informed Outsider. This attacker model represents an attacker who has information about the building layout and subsystem rooms. Thus, the *WeightTopo* value decreases to between 0.01 and 0.16 because the attacker is able to narrow down the number of rooms to enter. As shown in Figure 5c, the detection probability is correspondingly lower than for the *Ignorant Outsider* attacker model but still remains high, with the average detection probability higher than 0.7. Therefore, this attacker model is still detectable with relative ease.

Of the different subsystems, the users belonging to the TM subsystem have the lowest difference value and are spread out over a smaller range, as shown in Figure 5b. The reason is that the TM subsystem has the fewest rooms, so the difference in movement behavior between the TM users and the attacker would be lower than for other subsystems. The detection probability for TM users is also correspondingly the lowest, so this attacker model would choose to masquerade as the TM users to have a higher chance of avoiding detection by the physical movement-based IDS.

4.3.3 Insider. For this attacker model, we use other users' transition matrices to generate the attacker's malicious accesses. The *WeightTopo* metric in this case varies from 0 to 0.08, which is lower than for the two previous attacker models. In other words, this attacker has the most information about the users and can mimic their movement behavior more closely.

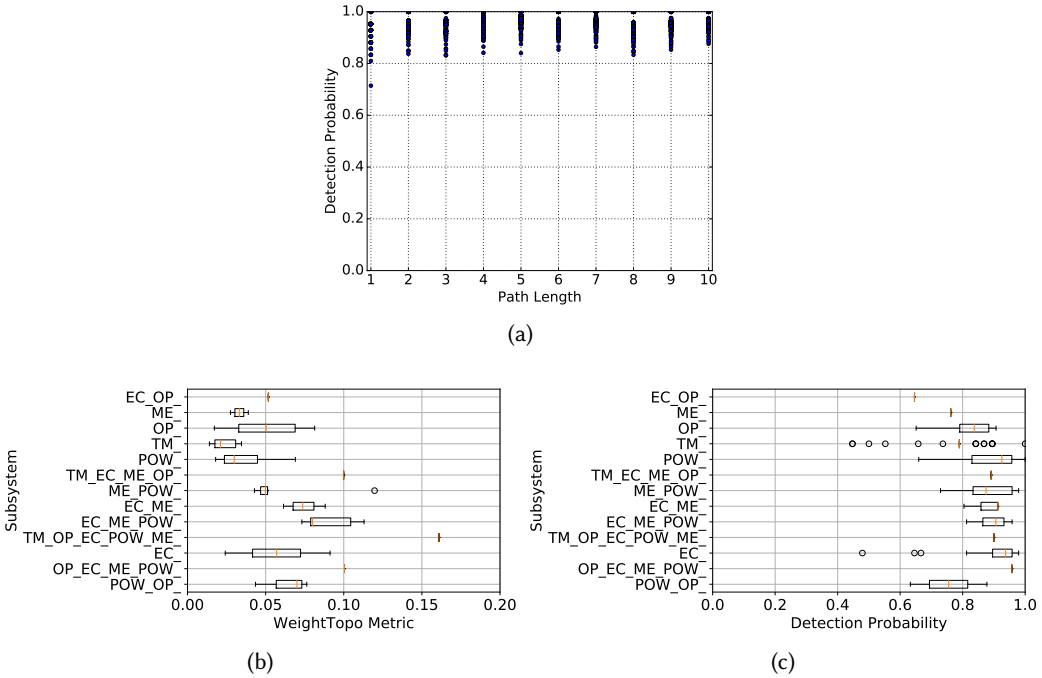
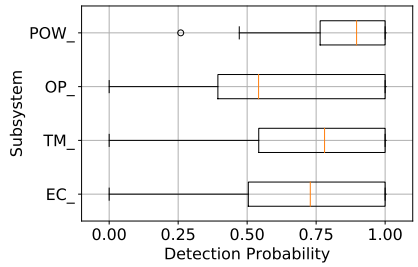


