Automatic ATM Fraud Detection as a Sequence-based Anomaly Detection Problem

Maik Anderka[1], Timo Klerx[1], Steffen Priesterjahn[2] and Hans Kleine Büning[1]
[1]*Department of Computer Science, University of Paderborn, Paderborn, Germany*
[2]*Wincor Nixdorf International GmbH, DE R&D ACT 53, Paderborn, Germany*

Keywords: ATM Fraud Detection, Sequence-based Anomaly Detection, Automatic Model Generation.

Abstract: Because of the direct access to cash and customer data, automated teller machines (ATMs) are the target of manifold attacks and fraud. To counter this problem, modern ATMs utilize specialized hardware security systems that are designed to detect particular types of attacks and manipulation. However, such systems do not provide any protection against future attacks that are unknown at design time. In this paper, we propose an approach that is able to detect known as well as unknown attacks on ATMs and that does not require additional security hardware. The idea is to utilize automatic model generation techniques to learn patterns of normal behavior from the status information of standard devices comprised in an ATM; a significant deviation from the learned behavior is an indicator of a fraud attempt. We cast the identification of ATM fraud as a sequence-based anomaly detection problem, and we describe three specific methods that implement our approach. An empirical evaluation using a real-world data set that has been recorded on a public ATM within a time period of nine weeks shows promising results and underlines the practical applicability of the proposed approach.

1 INTRODUCTION

Automated teller machines, ATMs, are subject to various attacks. The primary reason for this is the amount of cash inside the ATM safe (up to 500 000 EUR in a high volume ATM), but also the access to customer data that in turn gives access to cash. The total losses from ATM fraud during 2008 across Europe are estimated to 485.15 million EUR.[1] This makes clear that the prevention against attacks and fraud is a topic of highest importance, not only for financial institutes and bank customers but also for ATM manufacturers.

Modern ATMs comprise a variety of security technology. Internal devices either act as autonomous high security modules or are protected by encryption and the surrounding safe. Additionally, modern ATMs contain specialized security sensors ranging from shake sensors over gas sensors to cameras. The security state of an ATM is usually monitored by a software system to identify attacks and to react accordingly. The identification happens in a knowledge-based manner, i.e., based on a set of expert rules that need to be specified manually for individual attacks.

[1]According to the *European ATM Crime Report 2008* prepared by the European ATM Security Team, EAST (https://www.european-atm-security.eu).

The current security technology, however, is not able to identify novel types of attacks that are unknown at design time. Although respective security hardware and particular expert rules can be developed after a novel attack has become known, upgrading all ATMs in service is a time- and cost-intensive process. Moreover, such counteractions can only be initiated after the attack has happened, i.e., after loss or damage has already been incurred.

We propose a novel approach to detect ATM fraud that overcomes the mentioned limitations of current security solutions. Instead of modeling (known) attacks, we model the normal behavior of an ATM. This is based on the assumption that a significant deviation from the normal behavior is a strong indicator of an attack (which need not necessarily be known beforehand). Moreover, instead of manually specifying expert rules, we tackle the problem from a data-driven point of view by automatically generating a model of normal behavior based on the data stream of status information produced by the hardware devices and the software components inside an ATM. Compared to current security solutions, our approach has the following benefits:

1. It is able to detect both known attacks as well as novel attacks that are unknown at design time.

Anderka M., Klerx T., Priesterjahn S. and Kleine Büning H..
Automatic ATM Fraud Detection as a Sequence-based Anomaly Detection Problem.
DOI: 10.5220/0004922307590764
In *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2014), pages 759-764
ISBN: 978-989-758-018-5

2. It does not require particular (security) hardware, but uses the prevailing hardware equipment.
3. It can be applied on any machine, independent of the type, the equipment, and the manufacturer.
4. It minimizes human effort.

In this paper, we report on our current work on realizing the proposed approach using techniques from the research fields of anomaly detection (also known as outlier detection), pattern recognition, and automatic model generation. In an initial attempt to address the problem, we represent the continuous data stream that is produced inside an ATM as a discrete sequence of status events, and formulate the identification of anomalous behavior as the following *sequence-based anomaly detection* problem (Chandola et al., 2012): Given a set of normal training sequences, decide whether a test sequence is an anomaly with respect to the training sequences.

