Open ERP System Data For Occupational Fraud Detection

Julian Tritscher[1], Fabian Gwinner[2], Daniel Schlör[1], Anna Krause[1], and Andreas Hotho[1]

[1] University of Würzburg, Am Hubland, 97074 Würzburg, Germany
{tritscher, schloer, anna.kraus, hotho}@informatik.uni-wuerzburg.de
[2] fabian.gwinner@uni-wuerzburg.de

Abstract. Recent estimates report that companies lose 5% of their revenue to occupational fraud. Since most medium-sized and large companies employ Enterprise Resource Planning (ERP) systems to track vast amounts of information regarding their business process, researchers have in the past shown interest in automatically detecting fraud through ERP system data. Current research in this area, however, is hindered by the fact that ERP system data is not publicly available for the development and comparison of fraud detection methods. We therefore endeavour to generate public ERP system data that includes both normal business operation and fraud. We propose a strategy for generating ERP system data through a serious game, model a variety of fraud scenarios in cooperation with auditing experts, and generate data from a simulated make-to-stock production company with multiple research participants. We aggregate the generated data into ready to used datasets for fraud detection in ERP systems, and supply both the raw and aggregated data to the general public to allow for open development and comparison of fraud detection approaches on ERP system data.

Keywords: Data generation · Fraud detection · SAP.

1 Introduction

The association of certified fraud examiners defines occupational fraud as abusing one's occupation through the deliberate abuse of an employing organization's assets, and estimates that currently companies lose 5% of their revenue to this type of fraud [1]. To reduce the loss of revenue to occupational fraud, researchers have in the past suggested to use the data contained within ERP systems to detect fraudulent activity [18, 17, 12]. ERP systems are a core component for managing the flows of cash, materials, production and other resources within companies. They represent a large market with $37,679.26$ Mio. USD worldwide revenue in 2020, and support most medium-sized and large companies in their daily work [6, 20].

In spite of ERP systems providing many different views on an organization's workflow that could potentially aid the detection of fraudulent activity, research in this area is currently hindered by the fact that ERP system data, in general, is not available to the public. This proves problematic when attempting to reproduce published results, and compare performance of existing fraud detection approaches.

To address this issue, we propose an approach for generating synthetic ERP system data that extends previous research, generating data containing both normal operation and fraudulent activities, and making the resulting data publicly available.

Previous works on ERP system fraud detection may be divided into approaches that rely on entirely private data and frauds, private data with synthetically injected frauds, or entirely synthetic data and frauds. While there have been works that use real ERP system data to develop and evaluate fraud detection systems [18, 17, 12], details about the data and the data itself are kept under wraps to avoid revealing company trade secrets and privacy information. For scenarios where real frauds are not available, Islam et al. [8] generate synthetic fraud cases within private ERP system data through randomly creating changes to normal transactions while limiting changes and timings with given intervals. While this generates anomalous transactions through not yet observed peaks in single entries, the generated anomalies have no inherent meaning or interpretation with respect to real-life occurrences or fraud.

As an alternative that uses no private data, business researchers have in the past moved to developing data generators that are capable of generating both normal operation and frauds: Yannikos et al. [22] introduce 3LSPG, a generator that produces synthetic ERP data through a probabilistic approach using discrete time Markov chains. While the resulting data can mimic the transactions taken from an ERP system, the data's quality and realism are strongly dependent on the expert knowledge put into the simulation. With no data, code, and chosen simulation parameters available, modeling realistic ERP system data through this approach is challenging. Similarly, game based approaches may be used to model business processes with player interaction [21]. While this is a promising aspect for data generation, expert knowledge is required to ensure that the complex behavior of real ERP systems are mimicked in the resulting data.

Baader et al. [4] partially alleviate the need of expert knowledge by modeling normal and fraudulent behavior directly within an ERP system, thus being able to automate parts of the generation process that would be carried out by the ERP system in a real world scenario. Remaining business decisions that are not taken over by the ERP system such as procurement quantities or sales prices are simulated through random distributions. Baader and Krcmar [3] extend the approach in additional work, where, instead of generating fraud cases through random distributions, they obtain fraud scenarios through user participation with the white-collar hacking contest [15], a serious game developed to teach players the abuse of an ERP system and the detection thereof. While the resulting frauds may model realistic scenarios, they are modeled into an existing database in post, potentially causing unwanted divergence between normal and fraudulent data characteristics. Further, in contrast to other research areas that utilise synthetic data such as intrusion detection [14], all published synthetic ERP data generation methods to our knowledge do not publish their code and data, making reproducible and incremental research in ERP fraud detection difficult due to missing comparability.

