Validating Business Problem Hypotheses: A Goal-Oriented and Machine Learning-Based Approach

Validating Business Problem Hypotheses: A Goal-Oriented and Machine Learning-Based Approach

Robert Ahn[1], Sam Supakkul[2], Liping Zhao[3]
, Kirthy Kolluri[1], Tom Hill[4], and Lawrence Chung[1]

[1] University of Texas at Dallas, Richardson, Texas, USA.
{robert.sungsoo.ahn,kirthy.kolluri,chung}@utdallas.edu
[2] NCR Corporation, Atlanta, Georgia, USA. sam.supakkul@ncr.com
[3] University of Manchester, Manchester, UK. liping.zhao@manchester.ac.uk
[4] Fellows Consulting Group, Dallas, Texas, USA. tom.hill.fellow@gmail.com

Abstract. Validating an elicited problem to hinder a business goal is often more important than finding solutions in general. For example, validating the impact of a client's account balance toward an unpaid loan would be critical as a bank can take some actions to mitigate the problem. However, business organizations face difficulties confirming whether some business events or phenomena are causing a problem against a business goal. Some challenges to validate a problem are identifying testable factors for the identified problem, preparing data to validate, analyzing relationships between the factors and a problem, and reasoning the relationships towards high-level problems. Information systems developed to solve unconfirmed problems frequently tackle an erroneous problem, leading to some dissatisfying systems, consequently not achieving business goals. This paper proposes a goal-oriented and machine learning-based approach, *Gomphy*, for validating a business problem. The Gomphy presents an ontology and a process, a problem-related entity modeling method to identify relevant data features, a data preparation method, and an evaluation method of a problem for high-level problems. To illustrate our approach, we have validated problems behind an unpaid loan in one bank as an empirical study. We feel that at least the proposed approach helps validate business events negatively contributing to a goal, giving some insights about the validated problem.

1 Introduction

The assertion that *"A problem unstated is a problem unsolved"* expresses the importance of eliciting business needs and problems [1]. Understanding and validating a business problem likely to hinder a business goal is often more critical than developing solutions as it helps define system boundaries in the early phase of requirements engineering [2]. If the correct problems are validated first, a business can save precious time and cost to deal with erroneous problems [3].

However, business organizations face difficulties confirming whether an elicited business problem contributes to, how much degree, other high-level problems [4, 5]. Specifically, some challenging work might be identifying testable factors for the elicited problem, constructing a data set to test, and determining whether the identified problem has some relationships and how many degrees towards the high-level problem [6, 7]. Developing an information system with unconfirmed problems frequently leads to a system that is not useful enough to achieve business goals or is required to redevelop, costing valuable business resources.

Drawing on our previous work, GOMA [8] and Metis [9], this paper presents the *GOMPHY*, a Goal-Oriented and Machine learning-based approach using a Problem HYpothesis, to help validate business problems [10, 11]. Four technical contributions are made in this paper. Firstly, an ontology for modeling and validating a problem hypothesis is described. Secondly, a problem hypothesis-based entity modeling method is presented to help identify an entity, attributes, constraints, and relationships for a problem hypothesis. Thirdly, a mapping method is described from a problem hypothesis entity to a domain data feature in a source data model. Fourthly, an evaluation method for a problem contribution using ML explainability is elaborated to help understand a problem contribution towards a high-level problem.

This paper applies the proposed Gomphy approach to explore hypothesized business events behind an unpaid loan problem in one bank and validate the problem hypotheses towards the unpaid loan as an empirical study. Fig. 1 shows a high-level context diagram for the unpaid loan problem. We use the PKDD'99 Financial database [12] to represent data that the bank may have collected.

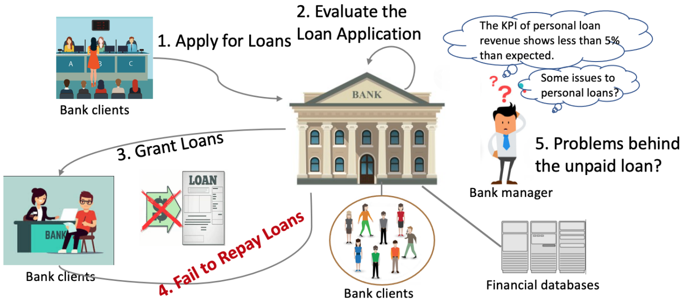

1. Apply for Loans
Bank clients
2. Evaluate the Loan Application
The KPI of personal loan revenue shows less than 5% than expected.
Some issues to personal loans?
BANK
5. Problems behind the unpaid loan?
Bank manager
3. Grant Loans
LOAN
Bank clients
4. Fail to Repay Loans
Bank clients
Financial databases

Fig. 1. Unpaid Loan Problem in a Bank

The rest of this paper is structured as follows. Section 2 presents the Gomphy approach, and Section 3 illustrates the Gomphy process in detail with an unpaid loan problem. Next, Section 4 describes three experiments performed, and Section 5 discusses related work, observations, and limitations. Finally, Section 6 summarizes the paper and future work.

2 The Gomphy Approach

The Gomphy approach, aiming to help validate business problems, consists of a domain-independent ontology, a series of steps based on goal orientation (GO) and Machine Learning (ML).

2.1 The Gomphy Ontology

The ontology consists of essential modeling concepts, relationships among modeling concepts, and constraints among the concepts and relationships, as shown in Fig. 2, where boxes and arrows represent the concepts and relationships.

