Predicting Erroneous Financial Statements Using a Density-Based Clustering Approach

Martha Tatusch
Department of Computer Science
HHU Düsseldorf
Düsseldorf, Germany
tatusch@hhu.de

Gerhard Klassen
Department of Computer Science
HHU Düsseldorf
Düsseldorf, Germany
klassen@hhu.de

Marcus Bravidor
Department of Business Administration
HHU Düsseldorf
Düsseldorf, Germany
bravidor@hhu.de

Stefan Conrad
Department of Computer Science
HHU Düsseldorf
Düsseldorf, Germany
stefan.conrad@hhu.de

ABSTRACT

In this paper, we present a novel machine-learning approach to detect and predict restated financial statements. Our approach is based on DBSCAN, a cluster analysis algorithm. In contrast to prior methods, we assume that firms which perform different than their peers (outliers) are more likely to restate. By modifying DBSCAN to also incorporate temporal variation of these differences, we optimize the algorithm to fit financial data. We test our model for US data and benchmark against prior findings in accounting research. Our results show that the modified version of DBSCAN is more efficient than prior approaches. Best results are obtained if we cluster based on only two or three features. We outperform prior approaches regarding the precision to identify restatements. As with prior results, detection error increases for material restatements.

CCS Concepts

•Computing methodologies→Machine learning→Learning paradigms→Unsupervised learning→Cluster analysis

•Computing methodologies→Machine learning→Learning paradigms→Unsupervised learning→Anomaly detection

•Applied computing→Operations research→Forecasting

•Applied computing→Law, social and behavioral sciences→Economics

Keywords

Clustering, Prediction, Anomaly Detection, Time Series Analysis, Financial Restatements

1. INTRODUCTION

Restatements weaken the reliability of financial information. They are an important signal for investors and have (negative) long-term financial consequences [1]. Even though the number of restatements declined from more than 800 in 2009 to around 550 in 2017 [2], the number is still troublesome. Therefore, it is important for financial statement users, particularly investors, to be able to identify potential restatements. In this paper, we aim to detect financial restatements with a modified, dynamic version of the DBSCAN [5] clustering algorithm.

To evaluate our model, we use data from Thomson-Reuters (TR) EIKON [8] for the 20-year period between 1998 to 2017. For benchmarking, we use four different sets of restatements. First, restatements are defined as changes in any notable financial statement position (see Appendix.A1) recorded in EIKON. Second, relevant restatements that change either sales, operating cash flow, net income or shareholders' equity. Third, relevant and material restatements which are similar to the second definition but the change for any position must be at least 5%. Note that the first definition is the least strict, with two and three increasing in strictness, respectively. The fourth definition has to be considered separately as it is given by restatements reported by Audit Analytics [1].

To detect financial restatements, we use our modified DBSCAN algorithm. The idea behind this approach is that similar firms should have similar attributes and changes in a similar fashion. Hence, once a firm behaves abnormally in a sense that it shows different development in attributes than its peers, we assume it to be an "outlier". The advantage of using an unsupervised machine-learning algorithm like DBSCAN is that we need no *ex ante* expectations on (a) why firms behave differently, and (b) the threshold in differences that makes a firm an "outlier". Firms are clustered within industry groups (defined as four-digit Thomson-Reuters Business Classification codes) and there are one up to six attributes (features) provided.

Our results show that our approach correctly classifies more than 50% of all firm-years as (non-)restatement years for all four restatement definitions. On first sight, this result falls short of prior approaches (e.g., [3] report an accuracy of more than 65%). However, DBSCAN excels in precision of the results. We report values of 65.6%, 52.7%, 33.5% and 16.4% for the four restatement

definitions, respectively. [3] score 0.7%. Put differently, our approach is many times more likely to correctly classify restatements (as opposed to non-restatements).