In this paper, we address these problems by extending the work of Baader and Krcmar [3] in multiple ways: We first extend the requirements for synthetic ERP data layed out by Baader et al. [4] to include further requirements of ERP fraud detection approaches [7]. We secondly propose to use an established serious game [9] to simulate not only fraudulent scenarios but also normal operation of a make-to-stock production company through user interaction in a real ERP system, generating both normal and fraudulent behavior simultaneously. Additionally, this allows us to extend the business processes investigated by previous data generators from the purchase-to-pay (P2P) process to modeling normal and fraudulent activity in the well established order-to-cash (O2C) process as well. Based on our extended requirements, we then design multiple fraud scenarios in cooperation with auditing experts. We conduct multiple runs of our proposed data generation scheme and produce ERP system data of multiple fiscal years of operation, extracting raw data from the ERP system. The resulting data contains many multi-relational tables that offer different views on the recorded company's business process. Since many fraud detection approaches require single tables to operate, we additionally create ready to use datasets from a subset of our multi-relational data that can be directly used for measuring and comparing the performance of fraud detection systems. We further extend these datasets by providing expert-created annotations for fraud cases that highlight the problematic entries of individual frauds for use in debugging and assessing performance of algorithms that focus on the detection of anomalous entries specifically.

In summary, our main contributions are as follows:

- We propose a strategy for data generation that simulates normal behavior and fraud jointly through user interaction within a real ERP system and is capable of modeling frauds and normal behavior in the P2P and O2C business processes.
- We conduct multiple simulation runs and construct ready to use datasets with detailed fraud annotations that allow for direct application and comparison of ERP fraud detection approaches.
- Finally, we provide both raw data and ready to use datasets to the general public to allow for open comparability of ERP fraud detection systems.[3]

The remaining paper is structured as follows: Section 2 outlines the requirements for data generation, introduces our data generator, showcases the modelled business scenario and chosen fraud cases, and details the data generation process. Section 3 presents an analysis on the collected data, while Section 4 aggregates parts of the data into ready to use datasets for direct use in fraud detection applications. Section 5 concludes the paper.

[3] Datasets are available under https://professor-x.de/erp-fraud-data.

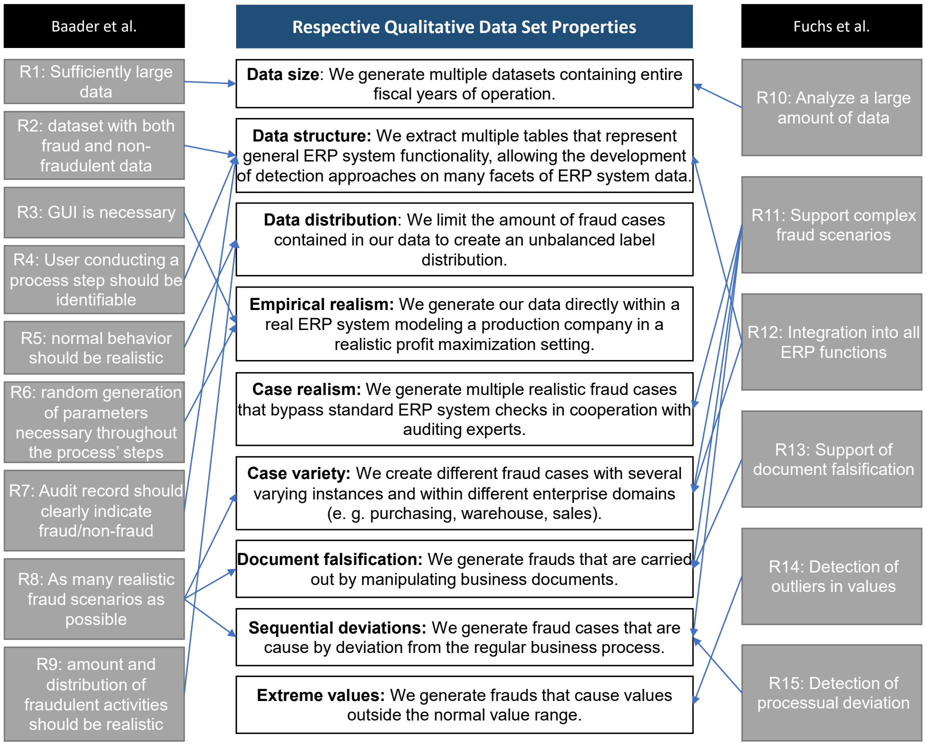
Baader et al.
Respective Qualitative Data Set Properties
Fuchs et al.
R1: Sufficiently large data
R2: dataset with both fraud and non-fraudulent data
R3: GUI is necessary
R4: User conducting a process step should be identifiable
R5: normal behavior should be realistic
R6: random generation of parameters necessary throughout the process' steps
R7: Audit record should clearly indicate fraud/non-fraud
R8: As many realistic fraud scenarios as possible
R9: amount and distribution of fraudulent activities should be realistic
Data size: We generate multiple datasets containing entire fiscal years of operation.
Data structure: We extract multiple tables that represent general ERP system functionality, allowing the development of detection approaches on many facets of ERP system data.
Data distribution: We limit the amount of fraud cases contained in our data to create an unbalanced label distribution.
Empirical realism: We generate our data directly within a real ERP system modeling a production company in a realistic profit maximization setting.
Case realism: We generate multiple realistic fraud cases that bypass standard ERP system checks in cooperation with auditing experts.
Case variety: We create different fraud cases with several varying instances and within different enterprise domains (e. g. purchasing, warehouse, sales).
Document falsification: We generate frauds that are carried out by manipulating business documents.
Sequential deviations: We generate fraud cases that are cause by deviation from the regular business process.
Extreme values: We generate frauds that cause values outside the normal value range.
R10: Analyze a large amount of data
R11: Support complex fraud scenarios
R12: Integration into all ERP functions
R13: Support of document falsification
R14: Detection of outliers in values
R15: Detection of processual deviation

