A Goal-Oriented Approach for Preparing a Machine-Learning Dataset to Support Business Problem Validation

Citation for published version (APA):
Ahn, R., Supakkul, S., Zhao, L., Kolluri, K., Hill, T., & Chung, L. (2021). *A Goal-Oriented Approach for Preparing a Machine-Learning Dataset to Support Business Problem Validation*. Paper presented at IEEE BigData 2021.

A Goal-Oriented Approach for Preparing a Machine-Learning Dataset to Support Business Problem Validation

1st Robert Ahn
Department of Computer Science
The University of Texas at Dallas
Richardson, Texas, USA
robert.sungsoo.ahn@utdallas.edu

2nd Sam Supakkul
NCR Corporation
Atlanta, Georgia, USA
sam.supakkul@ncr.com

3rd Liping Zhao
Department of Computer Science
The University of Manchester
Manchester, UK
liping.zhao@manchester.ac.uk

4th Kirthy Kolluri
Department of Computer Science
The University of Texas at Dallas
Richardson, Texas, USA
kirthy.kolluri@utdallas.edu

5th Tom Hill
Fellows Consulting Group, LLC
Dallas, Texas, USA
tom@fellowsconsultinggroup.com

6th Lawrence Chung
Department of Computer Science
The University of Texas at Dallas
Richardson, Texas, USA
chung@utdallas.edu

***Abstract*—Preparing a dataset representing business problems is an essential task in Machine Learning (ML). A suitable dataset is critical to accurate ML algorithms, which helps validate business problems. For example, preparing a dataset for predicting loan default in one bank would be vital in the ML project as bank staff may take some actions to mitigate the problem. However, preparing a dataset for identifying potential business problems is challenging. Some challenges might include determining possible events leading to problems, identifying testable factors of the events, and mapping a testable factor to data features to extract relevant data from source data. ML models using irrelevant or unimportant data may give incorrect predictions, negatively impacting problem validation, consequently not solving business problems. We present a goal-oriented approach for preparing an ML dataset to address this challenge. The approach provides an ontology and a process for guiding data preparation. In addition, it helps capture problematic business events, refine a business event to find a testable factor, map a testable factor to a database entity and features, and extract data from a database or Big data. We illustrate the approach using a retail banking application and a Financial database. The experimental results, we believe at least, show that the approach supports preparing a relevant ML dataset, helping validate business problems.**

***Index Terms*—Preparation, Dataset, Machine Learning, Validation, Problem Hypothesis, Goal-Orientation**

I. Introduction

Preparing an essential dataset representing business problems is vital in the Machine Learning (ML) project to mine some hidden patterns in data and discover insights leveraging the patterns in alleviating or mitigating business problems [1]–[3]. For example, preparing a relevant dataset from a banking database for predicting a client's loan default would be critical to the success of the ML project as the bank may take some actions to mitigate the problem with the prediction result.

However, preparing an ML dataset for identifying some events behind a business problem is challenging [4], [5]. Specifically, some challenges might be systematically exploring potential events leading to a business problem, identifying testable factors for the specified events, and mapping the testable factors to data features to extract relevant data from source data. Problem validation with an irrelevant or unimportant dataset may give inaccurate predictions, leading to dissatisfaction systems, consequently not solving business problems and failing to achieve business goals.

Drawing on our previous work, GOMA [6] and Metis [7], we present a goal-oriented data preparation approach, *DREGON*(Data pREparation using GOal-orieNtation) to support business problem validation. Four technical contributions are made in this paper. Firstly, a domain-independent ontology and a process for data preparation are described. Secondly, a method for capturing business events likely causing problems is presented. Thirdly, an entity modeling method identifying a testable factor of the captured business event is elaborated. Fourthly, a mapping method for connecting a testable factor to a database entity and features is shown.

This paper illustrates the proposed *Dregon* approach using a retail banking application and a Financial database. We suppose a hypothetical bank, the Case bank provides client services, such as offering loans and issuing credit cards. The bank has experienced an unpaid loan problem, where some clients failed to pay loan payments when due. However, it was challenging for the bank manager to know what specific clients' banking behaviors were behind this issue. So, the bank consulted a data analytics company to address this issue. The company hypothesized potential events impacting the loan problem against the bank's goals. It then prepared some data from the Financial database, performed an in-depth ML analysis by validating the hypothesized events, and suggested to the bank manager highly likely client's banking behaviors leading to the loan problem. This paper shows how goals, problems, hypothesized banking events, and some ML concepts, such as data features, a target label, and classification, can be systematically applied to prepare a dataset to validate potential banking events towards the unpaid loan. Our approach could help the bank manager make sound decisions among the alternative potential banking events and get confidence in mitigating the selected problem. Fig. 1 shows a high-level context diagram concerning the unpaid loan.

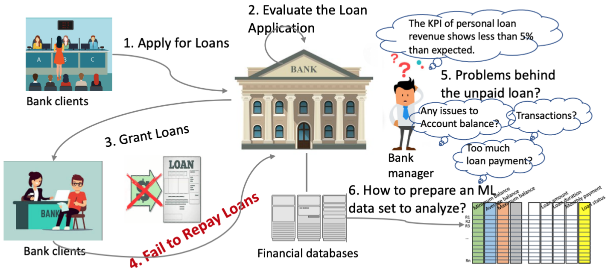


Fig. 1. Unpaid loan in Case bank (empirical study context)

The rest of this paper is structured as follows. Section II and III presents related work and the Dregon approach each. Section IV then illustrates the data preparation process in detail. Section V describes three experiments performed, and Section VI discusses observations and limitations. Finally, Section VII summarizes the paper and future work.