We contribute to the literature in several ways. First, prior models used to identify financial restatements usually relied on either extensive multiple regression models or supervised machine-learning approaches. Whereas the first require a lot of firm-level data, the latter are often time opaque and the results difficult to understand. We address both issues and implement an efficient clustering algorithm which can detect restatements based on two or three firm-level items. Second, DBSCAN was initially build to work with 'static' data. We introduce a modified, dynamic version of DBSCAN which can detect cluster outliers by changes over different periods. Put differently, our approach is suitable to track changes in yearly firm-level data.

The paper is structured as follows. In section 2, we briefly discuss some background information on financial restatements as well as prior studies in accounting and computational science research. In section 3, we introduce our modified, dynamic version of DBSCAN. Evaluation results and benchmarks are presented in section 4. The paper ends with some concluding remarks.

2. BACKGROUND

An important distinction in the first place is the difference between erroneous and fraudulent financial statements. Fraudulent statements are the product of (management's) intention to mislead the user. Erroneous statements are simply (partly) false. The reasons can be manifold: fraud / intention, clerical or technical errors, etc. Once such an (material) error is found, the company must file a restatement. In our case, we are not interested in the reason behind the error. Therefore, we look at restatements as an indicator for any kind of erroneous financial statement.

In their extensive survey of accounting research, [10] differentiate between the causes and consequences of financial statements. Especially smaller firms, growth firms, and firms with a low earnings and/or reporting quality are more likely to restate. Most of these studies use logistic regressions to identify the causes (e.g., [3]). In their methodological review on data mining techniques used to identify financial restatements, [4] show that most studies in this realm build upon artificial neural networks, Bayesian Belief Networks and other forms of supervised learning. We follow their call to explore other complementary techniques. In this case, cluster analysis as another form of unsupervised machine-learning.

3. MODEL DESCRIPTION

In the following, we explain our method, the related parameter and feature selection as well as necessary basics of the original DBSCAN algorithm.

3.1 DBSCAN

DBSCAN is an algorithm for discovering clusters in large databases with noise [5]. In contrary to other clustering algorithms, DBSCAN determines clusters with the spatial density property of the regarded data points. In addition, the problem of identifying the right number of clusters is omitted, since it is automatically established. The algorithm differentiates three types of data points:

- **Core points:** A point *p* is a core point if there are at least *minPts* points in the ε-neighborhood of it (including *p*). The ε-neighborhood of *p* is defined as the spatial region with center *p* and radius ε>0.
- **Density-reachable points:** A density-reachable point is a point that is located within the ε-neighborhood of a core point.
- **Noise points:** A noise point is a point that is neither a core point nor a density-reachable point.

A cluster consists of at least one core point and *minPts-1* density-reachable points. Core points of different clusters are not density-reachable regarding each other. Points that are not assigned to any cluster are interpreted as noise.

3.2 Our Approach

As outlined above, current approaches for the detection of financial restatements have two major shortcomings. First, they rely on an "estimate-predict" idea. Take the case of a logistic regression. In order to predict whether a firm-year is likely to be restated, one has to first estimate the model parameters, then reverse and fill in the "blanks" with firm-year specific data. This approach requires a lot of data (out of sample predictions) and judgment (e.g., thresholds). Second, to draw meaningful inferences, the variables in the prediction model have to be selected on *ex ante* expectations.

For our approach, we require no *ex ante* expectations. We assume that similar firms behave in a similar and comparable manner. Similar deviations or periodic changes represent shared economic characteristics. To cover a broad set of economic factors we use the set of variables from [3]. Furthermore, we consider real activities manipulation proxies (RAM, [9]), and accrual-based earnings management (AEM) based on the modified Jones-model ([7]; [6]). We expect these features to hold more information about restatements than usual balance sheet figures.

In order to avoid the comparison of highly distinctive companies and industry-specific circumstances such as seasonal changes, our algorithm is applied to every industry sector separately. We define industry sectors based on four-digit TR Business Classification codes. We analyzed about 30 features (see the full list in Appendix.A1) and targeted a solution with less input features than the state-of-the-art approaches. Therefore we looked into different feature sets. A feature set f for a year t is described as $f_t = a_{t1}, \ldots, a_{tn}$ with $n \in \{2, 3, 4, 5\}$. Then the development vector is calculated as $d_{t+1} = f_{t+1} - f_t$.

