Detecting money laundering transactions with machine learning

Martin Jullum, Anders Løland and Ragnar Bang Huseby
Norwegian Computing Center, Oslo, Norway, and
Geir Ånonsen and Johannes Lorentzen
DNB, Oslo, Norway

Abstract

Purpose – The purpose of this paper is to develop, describe and validate a machine learning model for prioritising which financial transactions should be manually investigated for potential money laundering. The model is applied to a large data set from Norway's largest bank, DNB.

Design/methodology/approach – A supervised machine learning model is trained by using three types of historic data: "normal" legal transactions; those flagged as suspicious by the bank's internal alert system; and potential money laundering cases reported to the authorities. The model is trained to predict the probability that a new transaction should be reported, using information such as background information about the sender/receiver, their earlier behaviour and their transaction history.

Findings – The paper demonstrates that the common approach of not using non-reported alerts (i.e. transactions that are investigated but not reported) in the training of the model can lead to sub-optimal results. The same applies to the use of normal (un-investigated) transactions. Our developed method outperforms the bank's current approach in terms of a fair measure of performance.

Originality/value – This research study is one of very few published anti-money laundering (AML) models for suspicious transactions that have been applied to a realistically sized data set. The paper also presents a new performance measure specifically tailored to compare the proposed method to the bank's existing AML system.

Keywords Machine learning, XGBoost, Supervised learning, Suspicious transaction

Paper type Research paper

1. Introduction

The true extent of money laundering transactions is unknown and uncertain, potentially because financial firms lack incentive and tools to estimate the extent of money laundering in their accounts (Reuter and Truman, 2004). In an old report to US Congress (1995), it was estimated that about 0.05-0.1 per cent of the transactions through the Society for Worldwide Interbank Financial Telecommunications system involved money laundering. A meta-analysis by United Nations Office on Drugs and Crime (2011) estimates that the total amount of money laundered through the financial system is equivalent to about 2.7 per cent

Journal of Money Laundering Control
Vol. 23 No. 1, 2020
pp. 173-186
Emerald Publishing Limited
1368-5201
DOI 10.1108/JMLC-07-2019-0055

of global gross domestic product, or US$1.6tn in 2009, while Walker (1999) estimates money laundering to account for US$2.85tn worldwide. As financial fraud of such an extent is a serious threat to societies and economies all over the world (Schott, 2006), it is crucial to detect as many of the fraudulent transactions as possible. The topic of the present paper is thus methodology which can identify the very few suspicious/fraudulent transactions from the very many legitimate transactions.

Historically, alert systems based on a set of fixed threshold rules were used to flag suspicious transactions that then underwent further manual control. Such systems are still being used today. See, for example, Demetis (2018) for an up-to-date study on the practicalities of a UK bank's efforts to combat money laundering.

There are three main problems with current AML systems:

(1) Keeping the rules up-to-date and relevant at all times, as well as deciding how to weigh the different rules, is a Sisyphean task.

(2) While such rules in principle can be found from data analysis (Gao and Ye, 2007; Alexandre and Balsa, 2015), rule-based systems are often too simplistic.

(3) As millions of transactions pass through a typical bank each day, reducing the number of false alerts is of paramount importance (Grint *et al.*, 2017). This is almost as important as increasing the number of correct alerts (Deng *et al.*, 2009).

Owing to these issues, new and bold anti-money laundering (AML) tools are needed.

Both Bolton and Hand (2002) and Sudjianto *et al.* (2010) provide excellent overviews of statistical methods for financial fraud detection.

1.1 Learning methods and previous work

In spite of the clear need for well founded, science-based AML methods, the literature on methods for detecting money laundering is fairly limited (Ngai *et al.*, 2011). The existing literature on AML methods can be grouped into two broad classes:

(1) unsupervised learning (Alexandre and Balsa, 2015; Sudjianto *et al.*, 2010); and

(2) supervised learning (Colladon and Remondi, 2017; Deng *et al.*, 2009; Liu *et al.*, 2008; Savage *et al.*, 2016; Sudjianto *et al.*, 2010).

For (1), the methods try to identify patterns in the data without information on which data correspond to money laundering and not. For (2), latter, the methods attempt to learn the patterns that differentiate between money laundering and legitimate operations by using data where the label/outcome (money laundering or not) is known.

Supervised learning is generally preferable when data with known outcome/labels are available. For AML that is problematic as, in contrast to other types of financial fraud, a financial institution rarely finds out if a money laundering suspect is actually guilty of crime. We can, however, get around this issue by modelling "suspicious" behaviour instead of actual money laundering. In Section 2, we argue that "suspicious" behaviour is actually what most financial institutions are indeed interested in. Thus, we only concentrate on supervised learning hereafter.

