Can merged datasets help in training money laundering detection models?

Paulius Savickas[1,3], Dovilė Kuizinienė[2,3], Mantas Bugarevičius[4], Žilvinas Kybartas[4] and Tomas Krilavičius[2,3]

[1] Vytautas Magnus University, Faculty of Economics and Management, K. Donelaičio street 52, LT-44244 Kaunas, Lithuania
[2] Vytautas Magnus University, Faculty of Informatics, Vileikos street 8, LT-44404 Kaunas, Lithuania
[3] Centre for Applied Research and Development, Lithuania
[4] UAB "Inventi", Vilnius, Lithuania

Abstract

Money laundering identification and prevention is one of the most important topics in the financial industry and financial crimes investigation. However, due to the high volume of transactions, personal data protection, and highly skilled white-collar criminals. Artificial intelligence and machine learning are already successfully used in different fintech applications as well, as crime prevention. Unfortunately, due to confidentiality and privacy regulations, AML cases and related data are hard to obtain, and different datasets include very different AML models. In most research, synthetically generated datasets with their own assumptions that do not know or reflect reality are used. For this reason, in this research, we try to improve AML models by merging different datasets with different features. We experiment with three publicly available, synthetically generated money transaction datasets and five different ML approaches: Random Forest, Generalized Linear Regression, XGBoost, Isolation Forest, and an ensemble of these methods. We use SMOTE for dataset balancing. The best model has achieved 95.98% accuracy with recognized 95.6% of legal payments and 84.4% of money laundering cases. This was achieved using an ensemble of all methods.

Keywords

AML, money laundering, machine learning algorithms, Random Forest, Isolation Forest, XGBoost, Generalized Linear Regression, merged data

1. Introduction

Money laundering is regulated by both the government authorities of financial crimes and the banks, since it involves much larger amounts of money than in the case of fraudulent payments. Every year, between 2% and 5% of global GDP is laundered, amounting to between 715 billion and 1.87 trillion euros [1]. In 1989, the Group of Seven (G-7) established the Financial Action Task Force (FATF) as an international group to combat money laundering on a global scale. Its mandate was broadened in the early 2000s to include countering terrorism financing [2].

Money laundering, poses a greater threat to society as a whole, yet it is rarely studied by researchers due to the high level of data confidentiality involved. Therefore, researchers for future topic development is using synthetically generated datasets. These datasets are created on different assumptions, then teaching on one set and testing on another does not achieve good results in recognizing money laundering cases. Hence, merged data set created, which is used for machine learning algorithms testing for ensuring better money laundering prevention.

While synthetic data has numerous advantages, it can be difficult to use appropriately. It's extremely challenging to ensure that it's as reliable as real-world data. It is possible to create a synthetic data set that does not accurately represent real-world scenarios when dealing with complex data sets containing a significant number of variables. This can lead to inaccurate decision-making due to incorrect insight development [3].

2. Literature review

Money laundering prevention is a matter for both government and financial institutions. Data in this area are highly sensitive and often difficult to access, for that reason, this problem is not widely discussed in the literature. All studies used cryptocurrency transaction or synthetically generated datasets. The following methods were analyzed: Random Forest, Logistic Regression, Decision Tree, XGBoost, Support Vector Machine, deep learning methods.

In the vast majority of studies reviewed, Random Forest showed best performance, compared to other methods, with an accuracy of 98.06%, 99%, 90.40%, 97.53%, and

IVUS 2022: 27th International Conference on Information Technology, May 12, 2022, Kaunas, Lithuania
paulius.savickas@card-ai.eu (P. Savickas); dovile.kuiziniene@vdu.lt (D. Kuizinienė); mantas.bugarevicius@inventi.lt (M. Bugarevičius); zilvinas@inventi.lt (Ž. Kybartas); tomas.krilavicius@vdu.lt (T. Krilavičius)

F1-score ranging from 0.76 to 0.83 [4][5][6][7][8].

The analysis suggests that Random Forest is the most appropriate method to address the problem of money laundering, however due to a lack of research, its effectiveness has not been validated. Furthermore, since it is unclear what assumptions the datasets were based on and whether they correlate to reality, merging several datasets allows for the most comprehensive coverage of those assumptions.

