ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-1045

On the Usage of Behavior Models to Detect ATM Fraud

Timo Klerx and **Maik Anderka** and **Hans Kleine Büning**[1]

Abstract. The detection of ATM fraud is a key concern for both financial institutes and bank customers but also for ATM suppliers. This paper deals with the algorithmic learning of an ATM's behavior model given the data stream of status information produced by standard mechatronic devices embedded in modern ATMs. During operation, the observed status information is compared with the learned reference model to detect abnormal behavior—assuming that a significant anomaly is a strong indicator of a fraud attempt. In contrast to previous work on automatic ATM fraud detection, we apply a class of models that also capture the timing behavior, thus covering a broader range of fraud and manipulation. In particular, we present an approach to learn a tailored behavior model, called *Probabilistic Deterministic Timed-Transition Automaton*, in order to enable the detection of time-based anomalies. We also report on preliminary results of an empirical evaluation using a real-world data set recorded on a public ATM, indicating the practical applicability of our approach.

1 Introduction

Automated teller machines (ATMs) are subject to various kinds of attacks and fraud. Common types of ATM fraud include skimming, card- or cash trapping, the installation of malicious software, and also various physical attacks. The automated detection of ATM fraud to prevent loss or damage is a relevant research topic. Previous studies only target single aspects like optical protection [6], image-based fraudster identification [8], and card fraud detection [7]. To the best of our knowledge, our prior study [1] is the first that endeavors to provide a holistic approach for the automatic detection of ATM fraud.

In our previous work, we defined ATM fraud detection as a sequence-based anomaly detection problem, and we utilized machine learning techniques to identify abnormal patterns in the discrete sequence of status events that is produced inside an ATM [1]. Although the anomaly detection effectiveness is quite promising, this approach is only able to detect a limited number of possible attacks because the time intervals between subsequent status events are disregarded. In practice, however, several types of attacks manifest themselves solely in an abnormal time behavior. Consider for example a skimming device mounted on the card reader that results in a slower card insertion process. This paper tries to close this gap by proposing a model-based anomaly detection approach that is able to detect not only sequence-based anomalies but also time-based anomalies.

2 Problem Formulation

We interpret an ATM as a discrete-state, event-driven system [3]. I.e. events occur at various time instants and cause transitions between system state. Events can be triggered by internal components, like sensors and actuators of mechatronic devices, or by customer input, like inserting the card and entering the PIN. This interpretation is in line with the observable system behavior in a practical application, where the ATM's Diagnosis and Serviceability module provides a timed sequence of status events that comprehensively describes the ATM's real-time behavior (for further information, refer to [1]).

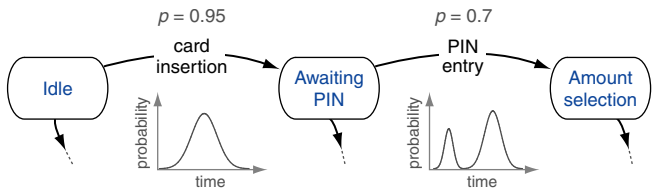

Figure 1. Example of a *Probabilistic Deterministic Timed-Transition Automaton* (PDTTA).

Given a timed event sequence of normal observations, the task is to learn a behavior model of an ATM that can be used to detect both sequence-based anomalies as well as time-based anomalies. To achieve this, we propose a tailored behavior model that explicitly capture the timing behavior, called *Probabilistic Deterministic Timed-Transition Automaton* (PDTTA). Figure 1 shows an example of a PDTTA. For each transition a probability p for taking the transition is given. Additionally, a probability density function describes the relative likelihood for the event's timing (e.g. the time it takes to enter the PIN). In the following, a PDTTA is formally defined:

Definition 1 *A Probabilistic Deterministic Timed-Transition Automaton is a tuple* $A = (S, s_o, \Sigma, T, \xi, \tau)$*, where*

