Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

A FARMING-FOR-MINING-FRAMEWORK TO GAIN KNOWLEDGE IN SUPPLY CHAINS

Joachim Hunker
Alexander Wuttke
Anne Antonia Scheidler
Markus Rabe

Department IT in Production and Logistics
TU Dortmund University
Leonhard-Euler-Straße 5
Dortmund, 44227, GERMANY

ABSTRACT

Gaining knowledge from a given data basis is a complex challenge. One of the frequently used methods in the context of a supply chain (SC) is knowledge discovery in databases (KDD). For a purposeful and successful knowledge discovery, valid and preprocessed input data are necessary. Besides preprocessing collected observational data, simulation can be used to generate a data basis as an input for the knowledge discovery process. The process of using a simulation model as a data generator is called data farming. This paper investigates the link between data farming and data mining. We developed a Farming-for-Mining-Framework, where we highlight requirements of knowledge discovery techniques and derive how the simulation model for data generation can be configured accordingly, e.g., to meet the required data accuracy. We suggest that this is a promising approach and is worth further research attention.

1 INTRODUCTION

Nowadays, an SC is a complex system that contains inherent and coherent effects. Due to this complexity, supply chain management (SCM) is confronted with finding answers to a multitude of different logistics tasks, e.g., finding the right means of transport or predicting the customer demand. Therefore, support is necessary to aid decision makers in SCM in answering specific questions regarding logistics tasks and to subsequently be able to make the right decisions in decision-making situations (Teniwut and Hasyim 2020). One of the key factors in supporting decisions in SCM is gaining and visualizing knowledge. One of the widely established methods in theory and practice is known under the term KDD (Rahman et al. 2011). A common understanding is to view KDD as a process model, consisting of a sequence of different phases, ranging from data collection to the visualization and interpretation of results (Fayyad et al. 1996). The core phase is known as data mining, which is often used as a synonym for KDD (Adriaans and Zantinge 1996). Applying successful data mining, e.g., to find useful and previously unknown patterns, relies heavily on a valid and preprocessed input data basis, which is usually stored in a database. For a more in-depth discussion, the reader is kindly referred to Hunker et al. (2020) for an overview of database support by data mining tools. There is a wide range of different data mining techniques, which have various different requirements on the input data, e.g., sample size or data accuracy. Commonly, the data basis consists of observational or ”real” data, which can lead to different flaws. Typical examples are low data quality, e.g., missing or out-of-range data (García et al. 2015), or patterns found in the data that are only correlative and not causal (Sanchez 2018). With the rise of computational power and the availability of Big Data infrastructures in the last two decades, new opportunities arose in this context, with one being simulation-based data generation which is known as data farming and was introduced by Brandstein and Horne (1998). Data farming aims at using a simulation model as a data generator by running multiple

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

experiments to generate a large scale of data as a result. It makes heavy use of experiment design and high performance computing (HPC) (Horne and Meyer 2005).

Up to now, the combination of data farming and data mining has been insufficiently investigated in both theory and practice. The focus of this paper lies on the balance between the data farming output and the specific requirements of the data mining input. This in particular is relevant with respect to data preparation. We developed a Farming-for-Mining-Framework, where we, in light of the aforementioned, highlight the impact of well-designed data farming experiments specifically geared towards the application of data mining techniques. We rely on the usefulness of the combination of data farming and data mining to generate knowledge for decision makers in SCM. A fundamental discussion of this field of research can be found at Kusiak (2006).

This paper extends previous research at the Department of IT in Production and Logistics at TU Dortmund University in the context of data farming and data mining. We kindly refer the reader to Rabe and Scheidler (2014), Rabe and Scheidler (2015), Scheidler (2017), and Scheidler and Rabe (2021) for further reading.

The remainder of this paper is structured as follows: Section 2 briefly introduces the related work covering the background of SCs, KDD, and data farming. In Section 3, we discuss our Farming-for-Mining-Framework with an emphasis on data mining requirements. Section 4 covers our experiments in detail and discusses insights gained by our experiments. The paper closes with Section 5, where we present our conclusion, highlight limitations, and give an outlook on possible further research opportunities.

2 THEORETICAL BACKGROUND

The following sections introduce the related work for this paper. First, we highlight the SC as our problem domain with a focus on transaction data. Based on this, we discuss data mining as the core phase in the process model of KDD and highlight requirements on data and tools. In the last section we briefly introduce simulation and data farming and present the state of research.

2.1 Supply Chains

In an SC, various independent economic entities, e.g., suppliers and manufacturers, act together to form a complex network (Christopher 1998). SCs are generally managed by SCM and involve a variety of tasks and decision situations, such as selecting the right means of transport or identifying future customer requirements (Lambert 2014). The emerging tasks as well as the associated processes make use of flows in SCs. Due to their complexity, these processes are supported by IT systems and generate a lot of observational data. Usually, the data are persistently stored in a database system, e.g., a relational database. Typical process data in SCs are transaction data. A transaction is a process in which an object passes from one position to another one, e.g., the exchange of goods or materials, and consists of various properties like timestamps (Moody and Kortink 2000). Observed transaction data introduce various shortcomings, for example, missing or incorrect data, or relationships found in the data that are correlated rather than causal in nature (Sanchez 2018). One of the key challenges in supporting decisions for SCM based on transaction data in SCs is finding correlations (Harland 1996). These correlations are relevant to the process of KDD, which aims to generate knowledge to support decisions in SCM in its process.

Nowadays, various advances and trends in information technology have a great impact on SCs, summarized under terms like digitization, Big Data, or Logistics 4.0 (Borgi et al. 2017). Due to these advances, SCs have become complex and dynamic networks that are difficult to design and manage (Serdarasan 2013). This also has a direct impact on SCM knowledge discovery and underlying processes, making tasks more challenging.