3. Methods

3.1. Machine learning methods

The selection of methods in this study was based on the literature review and best practices for money laundering prevention. Four supervised methods, namely, Random Forest, Generalized Linear Regression, Support Vector Machine and XGBoost, and one unsupervised method, namely, Isolation Forest, were used in the study.

3.1.1. Generalized Linear Model

The term "linear model" usually encompasses both systematic and random components in a statistical model, however for the purposes of this project the term was restricted to include only the systematic components:

$$Y = \sum_{i=1}^{m} \beta_i x_i, \tag{1}$$

when x_i is independent variables with known values, and β_i is parameters β_i values might be fixed (known) or unknown, requiring estimation. An independent variable can be quantitative, producing a single x-variate in the model, qualitative, producing a set of x-variates with values between 0 and 1, or mixed, producing a set of x-variates with values between 0 and 1.

3.1.2. Random Forest

Random Forest is a machine learning algorithm that constructs a multitude of decision trees during the training. The main principle of constructing a random forest is that a classifier is formed by combining solutions from binary decision trees made using diverse subsets of the original dataset and subsets containing randomly selected features from the feature set [9].

Constructing small decision trees that only have a few features takes up a little of the processing time, hence the majority of such trees' solutions can be combined into a single strong classifier.

3.1.3. XGBoost

XGBoost is a machine learning algorithm that implements frameworks based on Gradient Boosted Decision Trees [10]. XGBoost surpasses other machine learning algorithms by solving many data science problems faster and more accurately than its counterparts. Also, this algorithm has additional protection from overfitting.

3.1.4. Support Vector Machine

The aim of application of Support Vector Machine is to find the maximum separating line (if the case is two-dimensional) or a separating plane (if the case is three-dimensional), or a separating hyperplane (if the case is n-dimensional, $n > 3$) that would have the maximum distance between the nearest training data objects. For a hyperplane (could be a line or a plane) to be considered as the best, it needs to have the minimum classification error on previously unseen objects [11].

3.1.5. Isolation Forest

Let T be a node of an isolation tree. T is either an external-node with no child, or an internal-node with one test and exactly two daughter nodes (T_l,T_r). A test consists of an attribute q and a split value p such that the test $q < p$ divides data points into T_l and T_r [12].

Given a sample of data X = x_1 , ..., x_n of n instances from a d-variate distribution, to build an isolation tree (iTree), we recursively divide X by randomly selecting an attribute q and a split value p, until either: (i) the tree reaches a height limit, (ii) |X| = 1 or (iii) all data in X have the same values. An iTree is a proper binary tree, where each node in the tree has exactly zero or two daughter nodes. Assuming all instances are distinct, each instance is isolated to an external node when an iTree is fully grown, in which case the number of external nodes is n and the number of internal nodes is n 1; the total number of nodes of an iTrees is $2n$ 1; and thus the memory requirement is bounded and only grows linearly with n.

Anomaly detection's goal is to generate a ranking that indicates the degree of anomaly. As a result, sorting data points according to their path lengths or anomaly scores is one technique to find anomalies; anomalies are points at the top of the list. The following are the definitions of path length and anomaly score.

3.2. Class balancing

Unbalanced classes leads to machine learning algorithms classification issues. The unequal proportion of cases presented for each class of problem characterizes these issues. Synthetic Minority Oversampling Technique (SMOTE) is a well-known algorithm for dealing with

this problem, and its strategy is to artificially generate additional examples of the minority class by using the cases' closest neighbors. In addition, the majority of class examples are under-represented, resulting in a more balanced collection [13].

3.3. Data normalization method

We use Z-score normalization to normalize each column in the dataset separately, so the mean of the entire column becomes 0 and the standard deviation is 1. The following is the normalizing formula [14]:

$$x' = (x - \mu)/\sigma, \tag{2}$$

where μ is the population mean, and σ is the population standard deviation.