Deng *et al.* (2009) propose an active learning procedure through a sequential design method for prioritisation, using a combination of stochastic approximation and D-optimal designs to select the accounts for investigation. The method is applied to both transaction data from a financial institution (unfortunately with only 92 accounts) and a simulation study. Lopez-Rojas and Axelsson (2012) discuss the pros and cons of using synthetic data to detect anomalous transactions, as financial institutions can be reluctant to share data, and new financial services like mobile payment have not yet generated enough data. Liu *et al.* (2008) propose a sequence matching based algorithm to identify suspicious sequences in transactions. Other recent works aim at taking advantage of the inherit social networks (Savage *et al.*, 2016; Colladon and Remondi, 2017). Many of the commonly used solutions are proprietary to the technology provider and often completely opaque to the regulators (Grint *et al.*, 2017). Open research on AML methods is therefore needed. Some solutions are tailored to specific money laundering strategies, such as Riani *et al.* (2018), who target systematic mispricing to detect misinvoicing.

1.2 The present paper

In this paper, we develop a supervised machine learning method for discriminating between legitimate transactions and transactions that are suspicious in terms of money laundering. Our method improves the existing methodology, while also reducing the manual work. Our work stands out from earlier work in the field in a number of ways:

(1) We build our model directly on the transactions, instead of suspicious accounts (Deng *et al.*, 2009) or groups of so-called parties (Savage *et al.*, 2016).

(2) Most supervised AML methods assume that suspicious activities are marked as so by experts, while legitimate activities are actually just randomly sampled from the pool of regular customers – the latter is motivated by the fact that the chance of a random (or normal) activity being suspicious is almost zero (Liu *et al.*, 2008; Deng *et al.*, 2009; Savage *et al.*, 2016).

(3) We broaden the definition of legitimate transactions by including transactions both from:

- random customers; and
- AML alerts that did not result in an AML report.

(4) As we shall see, both types of legitimate transactions are crucial in developing a robust predictive model.

(5) While the data sources and modelling frameworks are somewhat limited in the AML literature (Ngai *et al.*, 2011), we summarise background customer data, transaction information and history and information about any earlier (semi-) suspicious behaviour into a fixed set of well-defined explanatory variables (data on matrix form). In trying to learn a binary outcome (suspicious or legitimate), we train a predictive model by using state-of-the-art supervised machine learning methods with proper model tuning.

(6) The literature seems to lack proper validation of the proposed modelling approaches. We develop a performance measure, allowing for direct comparison between the system currently implemented in the bank and our method, and carry out complete performance comparisons between both our alternative model variants and the existing alert-based system.

(7) Previous AML studies typically work on real but small (Deng *et al.*, 2009) or simulated data sets (Lopez-Rojas and Axelsson, 2012). We apply our methodology to an AML scenario in Norway with a data set that is both real and large. The performance results should thereby closely resemble the expected performance in a real-life scenario. To the best of our knowledge, no AML study of this extent has been published before. (The study of Savage *et al.* (2016) is comparable in size, but

considers two types of transactions only; large cash deposits and international funds transfers.)

Throughout this paper, we refer to each transaction as having a sending and a receiving *party*. These are the individuals or companies that control the sending and receiving accounts. In Section 2, we describe AML in Norway and motivate why it is beneficial to model suspicious *transactions* directly rather than accounts or parties. We also describe the data and how we summarise them into proper explanatory variables. We describe our model for a suspicious transaction in Section 3. In Section 4, we introduce our new performance measure and show the results from our performance study. We discuss and sum up in Section 5.

2. Data and anti-money laundering in Norway

A Norwegian financial institution is required by law (The Norwegian Money Laundering Act, Chapter 3, 2009) to make enquiries and report suspicious transactions to The National Authority for Investigation and Prosecution of Economic and Environmental Crime ("Økokrim" in Norwegian). In 2016, 8,776 suspicious transactions were reported [72 per cent of them reported by banks, and 23 per cent reported by payment solution providers (Økokrim, 2016)]. Whether or not a reported transaction leads to a lawsuit, and thus is defined as money laundering by the judicial power, is in principle not relevant for financial institutions. Their task is *solely* to monitor all transactions passing through their system and to classify each of them as suspicious or not. Although a transaction is reported to the authorities, the customer is in no way warned and may continue his/her financial operations as usual, until the authorities possibly take action. Thus, the very same customer could be reported several times. This is problematic when modelling suspicious accounts (Deng *et al.*, 2009) or parties (Savage *et al.*, 2016), as the same account/party would have multiple, conflicting labels. As reports are made on transaction level, this is unproblematic when modelling transactions directly.