2.2 Knowledge Discovery in Databases

KDD is a process for knowledge acquisition that is generally structured using process models. The process models vary depending on the focus and area of application, but show a large overlap in the essential phases (Kurgan and Musilek 2006). Figure 1 shows the well-established process model by Fayyad et al. (1996) with its relevant phases.

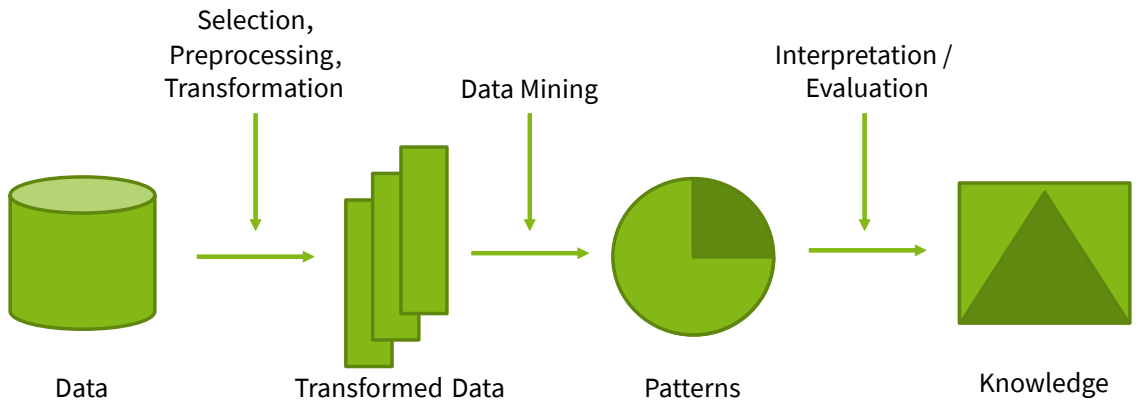

Figure 1: KDD according to Fayyad et al. (1996).

The essential phases include data selection, preprocessing and transformation, data mining, and evaluation at different levels of granularity. The applicability and success of the actual data mining process depends essentially on the input data (Crone et al. 2006), which are generally brought into a specific input format through preprocessing and transformation. Brachman and Anand state that "a KDD process cannot succeed without a serious effort to clean or scrub the data" (Brachman and Anand 1994, p. 6). Starting from a preproccessed data basis, the transformations of the data sets for the data mining processes can begin. For an overview of data mining in the context of SCs, the reader is kindly referred to Olson (2020) for an all-encompassing overview.

Transformation includes techniques such as reduction or projection, which are also listed under terms such as feature selection (Liu and Motoda 2001). The goal of the different techniques is to create a sufficient complexity of the data sets for the successful execution of data mining. Complexity of data sets is a concept that requires a multidimensional consideration. Since data sets today are potentially rather complex and extensive, many techniques pursue a so-called complexity reduction of the data. Looking at the data sets in terms of the complexity dimensions of number of attributes, domain of attribute occurrences, and number of entities, the SC environment is usually about complexity reduction. This is justified by the fact that extensive networks often hold a large amount of data on transitions, including, for example, historized orders. The impact generally concerns the number of attributes as well as the number of entities, unless they have been limited by a suitable sample in the selection phase. Attribute expressions also often require suitable preprocessing, since, for example, time values or exact route specifications are unsuitable as representatives of the continuous attribute expressions for data mining methods such as the rule learners. Complexity reduction in this context can be achieved by appropriate discretization, dimension reduction, aggregation, or numerical data reduction. The investigations in the area of complexity reduction are often very general and need a more specific investigation in the SC context. For example, standard techniques such as global discretization, which discretizes all continuous attributes, are not easily applicable to units of time or quantities in an SC (Frank and Witten 1999) and specific solutions such as local discretization (Liu and Motoda 2001) must be explored.

In summary, it can be stated that only suitable input data enable the applicability of specific data mining methods and that preprocessing and transformation are very costly. Here, it is worth considering whether

and in which cases it is expedient to adapt the input data to the data mining methods. Since simulation can be used to generate output data according to requirements, this seems to be a promising starting point.

2.3 Data Farming

Simulation is an established method in theory and practice for the modeling and analysis of complex systems (Law 2015), such as SCs (Rabe and Deininger 2012). Simulation can be defined as the "representation of a system with its dynamic processes in an experimentable model to reach findings, which are transferable to reality; in particular, the processes are developed over time" (Verein Deutscher Ingenieure 2014, p. 3). With the progress in information technology, e.g., the rise of computational power, new application possibilities for simulation emerge, for example in the context of data in SCs.

One way to address the above-mentioned flaws of observational data is simulation-based data generation, called data farming. Following the "Farming" metaphor, the idea is to use a simulation model for the targeted cultivation of data to maximize data output (Sanchez 2018). The term was coined by Brandstein and Horne (1998), evolving from Project Albert initiated by the US Marine Corps in 1998. Since then, most of the research has been conducted in the context of military and defense applications (Horne and Meyer 2016), e.g., in Forsyth et al. (2005), Kallfass and Schlaak (2012), and Lappi and Åkesson (2016), while some of the research publications have been transferring the ideas to other fields of research like medical science (Mayo et al. 2016) or manufacturing (Feldkamp et al. 2018).

Data farming is an iterative process that refers to using a simulation model as a data generator to generate vast amounts of data as output (Horne and Meyer 2005). Following the data farming loop of loops by Horne and Seichter (2014) that is presented in Figure 2, the process of data farming consists of different phases, which can be understood as a procedural model. It is separated into two main parts, the development of a simulation model and the experiment runs (Horne and Meyer 2016).