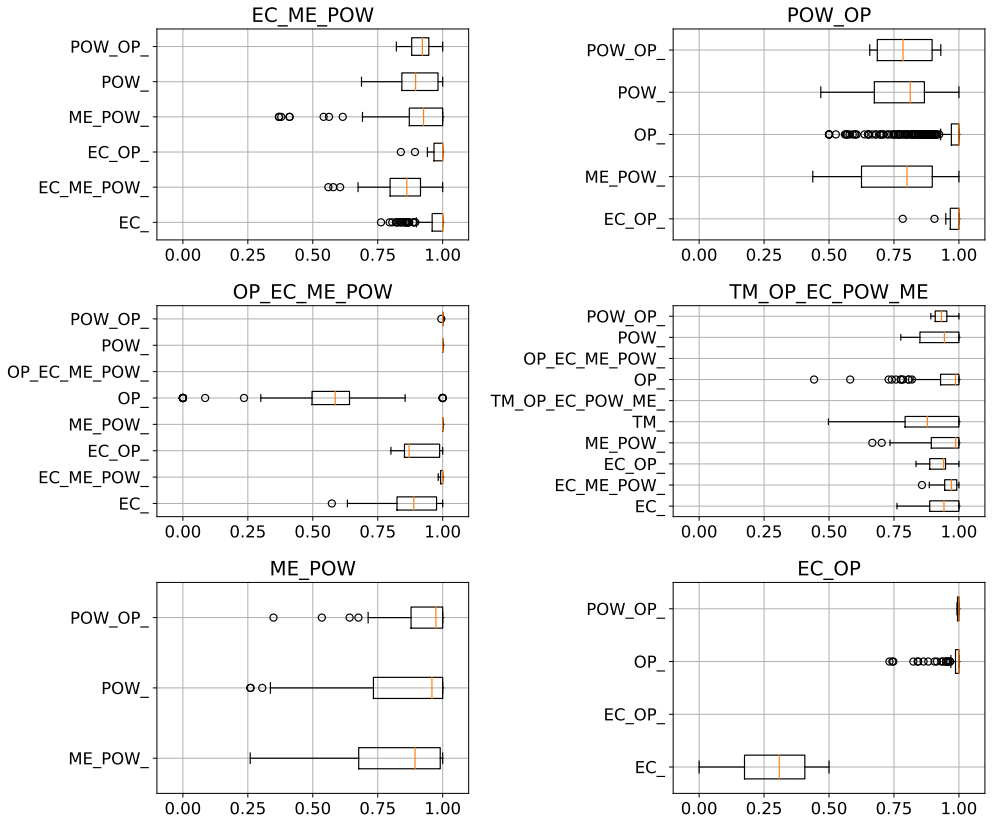
Fig. 5. (a) Detection probability of the *Amateur Outsider* attacker model as the malicious path length increases from one visited room to ten. (b) Range of *WeightTopo* values for the *Informed Outsider* attacker model. (c) Detection probability of the *Informed Outsider* attacker model.

For any user u_i associated with a subsystem $assign(u_i) = sys_i$, attackers u_a whose accesses are from a different subsystem $assign(u_a) = sys_j, sys_j \cap sys_i = \phi$ will always be detected with a 100% success rate. The reason is that the rooms associated with different subsystems are disjoint from each other, so the normal behavior models of users from one subsystem will completely deviate from those of another subsystem. Thus, the IDS will always detect attackers whose accesses are from a different subsystem than the user's. Therefore, attackers will choose to masquerade as users who share common subsystems with them. Figure 6a shows the detection probability for attackers who belong to the same subsystem as the users they are masquerading as, whereas Figure 6b shows the detection probability for attackers who belong to a subset of the user's subsystems. Attackers who belong to both the ME and POW subsystems would target users in the same subsystem or in the $\{EC, ME, POW\}$ and $\{POW, OP\}$ subsystems, since they have the lowest detection rates. Attackers who belong to the EC subsystem would target users in the $\{EC, OP\}$ subsystem and have a very low probability of being detected. Attackers who belong to the OP subsystem have the lowest detection rate out of all subsystems because the rooms that are visited by users in the OP subsystem are spread out through the station. Those attackers can target users from the same subsystem or users in $\{OP, EC, ME, POW\}$ and $\{TM, OP, EC, ME, POW\}$. Thus, attackers who belong to the OP subsystem have the largest variety of potential targets.