II. Related Work

The distinctive of our data preparation approach is to use a problem hypothesis for exploring alternative causes of a business problem in a goal-oriented manner, map the alternatives to data features of a database entity, and extract relevant data from a source database. The prepared data set is then entered into ML models to support business problem validation.

Problem analysis and data preparation have been studied to understand and solve real-world problems in two major areas: Requirements Engineering and Machine Learning [8], [9]. In Requirements Engineering, a fishbone diagram [10], Fault Tree Analysis (FTA) [11], Problem Frame [12], and (Soft-)Problem Interdependency Graph (PIG) [13] have been used to analyze root causes behind a problem. A fishbone diagram supports enumerating potential reasons for a problem and is typically used in a brainstorming session. FTA depicts a failure path and forms a logic diagram of failure. Problem Frame uses concepts including phenomena, shared phenomena, and domain requirements to analyze business problems and develop software solutions. PIG uses a (Soft−)problem concept to represent a stakeholder problem against stakeholder goals and provides refinement methods for a (Soft−)problem. While these techniques provide a sound, high-level model for analyzing business problems into sub-problems and some relationships, they lack mechanisms for connecting high-level concepts to the data features in a database and validating the identified problems using operational data in business.

In the area of Machine Learning, data is prepared in the structure or format that fits each machine learning task. As business databases may include noise, missing values, similar features, or redundant data, some low-quality data should be preprocessed or reduced for good prediction. There can be two kinds of preparation techniques, data preprocessing and data reduction [2]. The data preprocessing techniques may include data cleaning, transformation, integration, normalization, missing data imputation, and noise identification [14], [15]. In data reduction, the amount of data is downsized, while the reduced data still includes the essential structure of the original data. The data reduction techniques include feature selection, instance selection, discretization, feature extraction and/or instance generation [16], [17]. Although the data preprocessing and data reduction techniques in ML are useful in partly preparing data, these techniques often lack high-level concepts, such as goals and problems and their relationships, such as positive, negative contributions. These techniques are often used to identify low-level problems informally and do not provide traceability to higher-level problems [18]. Our approach prepares an ML data dataset to support business problem validation, adopting essential concepts of the goal-oriented and ML-based approach in a complementary manner.

III. The Dregon Approach

The Dregon approach provides a domain-independent ontology and a series of steps, helping prepare a dataset by exploring problems, identifying a testable factor and data features, and extracting data.

A. The Dregon Ontology

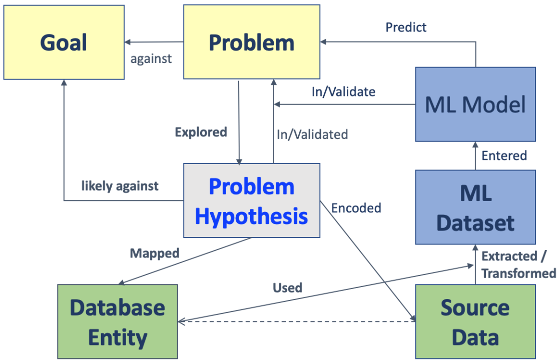


Fig. 2. The data preparation ontology at a high-level

The Dregon ontology, adopting key concepts from a goal-oriented [19] and ML-based approach [20], intends to help data preparation for validating a problem hypothesis. The ontology consists of essential modeling concepts, relationships among concepts, and constraints among the concepts and relationships. Fig. 2 shows a high-level ontology. The boxes and arrows represent concepts and relationships among concepts.

The more detailed Dregon ontology is shown in Fig. 3.(a). A few essential concepts needed for preparing a dataset are described. A *(Soft-)Goal* is defined as a goal that may not

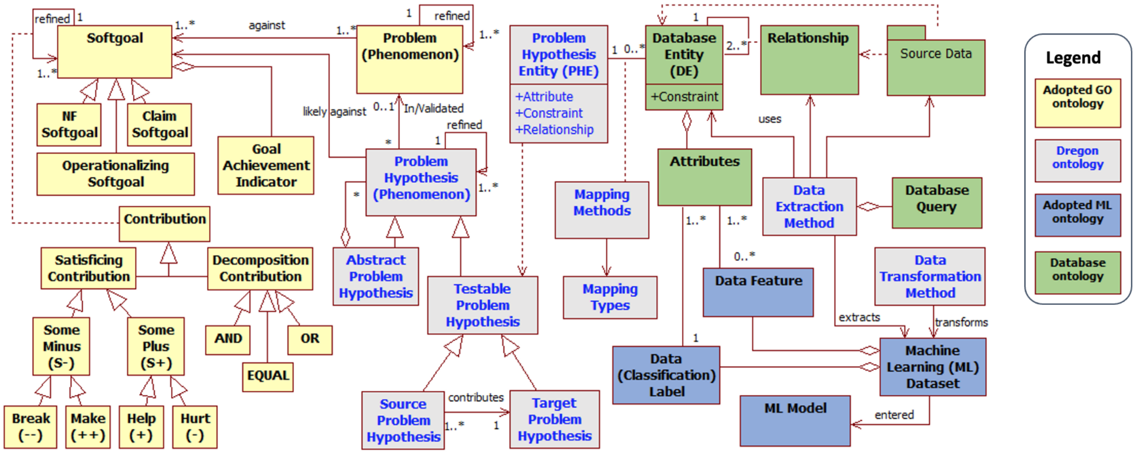


Fig. 3. The detailed data preparation ontology for a validating problem hypothesis

have a clear-cut criterion and can be specialized into a Non-Functional (NF) softgoal, an Operationalizing softgoal, and a Claim softgoal. While a *(Soft-)Problem* is a phenomenon against a softgoal, a *Problem Hypothesis* is a hypothesis that we believe a phenomenon is against a softgoal.