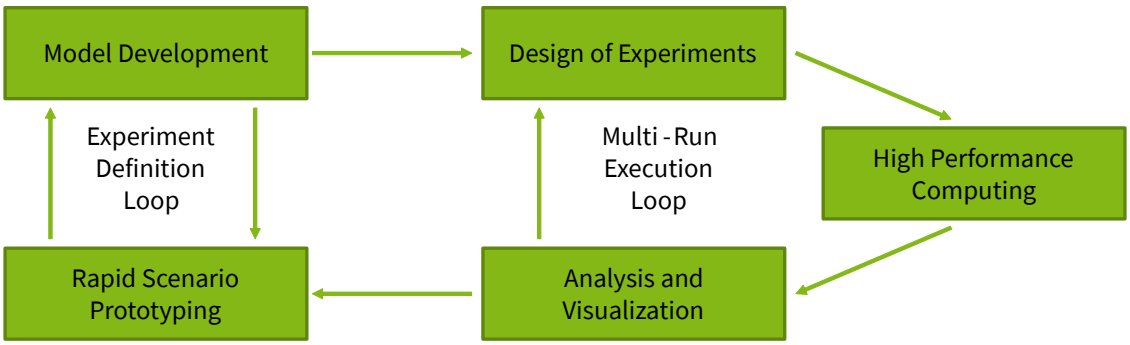

Figure 2: Data farming as the loop of loops according to Horne and Seichter (2014).

Due to the aforementioned complexity, for example in an SC, the developed simulation model contains a vast amount of factors that have to be changed accordingly (Horne and Seichter 2014). One of the key impacts to explore the developed simulation model in an effective and farmable way is design of experiments (Kleijnen et al. 2005). For example, instead of the common "trial-and-error" approach, which is often impractical or even impossible, the use of an appropriate design of experiments in a data farming study, for example 2^k Factorial Designs or the more efficient Latin Hypercube Designs, can reduce the number of experiments while ensuring balanced values of factors as well as a balanced model output (Sanchez 2006).

Conducting experiments using HPC leads to the generation of a vast amount of result data, which have to be analyzed accordingly to ultimately support decisions by decision makers in SCM. Following Horne and Meyer (2005), different methods can be used in the context of data farming, e.g., visualization or data mining. The process model in Figure 2 assumes that the analysis is part of data farming. However, as per our understanding, the analysis, e.g., KDD, connects directly to the data generation, which results in

two separate parts, data generation and analysis. A similar understanding can be identified for example in Feldkamp et al. (2018). This understanding is necessary for our Farming-for-Mining-Framework.

3 FARMING-FOR-MINING-FRAMEWORK

SCM is confronted with a multitude of different logistics tasks and decision situations which need to be supported adequately by using knowledge discovery techniques. Creating a valid, preprocessed, and transformed data basis for a successful knowledge discovery is a costly but necessary activity. In the context of SCs, data are complex, heavily interconnected, high in volume and can contain multiple flaws (see Section 2.1). Typically, when using real data and depending on the specifics of the used data mining techniques, the underlying data basis contains data that are not needed or cannot be used in raw form for the specific algorithm. To create a suitable data basis respectively subset, redundant or unsuitable data have to be prepared accordingly. A major problem in data mining is that insufficiently preprocessed data often lead to inherently wrong results. Since in our research the input data for data mining are artificial, data farming has to be aligned with the requirements of the used data mining techniques as well as the SCM tasks.

There is an interaction between the output generated by data farming and its input for data mining in the context of SCs. It is obvious that the output of data farming must be controlled with the objective that the data preparation effort for data mining methods can be significantly reduced or omitted completely. Thus, a balance between the data farming output and the data mining input is necessary. One of the decisive factors for the framework design are the complexity requirements of the specific data mining techniques. The complexity requirements of a data mining technique are multilayered (see Section 2.2). Our research shows that the following properties are viable for a variety of different data mining techniques, and consequently for a large part of SCM tasks. This concerns, for example, the following properties:

- **Data type**: Data mining techniques, for example FP-Growth or ID3, require specific data types in the attributes to be processed. As a consequence, the data type is an essential characteristic of the complexity. Typical representatives of data types in data mining algorithms are for example boolean, string, or integer.
- **Data type range**: Data mining techniques such as Support Vector Machines require a suitable non-linear function for the specific kernel transformations to map the input space into a high-dimensional space. As a consequence, the attribute domain must not contain null values and must be in well-defined ranges, for example, when using a sigmoid kernel.
- **Data volume (number of attributes)**: Data mining techniques, for example ID3 or CHAID, on the one hand, need the right number of attributes or can only work with a maximum number. On the other hand, the right attributes have to be selected. For example, a decision tree requires a set of attributes with a corresponding attribute value to generate an adequate decision tree for the problem domain.
- **Data volume (number of entities)**: Data mining techniques, for example k-Nearest-Neighbor, may require the labeling of entities. If the amount of data provided is too large, a sample has to be selected. This is accompanied by classical statistical problems, for example, whether the selected data subset is representative.
- **Relations (in and between entities)**: Inherent relations in and between entities must not be separated by automatic data mining techniques. For example, within the sampling of large input data sets these relations must be considered accordingly.

Therefore, it is necessary to integrate this in advance while conducting a possible study in an SC context and not leave it at random. This has not been considered in a summarizing framework so far. We assume a necessity of a common research domain between data farming output and data mining input in the context of SCM.

It follows that it is necessary to take these data mining requirements as well as specific SCM tasks into account while setting up a data farming experiment. This concerns in particular the design of experiments and the model design (see Section 2.3). With data farming, we are able to control the model output via parameterization in such a way that we can take the aforementioned properties into account in a suitable manner. For example, we can increase or decrease the number of attributes, or influence the domain space or the data types over the runtime of the experiments. Since the data mining input corresponds directly with the factors of the experiment design, we have identified two extremes of the property characteristics for our framework. The first is conducting the data farming experiment without consideration of the following data mining techniques with respect to answering a specific SCM task. The second is to fine-tune the design of experiments and the model in such way that data preparation before applying Mining techniques can be shortened accordingly. However, due to the structure of the Property Selection, we usually select characteristics in between the two extremes, because of the described balancing between data farming input and output. Due to the direct connection between input requirements of data mining techniques and simulation inputs of data farming, careful experiment planning and model development is required.

