Proc. of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME 2023)
19-21 July 2023, Tenerife, Canary Islands, Spain

Effective Automatic Feature Engineering on Financial Statements for Bankruptcy Prediction

Xinlin Wang
University of Luxembourg, SnT
xinlin.wang@uni.lu

Zsófia Kräussl
University of Luxembourg, DoF
zsofia.kraussl@uni.lu

Maciej Zurad
Yoba S.A., Luxembourg
maciej.zurad@gmail.com

Mats Brorsson *Senior Member, IEEE*
University of Luxembourg, SnT
mats.brorsson@uni.lu

***Abstract*—Feature engineering on financial records for bankruptcy prediction has traditionally relied significantly on domain knowledge and typically results in a range of financial ratios but with limited complexity and feature utilization due to manual design. It is often a time-consuming and error-prone procedure, confined to the domain experts' experience, without taking into account the characteristics of different data sets. In this paper, we propose an automated feature engineering approach to generate *effective*, *explainable*, and *extensible* model training features. The experiments have been conducted using a publicly available record of financial statements submitted to the Luxembourg Business Registers. This approach aims to improve bankruptcy prediction for professionals who may not possess the necessary engineering expertise or efficient data. The experimental results suggest that the proposed approach can provide valuable features for model training and in most of the cases, the model's outcomes outperforms predominantly as compared to the traditional approaches and the well-known approaches the models, thus can provide valuable features for model training.**

***Index Terms*—Automatic feature engineering, Bankruptcy prediction, Credit risk, Imbalanced data**

I. Introduction

Bankruptcy prediction models, whether based on financial experts or machine learning methods, are typically built on *financial ratios* [5]. While financial ratios have been demonstrated to have the ability in bankruptcy prediction [2], the range of these values can vary across different companies and sectors, making it challenging for credit analysts to make the decision. Additionally, financial statements can have different formats and accounting subjects, which further complicates the calculation of these ratios. Furthermore, not all companies can provide their complete historical data to calculate financial. This phenomenon often holds for small and medium-sized enterprises (SMEs). They cannot provide complete financial statements due the nature and/or maturity of their business, which leads to the inability to calculate financial ratios and to complete risk assessments, leading to a high level of declined credit requests [15].

This paper is inspired by the above-described constraints, thus, proposing and evaluating a novel automatic feature engineering approach, specifically suited for financial statements. The goal is to generate features to improve the quality of the input data for bankruptcy prediction, which is an important asset for credit requests of companies lacking the proper financial history and data. We also recreated the financial ratio approach [13], [19], [21], [30] and other representative feature generation approaches [14], [20] , and validated our solution design using published bankruptcy prediction models. Our results show that our automatic feature engineering approach not only performs significantly better to models based on financial ratios, but also outperforms other feature generation approaches. Our solution design resulted in the implementation of the automatic feature engineering method and algorithm [1].

In short, our contributions are:

- A novel automatic feature engineering algorithm replacing domain expert feature engineering in the business scenario,
- An implementation of the automatic feature engineering method algorithm ,
- A comparative study of some well known bankruptcy prediction models and different feature generation approaches using real data from Luxembourg Business Registers, and
- A performance evaluation of the impact of automatic feature engineering for bankruptcy prediction.

II. Related Work

With the development of machine learning and deep learning, plenty of studies have attempted to make a breakthrough by applying new models into bankruptcy prediction, and directly use well-calculated financial ratios to find the most predictive model and make predictions [4], [6], [9], [10], [22],

This work was supported by National Research Fund Luxembourg (FNR) under Grant 15403349 and Yoba S.A..

[1]https://pypi.org/project/auto-feature-engineering/

979-8-3503-2297-2/23/$31.00 © 2023 IEEE

[25]. In the early stage of applying machine learning methods for bankruptcy prediction, logistic regression [2] was once the most widely-used model in predicting bankruptcy. Even today, many financial institutions still adopt the logistic regression as the primary approach for building the credit scorecards because of its interpretability and stability [27] [23]. However, the results of using learning techniques have been shown to be contradictory at times, for instance, [22] and [4]. These discrepancies are due to the fact that machine learning models' results are highly dependent on the input data, whereas the financial data of firms, especially of SMEs, used to predict credit risk and bankruptcy are often unstructured and incomplete [18].

We aim to build a discipline-wide research of bankruptcy prediction, and complement it with this machine learning approach to improve the performance by focusing on input data. Our engineering exercise is considered from multiple viewpoints [3], since bankruptcy prediction is in the hand of different stakeholders (i.e. creditor and borrower) with different, often not well-aligned requirements and financial incentives. Taking commercial interest for solution design is well-documented in [7], which goes beyond technologically driven requirement elicitation and analysis of stakeholders. It recognizes the necessity of a commercially-driven requirement elicitation, and defines an ontology that allows, among others, conceptualizing the fundamentals of valuation principles of economic transactions. Authors in [8] provides a conceptual model to guide alignment of domain-specific requirements in order to achieve shared understanding among stakeholders for a solution design. Our analysis is predominantly inspired by these works, as our goal is to elicit stakeholder-specific data variables to enhance traditional bankruptcy prediction, which can then become input variables of implementable instructions. Thus, we choose to apply automatic feature engineering to enhance domain experts' analysis in an explainable and effective method.