3.4. Models evaluation

In this research, we use accuracy, sensitivity, specificity, F1 score, and AUC metrics to correctly evaluate the results of the models so that money laundering and legal payments can be clearly identified and compared with the results of other studies [15]. To compute them, a confusion matrix is needed. These metrics are calculated as follows [16][17]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Sensitivity = \frac{TP}{FN + TP} \tag{4}$$

$$Specificity = \frac{TN}{TP + FN} \tag{5}$$

$$F1 = 2x\frac{sensitivity \times specificity}{sensitivity + specificity} \tag{6}$$

$$AUC = \frac{1}{2}\sum_{i=1}^{m-1}(x_{i+1} - x_i) * (y_{i+1} + y_i) \tag{7}$$

4. Data

4.1. Datasets

For money laundering prevention, three different synthetically generated datasets are used, which are compared in Table 1. Comparing other datasets with *Paysim*, it has many records, which can influence machine learning algorithms to recognize only its data. For this reason, *Paysim* dataset is five time reduced by selecting every fifth row of this dataset.

Table 1
Money laundering datasets

Name	Transactions	Fraud %	Mean transaction	Median transaction
Mahootika [18]	2,340	60%	€2,508,583	€1,162,354
AMLSim [19]	1,323,234	0.16%	€115,988	€157
Paysim (1/5) [20]	1,272,524	0.13%	€179,953	€74,898

Mahootika synthetically generated dataset [18] covers five months (February 20, 2019 – July 20, 2019) of 2,340 transactions, 60% of which are money laundering. This dataset has seven attributes. The simulation is based on three processes of money laundering in financial transactions: 1) Money placement 2) Money layering 3) Money integration.

AMLSim dataset [19] consists of 1,048,575 transactions, of which 0.16% are money laundering cases. All of these transactions are made from 9,999 accounts to 9,999 receive accounts. This dataset consists of eight attributes. It is synthetically generated using the *AMLSim* simulator.

Paysim dataset [20] consists of 6,362,620 transactions, but we narrow it down to 1,272,524 transactions due to computer resources. 0.13% of these transactions are money laundering cases. All of these transactions are made from 1,272,159 accounts to 777,582 receive accounts. This dataset consists of 11 attributes. It is synthetically generated using the *Paysim* simulator.

Due to different assumptions made in the generated datasets, models trained with separate datasets perform poorly when evaluated with the other datasets. It is difficult to know which assumptions are closest to the real cases scenarios. Therefore, it is wise to train machine learning models with all the datasets merged together.

4.2. Additional attributes

There are limited overlapping attributes in all three datasets, for this reason the additional attributes are creating from time, action and amount. Hence, 11 additional overlapping attributes are created (Table 2). These additional attributes together with transaction amount and money laundering status are being used in further research.

Table 2
Additional attributes used in modeling

#	Additional attributes	*Mahootika*	*AMLSim*	*Paysim*
1	Action count	✓	✓	✓
2	Minimum amount	✓	✓	✓
3	Maximum amount	✓	✓	✓
4	Mean amount	✓	✓	✓
5	Median amount	✓	✓	✓
6	Coefficient variation amount	✓	✓	✓
7	Previous transactions average	✓	✓	✓
8	Same amount count	✓	✓	✓
9	Time difference	✓	✓	✓
10	Coefficient variation time difference	✓	✓	✓
11	Same amount time difference	✓	✓	✓

Table 3
Testing results based on merged training datasets

Train	Test	Metrics	Single methods				Ensemble	
			RF	LR	XGBoost	IF	ALL	LR out
Balanced 70% all datasets (unbalanced for IF model)	30% Mahootika	Accuracy	62,1%	***51,7%***	67,7%	76,4%	***85,2%***	62,1%
		Sensitivity	***0,7%***	72%	***15,3%***	***76,5%***	65,7%	0,8%
		Specificity	100%	39,2%	100%	76,3%	***97,2%***	***100%***
		F1	***1,5%***	53,2%	***26,5%***	***71,2%***	***77,2%***	1,5%
		AUC	***50,4%***	***55,6%***	***57,7%***	76,4%	***81,5%***	50,4%
	30% AMLSim	Accuracy	97,3%	58,1%	***99,9%***	***75,5%***	99,1%	98,8%
		Sensitivity	97,3%	***58,1%***	***99,9%***	***75,6%***	99,2%	98,8%
		Specificity	***100%***	83,4%	***100%***	12,1%	***84%***	***100%***
		F1	98,6%	***73,5%***	***100%***	***86%***	99,6%	99,8%
		AUC	***98,6%***	***70,7%***	***100%***	43,8%	91,4%	99,4%
	30% Paysim	Accuracy	92,6%	94,2%	92,6%	***99,2%***	94,3%	***99,9%***
		Sensitivity	92,6%	94,3%	92,6%	***99,3%***	94,4%	93,2%
		Specificity	***52,6%***	***48%***	52%	19,8%	***48%***	51,9%
		F1	96,1%	97%	96,1%	***99,6%***	97,1%	96,4%
		AUC	72,6%	***71%***	72,3%	59,5%	***71,2%***	***93%***
	30% all datasets	Accuracy	94,9%	75,8%	96,3%	87,1%	***96,7%***	96%
		Sensitivity	95%	75,8%	96,3%	87,2%	***96,8%***	95,6%
		Specificity	***84,6%***	59,8%	84,4%	32,1%	75,8%	84,4%
		F1	97,4%	86,2%	98,1%	93,1%	***98,3%***	97,9%
		AUC	89,8%	68%	***90,4%***	59,7%	86,3%	90%