By formulating the problem in this way, we can benefit from the large body of prior research in the well-developed field of sequence-based anomaly detection. Following Aggarwal (Aggarwal, 2013) three basic principles can be distinguished how to decide whether a test sequence is an anomaly: 1. the test sequence rarely occurs in the training sample (frequency-based), 2. its distance to most training sequences is very large (distance-based), and 3. its probability of being generated by some probabilistic generative model is low (model-based). In order to demonstrate the practical applicability of our approach, we implemented three specific sequence-based anomaly detection methods, where each of which is based on one of the basic principles. The three methods are threshold-based sequence time delay embedding (t-stide), a k-nearest neighbor approach (k-NN), and a hidden Markov model (HMM).

We empirically evaluate the three methods using a real-world data set that has been recorded on a public ATM within a time period of nine weeks. Since a representative sample of anomalous behavior is in general not available, we intersperse randomly generated artificial anomalies in the test data set. Given the best parameter combination, two of the three methods achieve an anomaly detection effectiveness of more than 0.8 in terms of F-measure.

The remainder of this paper is organized as follows. Section 2 provides background information on the data that is collected inside an ATM. Section 3 gives a formal problem definition and describes the three sequence-based anomaly detection methods. Section 4 presents the empirical evaluation and discusses the results. Finally, Section 5 concludes this paper and gives an outlook on future work.

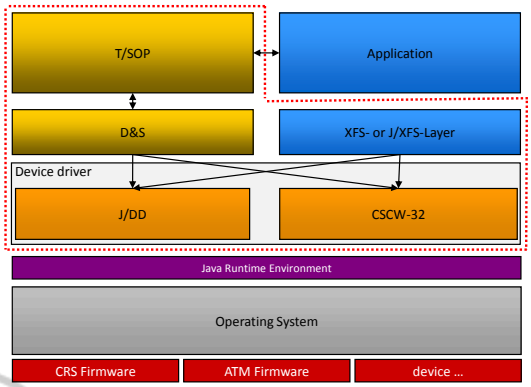

Figure 1: Software architecture of a Wincor Nixdorf ATM (modules ProBase/C and ProBase/J).

2 MONITORING ATM BEHAVIOR

Modern ATMs contain a variety of devices that produce a continuous stream of data. For example, every mechatronic system (e.g. for money and card transportation) is equipped with multi-sensory technologies that acquire real-time data for deriving current states, describing normal behavior, and realizing self-expression capabilities. Further examples include input devices (e.g. the PIN pad and soft keys or the touch screen) and monitoring electronics (e.g. anti-skimming units, demolition sensors, and cameras) that emit status events for process control and for health or security state assessment.

The data of all devices is usually processed and aggregated by a piece of software in the ATM PC. Figure 1 shows the architecture of the ATM software for a typical Wincor Nixdorf ATM.[2] The devices produce periodic and event-triggered messages, which are transferred from the devices' firmwares via their device drivers to the Diagnosis and Serviceability (D&S) module, among others. The D&S module can store all messages in a log file. Applications can obtain access to the log file for online processing during operation. In addition, the log file can be processed offline, e.g., on another machine (as it is the case for the experiments described in Section 4).

The messages that are comprised in the log file have the following structure:

```
<timestamp> <message ID> <payload>
```

The timestamp represents the moment when the D&S module received the message. The sending device as well as the reason why the message was sent is identi-

[2]This architecture is very similar for most ATM vendors due to the CEN/XFS standard that describes the common programming interface for ATM applications.

fied by the message ID. The payload contains device-specific data, such as sensor values. The log file comprehensively describes the ATM's real-time behavior in all monitored operation scenarios. Therefore, it is an appropriate source for (1) building a model of normal behavior and (2) monitoring the real-time behavior and trigger a security alert when detecting anomalies with respect to the model.

3 DETECTING ANOMALOUS BEHAVIOR

The idea is to identify attacks on ATMs by detecting anomalies in the monitored behavior. We start with a formal problem definition (Section 3.1). Afterwards, we describe the three sequence-based anomaly detection methods that we have implemented: frequency-based (Section 3.2), distance-based (Section 3.3), and model-based (Section 3.4).