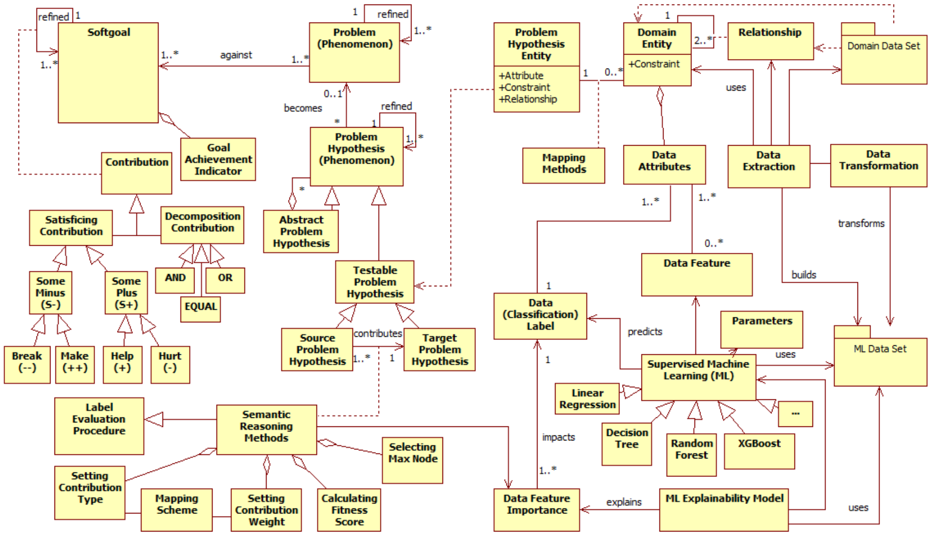


Fig. 2. The Gomphy Ontology for Validating Problem Hypotheses

Some essential concepts of Gomphy ontology are introduced. A *(Soft-)Goal* is defined as a goal that may not have a clear-cut criterion and a *(Soft-)Problem* as a phenomenon against a Goal. A *Problem Hypothesis* is a hypothesis that we believe a phenomenon is against a Goal. There are two kinds of Problem Hypothesis, an *Abstract Problem Hypothesis* and a *Testable Problem Hypothesis*. An Abstract Problem Hypothesis is conceptual, whereas a Testable Problem Hypothesis is testable using ML. A Testable Problem Hypothesis may be further refined, forming a *Source Problem Hypothesis* and a *Target Problem Hypothesis*.

A *Problem Hypothesis Entity* captures a Testable Problem Hypothesis, which is modeled using an entity-relationship model. A Problem Hypothesis Entity is mapped to relevant *Domain Entity*, *Domain Attributes*, *Domain Constraints*, and *Domain Relationships* in a source data model. The selected Domain Attributes are used to build an ML data set consisting of *Data Features* and a *Classification Label*.

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 [13]. The relationships between Problem Hypotheses and Problems are either *Validated* or *Invalidated*.

One crucial constraint about a problem hypothesis includes time-order among a target and a source 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 should be reasonably sensible [14].

2.2 The Gomphy Process

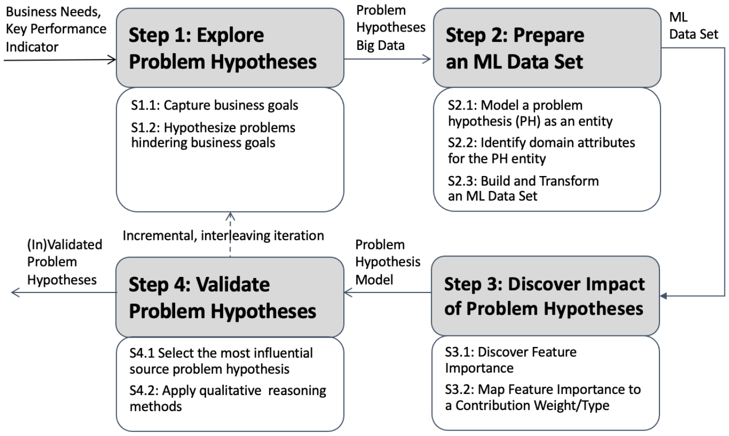

Business Needs, Key Performance Indicator
Step 1: Explore Problem Hypotheses
S1.1: Capture business goals
S1.2: Hypothesize problems hindering business goals
Problem Hypotheses Big Data
Step 2: Prepare an ML Data Set
S2.1: Model a problem hypothesis (PH) as an entity
S2.2: Identify domain attributes for the PH entity
S2.3: Build and Transform an ML Data Set
ML Data Set
Step 3: Discover Impact of Problem Hypotheses
S3.1: Discover Feature Importance
S3.2: Map Feature Importance to a Contribution Weight/Type
Problem Hypothesis Model
Step 4: Validate Problem Hypotheses
S4.1 Select the most influential source problem hypothesis
S4.2: Apply qualitative reasoning methods
Incremental, interleaving iteration
(In)Validated Problem Hypotheses

Fig. 3. The Gomphy process

The Gomphy process, shown in Fig. 3, intends to help guide the validation of a problem hypothesis, providing traceability among a goal, a problem, a data set, and ML. The process consists of four steps but should be understood as iterative, interleaving, and incremental in ML projects. The sub-steps of each step are described in detail in the following Section 3.

3 The Gomphy In Action

We suppose a hypothetical bank, the Case bank, offering client services, such as opening accounts, offering loans, and issuing credit cards. The bank has experienced an unpaid loan problem, where some clients failed to pay, when due, loan payments. However, it did not know what specific clients' banking behaviors were behind this issue. Since this is a hypothetical example, we used the PKDD'99 Financial database to represent data the bank may have collected [12].

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) issued to clients. Among the loan records, 606 loans were paid off within the contract period, and 76 were not. The Gomphy process is illustrated with the unpaid loan problem.

3.1 Step 1: Explore the Case Bank's Problem Hypotheses

Requirements engineers begin Step 1, understanding the banking domain, capturing and modeling the Case bank's goals. Potential problems are then hypothesized that could hinder Case bank's goals.