One direction of automatic feature engineering is *feature interaction*, which is extensively applied in the area of recommendation systems. There are plenty of explorations on automatic feature engineering to improve the performance of predicting click-through rate, such as the Factorization Machine (FM) [12], DeepFM [14], AutoInt [29], and AutoFIS [31] each of which build the extensive feature interactions to obtain a good result of the model. However, these methods are more suitable for the highly sparse categorical data and the business scenario of recommendation systems. Moreover, these features are deeply integrated with the deep learning models and therefore it is not meaningful to just take out the features as such. Furthermore these features typically lack interpretability. As financial business use-cases usually pay great attention to explainability, our approach takes this into account and addresses this issue.

The other direction is *feature combination*, of which deep feature synthesis (DFS) [20] is one of the most well-known algorithms. The overall idea is to arrange the original data in tables and to combine them with a greedy algorithm to search the whole feature space using SQL-like statements like "select...join...group by". An advantage of DFS is that it is interpretable which can give users a clearer insight into the business and it is good at handling relational data. The limitation of this method is that it will generate all the features according to the manual setting of the hyperparameters regardless of whether they are useful or not. It just implement the aggregation functions on the features so it cannot generate the features with the necessary depth. Additionally, modeling with redundant features can be costly and may lead to unfavorable outcomes. There are still some benefits of the DFS method and we use a variant of it by introducing a feature selection process based on feature importance to overcome this limitation.

Luo et al. [26] proposed the *AutoCross* method to generate an explainable feature set based on a beam search algorithm and it also proved to be effective on real world datasets. The authors treated all the original features as categorical features and they split the numerical features into bins in order to turn them into categorical features. In our case, the raw data comes from financial statements which are all numerical. This may lead to losing information from the raw data if we discretize the numerical data into categorical data.

Inspired by above-mentioned work, we propose an automatic feature engineering (AFE) algorithm to construct the features specifically targeting raw numerical data.

III. Methodology and Solution Design

A. Overview for Automatic Feature Engineering

Fig. 1 illustrates the entire process of automatic feature engineering (AFE) from raw financial records to a derived feature set for bankruptcy prediction. The first step involves pre-processing the raw data for feature construction, where we cope with extreme values by replacing infinity with finite extremes, and treat missing values as zero due to accounting subjects. Before proceeding to the next step, we determine the hyperparameters k_1 and k_2, representing the number of features selected from feature aggregation and feature crossing respectively, as well as $batch\_size$ representing the number of feature pairs resulting from one iteration of the first feature crossing round.

Feature generation process consists of two independent parts: aggregation and crossing. In the feature aggregation process, we generate the features by the aggregation method at one time and then select the most valuable ones. While feature crossing process consists of a loop of feature crossing and feature selection, new features are generated by the crossing method followed by the same selection criteria of feature aggregation. Subsequently, if the new generated feature set does not meet the termination condition, feature crossing continues. Otherwise, it stops and then we obtain the derived feature set by combining the outcomes from crossing process with the features generated by aggregation process.

B. Feature generation

For feature aggregation, we calculate statistical descriptive indicators for features of each company over n years and

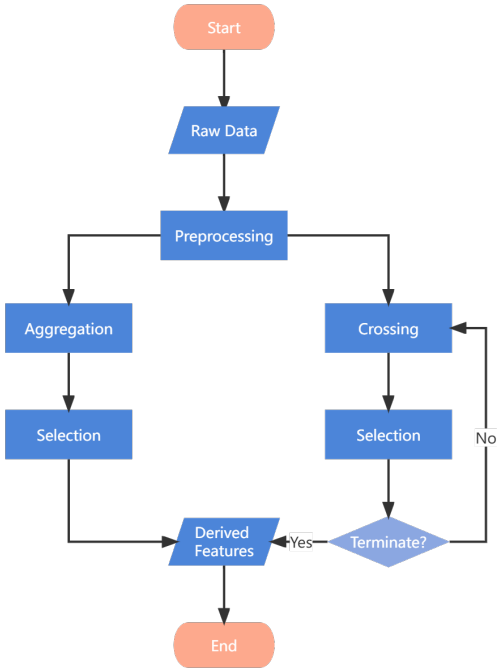

Fig. 1. Pipeline for automatic feature engineering process

use them as new features. Specifically, the maximum value (max), the minimum value (min), the sum (sum), the average ($mean$), the standard deviation (std), and the percentage change (pct_change) between the current and the previous year are used as the descriptive indicators. We adopt the feature importance from lightGBM to evaluate a feature's contribution for identifying the targets because of its fast and efficient computation, high reliability, and strong interpretability [24]. Top k_1 features are kept as a part of the final feature set.