Therefore, our framework aims at the suitable definition of data farming output as data input for data mining to support decisions in SCM. As described before, data mining techniques have specific requirements on the complexity of the input data, which have to be served by data farming. Data complexity is understood differently in different application domains. There is no standardization of the term, because data in different disciplines differ greatly from one another and a specific consideration is required. Therefore, we will concentrate our approach on data complexity in an SC. We refer with this understanding to the categorization of SC data in the work of Oedekoven (2011). We propose the comprehensive framework shown in Figure 3, displaying only the phases that are relevant for our paper.

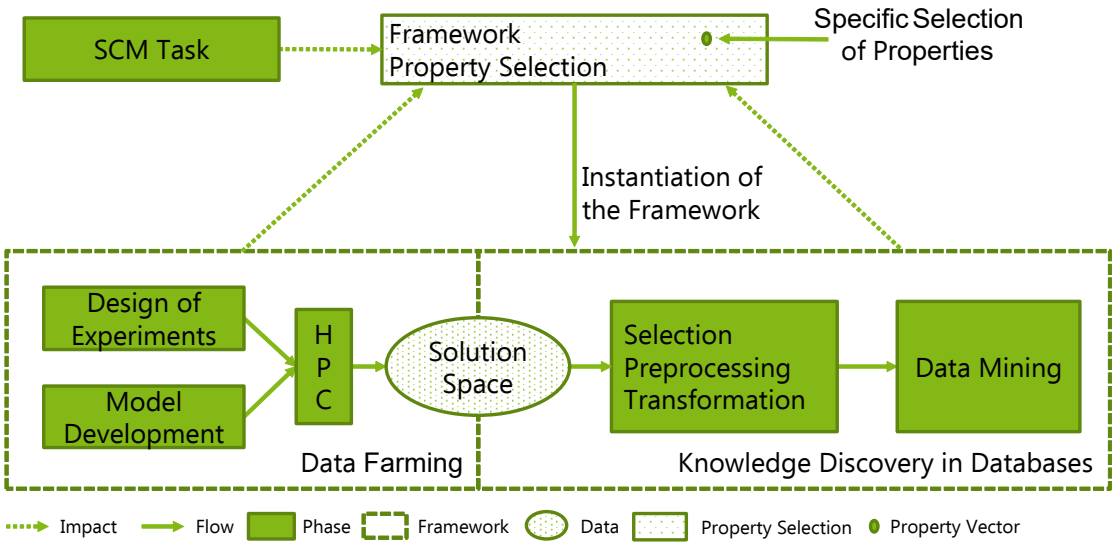

Figure 3: Relevant phases of the Farming-for-Mining-Framework.

The Farming-for-Mining-Framework consists of several elements. On the one hand, data farming and KDD, with the relevant phases, are combined in the framework. It also takes the features from data farming, especially the design of experiments and the model development phase, as well as from KDD, especially data preperation and data mining. The transition from farming to mining is marked by the Solution Space, which describes the output from data farming which is used as an input for KDD, respectively data mining. On the other hand the main element, which bundles the conceptual considerations as described above, is the Framework-Property-Selection. The characteristic of properties can mathematically be described as a vector and the Property Selection itself as a vector space. The framework starts with a specific task or a

decision situation in SCM which should be supported by our framework. The derivation of requirements resulting from an SCM task as well as from a data mining technique are taken into consideration. This results in a specific Property Selection for a specific set of data mining property characteristics, which will be used for the instantiation of our Farming-for-Mining-Framework. With the specific selection of properties, the simulation model is build and the experiments are designed. Using HPC, an artificial data basis is generated which is in the sequence used as an input for the KDD part of the framework. Depending on the selection of the specific properties, data preparation has to be implemented before running data mining.

As per our understanding, we propose to customize the data farming to the data mining requirements in such a way that data preparation can be reduced to a minimum. In summary, our concept proposes the general existence and necessity of such a Property Selection while using data farming for data mining. To validate our initial thoughts, we conducted experiments as a proof of work.

4 PROOF OF WORK

The following Section presents our proof of work for the Farming-for-Mining-Framework described in Section 3. First, we present the simulation model and the experiments we have conducted. Second, we briefly discuss the findings obtained.

4.1 Experiments

In our research, we developed a Farming-for-Mining-Framework which is programmed in C++. To create the simulation model and to run the experiments, the framework uses the established simulation software Plant Simulation (Bangsow 2020). To perform the analysis, we are using the known RapidMiner (Kotu 2015) as the data mining tool.

The objective of our experiments is highlighting to what extent the consideration of both the SCM task and data mining technique have an impact on the time consumed within the data preparation phase from KDD. This is done using two different instantiations of the framework (see Figure 3). Regarding KDD and data mining, we focused the experiments on a rule learner to demonstrate our initial thoughts (see Section 2.2). Based on this, SCM can decide upon procurement strategies and in particular on possible supply combinations based on customer demand.

For our proof of concept, we developed a discrete event model of a sufficiently complex, typical SC-network. The resulting simulation model is presented in Figure 4.