3.1 Problem Statement

Data Representation. The data base is the log file provided by the ATM's D&S module (described in Section 2). For the purpose of this paper, we represent the data as a discrete sequence:

Definition 1 (Discrete Sequence of Status Events). Let the ordered list $S = e_1 e_2 \dots e_i \dots$ be a discrete sequence of status events, where each event e_i corresponds to a message in the log file. The ordering of the sequence is based on the messages' timestamps. An event e_i is modeled by a single categorical attribute, namely the message ID.

We use this data representation in favor of a more complex one to underline the robustness of the proposed approach. Our data representation solely relies on the chronological order of the events (or messages) and on the information comprised in the message IDs. A more complex representation could also incorporate the time intervals between subsequent events as well as the message payloads.

Anomaly Detection. The formulation of the problem depends on whether specific events are considered as anomalies, or whether (small) sequences of events are considered as anomalies. ATMs typically operate in a transaction-based manner. This leads to commonly recurring series of user interactions and internal processes (e.g., card insertion → PIN entry → amount selection → payout → card return), which result in respective sequences of status events in the data. Analogously, an attack would result in an unusual event sequence. We therefore consider *sequences* of events as anomalies (instead of specific events), which is in line with the concept of so-called *combination outliers* (Aggarwal, 2013). Thus, event sequences form the unit elements for the subsequent analyses.

The log file provided by the ATM's D&S module, however, contains no transaction-related information. As a result, it is not possible to retrieve the intrinsic sequences from the data. We therefore apply a sliding windowing technique, to derive (artificial) sequences as contiguous windows of length w. This is illustrated in the following example for $w \in \{3, 4, 5, 6\}$:

$w = 3$: $e_1e_2e_3,\ e_2e_3e_4,\ e_3e_4e_5,\ e_4e_5e_6,\ \dots$
$w = 4$: $e_1e_2e_3e_4,\ e_2e_3e_4e_5,\ e_3e_4e_5e_6,\ \dots$
$w = 5$: $e_1e_2e_3e_4e_5,\ e_2e_3e_4e_5e_6,\ \dots$
$w = 6$: $e_1e_2e_3e_4e_5e_6 \dots$

Since the true length of the sequences is unknown, we evaluate different values of w in the experiments described in Section 4. Applying sliding windowing is an established procedure in the context of sequence-based anomaly detection; see e.g., (Hofmeyr et al., 1998) and (Warrender et al., 1999).

Given the above, we define the identification of anomalous behavior as the following sequence-based anomaly detection problem:

Definition 2 (Sequence-based Anomaly Detection). Given a set $\mathcal{T}$ of normal training sequences, determine an anomaly score for a given test sequence S_q with respect to $\mathcal{T}$.

Without loss of generality, let the anomaly score be between 0 and 1, where 1 corresponds to normal and 0 to anomaly. It will be assumed that the training sequences and the test sequence are of the same length (w). This assumption does not limit the scope of our research because the anomaly score of a longer test sequence can be computed by combining the anomaly scores of its subsequences; examples of respective combination techniques are locality frame count (Warrender et al., 1999) and leaky bucket (Ghosh et al., 1999). Since the training set $\mathcal{T}$contains only normal sequences, the detection of anomalies is essentially a one-class problem (Tax, 2001). Note that even if samples of anomalous behavior were available, they could not be exploited to properly characterize the universe of all possible attacks. In consequence, the respective anomaly detection methods need to operate in a semi-supervised fashion (Chapelle et al., 2006).

3.2 Frequency-based (t-stide)

A well-known sequence-based anomaly detection method is the threshold-based sequence time delay

embedding (t-stide), which has been proposed by Warrender et al. (Warrender et al., 1999). In t-stide, the anomaly score for a test sequence S_q is equal to the relative frequency of S_q in the training set $\mathcal{T}$. The idea is that rare sequences are likely to be anomalies. A test sequence is determined as anomaly if its anomaly score is smaller than a threshold t. The t-stide method has been successfully applied to various application, a popular example is operating system intrusion detection based on sequences of system calls (Warrender et al., 1999; Cabrera et al., 2001).