For the first round $i = 0$ of feature crossing, as in (1), we have $(f_1 f_2 \cdots f_n)$ representing the n features from the input dataset and S_0 representing the derived feature set after the first round of feature combination. The symbol $\odot$ represents four basic operands addition $(+)$, subtraction $(-)$, multiplication $(*)$, and division $(/)$, which aims to mimic the experts calculating financial ratios on each feature pair. Taking one element $(f_1 \odot f_2)$ as an example, this represents four value of new features $(f_1 + f_2), (f_1 - f_2), (f_1 * f_2), (f_1 / f_2)$.

$$
\begin{aligned}
S_0 &= \begin{pmatrix} f_1 & f_2 & \cdots & f_n \end{pmatrix}^T \odot \begin{pmatrix} f_1 & f_2 & \cdots & f_n \end{pmatrix} \\
&= \begin{pmatrix} (f_1 \odot f_1) & (f_1 \odot f_2) & \cdots & (f_1 \odot f_n) \\ (f_2 \odot f_1) & (f_2 \odot f_2) & \cdots & (f_2 \odot f_n) \\ \vdots & \vdots & \ddots & \vdots \\ (f_n \odot f_1) & (f_n \odot f_2) & \cdots & (f_n \odot f_n) \end{pmatrix}
\end{aligned} \tag{1}
$$

Succeeding to this, we follow the feature selection process. We unfold the feature set S_0 and put the new derived features into a LightGBM classification model to obtain their feature importance. We choose top k_2 features aligned with the results as the input features to the next feature generation round. Meanwhile, we add these k_2 features to the final feature set for the prediction model.

For the next rounds $i > 0$, as in (2), we have S_i as the new derived feature set. We then repeat the steps mentioned above and we will have k_2 new generated features for each order until it meets the termination condition. The detailed steps of automatic feature engineering is shown in the algorithm 1.

$$
\begin{aligned}
S_i &= \overbrace{\begin{pmatrix} f_a & f_b & \cdots & f_k \end{pmatrix}^T}^{k\ features\ from\ S_0} \odot \begin{pmatrix} f_1 & f_2 & \cdots & f_n \end{pmatrix} \\
&= \begin{pmatrix} (f_a \odot f_1) & (f_a \odot f_2) & \cdots & (f_a \odot f_n) \\ (f_b \odot f_1) & (f_b \odot f_2) & \cdots & (f_b \odot f_n) \\ \vdots & \vdots & \ddots & \vdots \\ (f_k \odot f_1) & (f_k \odot f_2) & \cdots & (f_k \odot f_n) \end{pmatrix}
\end{aligned} \tag{2}
$$

C. Termination condition

The algorithm for feature engineering offers two termination options. The first option is a maximum iteration limit for the feature generation loop to prevent infinite feature generation. This limit can be manually defined and it is set to the number of input features by default, allowing each feature to cross with every other feature once.

The other is automatic termination. As shown in algorithm 1, for every round of feature generation we train a LightGBM model to make the prediction on new generated features, and then compare the AUC value of current round with the previous round. If the AUC of current round is larger than the previous round, it represents that new generated features improves the performance of the prediction model so these features should be kept and added to the final feature set. If the AUC of current round is smaller than previous round, it indicates that new generated features cannot improve the prediction model so feature generation process should stop.

In our case, we adopt the default number of the maximum feature generation loop and also the automatic termination to achieve the minimal manual intervention and get the derived feature set.

IV. Experiments

To validate the performance, we conduct experiments based on our case data, which we described in Section II. The time period of our data is from 2011 (the earliest annual data available via LBR) to 2021, and it includes records of both bankruptcies and well-operating companies. We construct nine data sets with different historical periods (1-year, 2-year,..., and 9-year) to cover all time spans from 2011 to 2021. The n from 1 to 9 years means that the data contains balance sheets from n consecutive years of each company. The target label corresponds to the business status (bankrupt or non-bankrupt) of each company in the year after n consecutive years. The descriptive statistics of our datasets listing the number of positive samples (bankrupt companies), negative samples (non-bankrupt companies), positive rate (bankruptcy rate), the feature sizes of AFE approach and raw data per each historical period is shown in Table I.

Algorithm 1 The Automatic Feature Generation Process

Definitions:
F_{raw}: Input parameter. A vector with the original features in the raw data,
D: Input parameter. A random selection of rows from 70% of raw pre-processed data, also as the training data for the prediction models,
k = the number of highest ranked features,
$batch\_size$ = the number of feature pairs in each batch,
$f_x \odot f_y$ = operation that yields a set of four values: $f_x + f_y, f_x - f_y, f_x * f_y, f_x/f_y$,
AUC = area under receiver operating characteristic curve,
$AggIndicators(data, *arg)$ is a function to generate the descriptive indicators $*arg$ of each feature in $data$,
$TopFeatures(model, FC, k)$ is a function to rank the Feature Candidate set (FC) based on the feature importance calculated from $model$, and keep the top k.
lgb: LightGBM model.
Returns:
F = Constructed feature set