Fig. 1: Dataset properties derived from literature requirements.

2 Data Collection

2.1 Data Requirements

To create high-quality ERP system data and increase rigor, we develop requirements for our data based on prior work. We follow Baader et al. [4], who find several requirements for both their developed data generator and the resulting data. We also draw additional requirements from previously conducted design science research that aims to develop a fraud detection system in the ERP domain. Here, Fuchs et al. [7] aggregate requirements for detection systems that are able to highlight fraud in ERP systems. Some of their requirements describe design decisions that need to be respected during implementation of the fraud detection approach and are unaffected by the studied data (e.g. requiring adaptable or intelligent logic). Other requirements, however, describe scenarios in which fraud detection approaches should yield satisfying performance (e.g. detection of outliers in values). We argue that data should be created such that the performance of fraud detection approaches can be validated in these scenarios, and therefore identify these requirements as directly relevant for our data generation process.

Figure 1 gives an overview of the requirements in the preliminary work of Baader et al. [4] as well as the requirements we identify as relevant to our data generation process from Fuchs et al. [7]. We additionally note the resulting measures we take in our proposed data generation scheme to satisfy these requirements.

2.2 Data Generation through ERPsim

Similar to Baader and Krcmar [3] that use an existing serious game [15] to generate fraudulent ERP transactions through user interaction, we take a game-based data generation approach to meet our formal data generation requirements and employ a serious game, ERPsim [9], to record ERP system data. Within the ERPsim serious game, participants take control of a make-to-stock production company through the ability to plan overall sales, create purchase orders for raw materials, plan production of products, produce and deliver sales orders to fictitious customers, manage the accounting, and optionally take loans and manage debts. In our scenario, the ERP system is used for make-to-stock production of four products based on a market analysis and forecast. After production, products are stored in a warehouse and sold to customers. To simulate an in-game year for a single company, the game may be played with up to five players per company in the roles of material planner, production controller, sales manager, financial planner, and market analyst.

We choose ERPsim, as it allows for generating both normal operation and fraudulent activities through user interaction. Next to the added complexity that may be introduced to the data through multiple participants operating the company simultaneously, ERPsim also offers a realistic profit maximization scenario through a simulated market. Where business decisions such as deciding on purchase quantities are modeled through random distributions in prior data generators [4], ERPsim motivates participants to make economically sensible business decisions. The game is also conducted directly within an SAP S/4 HANA ERP system. While real life processes such as the delivery of raw materials and the production of goods are simulated, the resulting documents and transactions are recorded within the ERP system through the standardized processes that are also employed in real companies. Unlike previous work, this allows fraud scenarios to be committed directly in the ERP system interface during operation, rather than requiring anomalies to be synthetically injected into historical data of normal operation.

2.3 Modeled Business Scenario

As determined through our requirements analysis in Section 2.1, data for ERP fraud detection requires a large variety of realistic fraud cases. To create realistic fraud cases that translate well to different companies, we focus on creating frauds within two standardized business processes that are simulated within our data generator.

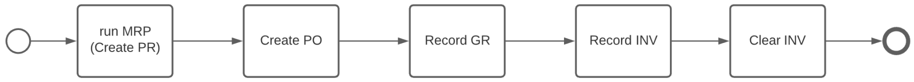

Fig. 2: Purchase-to-pay (P2P) process in the ERPsim simulation.

First, we select the widely used purchase-to-pay (P2P) process that has been the focus of previous data generation approaches [22, 4, 3]. In the P2P scenario, illustrated in Figure 2, a demand is created by the user's forecast. By performing the Material Requirement Planning (MRP) run, the user creates a purchase requisition (PR) for each demand. A buying agent then converts each PR into a purchase order (PO). As in the real world, saving a PO starts transferring the PO to the given supplier, where it gets converted into a customer order. While these steps are usually implemented via electronic data interchange between two ERP systems, in ERPsim the simulation middle-ware receives the PO and virtually ships the ordered goods after a random time within a defined time-frame. After this, the incoming goods need to be received at the production plant with a goods receipt (GR), and paid for by recording and clearing the invoice (INV).