Step 1.1 Capture the Case bank's goals After understanding the bank domain, one of the bank's goals, *Maximize revenue*$_{NFsoftgoal}$ [1] is captured as an NF (Non-Functional) softgoal to achieve at the top organizational level, which is AND-decomposed and operationalized by *Increase loan revenue*$_{OPsoftgoal}$ and *Increase fee revenue*$_{OPsoftgoal}$ as operationalizing softgoals, as shown in Fig. 4. The former is further AND-decomposed to more specific softgoals of *Increase personal loan revenue*$_{OPsoftgoal}$ and *Increase business loan revenue*$_{OPsoftgoal}$.

Here, 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) they intended to achieve [16] due to some clients' unpaid loans. So, the bank wanted to know which specific banking events of a client contribute to the unpaid loan causing the decrease of personal loan revenue. However, it was not easy for the bank staff to pinpoint the main causes of this problem.

Step 1.2: Hypothesize problems hindering the Case bank's goal We modeled that a client's *Unpaid loan*$_{OPsoftproblem}$ Breaks (–) the *Increase personal loan revenue*$_{OPsoftgoal}$. After understanding the loan process and analysis of the Financial database, we explored potential clients' banking behaviors against the unpaid loan. We hypothesized that a client's *Loan*$_{AbstractPH}$, *Account Balance*$_{AbstractPH}$, and *Transaction*$_{AbstractPH}$ might positively contribute to the *Unpaid loan*$_{OPsoftproblem}$.

An abstract problem hypothesis is further decomposed into a testable problem hypothesis. For example, *Balance of an Account*$_{AbstractPH}$ is divided into *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 using an OR-decomposition method.

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 *The minimum balance of an Account some positively contributes to an*

[1] The Gomphy concept is expressed in the notation from [15].

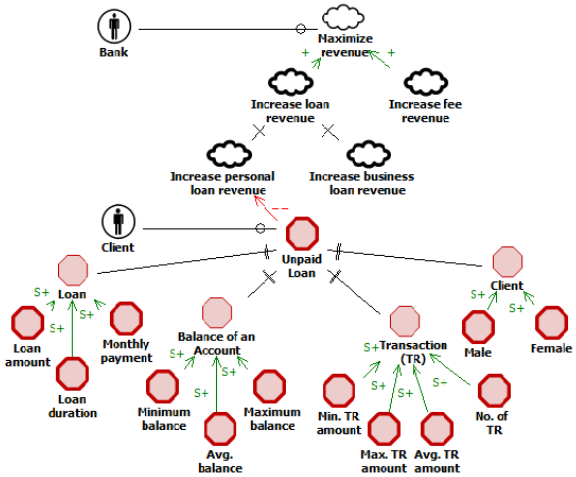


Fig. 4. Hypothesizing Problems for Unpaid Loan and Preparing an ML Data Set

unpaid loan in the loan duration$_{PH}$. Then, we can consider *Minimum balance of an Account*$_{SourcePH}$ as a source problem hypothesis (or an independent variable), *Some positively contributes*$_{PHcontribution}$ as a contribution relationship, and *Unpaid loan in the loan duration*$_{TargetPH}$ as a target problem hypothesis (or a dependent variable).

$$\begin{aligned} &\textit{Minimum balance of an Account}_{SourcePH} \\ \xrightarrow{Some-plus_{PHcontribution}}\ &\textit{Unpaid loan in the loan duration}_{TargetPH} \end{aligned} \tag{1}$$

3.2 Step 2: Prepare an ML Data Set

This step models the testable problem hypothesis as a problem hypothesis entity, identifies data attributes in the database, and constructs an ML data set.

Step 2.1: Model a problem hypothesis as an entity Considering captured goals and problem hypotheses in Step 1, we first model the elicited testable problem hypothesis as an entity using the entity-relationship model [17]. A problem hypothesis entity has attributes, constraints, and relationships. An *attribute* is a particular property of an entity, providing measurement value. A *constraint* is a condition restricting the value or state of a problem hypothesis. A *relationship* shows other entities associated with this entity.

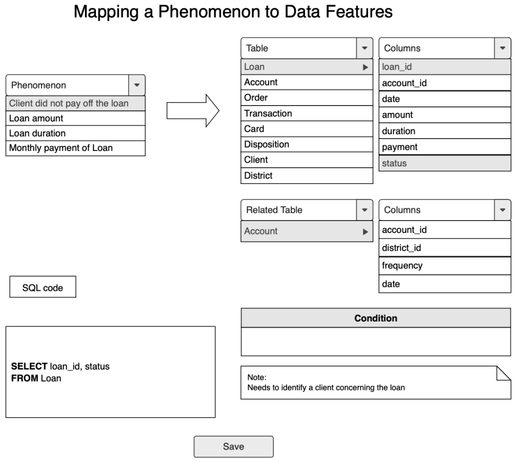
For example, the *Minimum balance of an Account*$_{SourcePH}$ in PH1 is modeled as a problem hypothesis entity of *Account*$_{PHE}$, having an attribute of *balance*$_{PHEattribute}$, a constraint of a *minimum balance*$_{PHEconstraint}$, and a relationship of a *Loan*$_{PHErelationship}$, as shown in Fig. 4.(a). Similarly, an *Un-*

paid Loan in the loan $duration_{TargetPH}$, is modeled as $Loan_{PHE}$, *loan* $status_{PHEattribute}$, *loan* $duration_{PHEconstraint}$, and $Account_{PHErelationship}$.

.

Step 2.2: Map an attribute of a problem hypothesis entity to domain attributes The attribute in the problem hypothesis entity may manually be mapped to domain attributes in the domain entity with tool support, as shown in Fig. 5. The tool first reads the database schema and shows the concerned domain entity and attributes. We then select a domain entity and check whether domain attributes are similar to the attributes of the problem hypothesis entity.