function AFE(F_{raw}, D)
  $F_{agg} \leftarrow AggIndicators(F_{raw}, *arg)$ ▷ $*arg$: a set of operations max, min, sum, mean, std, pct_change
  $FC \leftarrow F_{agg}$
  $FS_{agg} \leftarrow TopFeatures(lgb(D[FC]), FC, k_1)$
  $n_0 \leftarrow |F_{raw}|$
  $n\_batches \leftarrow n_0^2/batch\_size$
  $F_{cross} \leftarrow F_{raw}^T \odot F_{raw}$ ▷ See (1)
  $FS_0 \leftarrow \emptyset$
  for $b \leftarrow 0, n\_batches - 1$ **do**
    $FC \leftarrow F_{crossed}[batch]$
    $FS_b \leftarrow TopFeatures(lgb(D[FC]), FC, k_2)$
      ▷ Selecting all features in a batch and keep the top k
    $FS_0 \leftarrow FS_0 \cup FS_b$
  end for
  $FS_1 \leftarrow TopFeatures(lgb(D[FS_0]), FS_0, k_2)$ ▷ Select the k top most important feature pairs
  $AUC_0 \leftarrow 0$
  for $i \leftarrow 2, n_0 - 1$ **do**
    $F_{cross} \leftarrow F_{raw}^T \odot FS_{i-1}$
    $FC \leftarrow F_{crossed}$
    $FS_i \leftarrow FS_i \cup TopFeatures(lgb(D[FC]), FC, k)$
    $AUC_i \leftarrow Calculate\_AUC(lgb(D, FS_i))$
    if $AUC_i <= AUC_{i-1} | reached\_limit(i)$ **then**
      return FS_i
    end if
  end for
  $F \leftarrow FS_{agg} \cup FS_i$
end function

TABLE I. SUMMARY OF SUB-DATASETS

Dataset	Negative	Positive	Positive Rate	Num of AFE Features	Num of Input Features
1-year	31528	4351	12%	42	64
2-year	28570	3271	10%	58	128
3-year	26029	2348	8%	50	192
4-year	23783	1675	7%	34	256
5-year	21664	1133	5%	66	320
6-year	19689	737	4%	50	384
7-year	17560	440	2%	50	448
8-year	15036	229	2%	58	512
9-year	7913	88	1%	50	576

Hyperparameters of AFE approach can be chosen to fit the characteristics of different input data. The choice of hyperparameters involves a trade-off between model performance, training time, and resource consumption. We explore hyperparameters for 1-year, 2-year, and 3-year datasets to optimize the performance. We need to set k1 and k2 for selecting top features from the feature aggregation and feature crossing processes, as well as $batch\_size$ to prevent running out of memory during feature generation. k1 and k2 directly affect the number of features selected for the bankruptcy prediction model. After experiments on k1 varying from 5 to 50, k2 varying from 2 to 20 and $batch\_size$ varying from 100 to 512800, we set k1 to 15, k2 to 8 and $batch\_size$ to 30,000 for optimal model performance.

A. Comparison with feature sets generated by other methods

a) Financial ratios: Financial ratios are created and designed based on domain experts experience. Due to the limitation of lack of cash flow and profit & loss statements, we can only replicate 10 financial ratios based on the top 20 most frequently used financial ratios in bankruptcy prediction as discussed in [13] and 8 financial ratios from other studies[19], [21], [30]. The considered financial ratios are depicted in Table II. The variables marked $\star$ are derived from [13] and the ones marked † comes from[19], [21], [30].

TABLE II. THE SET OF FINANCIAL RATIOS USED IN THE PAPER

Variable	Financial Ratios	Description
f1*	current ratio	current assets ÷ current liabilities
f2*	debt to equity	debt ÷ equity
f3*	working capital to total assets	(current assets-current liabilities) ÷ total assets
f4*	total liabilities to total assets	total liabilities ÷ total assets
f5*	equity to total assets	equity ÷ total assets
f6*	quick ratio	(cash + marketable securities + accounts receivable) ÷ current liabilities
f7*	current assets to total assets	current assets ÷ total assets
f8*	cash to total assets	cash ÷ total assets
f9*	cash to current liabilities	cash ÷ current liabilities
f10*	long term debt to equity	long term debt ÷ equity
f11†	total assets growth rate	(total assets of current year - total assets of previous year) ÷ total assets of previous year
f12†	quick assets to total assets	(current assets-inventory-prepaid expenses) ÷ total assets
f13†	current assets to current liabilities	current assets ÷ current liabilities
f14†	(cash or marketable securities) to total assets	(cash + marketable securities) ÷ total assets
f15†	total debt to total assets	debt ÷ total assets
f16†	equity to fixed assets	equity ÷ fixed assets
f17†	current assets to total liabilities	current assets ÷ total liabilities
f18†	short-term liabilities to total assets	short-term liabilities ÷ total assets

First, we compare the performance of the representative bankruptcy prediction models, which are trained by the features generated by financial ratios and AFE, then we evaluate the prediction contribution of the features from these two different approaches.

b) DeepFM: DeepFM [14] is a prominent approach from the area of recommendation systems. It was improved from the factorization machine(FM). DeepFM model contains two parts, FM and DNN. FM model extracts low-order features and DNN model extracts high-order features so it can learn the low- and high-order feature interactions simultaneously. The output is the sum of the FM part and the DNN part as in (3). Since the input is the raw features and FM and DNN share these features, the training for DeepFM model is fast. Our focus here is on the results obtained using the default setting of DeepFM, rather than fine-tuning the model. It is a blackbox model so we can not get the exact features that generated by DeepFM and it returns the prediction as the model result.

$$\hat{y} = sigmoid(y_{FM} + y_{DNN}) \tag{3}$$

c) Deep feature synthesis: We also compare our approach with deep feature synthesis(DFS) mentioned in Section III. We deploy the `featuretools` package and set seven primitives: "sum","std","max","skew","min","mean" and "trend". These primitives are what we can have according to the input data. We compare DFS with AFE approach in two ways. One is to keep all the features that generated by DFS. The other is to select the same number of features from DFS as in AFE approach. This two-way comparison is to show the redundant features generated by DFS and the necessity of feature selection.