There are two kinds of problem hypotheses, an *Abstract Problem Hypothesis* and a *Testable Problem Hypothesis*. An abstract problem hypothesis is conceptual and not concrete enough to test, whereas a testable problem hypothesis is measurable and testable. A Testable Problem Hypothesis may be further refined, forming a testable *Source Problem Hypothesis* and a *Target Problem Hypothesis*. A *Problem Hypothesis Entity* is an entity representing a Testable Problem Hypothesis and may be mapped to a relevant *Database Entity* having *Attributes*, *Constraints*, and *Relationships* in a source data model. The identified database entities are used to extract data from source data using *Data Extraction Method*.

The Contribution relationships among goals, problems, and problem hypotheses are categorized into Decomposition types, such as *AND, OR, EQUAL*, or *Satisficing* types, such as *Make, Help, Hurt, Break, Some-Plus, Some-Minus* adopted from the NFR Framework [19]. The relationships between problem hypotheses and problems are either *Validated* or *Invalidated*.

One crucial constraint about a problem hypothesis includes time-order among a source and target problem hypothesis, where a source problem hypothesis must have occurred before the target problem hypothesis. Other constraints are a positive contribution from a source problem hypothesis to a target problem hypothesis, and the contribution relationship should be reasonably sensible [21].

B. The Dregon Process

The Dregon process, shown in Fig. 4.(a), consists of four steps, *Step 1: Explore business goals*, *Step 2: Hypothesize business problems*, *Step 3: Identify data features for a problem hypothesis*, and *Step 4 Extract and transform datasets*. The steps are necessary to prepare an ML data set systematically and should be understood as iterative, interleaving, and incremental in ML projects. The detailed sub-steps are described in the following Section IV.

IV. The Dregon In Action

This section shows how an ML dataset about the unpaid loan is constructed from the Case bank's Financial database by applying the Dregon ontology and process.

PKDD'99 Financial Database: The database contains records about banking services, such as Account (4,500 records), Transaction (1,053,620), Loan (682), Payment Order (6,471), and Credit cards (892) [22]. Among the loan records, 606 loans were paid off within the contract period, and 76 were not. Fig. 4.(b) shows the conceptual schema of the Financial database in the UML notation.

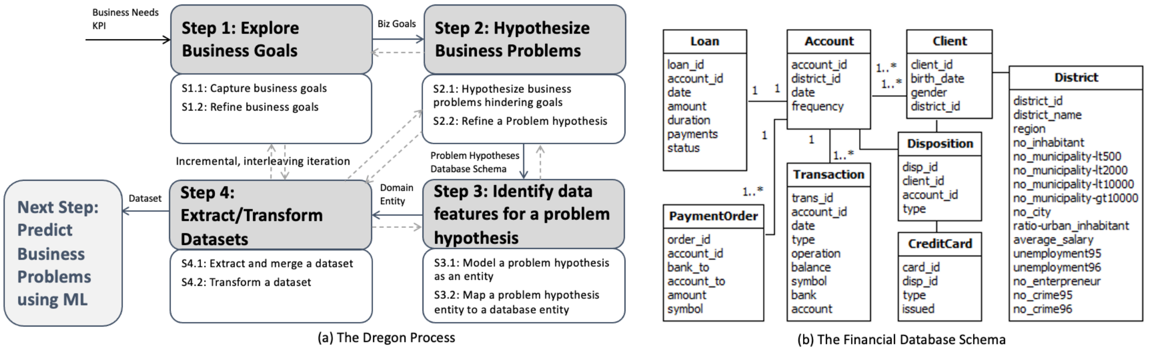
A. Step 1: Explore Business Goals

We begin Step 1, understanding and modeling the Case bank's goals, and then refining high-level goals into concrete and measurable goals.

1) Step 1.1: Capture the Case bank's goals: To better understand the relationships between the Case bank's goals and problems, we interview the bank manager and staff to understand and capture the Case bank's business goals and process. *Maximize revenue* [1] is captured as one of the bank's high-level goals and then is modeled as an NF softgoal, *Maximize revenue*$_{NFsoftgoal}$ to achieve, as shown in Fig. 5.(a).

2) Step 1.2: Refine the Case bank's goal: The modeled NF softgoal is AND-decomposed and operationalized by *Increase loan revenue*$_{OPsoftgoal}$ and *Increase fee revenue*$_{OPsoftgoal}$ as Operationalizing softgoals. The former is further AND-decomposed to more specific Operationalizing softgoals of *Increase personal loan revenue*$_{OPsoftgoal}$ and *Increase business loan revenue*$_{OPsoftgoal}$. The bank staff indicated during an interview that the personal loan revenue of this quarter is less than 5 percent for the Key Performance Indicator (KPI)

[1] The Dregon concept is expressed in the notation from [23] to show the modeling concepts in a class and an instance level.



(a) The Dregon Process

(b) The Financial Database Schema

Fig. 4. The data preparation process and the Financial database schema

[24]. This KPI indicates the unpaid loan is a problem hurting *Increase personal loan revenue*$_{OPsoftgoal}$.

B. Step 2: Hypothesize Business Problems Hindering Goals

In Step 2, we explore possible banking events leading to the unpaid loan and identify a testable factor of the problem hypothesis to validate.