The monitoring of suspicious transactions with respect to money laundering in a typical Norwegian bank goes through three stages: the alert stage; the case stage; and the reporting stage (Figure 1). These stages also apply to DNB, Norway's largest financial group, from which we have our data. All transactions with a customer of the bank go through the initial alert stage, a proprietary system based on a set of rules. Alerted transactions, which seem to be legitimate in terms of a simplified manual process, are left out of further investigation.

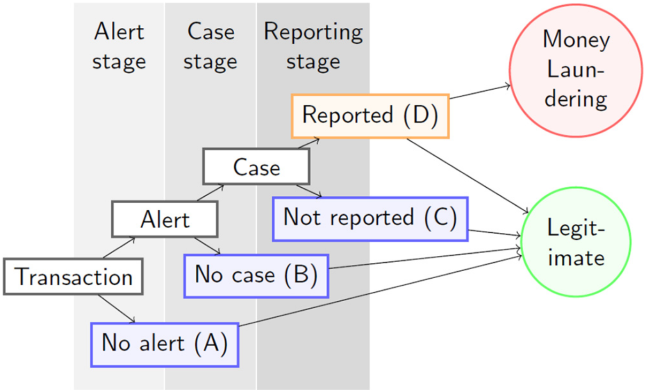

Figure 1.
Typical process of monitoring, investigating and reporting suspicious transactions in a bank

We refer to these as non-reported alerts or no case (B). The remaining alerts are gathered into cases built around a main suspect party and possible related parties. At this stage, several different alerts related to the very same party may be merged into a single case. Finally, these cases are thoroughly investigated by experienced inspectors who make the final decision on whether to report each case to the authorities. At each stage, decisions are made based on information about the transaction itself and the recent transaction history of bank accounts attached to the involved parties. Specifically, the manual investigation in the second and third stages benefit from other background info about the parties, unquantifiable financial information and experience. As discriminating between the reported transactions (D) and all other types of transactions (A), (B), (C) is the sole task for the bank, this is also what we aim to do.

2.1 Data sources and data refinement

In cooperation with the bank, we have been given access to anonymised data on a broad class of alerted transactions spanning from 1 April 2014 to 31 December 2016. We also know which of the transactions that led to a case and which of them ultimately led to reporting. To complement these alert-based transactions, we have access to a purely random selection of normal transactions, from the same time period, which were not given an alert or otherwise investigated. When available, we have background variables for both sending and receiving parties of each of these transactions. In addition, we have the full transaction history for all accounts the parties have access to, two months back in time. Different alerts may be related to the very same suspicious behaviour. In the present setting, alerts that pertain to the same party are merged into a single case in the case stage (Figure 1).

To create a data set as transparent as possible, and to limit the inherent dependence between the *modelled* transactions, we filtered out a number of transactions from our study set. In particular, we ensured that only *one* alerted transaction connected to a specific case is present in our data set. Moreover, the minimum time lag between two transactions from or to the same party is two days. The specifics here were chosen as a compromise to reduce the observation dependence, with a decent amount of data remaining. Note that this refinement was only applied to the *modelled* transactions, and *not* to the data included in the transaction history summaries described in Section 2.3.

2.2 Training and test data

To properly evaluate and compare the performance of different predictive models, we split the data in two. One set is used for training the predictive models, while the other is used only for evaluating the quality of the trained model. To mimic practical AML decisions, we use a time-based splitting rule. Our training set comprises transactions from 1 April 2014 to 30 June 2016, while the test set comprises transactions from 1 July 2016 to 31 December 2016. The time-based splitting should make the performance results representative for the performance that can be expected when using this methodology in practice.

The total amount of data available for training and testing is $n_{\text{train}} = 28{,}167$ and $n_{\text{test}} = 4{,}967$, respectively. For (A), (B), (C) and (D) (Figure 1) there are, respectively, 13,782, 12,746, 1,036 and 603 transactions for the training data, and 2,410, 2,186, 224 and 147 for the test data. For both training and test data, the number of normal transactions (A) were chosen to match the number of non-reported alerts (B) + (C). Other proportions may of course have been chosen here. We discuss this challenge in Section 5. We do not use the full training set ($n_{\text{train}} = 28{,}167$) in all model variants. This choice is described in detail in Section 4.