1) Models for evaluating different feature generation approaches: We evaluate the effectiveness of the features generated by two approaches by comparing the performance of the representative models trained from the features generated by these two approaches. Each of the nine datasets (1-year, 2-year,... and 9-year) are divided randomly into training and testing sets with the ratio of 7:3. Then we train the models mentioned below on these nine datasets. The following are the representative bankruptcy prediction models and their settings:

a) Logistic Regression (LR): It uses the sigmoid function to transit the result from linear model to classification result. [2] applied this model to make bankrupt prediction and it is widely used in this area. In this study, we use the LR function of Scikit-learn package (version 1.0.1), setting the $C = 0.1$ and *class_weight='balanced'*.

b) Random Forest (RF): It is a typical bagging ensemble model which performs well in the classification task. [17] compared RF with LR in individual credit risk prediction and the result showed that RF outperforms LR. In this study, we use the RF function of Scikit-learn package (version 1.0.1), setting the *max_depth=2* and *n_estimators = 10*.

c) LightGBM (LGB): It is an improved model based on extreme gradient boosting (XGBoost) and also a representative model of a boosting ensemble model [24]. [28] compared several models to predict bankruptcy and LightGBM is the best among all the models. In this study, we use the LightGBM package (version 3.2.2) and the Gridsearch method to find the best parameters for learning rate, max depth and number of leaves. We keep other parameters in default setting.

d) Multilayer perceptron (MLP): Inspired by the study [32], [33], we build a MLP model with four hidden layers. We use the ReLU function as the activation function and the dropout rate is 0.3. The loss function is the cross entropy and the learning rate of 0.01. We implemented the MLP model using PyTorch (version 1.10.0).

B. Feature performance indicators for comparing feature contribution

We evaluate the performance of features by feature importance and information value. We conduct feature importance ranking and calculate the information value by the `LGBMClassifier()` function with default parameters of `lightgbm` package on the combination dataset of features created by different approaches. We set the option of "importance_type" to "split" to calculated the importance, which is a split-based method. Feature importance is generally used to evaluate the contribution of each features in model training [11]. The higher the rank, the larger effect it has on the model. Information value is an indicator that can be used to measure the predictive power of an independent variable [1]. A higher information value means the feature has more predictive power. We adopt this indicator to evaluate features' performance because it is widely used in feature selection of credit risk assessment in the financial industry. It can be calculated as following [16]:

$$IV = \sum_{i=1}^{n} (\frac{G_i}{G} - \frac{B_i}{B}) * \ln \frac{G_i/G}{B_i/B} \tag{4}$$

where n is the number of bins of each feature, G_i and B_i represent the numbers of negative and positive samples of bin i, G and B are the total number of negative and positive samples in the population.

V. Results and Discussion

A. Comparison with financial ratios

1) Model performance: Fig. 2 shows the performance of mentioned models trained on automatic feature engineering (AFE) and financial ratios (FR) from Table II. The x-axis of the figure represents nine datasets. The y-asis represents the improvement of AUC from models trained by AFE compared to AUC from models trained by FR. We can see that the models trained by AFE have the outstanding advantages over the models trained by FR. In total, AFE outperforms FR in 35 out of 36 cases. Therefore, the models using automatic feature engineering approach has a better ability to predict bankruptcy under these scenarios.

2) Feature performance: We evaluate the contribution of each feature by putting them in the same bankruptcy prediction model. Fig. 3 shows a comparison of the feature importance between the features found by the AFE algorithm and the financial ratios. It is clear that the features created by AFE have higher rank than the features created by FR for all the nine datasets, which also implies why models on AFE tend to perform better.