Next, we looked at how closely an attacker should mimic a user's movement behavior to avoid being detected by the IDS. In Figure 7, we show the relationship between the *WeightTopo* value and detection probability for pairs of singleton subsystem sets. We excluded the ME subsystem and the



(a)



(b)

Fig. 6. The distributions of the detection probabilities for each pair of subsystems. The y-axes of the plots represent the subsystems of the attacker's generated accesses. The subsystems of the users are either (a) the same as the attacker's subsystem or (b) shown in the plot titles.

subsystem sets that contain more than one subsystem because each of those sets contains fewer than 20 users and as such is not statistically significant enough that we can assess the relationship. For each plot, we found the x-value or the difference value for which more than 95% of the points to the right of the value have a higher than 50% detection probability. The horizontal red line in

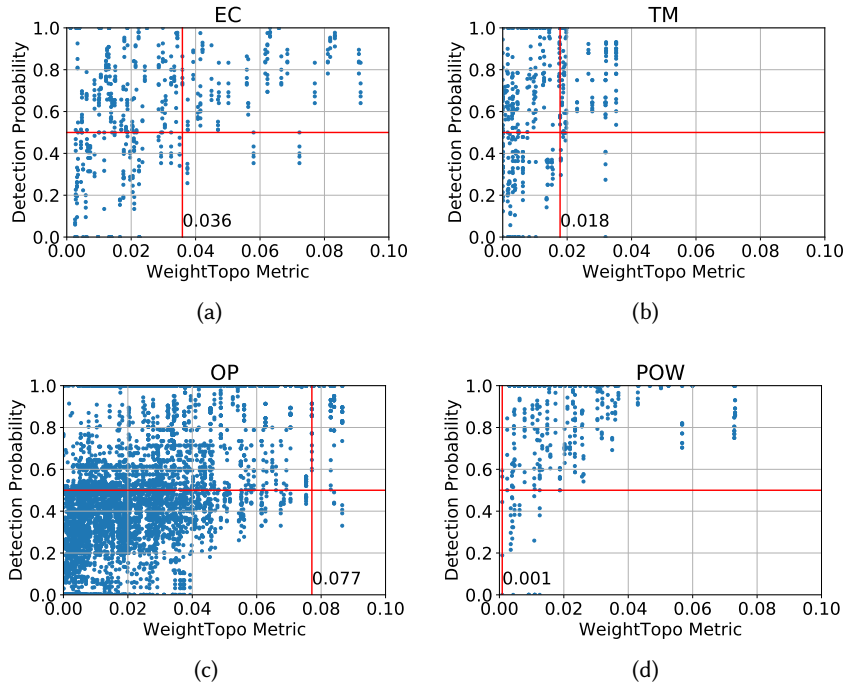


Fig. 7. The detection probability for attackers within the same subsystem vs. the *WeightTopo* metric.

each plot represents the 50% threshold, and the vertical red line represents the difference value that we found. In other words, we determined the minimum difference between an attacker and user needed in order for the IDS to detect the attacker's accesses with a probability higher than 0.5. So the higher the difference value, the easier it is for an attacker to mimic the user's movement behavior while remaining below the radar of the IDS.

Figure 7 shows the detection probability for attackers within the same subsystem. The subsystems in which it is the most difficult for an attacker to mimic a legitimate user while remaining relatively undetected are, in descending order, the POW subsystem, the TM subsystem, the EC subsystem, and then the OP subsystem. The reason is the distribution of the subsystem rooms within the station. For example, of all the subsystems, the POW subsystem's rooms are the most concentrated in a particular area. Thus, an attacker needs to mimic the exact movement behavior of a user in order to avoid being detected. The vertical red line for the OP subsystem, on the other hand, is the farthest to the right of all the vertical red lines. In other words, an attacker using his OP movement behavior can easily remain undetected if he has another OP user's access card. Thus, the attacker does not need to conform strictly to the user's normal movement behavior, and thus this attacker model will choose to masquerade as an OP user. This implies that the system needs to differentiate between OP users at a more fine-grained level by including either timing information or additional contextual information regarding the movement (e.g., clocking out vs. break time).

With a greater than 50% chance, our system is thus able to detect OP attackers whose *WeightTopo* values are greater than 0.077, POW attackers with difference values greater than 0.001, TM attackers whose difference values are greater than 0.018, and EC attackers whose difference values are greater than 0.036.

Finally, we look into the extreme cases in which the individual attacker's malicious accesses are not detected at all (i.e., the detection probability is zero) when he or she possesses an access card of a user within the same subsystem. We want to identify the individual users who can easily be imitated so that the system administrator can focus monitoring efforts on them.