In order to identify misstatements the development vectors of companies in the same section are clustered with DBSCAN. Finally, development vectors which are not assigned to a cluster are regarded as misstatements.

3.3 Feature & Parameter Selection

In order to determine the best feature set for our approach and the most suitable setting of DBSCAN, we set a few constraints and iterated through all possible combinations. Since the most values are between 0 and 1, the possibilities for the radius ε of DBSCAN were set to [0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0]. *minPts* was set to integers between 3 and 6.

To avoid the curse of dimensionality only subsets containing one to maximal five features were considered. In Appendix.A1 a complete list of all features is given. The original fields from EIKON (prefix “TR-”) have been normalized with the assets of the year before. All fields with the suffix “Error” contain the difference between the actual value and the expected regression value. For example, in *FF-CFOError* the difference between the actual cashflow from operating activities and the expected value of Roychowdhury's regression is stored. All regressions were calculated for every firm-year in every sector.

Note that the processed values and the original fields were considered individually. That means, that the subsets of all fields with the prefix "FF-" and all subsets of the fields with the prefix "TR-" have been tested. This was done to check whether, in our method, the popular proxies from [3], [6], [7] and [9] are more expressive than the original data.

4. EVALUATION

In the following, we will present our experiments, starting with the experimental setup, and discuss the results.

4.1 Experimental Setup

We analyzed 9300 companies with available data from 1998 to 2017. The data has been retrieved from TR EIKON which also holds restated figures.

In this paper, we observe four definitions of restatements:

- $Definition_{all}$: A statement is considered a restatement, if one or multiple financial values have been changed afterwards.
- $Definition_{relevant}$: A statement is considered a restatement, if at least one relevant financial value has been changed afterwards. Relevant values are: net income, shareholder's equity, operating cashflow, and sales.
- $Definition_{relevant5\%}$: A statement is considered a restatement, if one or multiple relevant financial values (see $Definition_{relevant}$) have been changed by at least 5% afterwards.
- $Definition_{audit}$: A statement is considered a restatement, if it is reported as restatement by Audit Analytics.

In case necessary data is missing, the firm-year is not included. This leads to different observations for different feature sets. All feature sets and parameters were evaluated on the HPC-Cluster of the Heinrich-Heine-University Düsseldorf. We analyzed about 500000 combinations of features and parameters. Every feature set was submitted as a job to the cluster.

4.2 Results

Apart from the type I and type II error, we consider three other measures:

- $accuracy = \frac{correct\ classified\ firm\ years}{all\ firm\ years}$
- $precision = \frac{correct\ classified\ as\ restatement}{all\ classified\ as\ restatements}$
- $recall = \frac{correct\ classified\ as\ restatement}{all\ real\ restatements}$

The detection of restatements is a challenging task. Since the data set is often very unbalanced, as there are only few misstatements, it is important to find a model that is not only strong in recognizing objects of one class.

The accuracy is not a suitable measure to verify this property. Suppose a dataset consists of 90% non-restatements and 10% restatements. If a model classifies every firm-year as non-restatement, an accuracy of 90% is achieved. This behavior is not desirable as the model does not make decisions based on features but on the fact, that most of the firm-years are non-restatements.

Precision and recall are more informative measures in this task. The precision indicates the relative value of how many elements of those classified as restatements have been correctly detected. The recall specifies the percentage of all restatements that have been recognized. If the recall is small, it is likely that the model rarely classifies objects as restatements. If the precision is small, this means that the probability, that a classification as a restatement is correct, is very low.

Table 1. Results for $Definition_{all}$.