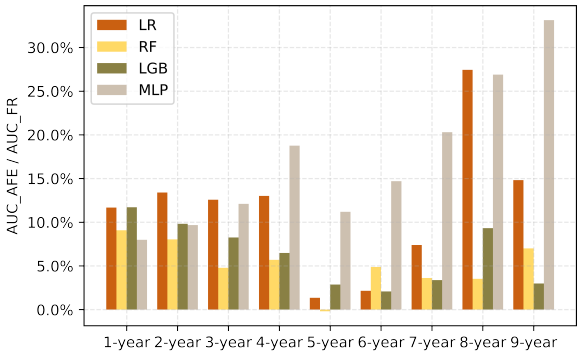

Fig. 2. AUC improvement of AFE compared to FR on each dataset

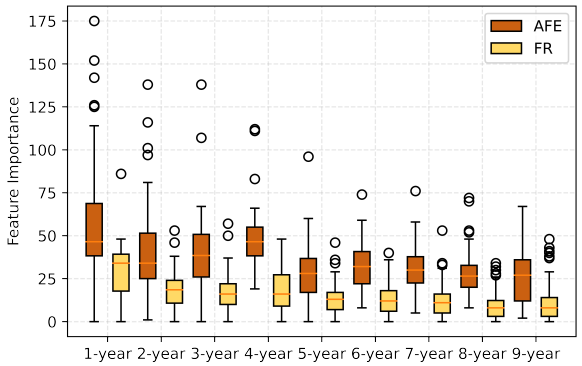

Fig. 3. Feature importance of AFE and FR on each dataset

Fig. 4 shows the information value, which indicates that most features created by AFE have higher information values than features created by FR approach among all datasets except for the 3-year dataset. Although the medium IV of 3-year features from AFE is slightly lower than 3-year features from FR, the 1st quartile of features from AFE is still larger than 3-year features from FR, which means AFE essentially contribute features with higher IV and could result in a better model performance. Hence, we can draw the conclusion that features created by AFE have better performance than features created by FR based on the results of both feature importance and information value.

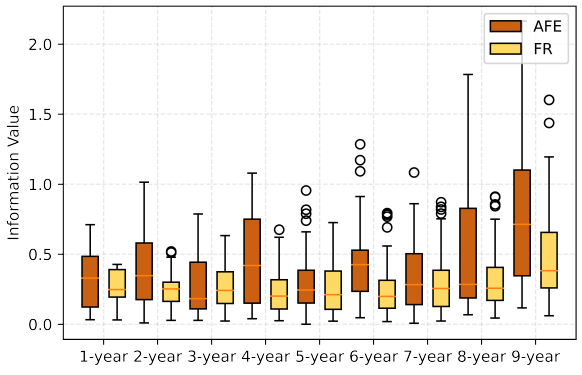

Fig. 4. Information value of features in AFE and FR on each dataset

B. Comparison with DeepFM

We compare the performance of the four models with the AUC of DeepFM result. From Table III, it shows that automatic feature engineering have the absolute advantages over all the nine datasets by all the four models. AUC of DeepFM for nine datasets are all between 0.6 and 0.7. But for LightGBM model on AFE, the AUC could reach more than 0.85 on 1-year and 2-year dataset and around 0.8 on other datasets, which is an impressive improvement compared to DeepFM. We also notice that LightGBM model on AFE have a better performance than logistic regression model and random forest on AFE. This is because not only the LightGBM model has effective predictive capabilities but also the hyperparameters are chosen based on the LightGBM model.

TABLE III. AUC OF MODELS TRAINED ON AFE AND DEEPFM

Dataset	AFE				DeepFM
	LR	RF	LGB	MLP	
1-year	0.7466	0.7940	**0.8713**	0.7658	0.6352
2-year	0.7542	0.7867	**0.8589**	0.7833	0.6502
3-year	0.7574	0.7725	**0.8458**	0.7872	0.6443
4-year	0.7797	0.7964	**0.8474**	0.8090	0.6560
5-year	0.6640	0.7517	**0.8206**	0.7292	0.6256
6-year	0.6734	0.8103	**0.8257**	0.7360	0.6303
7-year	0.6444	0.7693	**0.7997**	0.7120	0.6062
8-year	0.7722	0.7636	**0.8038**	0.7141	0.6262
9-year	0.7693	0.8207	**0.8301**	0.7993	0.6762

C. Comparison with deep feature synthesis

We compare automatic feature engineering with deep feature synthesis in two ways. Fig. 5 shows the comparison where we keep all the features generated by DFS. The x-axis of the figure represents nine datasets. The y-axis represents the improvement of AUC of models trained on AFE compared to AUC of models trained on DFS. From this figure, we can see that the models trained on AFE have a clear advantage over the models trained on DFS with all the features. We also identified that logistic regression model trained on AFE lacks to provide favorable results compared to random forest model and LightGBM model. This means that features generated by AFE performs well on the tree models rather than the linear models. We can consider this as the result of adopting the tree model to select features during the process of the automatic feature generation approach. In a nutshell, AFE outperforms DFS with all features in 28 out of 36 cases, indicating that it is still advantageous to use AFE features than DFS when training models.

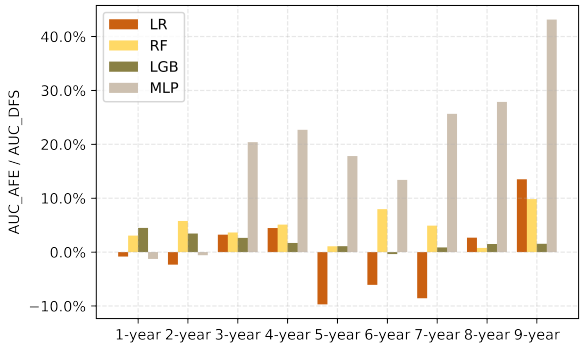

Fig. 5. AUC improvement of AFE compared to DFS with all the features on each dataset

Fig. 6 is the comparison when we select the same number of features from DFS as in AFE approach. The result is similar to

the Fig. 5. In summary, AFE outperforms DFS with selected features in 28 out of 36 cases. For some cases, DFS with selected features have better results than DFS with all features, which indicates that it is beneficial for model performance to drop the redundant features when training models.