As second business process, we select the well established order-to-cash (O2C) process that allows for modelling frauds in the sales department and has been suggested as future work for simulation by Baader et al. [4]. The O2C process is usually comprised of the activities from the customer ordering products to the payment of the order. As prices are an important component in our modeled fraud cases, we add the activities to determine sales prices into our O2C process. The resulting O2C process is illustrated in Figure 3 and consists of the market analysis and price calculation for determining the overall prices of sales products. Afterwards, customer or market specific discounts can be determined. Based on the resulting sales prices, a customer demand may be generated in the market and an order is generated. The orders then can be viewed, altered and confirmed. Confirming the order then triggers an activity for creating a delivery that ships the goods to the customer and generates an invoice. After a randomized time the customer pays the invoice and the accountant clears the open invoice, ending the process.

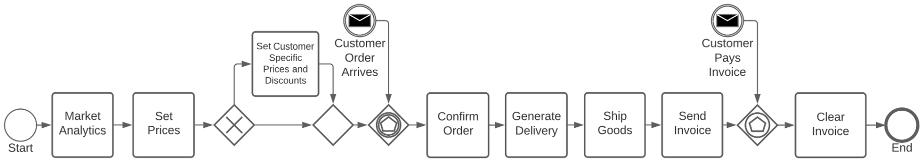

Fig. 3: Order-to-cash (O2C) process in the ERPsim simulation.

2.4 Modeled Fraud Cases

With the standardized business processes selected and introduced, we now turn to modeling fraudulent activity within these processes. Next to the requirement of a large variety of frauds, our analysis of Fuchs et al. [7] yields specific types of deviation that need to be included in the data. To satisfy the data requirements regarding fraud cases, we first analyse the fraud scenarios conducted by Baader et al. [4]. While analysing their modeled fraud scenarios, we found multiple cases are not possible in our realistic environment, as our ERP system is equipped with realistic control and audit mechanisms (e.g. due diligence, user authorisations, accounting checks), that are integrated in most ERP systems and are employed in many companies due to best practices or legal regulations [13, 11, 10, 2]. Since these control mechanisms prevent some cases of undesirable user behaviour, many fraud cases from Baader et al. [4] such as double payment, conto pro diverse transaction fraud, or non-purchase payment, would be blocked by the ERP system and would at most yield unsuccessful fraud attempts. Some further fraud cases of Baader et al. [4] rely on abusing quantity contracts for suppliers or value contracts for customers. As in our scenario purchase prices are driven by a simulated market where prices for raw materials change frequently, our use case does not provide any framework for contracts. Therefore, frauds involving contracts could not be simulated. The remaining fraud cases ”false invoice fraud” (in our scenario Larceny 1 and 2) and ”misappropriation fraud” (Larceny 4) are part of our modeled fraud cases.

After the preliminary selection of three fraud cases from Baader et al. [4], we select nine additional fraud scenarios to match our identified requirements. Here we select appropriate fraud scenarios from the ACFE’s report to the nations [1] in cooperation with business auditing experts. Since, in contrast to Baader et al. [4], our data generation approach is also capable of modeling the entire O2C business process, we additionally take care to include fraud scenarios that are conducted within the O2C process.

In total, we obtain twelve fraud cases that represent a broad spectrum of fraud on data level. Eight of the selected twelve fraud cases are part of the P2P process. In Figure 4 we visualize the fraudulent deviations (red activities) that

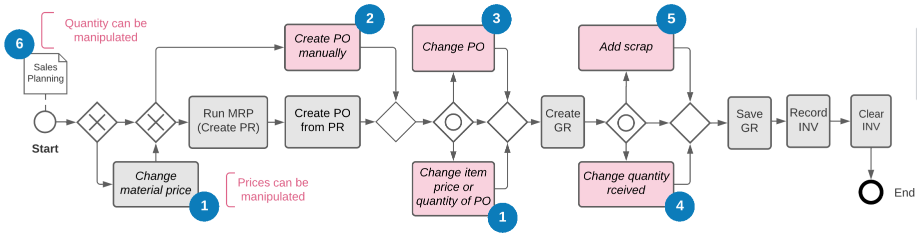

Fig. 4: P2P process steps with potential fraudulent deviations.

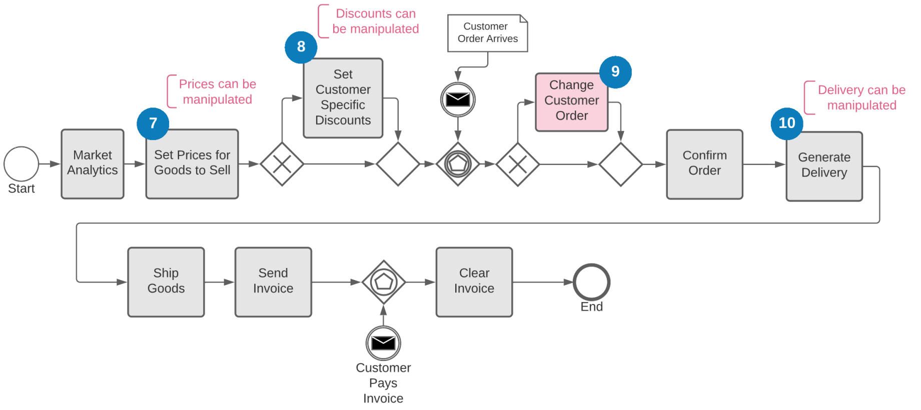

Fig. 5: O2C process steps with potential fraudulent deviations.

may be used by these fraud cases to deviate from the normal business process (grey activities).