- S *is a finite set of states, with* $s_0 \in S$ *the start state.*
- Σ *is a finite alphabet comprising all relevant events.*
- $T \subseteq S \times \Sigma \times S$ *is a finite set of transitions. E.g.* $\langle s, e, s' \rangle \in T$ *is the transition between states* $s, s' \in S$ *triggered by event* $e \in \Sigma$*.*
- $\xi: T \mapsto [0, 1]$ *is a transition probability function, which assigns each transition a probability value* p*.*
- $\tau: T \mapsto \Theta$ *is a time probability function, which assigns each transition a probability distribution* $\theta \in \Theta$*, with* Θ *being the set of all possible probability distributions. Every* $\theta \in \Theta$ *has the signature* $\theta: \mathbb{I} \mapsto [0, 1]$ *with* $\mathbb{I} \subseteq \mathbb{N}$ *a set of time values.*

Note that this model is similar to a probabilistic deterministic timed automaton (PDTA). But in contrast to a PDTA, the future execution in a PDTTA only depends on the events—the timing is inferred afterwards. Moreover, no specific acceptance states are required because for a given input sequence the automaton is used to compute the likelihood that the sequence is generated by the model. This likelihood is then interpreted as the sequence's anomaly score.

For this particular model class, model learning is still a challenge [5]. To the best of our knowledge, there are only two ap-

[1] Department of Computer Science, University of Paderborn, Germany, email: {timo.klerx, maik.anderka, kbcsl}@uni-paderborn.de

proaches that address the learning of PDTAs [9, 4]. However, both do not cover all types of anomalies because they model the timing behavior by rigid intervals instead of probability distributions.

3 Method

We propose a two-step approach to learn the model specified in Definition 1, given a set X that comprises timed event sequences $x = \langle e_1, i_1\rangle, \langle e_2, i_2\rangle, \ldots, \langle e_n, i_n\rangle$ of variable length:

Step 1. Learn an initial generative model. When omitting τ in Definition 1, the resulting automaton corresponds to a probabilistic deterministic finite automaton (PDFA). We take advantage of the fact that the learning of PDFAs can be considered as being state-of-the-art [5]. Hence, in this first step, a PDFA A' is learned from X using the well-known ALERGIA algorithm, described in [2]. ALERGIA is a state merging algorithm that first builds a prefix tree acceptor and then merges compatible states recursively.

Step 2. Learn the timing behavior and augment the initial model. For every timed sequence $x \in X$ we traverse A' from the initial state s_0 and whenever we take a transition $t = \langle s, e_k, s'\rangle \in T$, we add the associated time interval i_k to the set of intervals I_t that belongs to t. For every such I_t we fit a probability distribution θ_t that models the occurrences of time intervals in I_t. θ_t is associated with τ so that $\tau(t) = \theta_t$. We tested various approaches, e.g. we fitted common distributions (like normal, exponential, etc.) or first clustered the intervals $i \in I$ to find different modes and then fitted a common distribution to every cluster. Finally, we decided to use kernel density estimators with a Gaussian kernel because they lead to better results. Combining A' and τ gives us a PDTTA A.

Given the learned model A, the anomaly score $p_A(x)$ for an input sequence x is computed as follows: Analogous to step 2, we traverse A starting in s_0. For every tuple $\langle e_k, i_k\rangle \in x$ we take the transition $t = \langle s, e_k, s'\rangle \in T$. Additionally, we store all event probabilities $p_e(t) = \xi(t)$ that indicate how likely it is to traverse t in s, and the time probabilities $p_i(t) = \tau(t)(i)$ that indicate how likely it is that traversing t takes i time units. Then, we aggregate all event and time probabilities to $p_A(x)$, which is the probability of x being generated by A. Finally, $p_A(x)$ is compared to a threshold c. If $p_A(x)$ is lower than c we classify x as anomaly, otherwise as normal.