Table 4. Numbers of users who can imitate others, numbers of users who can be imitated by an attacker; and numbers of users who can both imitate and be imitated by others, for each subsystem.

	OP	**EC**	**POW**	**TM**
Total # of users	134	32	16	41
# of users who can be imitated by others	89	4	2	3
# of users who can imitate others	4	6	1	4
# of users who can both imitate and be imitated	34	0	0	0
Avg out-degree of users who can imitate others	8	4	2	1

First, we collated all the cases in which an attacker (whose movement behavior is represented by user u_i's transition matrix H_i) masquerades as a user u_j and the attacker's malicious accesses (generated using H_i) are not detected at all by the IDS. Then, we can represent each of those (attacker,user) pairs (u_i, u_j) as a directed edge $e(u_i, u_j)$. In the resulting graph, users who can only imitate others (attackers) have an in-degree of zero, whereas users who can only be imitated by others have an out-degree of zero. Users who can both imitate and be imitated by others have an in-degree and out-degree that are nonzero. We count the numbers of such users in the graph and tabulate them in Table 4.

We find that when an attacker can masquerade as another user, it implies that the attacker's transition matrix K is more sparse than the user's transition matrix H and that the attacker's visited places are a subset of the user's. In other words, the graph represents a hierarchy of users in which the innocent users at the top have a large transition matrix and a large number of visited rooms, and the pure attackers at the bottom have a sparse transition matrix and the smallest set of visited rooms.

As shown in Table 4, the OP subsystem has the highest number of individual attackers who can imitate other users. Only the OP subsystem has users who can be both attackers and innocent users. So the users in this subsystem have a more diverse movement behavior than those in other subsystems.

We also calculated the average out-degree for each vertex (excluding innocent users). If the average out-degree is higher, the attacker has a larger choice of access cards to compromise. For example, the TM subsystem has the lowest average out-degree of 1. Therefore, any of the four attackers with their given movement behavior must masquerade as one specific user so that their malicious accesses will remain undetected. On the other hand, the EC subsystem has an average out-degree of 4. The graph for this subsystem is fully bipartite (with the exception of one attacker), as shown in Figure 8. This implies that any of the five attackers can choose to compromise any of the four users and remain undetected. So a physical movement-based IDS will be better tailored to these pairs of users if it compares their timings of movement and durations of stay in rooms. Finally, the OP subsystem has the highest out-degrees of 8. Thus, users from the OP subsystem are the most easily imitated by attackers and should be more closely monitored.

4.3.4 Concluding Remarks. To summarize our results obtained from running the intrusion detection system in [6], we answer the three questions that we brought up in Section 1: (1) what factors help to characterize differences in movement behavior, (2) which users are more likely to be

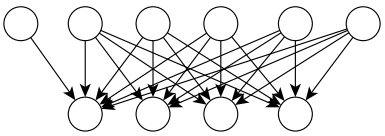


Fig. 8. Graph of (attacker,user) pairs for the EC subsystem.

targeted by attackers, and (3) how closely must an attacker mimic a user's movement behavior for the attacker to remain undetected.

We calculated the Pearson correlation between each of the three difference metrics and the detection probability. Metrics that took the building topology into consideration (*Topo* and *WeightTopo*) had a much higher correlation to the detection probability than did metrics that ignored the effect of topology (*Values*). Thus, the building topology is a major factor in characterizing differences in movement behaviors.

Summarizing our results, we find that different types of attackers will choose different users to masquerade as. The amateur outsider is very detectable regardless of which user he or she masquerades as. The informed outsider and the insider attackers will choose to masquerade as the TM users and OP users, respectively, since as those users, they have the lowest chances of being detected. Among the users who are assigned to multiple subsystems, the users assigned to the $\{OP, EC, ME, POW\}$ and $\{EC, OP\}$ subsystem sets are more likely to be targeted. We also identified specific users in each subsystem who can easily be imitated. Thus, these results give us insight into which users or types of users should be closely monitored.