1) Step 2.1: Hypothesize banking events hindering the Case bank's goal: We first model that a client's *Unpaid loan*$_{OPsoftproblem}$ $Breaks(--)$ the *Increase personal loan revenue*$_{OPsoftgoal}$. There could be many banking events related to the *Unpaid loan*$_{OPsoftproblem}$. To narrow the scope of business events to analyze, we then explore potential banking events that could positively contribute to the *Unpaid loan*$_{OPsoftproblem}$ and eventually hurt *Increase personal loan revenue*$_{OPsoftgoal}$. In other words, a goal and a problem are used as the context to search potential banking events.

After more understanding the loan process and analysis of the Financial database, we hypothesize that a client's *Poor Loan*$_{AbstractPH}$, *Abnormal Account Balance*$_{AbstractPH}$, and *Exceptional Transaction*$_{AbstractPH}$ might positively contribute to the *Unpaid loan*$_{OPsoftproblem}$ at an abstract level, as shown in Fig. 5.(a).

2) Step 2.2: Refine an abstract problem hypothesis into a testable problem hypothesis: The identified abstract problem hypothesis is further decomposed into a testable problem hypothesis that usually has a value of categorical or numeric type. For example, the *Balance of an Account*$_{AbstractPH}$ is OR-decomposed into the *Minimum balance of an Account*$_{TestablePH}$, *Average balance of an Account*$_{TestablePH}$, and *Maximum balance of an Account*$_{TestablePH}$ for the client's loan duration, which has a numeric balance.

Based on the goal and problem hypothesis graph, we can express one of the problem hypotheses in a conditional statement. Let PH1 be the problem hypothesis *If the minimum balance of an Account associated with a Loan is below a certain threshold, the status of Loan is likely to be unpayable for the loan duration.* Then, we can consider the *the minimum balance of an Account associated with a Loan is below a certain threshold*$_{SourcePH}$ as a source problem hypothesis (or an independent variable), *some positively contributes*$_{PHcontribution}$ as a contribution relationship, and an *status of Loan is likely to be unpayable for the loan duration*$_{TargetPH}$ as a target problem hypothesis (or a dependent variable).

$$\textit{Minimum balance of an account below a threshold}_{SourcePH} \xrightarrow{Some-plus_{PHcontribution}} \textit{Status of a loan unpayable for the loan duration}_{TargetPH} \quad (1)$$

C. Step 3: Identify Data Features for a Problem Hypothesis

We model a testable problem hypothesis as a problem hypothesis entity and map the entity to a database entity.

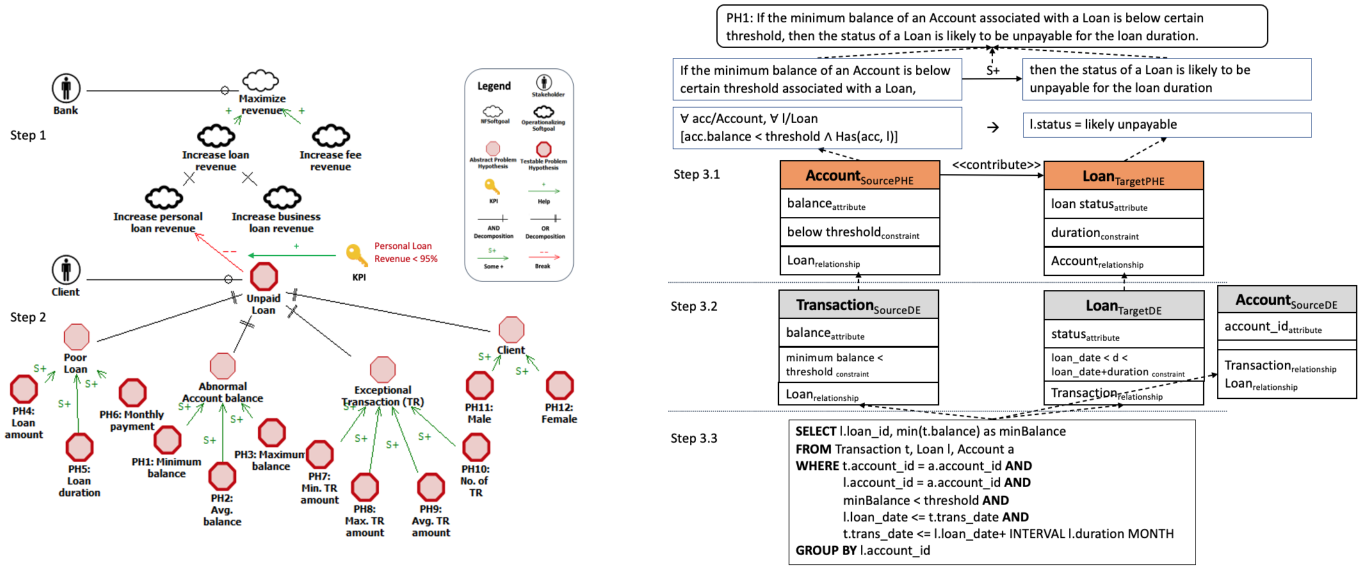
1) Step 3.1: Model a problem hypothesis as an entity: The elicited testable problem hypothesis is modeled as a problem hypothesis entity using the entity-relationship model [25] [26]. A problem hypothesis entity consists of attributes, constraints, and relationships. An *attribute* is a property of an entity having measurable value. A *constraint* is a condition restricting the value or state of a problem hypothesis. A *relationship* shows other entities associated with this entity.

For example, the *Minimum balance of an Account below a threshold*$_{SourcePH}$ in PH1 is modeled as a *Account*$_{SourcePHE}$, having *balance*$_{PHEattribute}$, *minimum balance, less than threshold*$_{PHEconstraint}$, and a *Loan*$_{PHErelationship}$. Similarly, the *Status of a loan unpayable for the loan duration*$_{TargetPH}$ is modeled as *Loan*$_{TargetPHE}$ having *status*$_{PHEattribute}$, *duration*$_{PHEconstraint}$, and *Account*$_{PHErelationship}$, as shown in Fig. 5.(b).