Original Data	**obs. \ pred.**	**Rest.**	**No Rest.**	∑
	Rest.	4066	3719	7785
	No Rest.	2140	2171	4311
	∑	6206	5890	12096
	Rest.	**52.2%**	47.8%	64.4%
	No Rest.	49.6%	**50.4%**	35.6%
	Precision:	65.6%		
	Recall:	52.2%		
	Accuracy:	51.6%		
Processed Data	**obs. \ pred.**	**Rest.**	**No Rest.**	∑
	Rest.	7816	7704	15520
	No Rest.	5976	6215	12191
	∑	13792	13919	27711
	Rest.	**50.4%**	49.6%	56.0%
	No Rest.	49.0%	**51.0%**	44.0%
	Precision:	56.7%		
	Recall:	50.3%		
	Accuracy:	50.6%		

Top: TR-AccountsPayable, TR-NetIncome, TR-TtlPlanExpectedReturn, $\varepsilon = 0.01$, minPts = 5. Bottom: FF-CH_CS, FF-CH_EMP, FF-CFF, FF-TAX, $\varepsilon = 0.1$, minPts = 5.

In Table 1, 2, 3 and 4 the best results after iterating all possibilities are shown for the four restatement definitions from 4.1. In this paper, results are considered as *good* if the hit rates are higher than the error values, and the precision and recall are near to their maximum. Unfortunately, many (original and processed) values are missing, so that only combinations with at least 10000 observations are discussed in the following.

Table 1 shows the results for $Definition_{all}$. The hit rates are highlighted using bold font. With this definition both, the original values from EIKON as well as the processed data lead to good results. It is notable, that with $Definition_{all}$ combined with the best settings, the restatements form the majority with 64.4% and 56.0%. The classification into restatement and non-restatement is balanced. This means that the model does not only focus on one class. The precision indicates that the model detects restatements with a certainty of 65.6% with the original data and 56.7% with the processed features. [3] use 7 different features (variables, model 1) and report a precision of $\frac{339}{48621} = 0.7\%$. Of course, the values are not directly comparable. One reason are the different data sets and the definition of the outcome variables (restatements vs. accounting and enforcement actions). On the other hand, [3] achieve a better recall with 68.6%. This means that they are more likely to detect restatements. However, this may be due to the fact that the number of restatements was smaller. Although there is a gap between the achieved precision with the original data and the processed proxies, both outcomes are competitive regarding the results in [3], as the recall is only slightly lower, but the precision is significantly higher.

Nevertheless, these results are not fully comparable due to different definitions.

Table 2. Results for *Definition*$_{relevant}$.

Original Data	**obs. \ pred.**	**Rest.**	**No Rest.**	**Σ**
	Rest.	2938	2487	5425
	No Rest.	2639	2644	5283
	Σ	5577	5131	10708
	Rest.	**54.2%**	45.8%	50.7%
	No Rest.	33.4%	**66.6%**	49.3%
	Precision:	52.7%		
	Recall:	54.2%		
	Accuracy:	52.1%		
Processed Data	**obs. \ pred.**	**Rest.**	**No Rest.**	**Σ**
	Rest.	5911	5725	11636
	No Rest.	7881	8194	16075
	Σ	13792	13919	27711
	Rest.	**50.8%**	49.2%	42.0%
	No Rest.	49.0%	**51.0%**	58.0%
	Precision:	42.9%		
	Recall:	50.8%		
	Accuracy:	50.9%		

Top: TR-NetSales, TR-TtlPlanExpectedReturn. Bottom: FF-CH_CS, FF-CH_EMP, FF-TAX. In both cases $\varepsilon = 0.01$ and minPts = 5.

The results for *Definition*$_{relevant}$ can be seen in Table 2. The original values as well as the processed proxies achieved good results with subsets of two and three features. The best performance, however, has been reached with the original data using two values: *TR-NetSales* and *TR-TtlPlanExpectedReturn*.

The data set and classification are nearly balanced. Nevertheless, it can be observed that non-restatements are better identified than restatements. Although only two original values are used, the hit rates can compete with those of [3]. Furthermore the precision is again significantly better. Using three proxies from [3] the hit rates are around 50%. However, we reach better precision values.