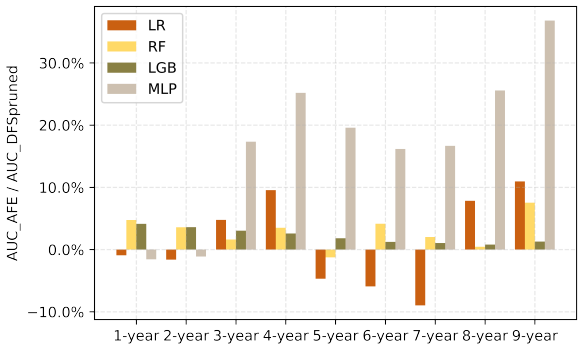

Fig. 6. AUC improvement of AFE compared to DFS with the selected features on each dataset

D. Comparison with raw data

To prove the necessity of feature engineering for the financial statements, we compare our approach with the raw data from financial statements. Because the data from financial statements are all numerical, we implement the same data preprocessing steps to handle the extreme values and the missing values as we adopt in automatic feature engineering.

The comparison of raw data and automatic feature engineering can be found in Fig. 7. The AFE have higher AUC in most of cases and at the same time, AFE has less advantage on the logistic regression models but in total AFE outperforms raw data with selected features in 26 out of 36 cases, which means feature engineering indeed improves the predictive ability of the features for model training.

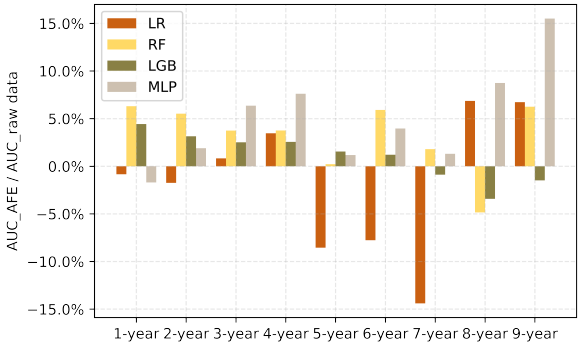

Fig. 7. AUC improvement of AFE compared to raw data on each dataset

E. Explainability and Extensibility

1) Explainability: All features created by the AFE algorithm take the form of simple arithmetic expressions. Taking the AFE feature with the highest feature importance from 1-year data set as an example: $(fid\_321 + fid\_322)$, where each term fid_x is the name of one of the original numerical features taken from the financial statement of a company. fid_321 and fid_322 represents the profit or loss of the current and previous year respectively. The sum of these two numbers shows the profit or loss of recent two years, which can be considered as an important factor that is related to the business status of a company.

2) Extensibility: For this particular solution design and experiment we adopted seven operands for the aggregation process and four operands for the crossing process. The automatic feature engineering process, however, could be extended by adding more operands for both the aggregation and crossing. It depends on the users to decide the number of operands, based on the available data set.

VI. Conclusion

In this paper we presented an automatic feature engineering approach to enhance bankruptcy prediction of companies that lack sufficient data due to incomplete financial statements for traditional risk assessment. Our design is centred around generating features for prediction improvement based on real-world financial statements. The results of our research shows that the models trained on features generated by automatic feature engineering outperform the models trained, among others, on features generated by the traditionally used financial ratios. Our research thus implies that automatic feature engineering can generate effective features for model training, which is an especially useful enhancing effect for the bankruptcy prediction and risk assessment of companies lacking sufficient data in a traditional crediting setup, such as SMEs. The presented results form the first completed phase of a longer time horizon. The case study, and thus our data is concentrating on the Luxembourgish market, thus potentially describe a biased sampling profile. Luxembourg is a small-scale economy within the European Union, with focal industry concentration. Therefore, we aim at geographically extending our data set in order to steer our solution design toward generalizability. As a direct consequence, our future work includes collecting more qualified samples, and running follow-up experiments to check if our conclusions remain valid across different markets.

References

[1] R. A. Howard, "Information value theory," *IEEE Transactions on systems science and cybernetics*, vol. 2, no. 1, pp. 22–26, 1966.

[2] J. A. Ohlson, "Financial ratios and the probabilistic prediction of bankruptcy," *Journal of accounting research*, pp. 109–131, 1980.

[3] A. Finkelstein, J. Kramer, B. Nuseibeh, L. Finkelstein, and M. Goedicke, "Viewpoints: A framework for integrating multiple perspectives in system development," *International Journal of Software Engineering and Knowledge Engineering*, vol. 2, no. 01, pp. 31–57, 1992.

[4] P. P. Pompe and A. Feelders, "Using machine learning, neural networks, and statistics to predict corporate bankruptcy," *Computer-Aided Civil and Infrastructure Engineering*, vol. 12, no. 4, pp. 267–276, 1997.

[5] S. A. Ross, R. Westerfield, and J. F. Jaffe, *Corporate finance*. Irwin/McGraw-Hill, 1999.

[6] A. F. Atiya, "Bankruptcy prediction for credit risk using neural networks: A survey and new results," *IEEE Transactions on neural networks*, vol. 12, no. 4, pp. 929–935, 2001.