2) Step 3.2: Map a problem hypothesis entity to a database entity: The attribute of the problem hypothesis entity (PHE) may manually be mapped to attributes of the database entity (DE) considering the constraints and relationships of the PHE. To guide systematic mapping, we identified five types of mappings from a PHE to a DE, as shown in Fig. 6.

The first type of mapping is from a *target PHE* to a *target DE*. The attribute and constraints of the target PHE are mapped to those of the target DE, where the attribute of target DE becomes a target or classification label. For example, *loan status*$_{PHEattribute}$ of *Loan*$_{TargetPHE}$ in Fig. 5.(b) is mapped to the *Loan*$_{DE}$ and *status*$_{DEattribute}$.

The second type is from a *source PHE* to a *target DE*. Here, we can notice that the mapped entity is the same target DE in



```
SELECT l.loan_id, min(t.balance) as minBalance
FROM Transaction t, Loan l, Account a
WHERE t.account_id = a.account_id AND
        l.account_id = a.account_id AND
        minBalance < threshold AND
        l.loan_date <= t.trans_date AND
        t.trans_date <= l.loan_date+ INTERVAL l.duration MONTH
GROUP BY l.account_id
```

(a) Modeling the banking goals and hypotheses of unpaid loan

(b) Identifying data features for a problem hypothesis

Fig. 5. Applying the Dregon process for an unpaid loan

the first type of mapping, but the attribute of a target DE is not a target label.

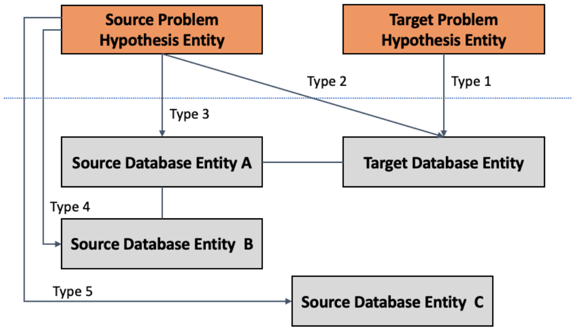
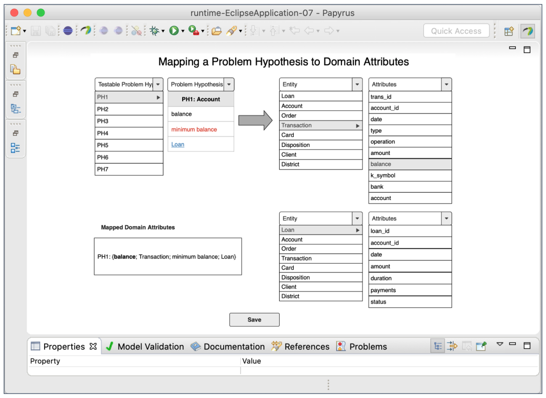


Fig. 6. Mapping types from a problem hypothesis entity into a database entity

The third type is from a *source PHE* to a *source DE*, where the source DE is directly associated with the target DE in the database schema. The attribute and constraints of the source PHE are mapped to those of source DE. The relationship of a source DE is the name of the target DE and vice versa.

The fourth type is from a *source PHE* to a *source DE* similar to the third type but the source DE is indirectly related to the target DE. In other words, there are other DEs between the source DE and the target DE. For example, for the $balance_{PHEattribute}$ of $Account_{SourcePHE}$ in Fig. 5.(b), we first select the Account entity of the database schema and check whether some attributes of the Account semantically match the $balance_{PHEattribute}$. If we could not find a relevant attribute of the Account, then we check the subsequent entities. While iterating domain entities, we could see a 'balance' attribute of the Transaction entity, representing a balance after the banking transaction. So, we mapped $Account_{PHE}$ to $Transaction_{DE}$ and $balance_{PHEattribute}$ to $balance_{DEattribute}$. As $Transaction_{DE}$ is not directly related with $Loan_{DE}$, we identify $Account_{DE}$ that is related with both $Loan_{DE}$ and $Transaction_{DE}$.



PH1: {**balance**; Transaction; minimum balance; Loan}

Fig. 7. Mapping a problem hypothesis entity into a database entity

This mapping may be streamlined with the Dregon prototype tool in Fig. 7. The tool first reads the Financial database schema and shows the concerned entity and attributes. Each entity may be selected and checked whether the entity's attributes are similar to that of the problem hypothesis entity.

D. Step 4: Extract and Transform an ML Dataset

In this step, we extract a dataset using the identified database entity and features, merge each dataset corresponding to the problem hypothesis, and transform the integrated dataset for ML processing.

1) Step 4.1: Extract and merge an ML dataset: The identified database entities corresponding to the source and target PHE are used to make a database query, as shown in Fig. 5.(b). For example, the data of *Minimum balance of an*

$\textit{Account below a threshold}_{SourcePH}$ in PH1 can be extracted using the identified $\textit{balance}_{DEattribute}$, $\textit{minimum balance} < \textit{threshold}_{DEconstraint}$, and $\textit{Loan, Account}_{DErelationship}$ in $\textit{Transaction}_{DE}$. SQL group function, min() may be used to select $\textit{minimum balance}_{DEconstraint}$. Also, to apply the relationship $\textit{Loan, Account}_{DErelationship}$, we need to identify a primary key and a foreign key relationship between $\textit{Loan}_{DE}$ and $\textit{Transaction}_{DE}$, which leads to identifying $\textit{Account}_{DE}$. The $\textit{loan duration}_{DEconstraint}$ of $\textit{Loan}_{DE}$ is also applied, as shown in Fig. 5.(b).