3.3 Distance-based (k-NN)

Chandola et al. (Chandola et al., 2008) propose a k-nearest neighbor (k-NN) approach for sequence-based anomaly detection. The anomaly score for a test sequence S_q is equal to the inverse distance of S_q to its k$^{\text{th}}$ nearest neighbor in the training set $\mathcal{T}$. If its anomaly score is lower than a threshold t, the test sequence is considered as anomaly. With other words, S_q is an anomaly if more than k training sequences are at a distance of t or less from S_q. In spite of its simplicity, this approach has been shown to be quite effective, and it is even able to outperform a complex clustering-based technique (Chandola et al., 2008). As a distance measure, we use the inverse of the normalized length of the longest common subsequence, which has been shown promising for anomaly detection in sequence data (Budalakoti et al., 2006).

3.4 Model-based (HMM)

The hidden Markov model (HMM) (Rabiner, 1989) is a particular probabilistic generative model that is widely used for sequence-based anomaly detection (Zhang et al., 2003; Florez-Larrahondo et al., 2005). In a first step, a model is learned by estimating the HMM parameters based on the training set $\mathcal{T}$. For this purpose, we use the segmental k-Means algorithm (Juang and Rabiner, 1990). The anomaly score for a test sequence S_q is equal to the probability of S_q being generated by the model. A test sequence is determined as anomaly if its anomaly score is lower than a threshold t.

4 EMPIRICAL EVALUATION

The goal of the evaluation is to assess the anomaly detection effectiveness of the three methods described above. We therefore use a log file that has been recorded as described in Section 2 on a public ATM in the period between June 2011 and April 2012. The log file comprises about 15 million messages, resulting in 1.6 GB file size. In the recorded period no attacks were registered, so we consider the monitored behavior as normal.

In a preprocessing step, we split the log file into weekly chunks to account for seasonality, which is a well-known effect in time series data. Consider for example an ATM that is placed in a shopping mall: It is likely that the normal usage patterns on a Saturday, where the mall is very well attended, differ from the normal usage patterns on a Sunday, where the mall is closed. We therefore perform the evaluation on the basis of whole weeks (and not days for instance).

Experiment Design. Each experiment is performed on a triple of three subsequent weeks, which serve as training set, validation set, and test set respectively. The training set is used for frequency computation (t-stide), for distance computation (k-NN), and for model generation (HMM), respectively. The validation set is used for parameter tuning, i.e., finding an appropriate threshold t for all three methods and finding a k for k-NN. The anomaly detection effectiveness under the best parameter combination is determined on the test set. We perform the evaluation for varying sequence lengths w using the sliding windowing technique described in Section 3. The anomaly detection effectiveness is measured in terms of precision and recall. The precision is the ratio between correctly detected anomalies and all detected anomalies. The recall is the ratio between detected anomalies and all anomalies.

As motivated earlier, no data of anomalous behavior (attacks) is available. Thus the anomaly detection methods can be evaluated only with respect to their recall; whereas a recall of 1 can be achieved easily by classifying all test sequences as anomaly. In order to evaluate the methods with respect to their precision one needs a representative sample of anomalies in the test set. A way out of this dilemma is the generation of uniformly distributed outlier examples (Tax, 2001). Here, we apply a more basic approach: Based on a random distribution, a certain portion of sequences in the test set is modified by replacing at least one of their events by some other events. The modified test sequence serves as (artificial) anomaly.

The proportion of anomalous sequences in the test set determines the "difficulty level" (Chandola et al., 2008); it applies that the less the proportion, the more difficult their detection. We experimented with different proportions ranging from 1% to 50%. Here, we report results for 1% of anomalous sequences in the test set because attacks on ATMs are expected to be relatively rare in reality.

Table 1: Effectiveness in terms of F-measure for the three sequence-based anomaly detection methods t-stide, k-NN, and HMM for varying sequence lengths $w \in \{2, ..., 19\}$. Bold numbers indicate the row maximum.