For example, for $balance_{PHEattribute}$ of $Account_{PHE}$, we first select Account domain entity and check whether domain attributes are semantically matching or similar to the $balance_{PHEattribute}$, as shown in Fig. 5. As we could not find a relevant attribute in the Account domain entity, we check the following entities. While iterating domain entities, we could find a 'balance' attribute in the Transaction entity, representing a balance after the banking transaction. So, we mapped $Account_{PHE}$ to the domain entity of $Transaction_{DE}$ and $balance_{PHEattribute}$ to the domain attribute of $balance_{DEattribute}$. The constraint and relationships of a problem hypothesis entity are similarly mapped to those of a domain entity.



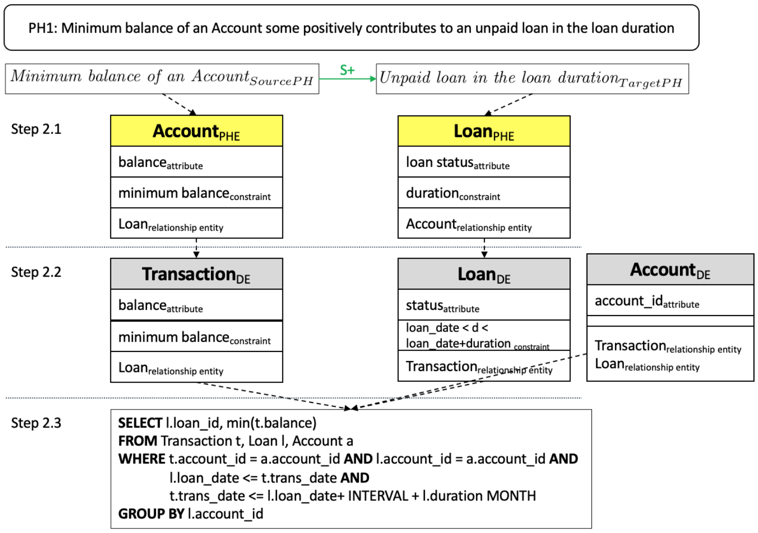
```
SELECT loan_id, status
FROM Loan
```

Note:
Needs to identify a client concerning the loan

Fig. 5. Mapping Attributes between Problem Hypothesis Entity and Domain Entity

Step 2.3: Extract and Transform a Data Set The identified domain attributes, constraints, and relationships corresponding to the source and target problem hypothesis entity are used to make a database query.

For example, the data of the $\textit{Minimum balance of an Account}_{SourcePH}$ can be extracted using the identified $\textit{balance}_{DEattribute}$, and $\textit{minimum balance}_{DEconstraint}$ in $\textit{Transaction}_{DE}$. SQL group function, min() may be used to select $\textit{minimum balance}_{DEconstraint}$. Also, to apply the relationship $\textit{Loan}_{DErelationship}$, we need to identify a primary key and a foreign key relationship between $\textit{Loan}_{DE}$ entity and $\textit{Transaction}_{DE}$. The $\textit{loan duration}_{DEconstraint}$ of $\textit{Loan}_{DE}$ is also applied.


PH1: Minimum balance of an Account some positively contributes to an unpaid loan in the loan duration
$\textit{Minimum balance of an Account}_{SourcePH}$
S+
$\textit{Unpaid loan in the loan duration}_{TargetPH}$
Step 2.1
$\text{Account}_{\text{PHE}}$
$\text{balance}_{\text{attribute}}$
$\text{minimum balance}_{\text{constraint}}$
$\text{Loan}_{\text{relationship entity}}$
Loan_{PHE}
$\text{loan status}_{\text{attribute}}$
$\text{duration}_{\text{constraint}}$
$\text{Account}_{\text{relationship entity}}$
Step 2.2
$\text{Transaction}_{\text{DE}}$
$\text{balance}_{\text{attribute}}$
$\text{minimum balance}_{\text{constraint}}$
$\text{Loan}_{\text{relationship entity}}$
Loan_{DE}
$\text{status}_{\text{attribute}}$
loan_date < d < loan_date+duration $_{\text{constraint}}$
$\text{Transaction}_{\text{relationship entity}}$
$\text{Account}_{\text{DE}}$
$\text{account\_id}_{\text{attribute}}$
$\text{Transaction}_{\text{relationship entity}}$
$\text{Loan}_{\text{relationship entity}}$
Step 2.3

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

Fig. 6. Step 2: Preparing an ML Data Set for Problem Hypothesis 1

The resulting data set for each source or target problem hypothesis is tentatively be stored in the database and then integrated into one data set for ML processing. After we get a data set, we may need to transform some features depending on feature value. The transformation may include scaling feature value, transforming categorical data to a numeric value, and others.

3.3 Step 3: Discover Impact of Problem Hypotheses using ML

The impact of a problem hypothesis towards the unpaid loan is uncovered using Supervised ML models and ML Explainability model, which decodes hidden feature patterns in the data set.

Step 3.1: Discover Feature Importance Using a Supervised ML and an ML Explainability model In this step, Supervised ML models with the domain features and data set are run to predict a loan [18, 19]. An ML Explainability model is then utilized to interpret features impacting the loan prediction and detect important features corresponding to clients' banking events.

First, four Supervised ML models, such as Linear Regression, Decision Tree, Random Forest, and XGBoost (eXtreme Gradient Boosting), were built with identified domain features and the data set. The ML models then predicted the loan instances as ‘Paid Loan’ or ‘Unpaid Loan.’ The accuracy of each ML model was 0.8892 (Logistic Regression), 08824 (Decision Tree), 0.8938 (Random Forest), and 0.9115 (XGBoost).