The remaining four fraud cases are part of the O2C process, where we visualize the fraudulent deviations in the same way in Figure 5.

An in-depth description of all frauds and their used deviations from the normal process are detailed in Table 1. To ease readability, we categorize our different fraud scenarios as follows. We denote frauds as *invoice kickback* frauds that attempt to falsify invoice documents in order to create an advantage for an accomplice supplier that may be shared with the fraudster. Frauds that focus on the theft of materials from the company are noted as *larceny* frauds. *Corporate injury* frauds describe frauds that do not yield monetary benefit for the fraudster, but instead aim to cause financial harm the company. Finally, *selling kickback* frauds represent fraudsters manipulating sales conditions for accomplice customers to gain potentially shared financial benefits.

2.5 Data Generation

Using our proposed data generation process and the collected frauds, we conduct multiple runs of the ERPsim game to generate ERP system data, with each run generating data of one fiscal year of operation. Runs are played by five research participants with an information systems background. Participants are instructed on the business process specifics of the company modeled within ERPsim and adopt the roles described in Section 2.2.

To model the proposed frauds during our data generation process, we assign one participant the role of fraudster who may introduce fraudulent activities throughout the data generation process. The fraudster is instructed by an auditing expert on how to conduct our chosen fraud cases within the ERP system interface. We further identified in Section 2.1 that the number and distribution

Table 1: Overview of chosen fraud scenarios with numbered deviations visualized in Figures 4 and 5.

Fraud	Description
Invoice Kickback 1 (P2P)	Item purchase prices are changed while or before creating a PO (1) for an existing PR, to get a kickback from the supplier later. Results are higher unit prices, causing an anomaly where the combination of quantities and amounts diverge from the normal data distribution.
Invoice Kickback 2 (P2P)	Instead of changing an existing PR as in Invoice Kickback 1, an old PO is copied manually (2) with higher prices, resulting in manually created transactions.
Larceny 1 (P2P)	Quantities of purchased items are changed in an existing PO (3). While recording the GR, the expected item quantity is changed back (4) to leave no leftover in stock. A rule-based ERP check found goods are missing and internally blocked this transaction, making this an unsuccessful fraud attempt.
Larceny 2 (P2P)	Similar to Larceny 1, but here the PO was released manually (2) and changed afterwards (1), and the GR was booked regularly. The system's rule-based approach was unsuccessful, leading to an increase in inventory without the items' physical presence. Although this could be detected during stocktaking, damage would have already been caused to the company. Structurally, this fraud causes anomalously missing values due to manual PO creation.
Larceny 3 (P2P)	In this case, goods are purchased regularly and a partial amount of waste or scrap is booked (5) through the quality inspection in the goods receipt activity, to hide the theft of goods. This case may also be applied in warehousing or production.
Larceny 4 (P2P)	Here, goods that are usually not needed for production or organizational processes are purchased through a manual PO (2). While many companies use a four eyes principle (two-man rule), we assume a collusion of purchaser and supervisor.
Larceny 5 (P2P)	In this fraud, products were ordered regularly, but the delivery address was changed for the PO (3) to a private address.
Larceny 6 (O2C)	Similar to larceny 5, but in the O2C process. The delivery address in the master data of a customer in the order was changed (9), so that the delivery of a corresponding customer order was delivered to the wrong address. The address was changed back afterwards.
Corporate Injury 1 (P2P)	This fraud represents extensively large purchases by changing the Sales Planning (6), leading to company damages through high spending and potential waste and overstocking of warehouses. Structurally, this fraud results in anomalous extreme values in POs.
Corporate Injury 2 (O2C)	Here, the employee committing fraud drastically lowered sales prices (7) to damage the company.
Selling Kickback 1 (O2C)	This fraud case was conducted by manipulating sales conditions (8) for specific customers, which allowed the customers to purchase products with lowered order prices via discounts.
Selling Kickback 2 (O2C)	Similar to Selling Kickback 1, the order prices are manipulated. In this case, beneficial sales conditions were given to a specific customer in the sales order document itself (9).

of committed frauds should be realistic. Schreyer et al. [17] argue that real audit scenarios have highly unbalanced class distribution between very few anomalous and vast amounts of regular entries. Judging the real number of occupational fraud cases that are expected to lie within a company's data, however, is challenging, since the number of employees engaging in fraud is unknown and even in detected frauds the large majority of cases contains active attempts to hide the fraudulent activity [1]. To limit the amount of frauds included within our data and obtain a heavily unbalanced class distribution, we therefore limit our fraudster to conducting two fraud cases per simulated month of operation.