The data of $\textit{Unpaid Loan for the loan duration}_{TargetPH}$ can be extracted using the following SQL code, which needs to join Loan and Account tables.

```sql
SELECT l.loan_id, l.status
FROM Loan l, Account a
WHERE l.account_id = a.account_id
```

Each dataset for the hypothesized business events is extracted, tentatively stored in the database, and then integrated into one dataset. Those datasets are then merged into one dataset based on the loan status, as shown in Fig. 8.

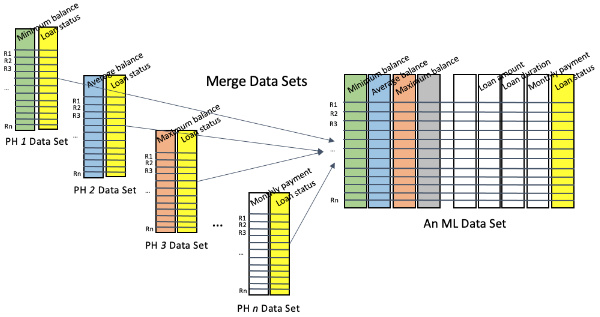


Fig. 8. Merging partial datasets into an ML dataset

2) *Step 4.2: Transform an ML dataset:* The merged dataset may need to be preprocessed for some feature, including filling in missing value, scaling feature values, converting categorical data to a numeric value, and others. The clean data are then entered into ML models. For example, we scaled the features of integrated data set using the data normalization method. We also used a one-hot encoding on the transaction type, mode, symbol features, and other nominal features.

V. Experimental Results

Three experiments were performed to see the strength and the weakness of the Dregon approach. In experiments 1 and 2, we prepared the ML dataset without the proposed approach, assuming all the features in the Financial database are potential banking events that could cause the unpaid loan. In experiment 3, we prepared the dataset for the validation of banking events towards the unpaid loan, following the Dregon process.

A. Experiment 1

For this experiment, we assumed all the attributes, except the table identifiers, of the entities in the Financial database schema as potential events causing unpaid loans without a goal and problem analysis, and selected the loan status as a target feature. The prepared ML dataset included 72 features with some transformation methods, such as hot encoding for the nominal features and 449,736 records based on the transaction id. The significant records are due to the $join$ operation among Account, Transaction, and Payment Order tables.

As some ML algorithms, such as Gradient Boosting Tree, provide feature importance [27], [28], we analyzed whether some features could be important factors towards the unpaid loan. Fig. 9 shows some crucial features predicted by the XGBoost model. However, it was not easy to get some ideas about whether the loan granted year and the credit card type, e.g., 'classic,' has some relationships towards the unpaid loan.

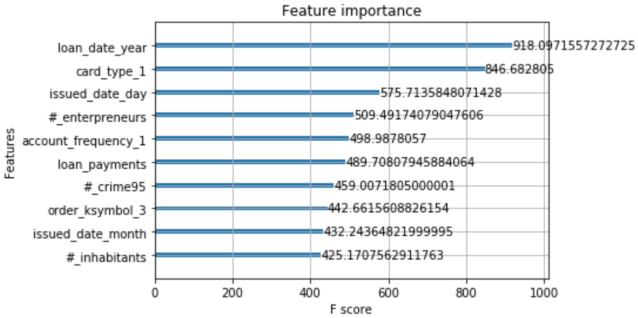


Fig. 9. Top important features in experiment 1

One critical issue of this approach is that one ML model, e.g., XGBoost, showed different prediction results for the same loan instance. For example, different transaction records, having the same Loan ID 233, showed different loan prediction results (i.e., paid and unpaid), which made the dataset poor in identifying a banking event for the unpaid loan.

Another issue is that this experiment included some unlikely features, such as no. of committed crimes '95. It was not easy to understand whether the no. of committed crimes is related to clients' loan payments as the feature is highly related to the community behavior, not a client's banking behavior.

B. Experiment 2

In experiment 2, we also assumed all the attributes in the database as potential problems without considering steps 1 and 2 of the Dregon process. However, we prepared the ML dataset centered on the loan ID to prevent duplicate data values of a loan record, unlike experiment 1. We used SQL group functions, such as Sum, Min, and Avg, to select records for the one-to-many relationships, for example, the relationship between Account and Transactions. The final dataset contained 682 records, including 72 features. Fig. 10 shows important features the Random Forest model provided, although it was challenging to understand whether these features positively contribute to the loan status.

A critical issue of this approach is that the prepared dataset did not consider some boundary constraint of the loan. For example, the loan duration of loan ID 1 is two years from 1993, but the dataset included records of 1996 and 1997, which

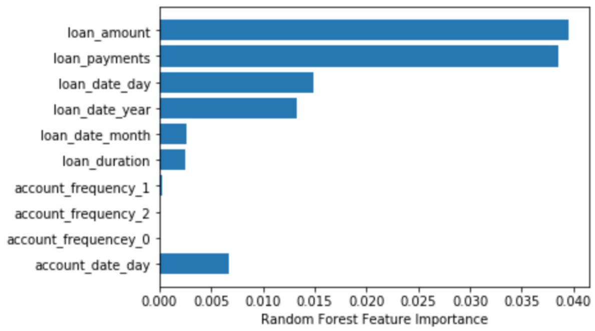


Fig. 10. Top important features in experiment 2

could violate the time order constraint between the source and target problem hypothesis and then give incorrect predictions leading to ineffective problem validation. The constraint of time order is essential in identifying a cause and effect relationship between banking events, but difficult to enforce this constraint in this experiment without some mechanisms.