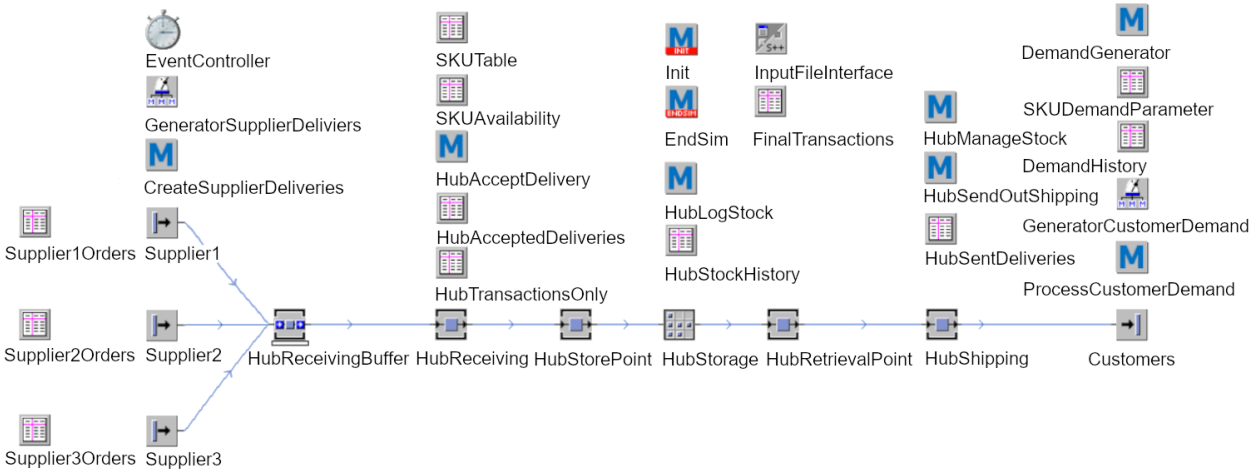

Figure 4: Plant Simulation model.

Such a network can be identified as a two-echelon network, with levels from source to consolidation hub and from consolidation hub to customers. In detail, the model contains three suppliers that act as a source and deliver Stock Keeping Units (SKU) based on customer demand. A total of nine different SKUs are differentiated in the model. Each supplier can deliver a subset of the SKUs. The model considers several parameters for sourcing, e.g., main supplier, stock levels, delivery time, and the reduction of individual deliveries to the central hub. The SKUs are loaded using standard containers and are delivered individually by the suppliers to the hub using a buffer to consolidate the deliveries. In the hub, SKUs are stored and, when demanded, handled and shipped to customers. The customers act as the sink of the model. Customers are modeled as a black box, but the demands for different SKUs are generated separately using a mathematical ruleset on an aggregated level for this purpose. These rules resemble parameterized probability density functions, more precisely, Normal distributions and Erlang-k distributions. These represent the expected mean demand for a given day. A Normal distribution is then in turn subsequently applied to these means. In our model, the delivery is finished once the SKUs reach the customers.

Our guiding task from the SCM for the experiments is the question in which combination the SKUs arrive at the hub for a given day with respect to customer demand. Taking this into consideration, an SCM can decide upon sourcing strategies, e.g., in respect to the main suppliers. In our experiment planning, we decided to show two selections, or vectors, in our Property Selection (see Figure 3). First, we instantiated our Farming-for-Mining-Framework without taking into consideration any requirements of specific data mining techniques. Second, we instantiated our framework specifically geared towards the use of a specific data mining technique to answer the given SCM task. This applies in particular to the consideration of the properties of the specific algorithm mentioned in Section 3 during model development and in the design of experiments. We will explain our experimental setup using both of the described cases in the following.

An SCM task of the type as described above can be answered using a rule learner as the data mining technique. In detail, we used the common FP-Growth algorithm to mine for frequent itemsets in the artificial data. This means that we have to carefully consider the requirements and the method of operation of this algorithm while setting up our experiments with our Farming-for-Mining-Framework. For example, the FP-Growth constructs a tree in the first step and in the second step searches for patterns based on the created tree. To do so, the algorithm makes use of valid transaction data. This is, for the second case, directly considered in the model.

As described above, the model contains several stochastic input parameters, which have been taken into consideration in a design of experiments. We used a Nearly Orthogonal Latin Hypercube design over other designs, for example a full factorial n^k design, since our prototype is of sufficient complexity for our proof and we can reduce the number of experiments while keeping balanced inputs factors. We used the spreadsheets provided by the SEED Center for data farming (please refer to Sanchez (2011) for more information). We parameterized the model using 27 factors. These are used to manage the expected demand of the nine different SKUs. In more detail, the demand for each SKU is controlled using three different factors. The design enables us to analyze the factors between a low and a high level. Examples are given in Figure 5, where two exemplary courses for a high and low level mean of SKU demand are shown.

First, it shows the progression of the mean demand in regards to the high level and its corresponding generated demand for an SKU based on the given mean (marked with 1). Second, it shows the progression for the low level analogous to the high level (marked with 2). The full design consists of 257 design points. We ran our experiments on a single machine. During the conduction of the experiments in Plant Simulation, the result data are written into a flat file. When the experiments are finished, the KDD part of our framework starts. Depending on the case, data preparation has to be conducted first. In the first case, we had to analyze the output data if and to what extend data preparation is necessary for the application of our algorithm, in our proof of work FP-Growth, and, therefore, be able to support decisions regarding the given SCM task at the end. For example, we had to discretize the numerical quantities for each delivery of SKUs to the hub. Within the second case, results in the sequence could be loaded directly into the analysis tool.