4.3. Data pre-processing

All three datasets were generated synthetically using different assumptions, therefore it is difficult to determine which are closest to real-world scenarios. All attributes are normalized separately using Z-score normalization to maintain the uniqueness of the data sets and to cover their assumptions. All datasets are divided into two parts: 70% training and 30% testing, and then all training and testing parts are merged separately into two datasets. The SMOTE approach is used to balance the training part except for Isolation Forest model. Only then machine learning methods testing is performed.

5. Results

To cover as many assumptions as possible, all data sets are merged into one, so that models could be applied to real data with the best possible results.

RF, LR, XGBoost, and IF models are trained with balanced merged training dataset. These trained models are tested with the rest of the merged test data of all datasets. To improve research results, the Ensemble is created. Two variants of ensemble are calculated:

1. Ensemble with all models.
2. Ensemble without LR model.

All the results obtained for the individual and combined datasets are shown in Table 3.

5.1. *Mahootika*

After testing the trained models on the *Mahootika* dataset, the best results has been obtained using Ensemble method on all the models. The accuracy of this Ensemble is 85.2% and the AUC is 81.5%.

5.2. *AMLSim*

Testing of the *AMLSim* dataset has shown even better results. XGBoost has achieved 99.9% accuracy and 100% AUC. The Ensemble method without the LR model has showed very similar results. The accuracy of this Ensemble method is 98.8% and the AUC is 99.4%.

5.3. *Paysim*

After testing the models on the *Paysim* testing part, the best Ensemble method is without the LR model and has an accuracy of 99.87% and detected 51.9% of all money laundering cases.

5.4. Merged dataset

Finally, the models have been applied to all merged datasets test's parts. The highest accuracy is achieved with the Ensemble method of all models - 96.7% and with XGBoost accuracy was 96.3% and AUC - 90.4%.

In conclusion, combining the datasets and using the Ensemble approach for all models is suitable due to bet-

ter models performance. The Ensemble without the LR model has achieved accuracy - 95.98%, with 84.4% of all money laundering cases correctly identified and only 4.4% of all legal payments wrongly identified. The XGBoost model's findings has been even better: accuracy is 96.3%, legitimate payments has been accurately recognized 96.3%, and money laundering cases has been detected, the same as the Ensemble method without the LR model.

6. Conclusion

Money laundering are not commonly addressed in the literature since it is strictly regulated by government entities and financial institutions. We use three publicly available synthetically generated datasets in this study, each with a different set of assumptions. Which ranged from 2,340 to 1,323,234 transactions with 0.13% to 60% of money laundering cases. A total of 11 additional attributes are generated for each dataset for further research. Each attribute of the datasets is normalized by the Z-score method. Then all three datasets are combined into one. The combined dataset is divided into training and testing parts and, if necessary, the data are balanced using the SMOTE method. Finally, the results of all the models are combined into the Ensemble method and the vast majority of models make a decision about instance.

After testing these models, we obtained an AUC of 72.6% to 100%, and both money laundering and legal payments have been well identified. To improve the models, we employ the Ensemble method, in which all methods vote is weighted equally and the class is determined by a majority of the classifiers votes for instance. Accuracy of this method ranged from 85.2% to 99.1%, and AUC from 71.2% to 91.4%.