In Table 3 the results for *Definition*$_{relevant5\%}$ are shown. The calculations show that original values as well as the processed ones achieve good results with subsets of two up to five elements. The pure financial ratios again delivered better scores than the processed proxies. This time restatements have a higher hit rate than non-restatements in both cases. The hit rates of the processed data are similar to the ones for *Definition*$_{all}$ in Table 1.

Last but not least the results for *Definition*$_{audit}$ are shown in Table 4. The best settings are very similar to the ones regarding *Definition*$_{relevant5\%}$. One reason for this could be that both definitions are quite granular. Only 13.8% of the data (15.3% respectively)

is considered as restatement. It is striking that for the first time better results can be achieved when using the processed data with a precision of 16.4%. However, the difference between the results of the different data is not very considerable.

One notable finding is that precision decreases for more granular or strict definitions of restatements. Hereby, the recall slightly increases. In our approach, the popular proxies for the processed

Table 3. Results for *Definition*$_{relevant5\%}$.

Original Data	**obs. \ pred.**	**Rest.**	**No Rest.**	∑
	Rest.	2077	1741	3818
	No Rest.	4129	4149	8278
	∑	6206	5890	12096
	Rest.	**54.4%**	45.2%	31.6%
	No Rest.	49.9%	**50.1%**	68.4%
	Precision:	33.5%		
	Recall:	54.4%		
	Accuracy:	51.5%		
Processed Data	**obs. \ pred.**	**Rest.**	**No Rest.**	∑
	Rest.	3754	3308	7062
	No Rest.	10038	10611	20649
	∑	13792	13919	27711
	Rest.	**53.2%**	46.8%	25.5%
	No Rest.	48.6%	**51.4%**	74.5%
	Precision:	27.2%		
	Recall:	53.2%		
	Accuracy:	51.8%		

Top: TR-AccountsPayable, TR-NetIncome, TR-TtlPlanExpectedReturn, $\varepsilon = 0.01$, minPts = 5. Bottom: FF-CFF, FF-CH_CS, FF-CH_EMP, FF-TAX, $\varepsilon = 0.1$, minPts = 5.

data generally perform worse than the original data for the restatements extracted from EIKON. Only when considering the Audit Analytics information, the processed data scores slightly better. Altogether, our model performs best on a broader definition with the original data (financial ratios). It could be shown, that our approach works better with the original data than with economic proxies, so that in total only up to four features are necessary to achieve the shown results.

5. CONCLUDING REMARKS

In this paper, we introduced a modified, dynamic version of the DBSCAN clustering algorithm. We use this algorithm to detect financial restatements. Overall, our approach is highly efficient. We reach more than 50% accuracy with just two or three features. Remarkably, the modified version of DBSCAN performs particularly in detecting restatement years as compared to non-restatement years.

Our results should be of interest to practitioners and standard-setters. We demonstrate that it is not the amount of data alone but the data processing method that can make a difference. Furthermore, we would like to point to the difficulty of assessing the superiority of one approach to another based on different evaluation criteria. Whereas our approach scores low values of

accuracy compared to [3], it is much better suited to identify restatements (precision).

One major shortcoming of our paper is the limited knowledge about the generalizability of results. More testing is required to analyze

the dependence of the results on the characteristics of the underlying sample as well as deriving evidence on the predictive power of the results.

Table 4. Results for *Definition*$_{audit}$.

Original Data	obs. \ pred.	Rest.	No Rest.	Σ
	Rest.	882	789	1671
	No Rest.	4739	5686	10425
	Σ	5621	6475	12096
	Rest.	**52.8%**	47.2%	13.8%
	No Rest.	45.5%	**54.5%**	86.2%
	Precision:	15.7%		
	Recall:	52.8%		
	Accuracy:	54.3%		
Processed Data	**obs. \ pred.**	**Rest.**	**No Rest.**	**Σ**
	Rest.	2264	2000	4264
	No Rest.	11563	12125	23688
	Σ	13827	14125	27952
	Rest.	**53.1%**	46.9%	15.3%
	No Rest.	48.8%	**51.2%**	84.7%
	Precision:	16.4%		
	Recall:	53.1%		
	Accuracy:	51.5%		

Top: *TR-AccountsPayable*, *TR-NetIncome*, *TR-TtlPlanExpectedReturn*, $\varepsilon = 0.01$, *minPts* = 4. Bottom: *FF-CH_CS*, *FF-CH_EMP*, *FF-TAX*, $\varepsilon = 0.05$, *minPts* = 3.