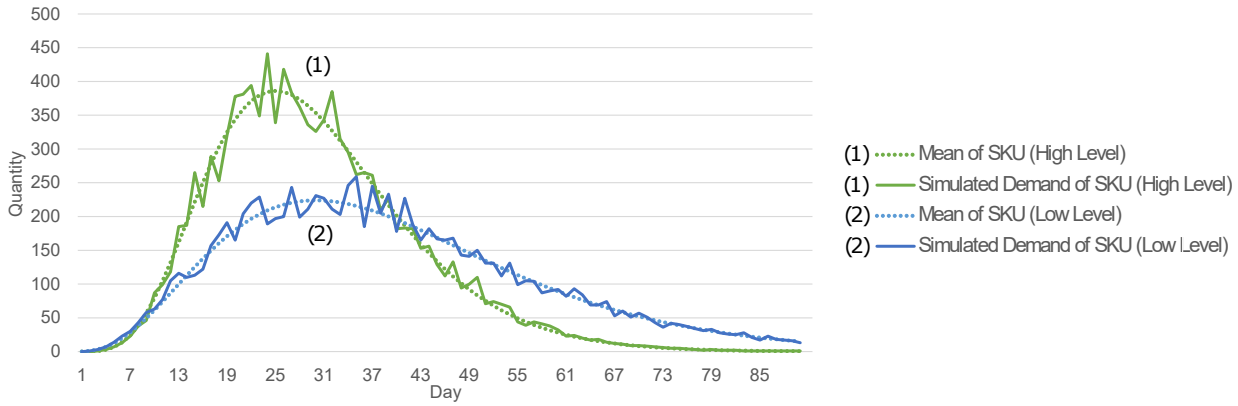

Figure 5: Examples for generated demands.

RapidMiner offers the use of FP-Growth, as well as multiple other algorithms, as building blocks wich are called ”Operators”, out of the box. Here, we draw the reader’s attention to the aspect that RapidMiner cannot only be located in the data mining phase of the KDD, as it offers multiple operators for the data preparation phases as well (see Figure 1). After loading the data, we ran the FP-Growth algorithm. The result of the algorithm is a frequent itemset. To learn a rule, we created associations, which can be understood as ”if-then-statements”. This is also done using RapidMiner building blocks. To display the results, e.g., for a decision process in the SCM, we used the included visualization techniques offered with the RapidMiner software. The resulting knowledge can be used by decision makers of SCM, for example to change the sourcing strategies.

4.2 Findings

Our experiments show that data mining, or KDD in general, should not be viewed as an isolated step within a data farming study. Basically, it can be stated that at the beginning of a data farming for data mining study, one is faced with two possible alternatives, which we have shown with our experiments. The first one is to prepare the input data using the data preparation phases of KDD, for example transformation, to prepare the data for the specific mining technique. The second one is to include the requirements of a specific data mining technique in the model development and design of experiments. This is accompanied by the requirements of the SCM task in a possible decision support scenario. A trade-of exists at this point. On the one hand, the artificial data (and subsequently model and experiment design) are specifically geared for the use of a certain data mining technique. If changes are necessary, for example due to a change in data mining, this entails correspondingly extensive work on the model and the experiment design. On the other hand, it seems promising that we can drastically save process time by reducing or even eliminating data preparation time before using a data mining technique. As a reminder, 61 % to 80 % of the time for the whole KDD is spent on data preparation (the reader is kindly referred to Munson (2012) for a survey among researchers and practitioners). In addition, the actual experiment time could be significantly reduced in this case while maintaining the same valid output. In our case, this was in the range of approx. 50 % time savings in computing time. In a nutshell, one has to decide between flexibility and reducing process time to create a valid data basis for data mining.

The developed Farming-for-Mining-Framework proved robust to be a first step to approach this trade-of in a suitable way. The requirements of SCM tasks, data mining techniques as well as model development and experiment design of data farming are taken into account in a Property Selection. This allows for a specific instantiation of our framework and the consideration of property characteristics with respect to the complexity requirements while at the same time ensuring flexibility to some extend and reducing or

removing data preparation time. Our research indicates that trading flexibility for time savings is beneficial and, on a side note, essential in light of, for example, realtime Big Data analysis.

To summarize, we can state that our experiments confirmed our initial thoughts, even though the case for our experiments conducted was sufficient, but simple to some extend.

5 CONCLUSION, LIMITATIONS, AND OUTLOOK

In this paper, we introduced a Farming-for-Mining-Framework in which we investigated the specific requirements of data mining techniques on artificial input data, which are solely based on the data output generated by data farming in the context of SCs. For this purpose, we identified research fields which have been analyzed and linked together. First, we briefly introduced SCs as our problem domain and highlighted the need for knowledge by decision makers of SCM. To gain knowledge, we relied on the established process of KDD and discussed the importance of valid and preprocessed input data for the data mining phase. In this context, we introduced the well-known approach of data farming to use a simulation model for the targeted generation of artificial input data. Our substantive considerations in this context have shown the need to carefully consider the dependence of valid input data for KDD and respectively the data mining techniques on the output of the data farming in light of SCM tasks. This concerns in particular the experiment design and the simulation model. We showed that mining techniques have multilayered complexity requirements regarding various different properties, which in turn have to be be taken into account specifically for each technique. To balance the data farming output with data mining input between two extremes, and to connect both of the domains, we introduced the concept of a Farming-for-Mining-Framework. A certain point, or vector, in the Property Selection represents a combination of characteristics of the data mining properties, which reflect the inherent dependency while using Farming for Mining. As proof of concept, we presented two different experiments within our Farming-for-Mining-Framework. Results showed that carefully considering requirements of data mining techniques within data farming is beneficial, since it can significantly reduce preprocessing and transformation efforts in the sequence and increase validity of the data basis.

Furthermore, and since our framework allows a first approach to the described research topic, our experiments show that the concept of our framework has limitations that can be narrowed down to the sufficient complexity in our experiments. We assume that with more complexity of the SCM tasks as well as the data mining techniques the optimal balancing point between flexibility and saving processing time in data preparation will move even more in the direction of time saving. This requires more complex experiments to be conducted. Moreover, this includes the extension beyond rule learners such as FP-Growth as data mining techniques. In addition, we assumed a given complexity as a black box for the data mining properties, which needed to be controlled by experiment design and model creation. This showed the lack of complexity categorization for the data used in data mining. Detailed experiments, especially with respect to ”real-life-situations”, have to be carried out in order to be able to derive further development steps for the Farming-for-Mining-Framework. For example, the results of our framework based on artificial data have to be tested and compared with the results based on real data in a specific SC scenario. Especially the consideration of further parameters for the initialization has to be implemented. Moreover, the practical use of knowledge based on artificial data is of future research interest. This includes applying the framework for different SCM tasks to be able to validate the practical relevance and trustworthiness of our framework.