Finally, we investigated how much an attacker's movement behavior needs to differ from a user's movement behavior for the IDS to detect the attacker's accesses with a relatively high probability. Our results show that for OP users, the insider attacker requires the least similarity to the user's movement behavior to remain undetected. The reason is that the rooms visited by those users are spread out through the station, and the paths taken to those rooms overlap a lot in terms of the spaces that are accessed. So additional contextual information about OP users' movement behavior (e.g., timing, physical constraints) is needed to help an IDS distinguish between normal and malicious movement.

5 RELATED WORK

Much related research focuses on evaluating the cyber threat in a system with respect to different attacker models, such as insiders, hackers, and nation-state attackers [22]. In the domain of cyber-physical systems, attacker models like those in [1, 21] include both cyber attacks and physical attacks (e.g., damaging equipment). In this paper, we focus on evaluating the threat of physical intrusions in terms of malicious physical movements. Building access control systems that use access cards or biometrics (e.g., fingerprints, iris, facial features) are only able to detect malicious movement when an unauthorized user tries to gain access to a space. However, those approaches are not able to detect the malicious behavior of an authorized user. Several different IDSes have been proposed for detection of malicious physical movement [5–9, 17, 20]. We take their work one step further by assessing the risks to a system presented by a variety of attackers who move in different manners, given a specific IDS that is installed to detect abnormal movement behavior.

We define three types of attacker models in terms of the amount of information they possess regarding the system and users' movement behavior. In particular, we model the attacker's movement using a transition matrix. Much like our work, [27] quantifies attacker knowledge in terms of

a transition counts matrix. The aim of the attacker in that work is to infer the exact user movement behavior or transition matrix and overcome the location privacy protection. The attacker in our work has a similar goal of wanting to mimic user movement behavior so as to remain undetected by the IDS. However, in [27], the authors quantify success in terms of whether the attacker can identify the specific user associated with the movement, whereas in our case, success is measured by the detection rate.

We measure the amount of information the attacker possesses by calculating the difference between the attacker's and user's movement trajectories, i.e., the difference between their Markov models. The difference in movement trajectories has been explored in domains such as social spaces and geographic information systems [10, 25]. Those approaches focus on comparing trajectories that are in free space by using features such as topological distances and timing [26, 28]. However, we focus on movement within a confined space. Also, we compare a set of trajectories instead of a single trajectory.

Hidden Markov Model (HMM)-based clustering of sequences is a domain of work that is similar to calculation of differences between Markov models [11, 19]. For example, [12] involves clustering of time series of health data based on the distances between HMMs. The techniques used in those domains to calculate differences between HMMs involve calculation of the likelihood of obtaining a trajectory based on the model, and calculation of the KL divergence on the rows in the transition matrix. However, those techniques do not take into account the topology of the underlying space. For example, consider the approach taken by Porikli and Haga [24] to measure the difference between trajectories, which is the prior work most similar to our *WeightTopo* metric. They represent the trajectory as a set of coordinates, orientation, and speed. The trajectory is then fed into an HMM. They compare two trajectories by computing the cross-fitness of the trajectories to each other's trained HMMs. Their application domain is in geographical spaces that are not delineated by rooms. If we wanted to extend their approach to our domain of indoor movement trajectories, the coordinates could refer to either rooms or general spaces in the building. Since a user can take multiple paths to reach a single room, modeling of the movement at the level of spaces is too low-level and would result in an overfitting of the HMM. Constructing an HMM by using the sequence of rooms would then lead to the same problem as using a simple Frobenius norm of two transition matrices, since topology is not taken into account.

6 CONCLUSION

The physical security of an organization is an important aspect of its overall security posture. In this paper, we defined three different attacker models based on the amount of information they possess regarding the building layout and legitimate users' movement behavior. To analyze how they impact the performance of an intrusion detection system, we defined a metric *WeightTopo* to better characterize the difference between transition matrices that represent different physical movement behaviors. Finally, we evaluated our approach on a railway station in which users are assigned to different subsystem sets or roles. Our results indicate that for different attacker models, different user roles will be imitated and this informs our decision to increase monitoring for those roles. Our defined metric thus allows us to quantitatively compare how well detection systems work for different user roles and under different attacker models.