Method	**Sequence length**, w																	
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
t-stide	0.76	**0.85**	0.84	0.79	0.72	0.67	0.60	0.56	0.50	0.47	0.42	0.38	0.35	0.32	0.29	0.27	0.24	0.22
k-NN	0.07	0.16	**0.21**	0.15	0.13	0.13	0.11	0.12	0.11	0.11	0.08	0.09	0.08	0.10	0.08	0.09	0.07	0.11
HMM	0.77	0.83	0.80	0.83	0.78	0.80	**0.84**	0.81	0.75	0.80	0.79	0.79	0.71	0.72	0.69	0.77	0.76	0.77

Results and Discussion. Table 1 shows the anomaly detection effectiveness for the three methods in terms of F-Measure, which is the harmonic mean of precision and recall. The table reports the F-measure values achieved on the test set using the respective parameter combination that have been found to perform best on the validation set. The parameters of each method are tuned individually for each value of w because this would also be the case in practice where w is assumed to be fixed but unknown (cf. Section 3).

According to Table 1, the effectiveness of all three methods varies depending on the sequence length w. The degree of variation, however, differs among the methods. The t-stide method is very sensitive to the sequence length, and its effectiveness deteriorates with increasing values of w. A possible explanation for this behavior is that a rarely occurring sequence that is labeled as anomaly for high values of w may be split into smaller frequently occurring subsequences that are labeled as normal for lower values of w. The other two methods perform nearly constant, and there is no discernible trend for increasing or decreasing values of w. This observation can be explained as follows: In case of the k-NN method, the normalized length of the longest common substring is used as distance measure, which is independent of the length of the input. In case of the HMM method, the same model is build for different values of w, and for changing sequence lengths only the threshold t has to be adjusted for probability computation.

All three methods have in common that the respective highest F-measure values are achieved at small sequence lengths, with w between 3 and 8. This gives some indication of the true length of the intrinsic sequences in the data, and it is consistent with the information we got from Wincor Nixdorf experts. Knowing this is helpful for future research, e.g., to develop dedicated anomaly detection methods and to perform an accurate evaluation by focusing on respective values of w.

Independent of the sequence length, the reported F-measure values achieved by the k-NN method are relatively low, ranging from 0.07 to 0.21. By contrast, t-stide and HMM achieve quite good F-measure values for certain sequence lengths, 0.85 for $w = 2$ and 0.84 for $w = 8$, respectively. Altogether, the HMM method performs slightly better than t-stide, which is inline with the findings of Warrender et al. (Warrender et al., 1999). We explain the poor performance of the k-NN method by the choice of the distance measure; although Budalakoti et al. (Budalakoti et al., 2006) report promising results, the normalized length of the longest common substring seems to be too restrictive for capturing the distance (or similarity) between sequences of status events.

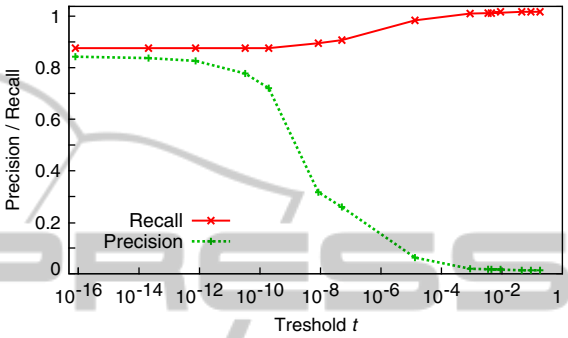

Figure 2: Precision and recall for the HMM method with $w = 8$ over the anomaly score threshold t.

Up to now, we have discussed the anomaly detection effectiveness in terms of F-measure. As already mentioned, F-measure equals to the harmonic mean of precision and recall. In a practical application, however, it is often desirable to tune either precision or recall. For example, if an alert should be triggered at the slightest sign of an anomaly so that no attack is missed (high recall) or if an alert should be triggered only when the anomaly detection confidence is high so that the number of false alerts is minimized (high precision). The tradeoff between precision and recall can be controlled by the threshold t. Figure 2 exemplifies this for the HMM method and for $w = 8$. Note that the probability of following a certain path in a HMM can become considerably small, and hence, the x-axis in Figure 2 is in log scale to account for very small values of t. The highest F-measure value (0.84), which is also reported in Table 1, corresponds to a threshold of $t = 10^{-16}$. For higher values of t the recall increases, i.e., more anomalies are detected; but at the same time, the precision decreases, which results

in an increasing number of false positives (normal sequences that are erroneously detected as anomalies).