C. Experiment 3

In this experiment 3, the Dregon approach was applied to prepare an ML dataset to validate business events behind the unpaid loan. The banking events were hypothesized as four groups, including Loan, Account, Transaction, and Client, as shown in Fig. 5.(a).

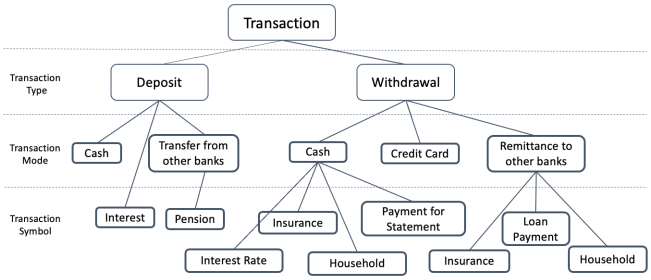


Fig. 11. Deposit and withdrawal classification in Transaction

While preparing a dataset, we could discover that the balance depends on the transaction *type* (deposit or withdrawal), *operation* (mode of a transaction), and *symbol* (characterization of the transaction) features in the Transaction entity. We could organize the structure of deposit and withdrawal transactions and analyze these features, as shown in Fig. 11, to get insights into transaction impact [29]. We then hypothesized deposit and withdrawal of transactions leading to the balance change. In a usual ML approach, these category features would be hot-encoded, like in experiments 1 and 2.

Based on the modeled problem hypotheses, six hundred eighty-two (682) loan records with 25 features were prepared. We then ran ML models to predict whether each loan could be paid off or not. Fig. 12 shows performance results for some ML models with the constructed dataset. The accuracy of ML models was overall satisfactory, and XGBoost gave the highest accuracy (0.91). We could also identify significant features corresponding to some problem hypotheses towards the unpaid loan.

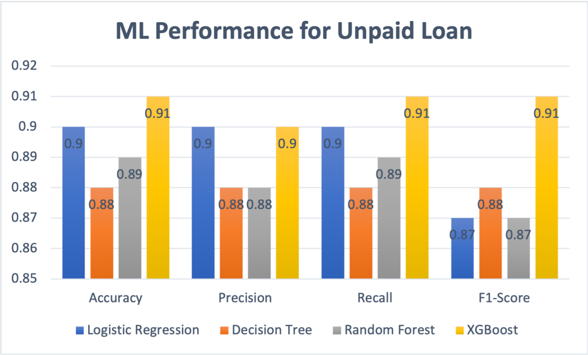
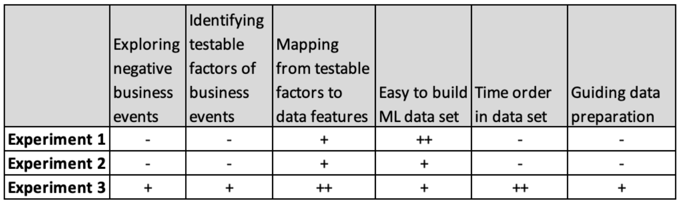


Fig. 12. ML performance using the prepared dataset in experiment 3

The trade-off analysis for the experiments is shown in Fig. 13. Experiment 1 is easier to prepare an ML dataset assuming all the features as problem hypotheses. However, its results are not easy to understand, even giving different predictions for the same loan case, thus not trustworthy. Experiment 2 shows more sensible results than experiment 1 but still challenging to understand the relationships among the problem hypotheses and a target label. It needs to apply some systematic process for asserting constraints of time order. Experiment 3 provides more sensible and understandable relationships among the banking events and an unpaid loan with fewer features than experiments 1 and 2. Although experiment 3 may take some time to prepare data, it helps identify potential banking events causing the unpaid loan and get some insights into the hypothesized banking events. In addition, it helps understand some implicit patterns of the data features otherwise overlooked.



	Exploring negative business events	Identifying testable factors of business events	Mapping from testable factors to data features	Easy to build ML data set	Time order in data set	Guiding data preparation
Experiment 1	-	-	+	++	-	-
Experiment 2	-	-	+	+	-	-
Experiment 3	+	+	++	+	++	+

Fig. 13. Experiments comparison of data preparation

VI. Discussion and Observation

In constructing a problem hypothesis concerning the unpaid loan, it may not be easy to keep the time constraint between a source and a target problem hypothesis. To prevent the violation of this constraint, we used a date feature and potential banking events together to ensure the time order constraint between a source and a target problem hypothesis.

Some problem hypotheses may not be mapped to data features of the database schema, such as the fifth type mapping in Fig. 6, due to unmatched data features or type and cannot be validated. In that case, the data for the problem hypothesis may need to be acquired from external data sources [30].

Our data preparation approach may be applied to identify potential business problems in other business domains, such as logistics, telecommunication, or healthcare. However, as the data preparation in this empirical study is the first attempt and ML performance depends on ML algorithms, their parameters, data characteristics, and others, more empirical studies are needed to show the usefulness of our approach.

Limitations Problem hypotheses are conceived and manually constructed, which tends to be error-prone and ineffective in managing different hypotheses. Some guiding template or tool support may help refine a problem hypothesis into a source and target problem hypothesis and a relationship. The mapping process between a problem hypothesis entity and a database entity is partially supported with a prototype mapping tool. However, the tool needs more work to automate the presented approach. The process also needs to be fully formalized to define precise semantics.