REFERENCES

Adriaans, P., and D. Zantinge. 1996. *Data Mining*. Boston: Addison Wesley Professional.

Bangsow, S. 2020. *Tecnomatix Plant Simulation*. Cham: Springer International Publishing.

Borgi, T., N. Zoghlami, and M. Abed. 2017. “Big Data for Transport and Logistics: A Review”. In *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, 44–49: IEEE.

Brachman, R. J., and T. Anand. 1994. “The Process of Knowledge Discovery in Databases: A First Sketch”. In *Knowledge Discovery in Databases*, edited by U. M. Fayyad and R. Uthurusamy, 1–11. Menlo Park: AAAI.

Brandstein, A. G., and G. E. Horne. 1998. "Data Farming: A Meta-technique for Research in the 21st Century". *Maneuver Warfare Science*:93–99.

Christopher, M. 1998. *Logistics and Supply Chain Management: Strategies for Reducing Cost and Improving Service*. 2nd ed. London: Financial Times.

Crone, S. F., S. Lessmann, and R. Stahlbock. 2006. "The Impact of Preprocessing on Data Mining: An Evaluation of Classifier Sensitivity in Direct Marketing". *European Journal of Operational Research* 173(3):781–800.

Fayyad, U. M., G. Piatetsky-Shapiro, and P. Smyth. 1996. "From Data Mining to Knowledge Discovery in Databases". *AI Magazine* 17(3):37–54.

Feldkamp, N., S. Bergmann, S. Strassburger, E. Borsch, M. Richter, and R. Souren. 2018. "Combining Data Farming and Data Envelopment Analysis for Measuring Productive Efficiency in Manufacturing Simulations". In *Proceedings of the 2018 Winter Simulation Conference (WSC)*, edited by M. Rabe, A. A. Juan, A. Mustafee, S. J. Skoogh, and B. Johansson, 1440–1451. Piscataway, New Jersey: IEEE.

Forsyth, A. J., G. E. Horne, and S. C. Upton. 2005. "Marine Corps Applications of Data Farming". In *Proceedings of the 2005 Winter Simulation Conference (WSC)*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joins, 1077–1081. Piscataway, New Jersey: IEEE.

Frank, E., and I. H. Witten. 1999. "Making Better Use of Global Discretization". In *Proceedings of 16th international conference on machine learning*, edited by I. Bratko and S. Dzeroski, 115–123. San Francisco: Morgan Kaufmann.

García, S., J. Luengo, and F. Herrera. 2015. *Data Preprocessing in Data Mining*, Volume 72. Cham: Springer.

Harland, C. M. 1996. "Supply Chain Management: Relationships, Chains and Networks". *British Journal of Management* 7(1):63–80.

Horne, G., and T. Meyer. 2016. "Data Farming Process and Initial Network Analysis Capabilities". *Axioms* 5(1):1–17.

Horne, G., and S. Seichter. 2014. "Data Farming in Support of NATO Operations - Methodology and Proof-of-Concept". In *Proceedings of the 2014 Winter Simulation Conference (WSC)*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and Miller J. A., 2355–2363. Piscataway, New Jersey: IEEE.

Horne, G. E., and T. E. Meyer. 2005. "Data Farming: Discovering Surprise". In *Proceedings of the 2005 Winter Simulation Conference (WSC)*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joins, 1082–1087. Piscataway, New Jersey: IEEE.

Hunker, J., A. A. Scheidler, and M. Rabe. 2020. "A Systematic Classification of Database Solutions for Data Mining to Support Tasks in Supply Chains". In *Data Science and Innovation in Supply Chain Management : How Data Transforms the Value Chain*, edited by W. Kersten, T. Blecker, and C. Ringle, 395–425. epubli. https://doi.org/10.15480/882.3121.

Kallfass, D., and T. Schlaak. 2012. "NATO MSG-088 Case Study Results to Demonstrate the Benefit of Using Data Farming for Military Decision Support". In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 1–12. Piscataway, New Jersey: IEEE.

Kleijnen, J. P. C., S. M. Sanchez, T. W. Lucas, and T. M. Cioppa. 2005. "State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments". *INFORMS Journal on Computing* 17(3):263–289.

Kotu, V. 2015. *Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner*. Waltham, MA: Morgan Kaufmann.

Kurgan, L. A., and P. Musilek. 2006. "A Survey of Knowledge Discovery and Data Mining Process Models". *The Knowledge Engineering Review* 21(1):1–24.

Kusiak, A. 2006. "Data Farming: Concepts and Methods". In *Data Mining and Knowledge Discovery Approaches Based on Rule Induction Techniques*, edited by E. Triantaphyllou and G. Felici, Volume 6 of *Massive Computing*, 279–304. Springer US.

Lambert, D. M. 2014. *Supply Chain Management: Processes, Partnerships, Performance*. Fourth ed. Ponte Vedra Beach, Florida: Supply Chain Management Institute.

Lappi, E., and B. Åkesson. 2016. "Tactical Size Unit as Distribution in a Data Farming Environment". *Axioms* 5(1):1–11.

Law, A. M. 2015. *Simulation Modeling and Analysis*. Fifth ed. New York: McGraw-Hill Education.

Liu, H., and H. Motoda. 2001. *Instance Selection and Construction for Data Mining*. Boston: Springer US.

Mayo, C. S., M. L. Kessler, A. Eisbruch, G. Weyburne, M. Feng, J. A. Hayman, S. Jolly, I. El Naqa, J. M. Moran, M. M. Matuszak, C. J. Anderson, L. P. Holevinski, D. L. McShan, S. M. Merkel, S. L. Machnak, T. S. Lawrence, and R. K. ten Haken. 2016. "The Big Data Effort in Radiation Oncology: Data Mining or Data Farming?". *Advances in Radiation Oncology* 1(4):260–271.