A TABLE OF SYMBOLS

Symbol	Description
U	Users in the system
G	Building topology graph
E	Set of edges in building topology graph
$e(v_1, v_2)$	Edge connecting vertices v_1 and v_2 in building topology graph
S	Spaces in building
$\mathbf{R}$	Rooms in building
$\mathbf{C}$	Common areas in building
$Subsys$	List of subsystems
$label$	Labeling of vertex with subsystems associated with it
$assign$	Assignment of users to subsystems
$Perm$	Rooms that user is allowed to access
Mov	Movement behavior model of users
Kwg	Information possessed by attacker
M	Transition matrix
H	User's movement transition matrix
K	Attacker's movement transition matrix
$\mathbf{e}$	Noise vector
WeightTopo	Our metric to quantify similarity of attacker's and user's movement behavior
diff(.)	Difference function between two transition matrices
h_i, k_i	Row i of transition matrix
r_s	Source room
$L^M_{r_s}$	Set of possible rooms to be visited next after r_s based on transition matrix M
$P^M_{r_s}$	Set of spaces on the shortest paths between r_s and vertices in $L^M_{r_s}$
$path(.)$	Shortest path between the two vertices
$Sc_M(v)$	The score of a space v based on transition matrix M
H_{max}	Constructed user transition matrix that produces highest difference values
K_{max}	Constructed attacker transition matrix that produces highest difference values
$d(.)$	Geodesic distance, i.e., number of edges in a shortest path between the vertices

REFERENCES

[1] S. Adepu and A. Mathur. 2016. Generalized Attacker and Attack Models for Cyber Physical Systems. In Proc. IEEE 40th Annual Computer Software and Applications Conference, Vol. 1. 283–292.

[2] Graeme Baker. 2008. Schoolboy hacks into city's tram system. The Telegraph (January 2008). http://www.telegraph.co.uk/news/worldnews/1575293/Schoolboy-hacks-into-citys-tram-system.html

[3] James Bayne. 2002. An Overview of Threat and Risk Assessment. Technical Report. SANS Institute. https://www.sans.org/reading-room/whitepapers/auditing/overview-threat-risk-assessment-76

[4] B. Biggio, Z. Akhtar, G. Fumera, G. L. Marcialis, and F. Roli. 2012. Security evaluation of biometric authentication systems under real spoofing attacks. IET Biometrics 1, 1 (March 2012), 11–24. https://doi.org/10.1049/iet-bmt.2011.0012

[5] Robert P Biuk-Aghai, Yain-Whar Si, Simon Fong, and Peng-Fan Yan. 2012. Individual movement behaviour in secure physical environments: Modeling and detection of suspicious activity. In Behavior Computing, Longbing Cao and Philip S. Yu (Eds.). Springer, 241–253.

[6] Carmen Cheh, Binbin Chen, William G. Temple, and William H. Sanders. 2017. Data-Driven Model-Based Detection of Malicious Insiders via Physical Access Logs. In Proc. 14th International Conference on Quantitative Evaluation of Systems, Nathalie Bertrand and Luca Bortolussi (Eds.). Springer International Publishing, 275–291. https://doi.org/10.1007/978-3-319-66335-7_17

[7] Michael Davis, Weiru Liu, Paul Miller, and George Redpath. 2011. Detecting Anomalies in Graphs with Numeric Labels. In Proc. 29th ACM Conf. on Information and Knowledge Management. 1197–1202.

[8] William Eberle and Lawrence Holder. 2007. Anomaly Detection in Data Represented as Graphs. Intelligent Data Analysis: An International Journal 11, 6 (2007), 663–689.