In this setting, we conduct multiple data generation runs in the SAPs R/3 on HANA ERP system together with ERPsim R11.2 with a group of 5 research participants. We let the group play the game twice, obtaining a run of exclusively normal operation (normal 1) as well as a run that has fraudulent activities incorporated next to normal business processes (fraud 1). To obtain differing company characteristics such as varying business strategies and user behavior, we additionally select a second group of participants to generate data. Our second group of participants generate one run of normal operation (normal 2) and two individual runs containing different fraud cases (fraud 2, 3), resulting in 3 datasets, each simulating one financial year.

To further increase the complexity of our generated data, we extend the normal business procedure of ERPsim by modeling specific events with our second participant group. As some of our modeled fraud cases involve process steps that are usually not part of the ERPsim game, such as booking of scrap or giving customer discounts, we add these behaviors as additional activities, build them as repetitive tasks into the process and track them similarly to the conducted fraud cases. For scrap we add regular manual scrap bookings of received goods, to simulate problems in the delivery or warehouse operations, while limiting the bookings to small amounts of broken materials. To simulate a regularity in customer discounts we add promotional campaigns, that give customer groups a small discount for their purchases within a given time frame, through setting the appropriate discounts for the distribution channels within the ERP system.

3 Analysis of Generated Data

Within our proposed data generation scheme, we conducted five separate runs of the ERPsim serious game with both exclusively normal and partially fraudulent business operation. In Table 2 we report some financial characteristics of the simulated company over the conducted runs, with each run lasting one fiscal year. When comparing purchasing costs and turnover costs, we observe that all companies were capable of achieving a considerable added value through the procurement, production, and sales strategies employed by the data generation participants. Our first participant group was capable of achieving higher added values within their runs normal 1 and fraud 1, which can be attributed to a largely differing business strategy compared to our second group. While our second group specifically targeted large resellers in their runs (normal 2, fraud

Table 2: Data characteristics of the recorded runs of ERPsim **excluding fraud*

Dataset	Turnover volume (qty)	Turnover costs (€)	Purchasing volume (kg)	Purchasing costs*(€)	No. of customers	No. of sales	No. of purchases
normal 1	3,382,416	19,482,620,84	2,764,010	6,088,646.20	194	9,329	552
fraud 1	2,655,058	13,740,944.11	6,598,400	3,654,801.50	194	6,605	222
normal 2	2,925,000	12,901,063.42	2,915,200	6,424,043.80	71	6,243	257
fraud 2	3,026,318	14,154,144.48	3,667,800	7,090,319.80	71	6,793	287
fraud 3	3,244,421	14,820,360.15	4,727,500	7,133,712.35	71	7,113	280

2, fraud 3) through producing exclusively large product sizes and was capable of serving the market of the 71 large resellers within ERPsim, our first group produces additional small product sizes that are sold also to smaller retails which left them with a higher number of customers. This also explains the high turnover volume in comparison to the purchasing volume of run normal 1, as turnover volume here also included smaller packaging. Overall, we find that the different participant groups indeed generated data with varying characteristics through the choice of different business strategies.

Beyond the economic characteristics of the simulated companies, we report the number of fraud cases and added additional events within the generated data in Table 3. As described in Section 2.5, the total amount of fraud cases was kept low to retain an unbalanced data distribution. All fraud runs contain fraud scenarios within the P2P and O2C business process. Our first participant group focused on several scenarios of larceny fraud. Our second group, on the other hand, modeled multiple fraud scenarios that hide their activities through scrap bookings, and frauds that achieve profit through fraudulent discounts. In their runs (normal 2, fraud 2, fraud 3) we also simulated regular scrap bookings and sales events as described in Section 2.5.

Overall, both groups modeled a variety of complex fraud scenarios within their generated data, with different distributions of fraud cases.

Table 3: Fraud cases and events occurring within the recorded runs of ERPsim.

Dataset	Invoice Kickback 1	Invoice Kickback 2	Larceny 1	Larceny 2	Larceny 3	Larceny 4	Larceny 5	Larceny 6	Corporate Injury 1	Corporate Injury 2	Selling Kickback 1	Selling Kickback 2	Scrap	Sale
normal 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fraud 1	1	0	1	1	0	0	4	1	1	1	0	0	0	0
normal 2	0	0	0	0	0	0	0	0	0	0	0	0	3	2
fraud 2	2	2	1	2	2	2	1	0	1	1	1	2	2	1
fraud 3	2	1	2	1	2	2	1	0	1	1	1	2	1	1

4 Dataset Construction

In the previous chapters, we discussed the generation and analysis of data from our synthetic data generation approach. We extract this data from the ERP system into many different tables to allow researchers to design fraud detection systems that integrate into all ERP functions as discussed in Section 2.1. Many fraud detection approaches, however, (especially ones utilizing machine learning [16]) are not capable of integrating data from multiple tables and instead require a single joint dataset to operate. Since ERP system data is inherently multi-relational due to it tracking many different views on the underlying business activities, exactly reproducing specific joins may prove challenging. To allow for easy reproducibility and comparisons of ERP fraud detection approaches, we therefore focus in this section on providing single joint datasets that may be directly used for detecting fraudulent transactions.