Next, we utilized the SHAP (Shapley Additive exPlanations) model to get an intuitive and consistent feature value among ML Explainability models [20]. The more accurate an ML model, the more we can get confidence about feature importance value. The XGBoost model was given as input to the SHAP model.

To analyze the feature importance for prediction results, we first collected predicted instances of unpaid loans. Fig. 7 shows the SHAP value of some important features for one case. After that, we summed up the feature values of all the unpaid loans to detect the feature impact of all unpaid loans.

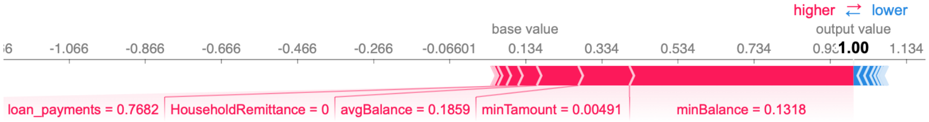


Fig. 7. Feature importance for one unpaid loaner

Step 3.2: Update a Contribution Weight and Type with Feature Importance in a Problem Hypothesis Model The collected feature importance value can be considered a contribution weight towards the target feature, i.e., unpaid loan. The contribution weight and hypothesized Contribution type of each leaf-level problem hypothesis are updated based on the detected feature importance value using Formula 2 and 3, as shown in Fig. 8.

$$weight(P_s, P_t) = I_{s,t} \tag{2}$$

$$ctr\_type(I_{s,t}) = \begin{cases} \text{S+} & \text{if } I_{s,t} \geq 0 \\ \text{S-} & \text{if } I_{s,t} < 0 \end{cases} \tag{3}$$

For example, the Contribution weight and type of the leaf node, a minimum balance, are updated with the value ’15.32’ and ’S+’. Similarly, the contribution weight and type of other leaf nodes are updated accordingly.

Next, in order to know the direct and indirect impact of leaf-level problem hypotheses towards a high-level problem in the problem hypothesis model, we first calculate the fitness score of a source problem hypothesis using Formula 4.

$$score(P_s) = \left(\sum_{t=1}^{\#targets} weight(P_t) \times weight(P_s, P_t) \right) \tag{4}$$

We assume that the weight of each problem hypothesis is 0.2 and adopting a weight-based quantitative selection pattern [21]. For example, the fitness score of $Minimum\ balance\ of\ an\ Account_{SourcePH}$ is calculated as (0.2 * 15.32 =) 3.064.

3.4 Step 4: Validate Problem Hypotheses

This step selects the most critical problem hypothesis as a validated problem hypothesis among many alternative hypotheses and evaluates the impact of the validated problem on other high-level problems in a problem hypothesis model, as shown in Fig. 8.

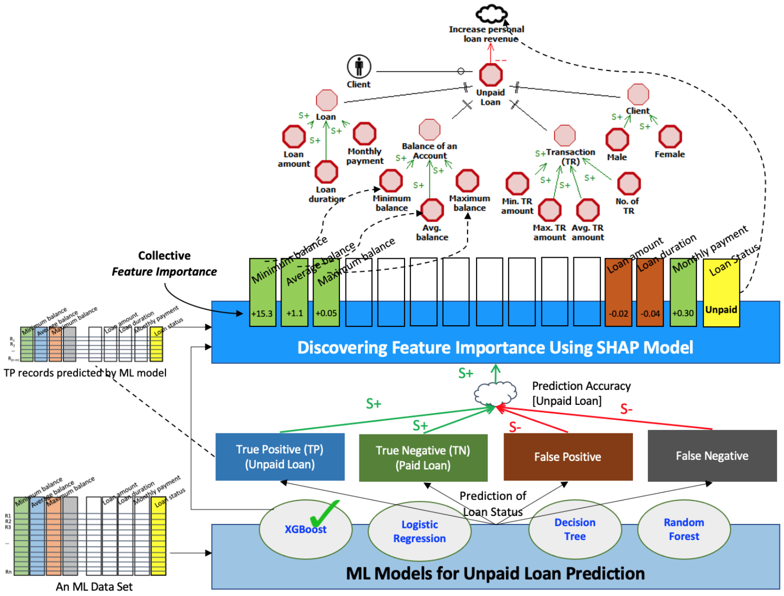


Fig. 8. Step 4: Validating a Problem Hypothesis

Step 4.1: Select the most influential source problem hypothesis Among the alternative problem hypotheses contributing to the target problem in the problem hypothesis model, we select a problem hypothesis having the highest fitness score in the leaf nodes. Banking staff may give a qualitative priority for some problem hypotheses, depending on some schemes, such as 'normal', 'critical', or 'very critical'. Here, we assume a 'normal' priority of problem hypothesis.

For example, the $Minimum\ balance\ of\ an\ Account_{SourcePH}$ in the problem hypothesis model was selected by Formula 5 as it has the highest fitness score among the leaf problem hypotheses under *Loan*.

$$selection(P_t) = max\Big(score(P_s)\Big)_{s=1}^{\#sources} \tag{5}$$

The selected problem hypothesis is considered a validated problem hypothesis by Formula 6, as it is most likely to be the cause for the target problem hypothesis. It means the *Minimum balance of an Account*$_{SourcePH}$ is likely to be the most important cause of *Low balance.*

$$validated(selection(P_i)) \rightarrow validated(P_i) \tag{6}$$

Step 4.2: Apply qualitative reasoning methods to reason the validation impact towards a high-level problem Once the most likely problem hypothesis is validated, as shown by 'check mark' in Fig. 8, qualitative reasoning, e.g., the label propagation procedure [13], is carried out to determine the validated problem's impact upward a problem.