6. ACKNOWLEDGMENTS

This work was partly supported by the Jürgen Manchot Foundation which funds the research Group *Decision-making with the help of Artificial Intelligence* at HHU Düsseldorf.

7. REFERENCES

[1] AuditAnalytics: https://www.auditanalytics.com

[2] AuditAnalytics: 2017 financial restatements review, https://www.auditanalytics.com/blog/2017-financial-restatements-review/

[3] Dechow, P., Ge, W., Larson, C., Sloan, R.: Predicting material accounting misstatements. *Contemporary Accounting Research* 28, 1 (2010), 17–82.

[4] Dutta, I., Dutta, S., Raahemi, B.: Detecting financial restatements using data mining techniques. *Expert Systems with Applications* 90 (2017), 374–393.

[5] Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd Int. Conference on Knowledge Discovery and Data Mining* (1996). 226–231.

[6] Jones, J.J.: Earnings management during import relief investigations. *Journal of Accounting Research* 29, 2 (1991), 193–228.

[7] Kothari, S.P., Leone, A.J., Wasley, C.E.: Performance matched discretionary accrual measures. *Journal of Accounting and Economics* 39 (2005), 163–197.

[8] Reuters, T.: Eikon financial analysis and trading software

[9] Roychowdhury, S.: Earnings management through real activities manipulation. *Journal of Accounting and Economics 42* (2006), 335–370

[10] Sievers, S., Sofilkanitsch, C.: Financial Restatements: Trends, Reasons for Occurrence, and Consequences. A Survey of the Literature.

APPENDIX

Table A1. List of features included

Fields		Source
FF-CFF FF-CH_CM FF-CH_EMP FF-CH_INV FF-CH_REC FF-EXFIN FF-LEASEDUM FF-OPLEASE FF-RSST_ACC FF-TAX	FF-CH_BACKLOG FF-CH_CS FF-CH_FCF FF-CH_PENSION FF-CH_ROA FF-ISSUE FF-LEVERAGE FF-PENSION FF-SOFT_ASSETS FF-WC_ACC	Dechow et al. [3]
FF-CFOError FF-DISEXPError FF-PRODError	FF-COGSError FF-INVError	Roychow-dhury [9]
FF-ACC_JONESError		Jones [6]
FF-ACC_KOTHARIError		Kothari et al. [7]
TR-Employees TR-LTDebt TR-LTDebtIssued TR-LTDebtNet TR-LTInvestments TR-NetIncome TR-TaxDefTot TR-Revenue TR-NetSales TR-TotalEquity TR-TotalInventory TR-ValueBacklog	TR-NetIncomeAfterTaxes TR-NetIncomeBeforeTaxes TR-PreferredStockNet TR-SgaExpenseTotal TR-ShortTermInvestments TR-TotalCurrLiabilities TR-TotalLiabilities TR-TotalOperatingExpense TR-TotalReceivablesNet TR-TtlPlanExpectedReturn TR-TtlPreferredSharesOut TR-CostOfRevenue	EIKON [8]
TR-ResearchAndDevelopment TR-SaleIssuanceOfCommonPreferred TR-TotalEquityAndMinorityInterest TR-TotalOperatingLeasesSuppl TR-AdvertisingExpense TR-CapitalExpenditures TR-CapitalLeaseObligation TR-CashandEquivalents TR-CashAndSTInvestments TR-CashFromFinancingAct TR-CashFromOperatingAct TR-CommonStockNet TR-CostOfRevenueTotal TR-DepreciationAmort TR-IncomeTaxesPayable TR-LTDebtMaturingYear1 TR-PptyPlantEqpmtTtlGross TR-SaleIssuanceOfCommon		EIKON [8]