For our joint datasets, we focus on the financial accounting data of the P2P business process in the SAP database tables RBKP, RSEG, BKPF and BSEG. We choose the financial accounting information to detect fraudulent behavior since this information is usually used by auditors in real-world auditing procedures [19]. For each run, we combine the respective tables to one dataset, obtaining a single table with financial accounting data featuring invoice, credit, general ledger (G/L) account posting and material movement transactions. We remove duplicate columns, empty columns, columns containing always the same value, and identifier columns, since they offer no usable information to many automatic fraud detection algorithms. For the remaining columns, we provide information on whether the columns contain numerical or categorical data.

As a single fraud case may create multiple transactions within the ERP system, we mark all invoice, credit, G/L account posting and material movement transactions as fraudulent depending on whether they are part of one of our fraud scenarios from Section 2.4, thus obtaining fraud labels for all transactions in our data. Further, since in this join of our data we focus on financial accounting information that does not contain sales information such as final product sales prices or delivery information such as delivery addresses, we exclude the fraud cases belonging to the O2C process and the fraud cases based on delivery details since they are structurally indistinguishable from normal activity within the joint accounting tables. The final datasets with the resulting distribution of fraudulent and total transactions are listed in Table 4.

While analysing our datasets, we found that many fraud cases are only detectable due to few anomalous entries in otherwise sparse and largely regular table data. This observation is also used in several well known approaches such as Benford's law or Extreme Value Analysis, that detect anomalies by the frequency of numbers or the probability of higher values in a distribution of a single feature [5]. With fraud detection approaches specifically targeting few data entries of entire transactions, a fine granular labeling process that makes single fraud cases traceable on a feature level allows for gaining insights into fraud detection performance. We therefore conduct an additional labeling process and provide additional expert feature-level annotations for each fraudulent transac-

Table 4: Transaction types and fraud cases of normal and partially fraudulent data.

Dataset	Invoice	Credit	G/L Acc. Posting	Material Receipt	Material Withdrawl	Total Transact.	Invoice Kickback 1	Invoice Kickback 2	Larceny 1	Larceny 2	Larceny 3	Larceny 4	Corporate Injury	Frauds in Total
normal 1	7511	7050	28845	769	10495	54677	0	0	0	0	0	0	0	0
fraud 1	5212	5181	20828	447	7758	39430	4	0	2	4	0	0	14	24
normal 2	3231	3129	18271	425	7280	32337	0	0	0	0	0	0	0	0
fraud 2	4154	4007	20186	469	7960	36778	6	18	2	4	10	6	4	50
fraud 3	4080	3841	20678	464	8344	37407	24	6	8	10	26	4	8	86

tion: Within the fraudulent transactions, we identify and highlight all features that hint at the underlying fraud case due to anomalous column entries. The resulting annotations are supplied alongside the joint datasets and may be used for in-depth prototyping and evaluation of fraud detection approaches.

5 Conclusion

With companies keeping a lock on ERP system data due to privacy and trade secrets concerns, researchers in the area of occupational fraud detection have in the past turned to synthetic data generation for validating their work. Previous works in this area however did not provide data to the public, limiting open and reproducible research on detecting fraud in ERP systems.

In this paper, we proposed a strategy for generating ERP system data through an existing serious game, modeled a variety of occupational fraud cases within the serious game's real ERP system interface, and recorded multiple runs of both normal and fraudulent operation of a simulated make-to-stock production company. We gave an overview of the resulting data, and provided additional joint datasets that can be directly used for applying and comparing fraud detection approaches. Our obtained data is free from privacy and secrecy concerns and is publicly available for reproducible research on fraud detection in ERP systems.

With ERP system data of both normal and fraudulent transactions now openly available, benchmarking existing fraud detection approaches and carrying out rigorous comparisons is now a promising point for future work. For this, we also plan on further extending the aggregation of joint datasets to enable accessible fraud detection from different perspectives of our generated ERP system data. Further, while we provide a variety of different occupational fraud scenarios, future efforts should be directed towards the aggregation of additional novel fraud cases that allow for further validation of fraud detection performance. Finally, since collection and annotation of datasets containing occupational fraud requires great effort, automated large scale acquisition of data for occupational fraud detection (e.g. through Active Learning) is a promising area for future work. With this study we took a first step towards open research on ERP fraud detection and encourage future research and practical applications by providing the generated data to the general public.