Moreover, there has been made one modification for Ensemble method, by LR model exclusion. In this case, we have achieved 95.98% accuracy and the model has recognized 95.6% of legal payments, and 84.4% of money laundering cases.

Machine-learning-based methods have been adapted to address the problem of money laundering prevention. The results has shown that these methods detects potential money laundering cases and reduce the number of payments reviewed. However, it is not known which dataset generation assumptions would be closest to our market due to this reason it is recommend in implementation stage to make data verification by experts.

Further research should include deep learning methods and other class balancing techniques.

References

[1] Europol, Crime area: money laundering, 2021. URL: https://www.europol.europa.eu/crime-areas-and-statistics/crime-areas/economic-crime/money-laundering.

[2] Aniruddha, Financial action task force (fatf), 1989. URL: https://www.fatf-gafi.org/about/historyofthefatf/.

[3] I. Kot, The pros and cons of synthetic data, 2021. URL: https://www.dataversity.net/the-pros-and-cons-of-synthetic-data/#.

[4] I. Alarab, S. Prakoonwit, M. I. Nacer, Comparative analysis using supervised learning methods for anti-money laundering in bitcoin, in: Proceedings of the 2020 5th International Conference on Machine Learning Technologies, ACM, 2020-06-19, pp. 11–17. URL: https://dl.acm.org/doi/10.1145/3409073.3409078. doi:10.1145/3409073.3409078.

[5] O. Raiter, Applying supervised machine learning algorithms for fraud detection in anti-money laundering (2021) 13.

[6] E. Badal-Valero, J. A. Alvarez-Jareño, J. M. Pavía, Combining benford's law and machine learning to detect money laundering. an actual spanish court case 282 (2018-01) 24–34. URL: https://linkinghub.elsevier.com/retrieve/pii/S0379073817304644. doi:10.1016/j.forsciint.2017.11.008.

[7] V. Huyen, Machine learning in money laundering detection (2020-05-10).

[8] J. Lorenz, M. I. Silva, D. Aparício, J. T. Ascensão, P. Bizarro, Machine learning methods to detect money laundering in the bitcoin blockchain in the presence of label scarcity (2021-10-05). URL: http://arxiv.org/abs/2005.14635. arXiv:2005.14635.

[9] J. Le, Decision trees in r, DataCamp (2018). https://www.datacamp.com/community/tutorials/decision-trees-R.

[10] xgboost developers, xgboost Release 1.2.0-SNAPSHOT, 2020. URL: https://buildmedia.readthedocs.org/media/pdf/xgboost/latest/xgboost.pdf.

[11] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, Second Edition (The Morgan Kaufmann Series in Data Management Systems), 2 ed., Morgan Kaufmann, 2006.

[12] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 Eighth IEEE International Conference on Data Mining, IEEE, 2008, pp. 413–422. URL: http://ieeexplore.ieee.org/document/4781136/. doi:10.1109/ICDM.2008.17.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique 16 (2002) 321–357.

URL: https://www.jair.org/index.php/jair/article/view/10302. doi:10.1613/jair.953.

[14] S. K. Patro, K. K. Sahu, Normalization: A preprocessing stage (2015) 20–22. URL: http://www.iarjset.com/upload/2015/march-15/IARJSET%205.pdf. doi:10.17148/IARJSET.2015.2305.

[15] J. T. Townsend, Theoretical analysis of an alphabetic confusion matrix 9 (1971-01) 40–50. URL: http://link.springer.com/10.3758/BF03213026. doi:10.3758/BF03213026.

[16] R. Kohavi, F. Provost, Glossary of terms 30 (1998) 271–274. URL: http://link.springer.com/10.1023/A:1017181826899. doi:10.1023/A:1017181826899.

[17] Y. Sasaki, The truth of the f-measure (2007) 5.

[18] M.Mahootika, "money laundering data", *M. Mahootika data set.* [online]. https://www.kaggle.com/maryam1212/money-laundering-data/metadata. accessed on: February 27th, 2022 (2022).

[19] "money laundering data", *AMLSim data set.* [online]. https://www.kaggle.com/anshankul/ibm-amlsim-example-dataset. accessed on: February 27th, 2022 (2022).

[20] "aml detection", *PaySim data set.* [online]. https://www.kaggle.com/x09072993/aml-detection. accessed on: February 27th, 2022 (2022).