[9] William Eberle, Lawrence Holder, and Jeffrey Graves. 2009. Detecting Employee Leaks Using Badge and Network IP Traffic. In Proc. IEEE Symposium on Visual Analytics Science and Technology.
[10] Andre Salvaro Furtado, Despina Kopanaki, Luis Otavio Alvares, and Vania Bogorny. 2016. Multidimensional Similarity Measuring for Semantic Trajectories. Transactions in GIS 20, 2 (2016), 280–298.
[11] D. García-García, E. Parrado Hernández, and F. Díaz de María. 2009. A New Distance Measure for Model-Based Sequence Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 31, 7 (July 2009), 1325–1331. https://doi.org/10.1109/TPAMI.2008.268
[12] Shima Ghassempour, Federico Girosi, and Anthony Maeder. 2014. Clustering multivariate time series using hidden Markov models. International journal of environmental research and public health 11, 3 (2014), 2741–2763.
[13] Shelby Grad. 2009. Engineers who hacked into L.A. traffic signal computer, jamming streets, sentenced. Los Angeles Times (1 December 2009). http://latimesblogs.latimes.com/lanow/2009/12/engineers-who-hacked-in-la-traffic-signal-computers-jamming-traffic-sentenced.html
[14] Andy Greenberg. 2015. OPM now admits 5.6M Feds' Fingerprints were stolen by hackers. Wired (23 September 2015). https://www.wired.com/2015/09/opm-now-admits-5-6m-feds-fingerprints-stolen-hackers/
[15] Alex Hern. 2014. Hacker fakes German minister's fingerprints using photos of her hands. The Guardian (30 December 2014). https://www.theguardian.com/technology/2014/dec/30/hacker-fakes-german-ministers-fingerprints-using-photos-of-her-hands
[16] Alex Hern. 2017. Samsung Galaxy S8 iris scanner fooled by German hackers. The Guardian (23 May 2017). https://www.theguardian.com/technology/2017/may/23/samsung-galaxy-s8-iris-scanner-german-hackers-biometric-security
[17] Mark J. Hoesl. 2014. Integrated Physical Access Control and Information Technology Security. (2014). U.S. Patent No. 6641090 B2, granted on Jun 17 2014.
[18] Stephen Irwin. 2014. Creating a Threat Profile for Your Organization. Technical Report. SANS Institute. https://www.sans.org/reading-room/whitepapers/threats/creating-threat-profile-organization-35492
[19] Ming Ji, Fei Wang, Jia Ning Wan, and Yuan Liu. 2015. Literature Review on Hidden Markov Model-Based Sequential Data Clustering. In Mechatronics Engineering and Modern Information Technologies in Industrial Engineering (Applied Mechanics and Materials), Vol. 713. Trans Tech Publications, 1750–1756. https://doi.org/10.4028/www.scientific.net/AMM.713-715.1750
[20] Himanshu Khurana, Valerie Guralnik, and Robert Shanley. 2014. System and Method for Insider Threat Detection. (2014). U.S. Patent No. 8793790 B2, granted on Jul 29 2014.
[21] L. Langer, P. Smith, and M. Hutle. 2015. Smart grid cybersecurity risk assessment. In International Symposium on Smart Electric Distribution Systems and Technologies. 475–482.
[22] Richard P Lippmann and James F Riordan. 2016. Threat-Based Risk Assessment for Enterprise Networks. Lincoln Laboratory Journal 22, 1 (2016), 33–45.
[23] Wayne Penny. 2019. Biometrics: A Double Edged Sword - Security and Privacy. Technical Report. SANS Institute. https://www.sans.org/reading-room/whitepapers/authentication/biometrics-double-edged-sword-security-privacy-137
[24] F. Porikli and T. Haga. 2004. Event Detection by Eigenvector Decomposition Using Object and Frame Features. In Proc. 2004 Conference on Computer Vision and Pattern Recognition Workshop. 114–123.
[25] Peter Ranacher and Katerina Tzavella. 2014. How to compare movement? A review of physical movement similarity measures in geographic information science and beyond. Cartography and Geographic Information Science 41, 3 (2014), 286–307.
[26] Mohammad Sharif and Ali Asghar Alesheikh. 2017. Context-awareness in similarity measures and pattern discoveries of trajectories: A context-based dynamic time warping method. GIScience & Remote Sensing 54, 3 (2017), 426–452. https://doi.org/10.1080/15481603.2017.1278644
[27] R. Shokri, G. Theodorakopoulos, J. Y. Le Boudec, and J. P. Hubaux. 2011. Quantifying Location Privacy. In IEEE Symposium on Security and Privacy. 247–262. https://doi.org/10.1109/SP.2011.18
[28] Kevin Toohey and Matt Duckham. 2015. Trajectory Similarity Measures. SIGSPATIAL Special 7, 1 (May 2015), 43–50. https://doi.org/10.1145/2782759.2782767
[29] Yi Xu, True Price, Jan-Michael Frahm, and Fabian Monrose. 2016. Virtual U: Defeating Face Liveness Detection by Building Virtual Models from Your Public Photos. In 25th USENIX Security Symposium (USENIX Security 16). USENIX Association, Austin, TX, 497–512. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/xu

Received February 2007; revised March 2009; accepted June 2009