In the goal and problem hypothesis model, if the *Minimum balance of an Account*$_{SourcePH}$ and *Some positively to contribute*$_{PHcontribution}$ are satisficed, then the *Balance of an Account* is satisficed or check marked. The reasoning propagation shows that the *Balance of an Account*$_{SourcePH}$ positively contributes to *Unpaid Loan*, which *Breaks* the goal, *Increase personal loan revenue.*

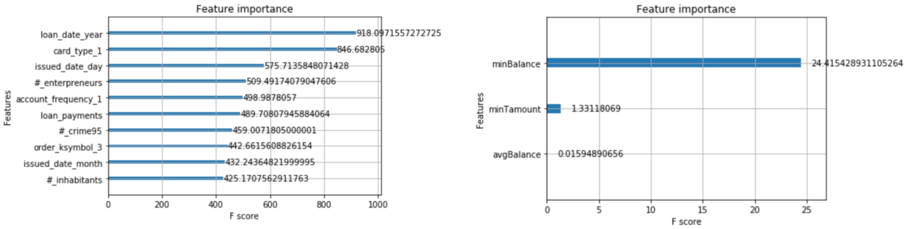
4 Experimental Results

We describe three experiments we performed to see the strength and the weakness of our approach. Experiments 1 and 2 were performed without the Gomphy, assuming all the features (or attributes) in the Financial database are potential banking events that could cause an unpaid loan. Experiment 3 was performed with the Gomphy approach. We validated whether the features as important ML predicted are reasonable towards the unpaid loan.

4.1 Experiment 1

One way to validate banking events causing unpaid loans is to assume all the features (or attributes) in the database as potential events and validate the events using ML.

For this experiment, the ML data set were prepared by selecting all the features, except the table identifiers in the Financial database. The prepared ML data set included 72 features and 449,736 records based on the transaction id. The big records are due to the Join operation among Account, Transaction, and Payment Order tables, where Transaction tables contain more than 1 million records. As some ML algorithms such as Gradient Boosting Tree provide feature importance, we analyzed whether the provided essential features could be possible banking events leading to the unpaid loan. Fig. 9.(a) shows some important features predicted by the XGBoost prediction model.



(a) Top 10 feature importance in XGBoost Exp. 1

(b) Top 3 feature importance in XGBoost in Exp.2

Fig. 9. Top Important Features in Experiment 1 and 2

One critical issue of this approach is that the ML (e.g., XGBoost) models showed different prediction results for the same loan instance (e.g., Load ID: 233). The data set was prepared based on the transaction IDs, and some transaction IDs have identical Loan IDs. However, ML models showed different loan prediction results (i.e., paid and unpaid) for some transactions associated with the same Loan IDs. We noticed that this different prediction could cause some confusion in identifying a banking event for the unpaid loan.

Another issue is that Experiment 1 showed some unlikely features, such as no. of entrepreneurs per 1000 inhabitants and no. of committed crimes '95 as essential features, which is not highly likely to make sense for factors impacting the unpaid loan. It was not easy to understand whether the no. of committed crimes is related to clients' loan payments.

4.2 Experiment 2

In this experiment 2, we also assumed all the features in the database as potential events to validate. However, the ML data set was prepared based on the loan ID, unlike experiment 1, to analyze the important features produced by ML models. To prepare a loan-based data set, we used SQL group functions, such as Sum, Min, and Avg, to select records for the one to many relationships between Account and Transactions. The final data set contained 682 records including 72 features. Four ML models were built to predict the loan instances. Fig. 9.(b) shows the some important features for the prediction.

A critical issue of this approach is that the prepared data set did not consider the boundary of the records within the loan duration. For example, when the loan duration of loan ID 1 is two years from 1993, the data set included records of 1996 and 1997, which may give incorrect predictions.

Among the given three important features, minimum balance, minimum transaction amount, and average balance, it was noticeable that 'the minimum amount of transaction' could cause the unpaid loan. If a client is not enough balance in his/her account, the transaction amount could be small, but other banking events seemed to be needed to get a deep understanding of this minimum amount of transaction.

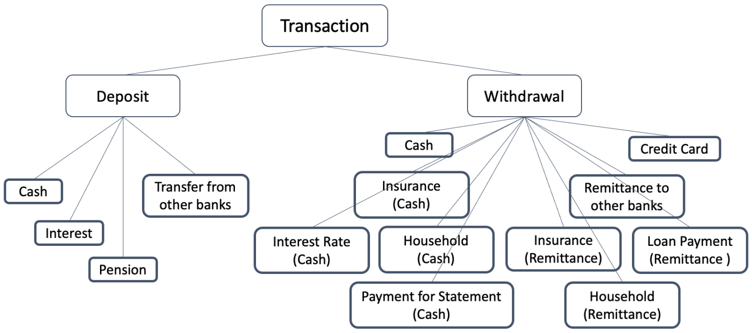
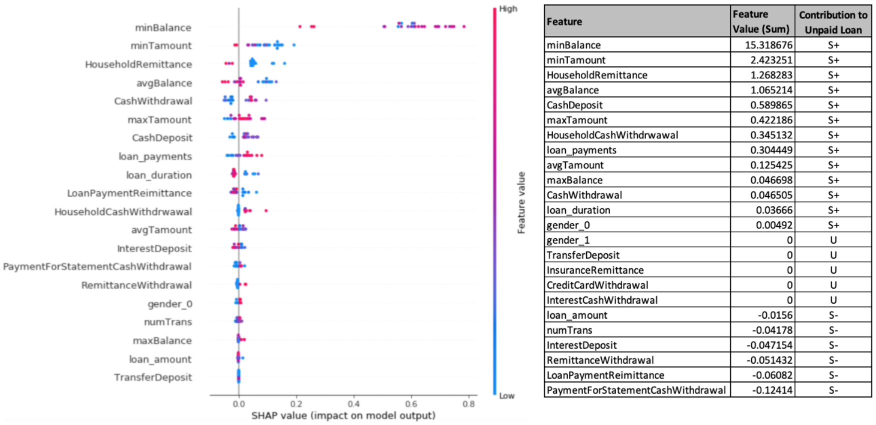


Fig. 10. Deposit and Withdrawal Classification in Transactions

4.3 Experiment 3

In this experiment 3, we applied the Gomphy approach to validate the banking behaviors towards the unpaid loan. The banking events were hypothesized as four groups, including Loan, Account, Transaction, and Client. The hypothesis is further analyzed into testable problem hypotheses, as shown in Fig. 8.