[7] J. Gordijn and J. Akkermans, "Value-based requirements engineering: Exploring innovative e-commerce ideas," *Requirements engineering*, vol. 8, no. 2, pp. 114–134, 2003.

[8] Z. Derzsi and J. Gordijn, "A framework for Business/IT alignment in networked value constellations.," in *BUSITAL*, 2006.

[9] C.-F. Tsai and J.-W. Wu, "Using neural network ensembles for bankruptcy prediction and credit scoring," *Expert systems with applications*, vol. 34, no. 4, pp. 2639–2649, 2008.

[10] L. Nanni and A. Lumini, "An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring," *Expert systems with applications*, vol. 36, no. 2, pp. 3028–3033, 2009.

[11] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer, "Permutation importance: A corrected feature importance measure," *Bioinformatics*, vol. 26, no. 10, pp. 1340–1347, 2010.

[12] S. Rendle, "Factorization machines," in *2010 IEEE International conference on data mining*, IEEE, 2010, pp. 995–1000.

[13] V. Boguslauskas, R. Mileris, and R. Adlytė, "The selection of financial ratios as independent variables for credit risk assessment," *Economics and management*, vol. 16, no. 4, pp. 1032–1040, 2011.

[14] G. Wang, J. Hao, J. Ma, and H. Jiang, "A comparative assessment of ensemble learning for credit scoring," *Expert systems with applications*, vol. 38, no. 1, pp. 223–230, 2011.

[15] M. Cowling, W. Liu, and A. Ledger, "Small business financing in the uk before and during the current financial crisis," *International Small Business Journal*, vol. 30, no. 7, pp. 778–800, 2012.

[16] N. Siddiqi, *Credit risk scorecards: developing and implementing intelligent credit scoring*. John Wiley & Sons, 2012, vol. 3.

[17] J. Kruppa, A. Schwarz, G. Arminger, and A. Ziegler, "Consumer credit risk: Individual probability estimates using machine learning," *Expert Systems with Applications*, vol. 40, no. 13, pp. 5125–5131, 2013.

[18] N. Gordini, "A genetic algorithm approach for smes bankruptcy prediction: Empirical evidence from italy," *Expert systems with applications*, vol. 41, no. 14, pp. 6433–6445, 2014.

[19] Q. Yu, Y. Miche, E. Séverin, and A. Lendasse, "Bankruptcy prediction using extreme learning machine and financial expertise," *Neurocomputing*, vol. 128, pp. 296–302, 2014.

[20] J. M. Kanter and K. Veeramachaneni, "Deep feature synthesis: Towards automating data science endeavors," in *2015 IEEE international conference on data science and advanced analytics (DSAA)*, IEEE, 2015, pp. 1–10.

[21] M. Zięba, S. K. Tomczak, and J. M. Tomczak, "Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction," *Expert systems with applications*, vol. 58, pp. 93–101, 2016.

[22] F. Barboza, H. Kimura, and E. Altman, "Machine learning models and bankruptcy prediction," *Expert Systems with Applications*, vol. 83, pp. 405–417, 2017.

[23] A. Bequé, K. Coussement, R. Gayler, and S. Lessmann, "Approaches for credit scorecard calibration: An empirical analysis," *Knowledge-Based Systems*, vol. 134, pp. 213–227, 2017.

[24] G. Ke, Q. Meng, T. Finley, *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[25] T. Hosaka, "Bankruptcy prediction using imaged financial ratios and convolutional neural networks," *Expert systems with applications*, vol. 117, pp. 287–299, 2019.

[26] Y. Luo, M. Wang, H. Zhou, *et al.*, "Autocross: Automatic feature crossing for tabular data in real-world applications," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1936–1945.

[27] Y. Shi and X. Li, "An overview of bankruptcy prediction models for corporate firms: A systematic literature review," *Intangible Capital*, vol. 15, no. 2, pp. 114–127, 2019.

[28] H. Son, C. Hyun, D. Phan, and H. J. Hwang, "Data analytic approach for bankruptcy prediction," *Expert Systems with Applications*, vol. 138, p. 112 816, 2019.

[29] W. Song, C. Shi, Z. Xiao, *et al.*, "Autoint: Automatic feature interaction learning via self-attentive neural networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1161–1170.

[30] Y. Zhu, L. Zhou, C. Xie, G.-J. Wang, and T. V. Nguyen, "Forecasting smes' credit risk in supply chain finance with an enhanced hybrid ensemble machine learning approach," *International Journal of Production Economics*, vol. 211, pp. 22–33, 2019.

[31] B. Liu, C. Zhu, G. Li, *et al.*, "Autofis: Automatic feature interaction selection in factorization models for click-through rate prediction," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 2636–2645.

[32] B. Prasetiyo, M. Muslim, N. Baroroh, *et al.*, "Artificial neural network model for banckrupty prediction," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1567, 2020, p. 032 022.

[33] G. Lombardo, M. Pellegrino, G. Adosoglou, S. Cagnoni, P. M. Pardalos, and A. Poggi, "Machine learning for bankruptcy prediction in the american stock market: Dataset and benchmarks," *Future Internet*, vol. 14, no. 8, p. 244, 2022.