References

[1] ACFE. "Occupational Fraud 2022: A Report to the nations". In: *Report To the Nations* (2022). [Online; accessed 01. Jun. 2022]. URL: https://legacy.acfe.com/report-to-the-nations/2022/.

[2] Sarbanes-Oxley Act. "Sarbanes-oxley act". In: *Washington DC* (2002).

[3] Galina Baader and Helmut Krcmar. "Reducing false positives in fraud detection: Combining the red flag approach with process mining". In: *Int. Journal of Accounting Information Systems* 31 (2018), pp. 1–16.

[4] Galina Baader et al. "Specification and Implementation of a Data Generator to simulate Fraudulent User Behavior". In: *International Conference on Business Information Systems*. Springer. 2016, pp. 67–78.

[5] Lucio Barabesi, Andrea Cerioli, and Domenico Perrotta. "Forum on Benford's law and statistical methods for the detection of frauds". In: *Stat. Methods Appl.* 30.3 (Sept. 2021), pp. 767–778. ISSN: 1613-981X. DOI: 10.1007/s10260-021-00588-0.

[6] Federal Statistical Office of Germany. *ICT indicators for enterprises: Germany, years, employee size classes (52911-0003)*. [Online; accessed 12. Nov. 2020]. Nov. 2020. URL: https://www-genesis.destatis.de/genesis//online?operation=table&code=52911-0003&bypass=true&levelindex=0&levelid=1605691930837#abreadcrumb.

[7] Anna Fuchs et al. "A Meta-Model for Real-Time Fraud Detection in ERP Systems". In: *Proceedings of the 54th Hawaii International Conference on System Sciences*. 2021, p. 7112.

[8] Asadul Khandoker Islam et al. "Fraud detection in ERP systems using scenario matching". In: *IFIP Int. Information Security Conf.* Springer. 2010.

[9] PM Léger et al. "ERPsim". In: *ERPsim Lab (erpsim. hec. ca), HEC Montreal, Montreal, Qc* (2007).

[10] Susan S Lightle and Cynthia Waller Vallario. "Segregation of duties in ERP: an automated assessment tool enables internal auditors at Mead-Westvaco to enhance their SOD control reviews throughout the enterprise". In: *Internal auditor* 60.5 (2003), pp. 27–30.

[11] Ronny S Mans et al. "Application of process mining in healthcare–a case study in a dutch hospital". In: *International joint conference on biomedical engineering systems and technologies*. Springer. 2008, pp. 425–438.

[12] William Ferreira Moreno Oliverio et al. "A Hybrid Model for Fraud Detection on Purchase Orders". In: *Intelligent Data Engineering and Automated Learning – IDEAL 2019*. Cham, Switzerland: Springer, Oct. 2019, pp. 110–120. ISBN: 978-3-030-33606-6. DOI: 10.1007/978-3-030-33607-3_13.

[13] Mihaela Osaci et al. "SAP authorization based on the four eyes principle". In: *Annals of the Faculty of Engineering Hunedoara* 16.2 (2018), pp. 43–46.

[14] Markus Ring et al. "A survey of network-based intrusion detection data sets". In: *Computers & Security* 86 (2019), pp. 147–167.

[15] Michael Schermann and Scott R Boss. "The White-Collar Hacking Contest: A Novel Approach to Teach Forensic Investigations in a Digital World".

In: *2014 Dewald Roode Workshop on Information Systems Security Research*. 2014.

[16] Marco Schreyer et al. “Detection of accounting anomalies in the latent space using adversarial autoencoder neural networks”. In: *2nd KDD Workshop on Anomaly Detection in Finance*. ACM. 2019.

[17] Marco Schreyer et al. “Detection of anomalies in large scale accounting data using deep autoencoder networks”. In: *arXiv preprint arXiv:1709.05254* (2017).

[18] Kishore Singh and Peter Best. “Interactive visual analysis of anomalous accounts payable transactions in SAP enterprise systems”. In: *Managerial Auditing Journal* (2016).

[19] Kishore Singh and Peter J Best. “Design and implementation of continuous monitoring and auditing in SAP enterprise resource planning”. In: *Int. Journal of Auditing* 19.3 (2015), pp. 307–317.

[20] Statista. *Statista - ERP-market worldwide*. [Online; accessed 12. Nov. 2020]. Nov. 2020. URL: https://www.statista.com/outlook/14210/100/enterprise-resource-planning-software/worldwide.

[21] Julian Tritscher et al. “A financial game with opportunities for fraud”. In: *2021 IEEE Conference on Games (CoG)*. IEEE. 2021, pp. 1–5.

[22] York Yannikos et al. “3LSPG: Forensic tool evaluation by three layer stochastic process-based generation of data”. In: *Int. Workshop on Computational Forensics*. Springer. 2010.