While analyzing the database, we could understand 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, which showed the deposit and withdrawal transaction, as shown in Fig. 10. So, we 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.



(a) Important features with SHAP in summary graph

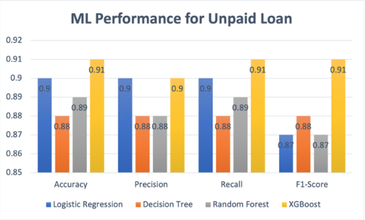
Feature	Feature Value (Sum)	Contribution to Unpaid Loan
minBalance	15.318676	S+
minTamount	2.423251	S+
HouseholdRemittance	1.268283	S+
avgBalance	1.065214	S+
CashDeposit	0.589865	S+
maxTamount	0.422186	S+
HouseholdCashWithdrwawal	0.345132	S+
loan_payments	0.304449	S+
avgTamount	0.125425	S+
maxBalance	0.046698	S+
CashWithdrawal	0.046505	S+
loan_duration	0.03666	S+
gender_0	0.00492	S+
gender_1	0	U
TransferDeposit	0	U
InsuranceRemittance	0	U
CreditCardWithdrawal	0	U
InterestCashWithdrawal	0	U
loan_amount	-0.0156	S-
numTrans	-0.04178	S-
InterestDeposit	-0.047154	S-
RemittanceWithdrawal	-0.051432	S-
LoanPaymentReimittance	-0.06082	S-
PaymentForStatementCashWithdrawal	-0.12414	S-

(b) Features Value and Contribution Type

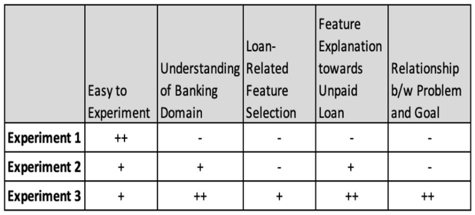
Fig. 11. Important Features and Contribution Type in Experiment 3

After the data preparation step, ML models were then constructed to predict the loan. Next, the ML Explainability model was applied to the ML model with the separately collected unpaid loan data set to understand better the impact of the features on the unpaid loan. The accuracy of the ML models using the extracted data is shown in Fig. 12.(a).

Fig. 11, produced by the SHAP model, shows some important features for the unpaid loan, including *minimum balance*, *minimum transaction amount*, *remittance withdrawal for household cost*, and others. Unlike the features in experiments 1 and 2, the selected features could be the factors leading to the unpaid loan. –more description?



(a) Exp. 3-ML Performance for Unpaid Loan



	Easy to Experiment	Understanding of Banking Domain	Loan-Related Feature Selection	Feature Explanation towards Unpaid Loan	Relationship b/w Problem and Goal
Experiment 1	++	-	-	-	-
Experiment 2	+	+	-	+	-
Experiment 3	+	++	+	++	++

(b) Comparison of Experiments

Fig. 12. Comparison of Experiments

5 Discussion and Related Work

Problem analysis and validation have been studied to understand business events behind real-world business problems in two major areas, including Requirements Engineering and Machine Learning. The distinctive of our approach is to use a concept of a problem hypothesis to explore causes of a problem, identify data features, construct a data set, refute or confirm the problem hypotheses using a goal-oriented and ML-based approach.

In Requirements Engineering, *a Fishbone diagram* has been used for identifying possible causes for a problem or an effect [22]. This technique helps enumerate potential causes for a problem, usually utilized in a brainstorming session. However, the lack of a clear relationship between a cause and an effect, e.g., logical connectives, such as 'AND,' or 'OR', makes problem validation difficult. *Fault Tree Analysis (FTA)* is a top-down, deductive analysis that visually depicts a failure path or failure chain [23]. FTA provides Boolean logic operators, which help create a series of True or False statements. When linked in a chain, these statements form a logic diagram of failure. However, FTA does not provide relationship direction and degrees, such as positive, negative, full, and partial, making it challenging to validate business problems using ML. (Soft-)Problem Inter-dependency Graph (PIG) uses a (Soft−)problem concept to represent a stakeholder problem against stakeholder goals. In PIG, a problem is refined into sub-problems and then traced to corresponding solutions [24]. However, PIG lacks a mechanism to connect the sub-problems to data features in a database. While the Fishbone diagram, FTA, and PIG provide a sound high-level model, they need validation mechanisms for eliciting causes behind business problems.

In the area of Machine Learning, some ML algorithms, such as Linear Regression and Decision Trees, provide feature importance value concerning their predictions. When ML models predict a numerical value in the regression model or a target label in the classification, relative feature importance scores are calculated for the features in the data set [3]. Explainable or interpretable machine learning models also provide feature importance [25]. LIME(Local Interpretable Model-agnostic Explanations) explains individual predictions, but there is some instability of the explanations, which may hurt validating business problems [26]. SHAP (SHapley Additive exPlanations) outputs intuitive feature value that helps to understand and validate business problems. However, SHAP may take a long computational time [20]. Although feature importance in ML algorithms could be utilized to get insights about business problems, there are some issues identifying business problems and preparing the data set, such as mapping business events to data features. The data features are often selected on informal identification of a low-level problem, which makes it difficult to understand transparent relationships between the low-level problem and high-level business problems [27].