4 Analysis and Results

To evaluate the anomaly detection effectiveness of our approach, we use a data set that has been recorded on a Wincor Nixdorf ATM in a ten month period comprising timed sequences of more than 15 million status events in total. In the recorded period no attacks were registered, so we consider the monitored behavior as normal. We perform the evaluation on a weekly basis to account for seasonality effect: The data of an individual week is used for model learning (training) and the data of the respective subsequent week is used for evaluation (testing). To assess the anomaly detection effectiveness, data of normal *and* abnormal behavior is required. Note that, in our use case, data of monitored attacks on ATMs is in general not available for reasons of security and secrecy. We therefore intersperse artificial anomalies in the testing data by randomly choosing a certain proportion of sequences and multiplying a random subset of time intervals in each sequence by a given factor. [2]

[2] Part of our current research is to investigate different strategies to derive anomaly examples, which includes the generation of uniformly distributed outliers and the explicit specification of known attacks by domain experts.

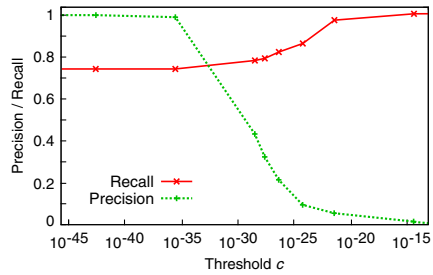

Figure 2. Anomaly detection effectiveness in terms of precision and recall over the anomaly threshold c. The x-axis is in log scale since the probability of traversing a particular path in the PDTTA can become considerably small.

We can only give a glimpse of the results here. Figure 2 shows the performance in terms of precision and recall using data with a proportion of 1% of artificial anomalies (note that attacks on ATMs are expected to be rare). The precision is the ratio between correctly detected anomalies and all detected anomalies. The recall is the ratio between detected anomalies and all anomalies. The threshold c allows for controlling the precision/recall tradeoff: For higher values of c more anomalies are detected (increasing recall), but also more normal sequences are falsely labeled as anomaly (decreasing precision). Altogether, the results show that, for small values of c, a precision close to 1 can be achieved while maintaining a reasonable recall (≈ 0.75), which indicates the practical applicability of our approach.

Although this paper focuses on ATM fraud detection, our approach can by applied to detect anomalies in other technical system, such as production plants, communication networks, or software systems.

Acknowledgements

This work was supported by Wincor Nixdorf International, and partly funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster *it's OWL*.

References

[1] M. Anderka, T. Klerx, S. Priesterjahn, and H. Kleine Büning, 'Automatic ATM fraud detection as a sequence-based anomaly detection problem', in *Proc. of the Intern. Conference on Pattern Recognition Applications and Methods (ICPRAM'14)*. SciTePress, (2014).

[2] R. C. Carrasco and José Oncina, 'Learning Stochastic Regular Grammars by Means of a State Merging Method', in *Proc. of the Intern. Colloquium (ICGI'94)*. Springer, (1994).

[3] C. Cassandras and S. Lafortune, *Introduction to discrete event systems*, Springer, 2008.

[4] A. Maier, A. Vodencarevic, O. Niggemann, R. Just, and M. Jäger, 'Anomaly detection in production plants using timed automata', in *Proc. of the Intern. Conference on Informatics in Control, Automation and Robotics (ICINCO'11)*. SciTePress, (2011).

[5] O. Niggemann, B. Stein, A. Vodencarevic, A. Maier, and H. Kleine Büning, 'Learning behavior models for hybrid timed systems', in *Proc. of the Intern. Conference on Artificial Intelligence (AAAI'12)*. AAAI, (2012).

[6] S. Priesterjahn, 'Robust and usable optical ATM surface protection', in *Proc. of the Optical Document Security Conference*. To appear, (2014).

[7] B. Reardon, K. Nance, and S. McCombie, 'Visualization of ATM usage patterns to detect counterfeit cards usage', in *Proc. of the Hawaii Intern. Conference on System Science (HICSS'12)*. IEEE, (2012).

[8] J. K. Suhr, Sungmin Eum, H. G. Jung, G. Li, G. Kim, and J. Kim, 'Recognizability assessment of facial images for automated teller machine applications', *Pattern Recognition*, **45**(5), (2012).

[9] S. Verwer, M. de Weerdt, and C. Witteveen, 'A likelihood-ratio test for identifying probabilistic deterministic real-time automata from positive data', in *Proc. of the Intern. Colloq. Conference on Grammatical Inference: Theoretical Results and Applications (ICGI'10)*. Springer, (2010).