Moody, D. L., and M. A. Kortink. 2000. "From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design". In *Proceedings of the 2nd International Workshop on Design and Management of Data Warehouses*, edited by M. A. Jeusfeld, H. Shu, M. Staudt, and G. Vossen, 1–12. Stockholm: DMDW.

Munson, M. A. 2012. "A Study on the Importance of and Time Spent on Different Modeling Steps". *ACM SIGKDD Explorations Newsletter* 13(2):65–71.

Oedekoven, D. 2011. *Nutzenpotenziale harmonisierter Stammdaten in den Prozessen der Auftragsabwicklung von Auftragsfertigern*. Aachen: Apprimus.

Olson, D. L. 2020. "A Review of Supply Chain Data Mining Publications". *Journal of Supply Chain Management Science* (1):15–26.

Rabe, M., and M. Deininger. 2012. "State of Art and Research Demands for Simulation Modeling of Green Supply Chains". *International Journal of Automation Technology* 6(3):296–303.

Rabe, M., and A. A. Scheidler. 2014. "An Approach for Increasing the Level of Accuracy in Supply Chain Simulation by Using Patterns on Input Data". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 1897–1906. Piscataway, New Jersey: IEEE.

Rabe, M., and A. A. Scheidler. 2015. "Farming for Mining - Entscheidungsunterstützung mittels Simulation im Supply Chain Management". In *Simulation in Production and Logistics 2015*, edited by M. Rabe and U. Clausen, 671–679. Stuttgart: Fraunhofer IRB.

Rahman, F. A., M. I. Desa, and A. Wibowo. 2011. "A Review of KDD-Data Mining Framework and its Application in Logistics and Transportation". In *2011 7th International Conference on Networked Computing and Advanced Information Management (NCM)*, edited by Y. Cho, S. Kawata, and F. Ko, 175–180. Piscataway, New Jersey: IEEE.

Sanchez, S. M. 2006. "Work Smarter, not Harder: Guidelines for Designing Simulation Experiments". In *Proceedings of the 2006 Winter Simulation Conference (WSC)*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 47–57. Piscataway, New Jersey: IEEE.

Sanchez, S. M. 2011. "NOLHdesigns Spreadsheet". http://harvest.nps.edu/, accessed 29.03.2021.

Sanchez, S. M. 2018. "Data Farming: Better Data, Not Just Big Data". In *Proceedings of the 2018 Winter Simulation Conference (WSC)*, edited by M. Rabe, A. A. Juan, A. Mustafee, S. J. Skoogh, and B. Johansson, 425–439. Piscataway, New Jersey: IEEE.

Scheidler, A. A. 2017. *Methode zur Erschließung von Wissen aus Datenmustern in Supply-Chain-Datenbanken*. 1 ed, Volume 1 of *Schriftenreihe Fortschritte in der IT in Produktion und Logistik*. Göttingen: Cuvillier.

Scheidler, A. A., and M. Rabe. 2021. "Integral Verification and Validation for Knowledge Discovery Procedure Models". *International Journal of Business Intelligence and Data Mining (IJBIDM)* 18(1):73–87.

Serdarasan, S. 2013. "A Review of Supply Chain Complexity Drivers". *Computers & Industrial Engineering* 66(3):533–540.

Teniwut, W. A., and C. L. Hasyim. 2020. "Decision Support System in Supply Chain: A Systematic Literature Review". *Uncertain Supply Chain Management*:131–148.

Verein Deutscher Ingenieure 2014. *VDI 3633 - Simulation of Systems in Materials Handling, Logistics and Production: Fundamentals*. Berlin, Germany: Beuth Verlag.

AUTHOR BIOGRAPHIES

JOACHIM HUNKER is a researcher at the department IT in Production and Logistics at the TU Dortmund University. He holds a Master of Science in Logistics, Infrastructure, and Mobility with a focus on IT in Logistics from the Technical University of Hamburg. He graduated with a master thesis on a hybrid-scheduling-approach of assembly lines of car manufacturers. His research focuses on simulation-based data generation and data analytics in logistics. His email address is joachim.hunker@tu-dortmund.de.

ALEXANDER WUTTKE is a research assistant at the department IT in Production and Logistics and a master's student of mechanical engineering with an emphasis on IT in Production and Logistics at the TU Dortmund University. His research interests focus on data processing in production and logistics. His email address is alexander2.wuttke@tu-dortmund.de.

ANNE ANTONIA SCHEIDLER is a researcher at the department IT in Production and Logistics at the TU Dortmund University. Until 2012, she worked as IT consultant for different large companies. Key areas of her activity were business processes modeling and data concepts. She graduated in 2017 with a PhD thesis on methods for knowledge discovery using data patterns. Currently, her research focus is on Data Mining, concepts for input data, data in simulation, and supply chain information. Her e-mail address is anne-antonia.scheidler@tu-dortmund.de.

MARKUS RABE is a full professor for IT in Production and Logistics at the TU Dortmund University. Until 2010 he had been with Fraunhofer IPK in Berlin as head of the corporate logistics and processes department, head of the central IT department, and a member of the institute direction circle. His research focus is on information systems for supply chains, production planning, and simulation. Markus Rabe is vice chair of the "Simulation in Production and Logistics" group of the simulation society ASIM, member of the editorial board of the Journal of Simulation, member of several conference program committees, has chaired the ASIM SPL conference in 1998, 2000, 2004, 2008, and 2015, Local Chair of the WSC'2012 in Berlin and Proceedings Chair of the WSC'2018 and WSC'2019. More than 200 publications and editions report from his work. His email address is markus.rabe@tu-dortmund.de.