Modeling a problem hypothesis as an entity consisting of attributes, constraints, and relationships helps identify relevant domain attributes in the domain entity. As a problem hypothesis is usually constructed in a class level capturing a business phenomenon, not an instance level, a problem hypothesis entity helps find data attributes in the database.

The feature importance in ML shows only one level relationship between data features and a target label, as shown in experiments 1 and 2. However, there may be more intermediate relationships than one among business events. Goal and problem analysis helps narrow this gap, with the help of ML

Limitations This paper has some limitations. 1) Correlation among problem hypotheses and goals could be considered to understand the business phenomena better, but the correlation analysis was not explored yet. 2) ML prediction as a solution approach to mitigate or alleviate the validated problems could be explored, but this work does not deal with that. 3) The Gomphy process is partially supported with Gomphy Assistant, although the Assistant needs more work to automate the presented approach.

6 Conclusion and Future Work

This paper has presented the Gomphy approach to validate potential business problems using goal-orientation and Machine Learning. After identifying likely problems against goals, Gomphy prepares the data set corresponding to problem hypotheses and discovers essential features for a target problem using ML, and evaluates whether those important features make sense to the problem. The empirical study was performed for the client's unpaid loan with the Financial database. Three technical contributions were presented: First, Gomphy domain-independent ontology; Second, a method of modeling a problem hypothesis as a problem hypothesis entity; Third, a mapping method identifying relevant features in a database using attributes, constraints, and relationships in the problem hypothesis entity, and finally, an evaluation method validating the problem hypothesis by selecting the most important features provided by ML.

Future work includes an in-depth study about correlations of features and applying it to the Gomphy approach, exploring solutions for the validated problems using ML, and developing a reliable Gomphy assistant tool.

References

1. Ross, D.T., Schoman, K.E.: Structured analysis for requirements definition. IEEE Transactions on Software Engineering **SE-3**(1) (1977) 6–15
2. Nuseibeh, B., Easterbrook, S.: Requirements engineering: a roadmap. In: Proceedings of the Conference on the Future of Software Engineering. (2000) 35–46
3. Brownlee, J.: Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python. Machine Learning Mastery (2020)
4. Davenport, T.H., Bean, R.: Big data and ai executive survey (2020). NewVantage Partners (NVP), Tech. Rep (2020)
5. Zhou, L., Pan, S., Wang, J., Vasilakos, A.V.: Machine learning on Big data: Opportunities and challenges. Neurocomputing **237** (2017) 350–361
6. Asay, M.: 85% of Big data projects fail, but your developers can help yours succeed. techrepublic (2017)
7. Nalchigar, S., Yu, E.: Business-driven data analytics: a conceptual modeling framework. Data & Knowledge Engineering **117** (2018) 359–372
8. Supakkul, S., Zhao, L., Chung, L.: GOMA: Supporting Big data analytics with a goal-oriented approach. In: 2016 IEEE International Congress on Big Data (BigData Congress). (June 2016) 149–156
9. Supakkul, S., Ahn, R., Gonçalves, R.J., Villarreal, D., Zhao, L., Hill, T., Chung, L.: Validating goal-oriented hypotheses of business problems using machine learning. In: Int. Conf. on Big Data, Springer (2020)
10. Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J.: Non-functional requirements in software engineering. Volume 5. Springer Science & Business Media (2012)
11. Binkhonain, M., Zhao, L.: A review of machine learning algorithms for identification and classification of non-functional requirements. Expert Systems with Applications: X **1** (2019) 100001
12. Berka, P., Sochorova, M.: Discovery challenge guide to the financial data set, pkdd-99 (1999)
13. Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: A process-oriented approach. IEEE Transactions on software engineering **18**(6) (1992) 483–497
14. Pearl, J., Verma, T.S.: A theory of inferred causation. In: Studies in Logic and the Foundations of Mathematics. Volume 134. Elsevier (1995) 789–811
15. Rolland, C., Souveyet, C., Achour, C.B.: Guiding goal modeling using scenarios. IEEE transactions on software engineering **24**(12) (1998) 1055–1071

16. Wu, H.Y., Tzeng, G.H., Chen, Y.H.: A fuzzy MCDM approach for evaluating banking performance based on balanced scorecard. Expert systems with applications **36**(6) (2009) 10135–10147
17. Chen, P.P.S.: The entity-relationship model—toward a unified view of data. ACM transactions on database systems (TODS) **1**(1) (1976) 9–36
18. Zheng, A., Casari, A.: Feature engineering for machine learning: principles and techniques for data scientists. " O'Reilly Media, Inc." (2018)
19. Li, J.J., Tong, X.: Statistical hypothesis testing versus machine learning binary classification: Distinctions and guidelines. Patterns **1**(7) (2020) 100115
20. Lundberg, S., Erion, G., Chen, H., DeGrave, A., Prutkin, J., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.I.: From local explanations to global understanding with explainable AI for trees. Nature machine intelligence **2** (2020) 56–67
21. Supakkul, S., Hill, T., Chung, L., Tun, T.T., do Prado Leite, J.C.S.: An NFR pattern approach to dealing with NFRs. In: 2010 18th IEEE International Requirements Engineering Conference, IEEE (2010) 179–188
22. Ishikawa, K.: Introduction to quality control. Productivity Press (1990)
23. Vesely, W., Goldberg, F., Roberts, N., Haasl, D.: Fault tree handbook. Technical report, Nuclear Regulatory Commission Washington DC (1981)
24. Supakkul, S., Chung, L.: 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) 389–394
25. Molnar, C.: Interpretable machine learning. Lulu. com (2020)
26. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. (2016) 1135–1144
27. Berry, M.J., Linoff, G.S.: Data mining techniques: for marketing, sales, and customer relationship management. John Wiley & Sons (2004)