VII. Conclusion and Future Work

This paper has presented a goal-oriented ML data preparation approach to support the validation of a business problem. Starting with modeling business goals, we explored potential business events against goals, modeled the events as a testable problem hypothesis entity, identified data attributes along with constraints and relationships, and built an ML dataset from a source database. Specifically, this paper presented 1) a domain-independent ontology and a process for guiding the preparation of an ML dataset, 2) a method to capturing potential business events in the context of goals and problems, 3) a modeling method of a problem hypothesis entity to help to identify a testable factor, constraints, and relationships of the captured business event, and 4) a mapping method and mapping types from a problem hypothesis entity to a database entity. The experiment, we feel, shows that our approach helps prepare an appropriate ML dataset, enforce time order constraints, and provide traceability from problem hypotheses to data features.

There are several lines of future work. Tool design and support, such as a template, helping to manage a problem hypothesis are needed. Formalization of the mapping process is planned using first-order logic and the development of a fully-fledged tool also would be helpful to automate the mapping between a problem hypothesis entity and a database entity. We also plan to apply the Dregon approach to other domains, such as the public health domain, to see the strength and weaknesses of our work.

References

[1] D. Pyle, *Data preparation for data mining*. Morgan Kaufmann, 1999.

[2] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*. Springer, 2015, vol. 72.

[3] J. Brownlee, *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python*. Machine Learning Mastery, 2020.

[4] S. LaValle, E. Lesser, R. Shockley, M. Hopkins, and N. Kruschwitz, "Big data, analytics and the path from insights to value," *MIT sloan m. review*, vol. 52, pp. 21–32, 2011.

[5] T. H. Davenport and R. Bean, "Big data and ai executive survey (2020)," *NewVantage Partners (NVP), Tech. Rep*, 2020.

[6] S. Supakkul, L. Zhao, and L. Chung, "GOMA: Supporting Big data analytics with a goal-oriented approach," in *2016 IEEE International Congress on Big Data (BigData Congress)*, June 2016, pp. 149–156.

[7] S. Supakkul, R. Ahn, R. J. Gonçalves, D. Villarreal, L. Zhao, T. Hill, and L. Chung, "Validating goal-oriented hypotheses of business problems using machine learning," in *Int. Conf. on Big Data*. Springer, 2020.

[8] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol. 1. Springer-Verlag London, UK, 2000.

[9] R. Lukyanenko, A. Castellanos, J. Parsons, M. C. Tremblay, and V. C. Storey, "Using conceptual modeling to support machine learning," in *International Conference on Advanced Information Systems Engineering*. Springer, 2019, pp. 170–181.

[10] K. Ishikawa, *Introduction to quality control*. Productivity Press, 1990.

[11] B. Vesely, "Fault tree analysis (fta): Concepts and applications," *NASA HQ*, 2002.

[12] M. Jackson, *Problem frames: analysing and structuring software development problems*. Addison-Wesley, 2001.

[13] S. Supakkul and L. Chung, "Extending problem frames to deal with stakeholder problems: An agent-and goal-oriented approach," in *Proceedings of the 2009 ACM symposium on Applied Computing*, 2009, pp. 389–394.

[14] J. Luengo, S. García, and F. Herrera, "On the choice of the best imputation methods for missing values considering three groups of classification methods," *Knowledge and information systems*, vol. 32, no. 1, pp. 77–108, 2012.

[15] J. A. Sáez, J. Luengo, and F. Herrera, "Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification," *Pattern Recognition*, vol. 46, no. 1, pp. 355–364, 2013.

[16] H. Liu and H. Motoda, *Computational methods of feature selection*. CRC Press, 2007.

[17] K. Yu, L. Liu, and J. Li, "A unified view of causal and non-causal feature selection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 4, pp. 1–46, 2021.

[18] M. J. Berry and G. S. Linoff, *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 2004.

[19] J. Mylopoulos, L. Chung, and B. Nixon, "Representing and using nonfunctional requirements: A process-oriented approach," *IEEE Transactions on software engineering*, vol. 18, no. 6, pp. 483–497, 1992.

[20] P. Norving and S. Russell, *Artificial intelligence: a modern approach, Global Edition*. Pearson Education Limited, 2021.

[21] J. Pearl and T. S. Verma, "A theory of inferred causation," in *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1995, vol. 134, pp. 789–811.

[22] P. Berka and M. Sochorova, "Discovery challenge guide to the financial data set, pkdd-99," 1999.

[23] C. Rolland, C. Souveyet, and C. B. Achour, "Guiding goal modeling using scenarios," *IEEE transactions on software engineering*, vol. 24, no. 12, pp. 1055–1071, 1998.

[24] H.-Y. Wu, G.-H. Tzeng, and Y.-H. Chen, "A fuzzy MCDM approach for evaluating banking performance based on balanced scorecard," *Expert systems with applications*, vol. 36, no. 6, pp. 10 135–10 147, 2009.

[25] P. P.-S. Chen, "The entity-relationship model—toward a unified view of data," *ACM Transactions on Database Systems (TODS)*, vol. 1, no. 1, pp. 9–36, 1976.

[26] B. Carlo, S. Ceri, and N. Sham, *Conceptual Database Design: An Entity-Relationship Approach*. Benjamin/Cummings, 1992.

[27] M. Binkhonain and L. Zhao, "A review of machine learning algorithms for identification and classification of non-functional requirements," *Expert Systems with Applications: X*, vol. 1, p. 100001, 2019.

[28] C. Molnar, *Interpretable machine learning*. Lulu. com, 2020.

[29] B. Rajagopalan and M. W. Isken, "Exploiting data preparation to enhance mining and knowledge discovery," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 31, no. 4, pp. 460–467, 2001.

[30] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data lifecycle challenges in production machine learning: a survey," *ACM SIGMOD Record*, vol. 47, no. 2, pp. 17–28, 2018.