5 CONCLUSIONS AND RESEARCH OUTLOOK

Our ATM fraud detection approach is based on the assumption that a significant deviation from the normal behavior is a strong indicator of an attack. To the best of our knowledge, we are the first who utilize the data stream produced inside an ATM to automatically generate a model of normal behavior, which is then used to detect anomalies (or attacks respectively). The formulation of this approach as a sequence-based anomaly detection problem and the empirical evaluation of three respective methods using a real-world data set show its practical applicability.

This paper constitutes a proof of concept. Our current research targets the further elaboration of the problem formulation based on the lessons learned and the investigation of tailored anomaly detection methods. In particular, this includes the following aspects:

- Incorporating the time intervals between subsequent events by representing the data stream as a continuous sequence (or time series).
- Exploiting the information comprised in the message payload by applying a multidimensional event model (Budalakoti et al., 2006).
- Analyzing training and test sequences of different length by combining the anomaly scores of subsequences (Ghosh et al., 1999; Warrender et al., 1999).
- Investigating unsupervised anomaly detection approaches, i.e., training and test sequences are not differentiated (Leung and Leckie, 2005; Zhang and Zulkernine, 2006).

Regarding the evaluation, we plan to investigate different strategies to derive anomaly examples, which includes the generation of uniformly distributed outliers (Tax, 2001) as well as the explicit specification of known attacks by domain experts.

ACKNOWLEDGEMENTS

This work was partly funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster “Intelligent Technical Systems OstWestfalenLippe” (it’s OWL).

REFERENCES

Aggarwal, C. (2013). *Outlier Analysis*. Springer.

Budalakoti, S., Srivastava, A., Akella, R., and Turkov, E. (2006). Anomaly detection in large sets of high-dimensional symbol sequences. Technical Report TM-2006-214553, NASA Ames Research Center.

Cabrera, J., Lewis, L., and Mehra, R. (2001). Detection and classification of intrusions and faults using sequences of system calls. *ACM SIGMOD Record*, 30(4).

Chandola, V., Banerjee, A., and Kumar, V. (2012). Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5).

Chandola, V., Mithal, V., and Kumar, V. (2008). Comparative evaluation of anomaly detection techniques for sequence data. In *Proceedings of the 8th IEEE Conference on Data Mining (ICDM’08)*. IEEE.

Chapelle, O., Schölkopf, B., and Zien, A., editors (2006). *Semi-Supervised Learning*. MIT Press.

Florez-Larrahondo, G., Bridges, S., and Vaughn, R. (2005). Efficient modeling of discrete events for anomaly detection using hidden Markov models. In *Proceedings of the 8th Conference on Information Security (ISC’05)*. Springer.

Ghosh, A., Schwartzbard, A., and Schatz, M. (1999). Learning program behavior profiles for intrusion detection. In *Proceedings of the USENIX Workshop on Intrusion Detection and Network Monitoring (ID’99)*. USENIX Association.

Hofmeyr, S., Forrest, S., and Somayaji, A. (1998). Intrusion detection using sequences of system calls. *Journal of Computer Security*, 6(3).

Juang, B. and Rabiner, L. (1990). The segmental K-means algorithm for estimating parameters of hidden Markov models. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(9).

Leung, K. and Leckie, C. (2005). Unsupervised anomaly detection in network intrusion detection using clusters. In *Proceedings of the 28th Australasian Conference on Computer Science (ACSC’05)*. Australian Computer Society, Inc.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2).

Tax, D. (2001). *One-class Classification: Concept-learning in the Absence of Counter-examples*. Ph.d. thesis, Delft University of Technology.

Warrender, C., Forrest, S., and Pearlmutter, B. (1999). Detecting intrusions using system calls: Alternative data models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy (PS’99)*. IEEE.

Zhang, J. and Zulkernine, M. (2006). Anomaly based network intrusion detection with unsupervised outlier detection. In *IEEE International Conference on Communications (ICC’06)*. IEEE.

Zhang, X., Fan, P., and Zhu, Z. (2003). A new anomaly detection method based on hierarchical HMM. In *Proceedings of the 4th Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT’03)*. IEEE.