2.3 Background variables and transaction variables

To model the transactions that should be reported and those that should not, with the supervised learning methodology, we first need to transform all data to a matrix form where each column is a well-defined explanatory variable which means the same across all labelled observations/rows that we are modelling (Whitrow *et al.*, 2009). We consider four different types of explanatory variables:

1. background information about the sending and receiving parties;
2. summary of the transaction history for the sending and receiving parties;
3. summary of information about the suspicious transaction itself; and
4. summary of the outcome of previous alerts and cases where the sending or the receiving party are involved.

The background information for both sending and receiving parties is summarised into $k_{\text{background}} = 30$ explanatory variables of the following type:

- indicator of any previous bankruptcies registered to the party, and the number of years since the last one;
- number of years since the first and last customer relationship was established;
- the type and number of customer relationships registered to the party (individual, corporation or both);
- the sex and ten-year age group of the party (for individuals);
- the nationality of the party;
- the activity level of the party;
- number of years since corporation was established; and
- industry and sector type (for corporations).

We summarise the transaction history for the previous two months for both sending and receiving parties with analogous variables for both debited and credited transactions. The $k_{\text{trans history}} = 1{,}716$ unique explanatory variables constitute:

- The maximum and total amount, and the number of transactions in each of over 100 different currencies.
- The maximum and total amount, and the number of transactions of each of almost 30 different transaction types, such as cash deposit, store purchase, salary, interest rate, manual payment with/without message, subscription payment, pension payment.

We summarise the information about the specific transaction we are modelling into the $k_{\text{current trans}} = 3$ unique explanatory variables: the amount and currency being transferred, and the type of transaction being transferred (like for the transaction history).

Finally, the outcome of any previous alerts and cases for both sending and receiving parties are summarised into $k_{\text{prev behaviour}} = 18$ explanatory variables comprising:

- The proportion of the previous transactions which led to an alert, case and reported case, respectively, registered with the specified party as sender, receiver and either of them.

In total, we have $k_{\text{background}} + k_{\text{trans history}} + k_{\text{current trans}} + k_{\text{prev behaviour}} = 1{,}767$ explanatory variables. After recoding categorical variables as dummy variables (Garavaglia and Sharma, 1998), and removing non-informative variables (i.e. explanatory variables taking only a single value across the training data), we are left with about 1,100 numerical explanatory variables for the model.

3. A model for a suspicious transaction

3.1 Predictive modelling for reporting

For each transaction introduced in Section 2, let Y_i take the value 1 if transaction i was reported to the authorities, and 0 if not. Let $\boldsymbol{x}_i$ denote vectors containing the numerical explanatory variables related to transaction i described in Section 2.3. We attempt to model the probability that a transaction is reported, given its associated explanatory variables (i.e. $\Pr(Y_i = 1 | \boldsymbol{x}_i)$, by $f(\boldsymbol{x}_i) \in [0, 1]$ for some function f (). This is usually done by aiming at minimising the logistic loss:

$$L\big(Y_i, f(\boldsymbol{x}_i)\big) = Y_i \log\big(f(\boldsymbol{x}_i)\big) + (1 - Y_i)\log\big(1 - f(\boldsymbol{x}_i)\big) \tag{1}$$

To fit the model, we use the machine learning library XGBoost (Chen and Guestrin, 2016). XGBoost is very fast, scales to large data sets and also has a graphics processing unit module (Mitchell and Frank, 2017), which can reduce the training time even further (at least an order of magnitude less than the standard central processing unit version).

XGBoost is built around an optimised parallelised (gradient) boosting framework, with tree models as so-called base models. A tree model may be viewed as a decision tree with branches based on explanatory variables and function values in the leaf nodes [Hastie *et al.* (2009), Ch 9.2]. Tree models incorporate non-linearities and interactions directly, do not require much pre-processing, and may be trained in a quite simple and greedy fashion. Owing to their limited predictive power, they are, however, seldom used as stand-alone models, but they suit perfectly as base models in an ensemble model like boosting. Boosting combines base models of a model ensemble to obtain a model which better fits the training data. This is done by iteratively adding new base models to the ensemble, to constantly try to repair the poorest fitting parts of the model. Gradient boosting is a certain type of boosting algorithm which approximates the loss function [here the logistic loss in equation (1)] using its gradient.

In this particular application, we used ten-fold cross validation (CV) [Hastie *et al.* (2009), Ch 7.10] to train the model with the mean AUC (see Section 4.1 for the definition) as the stopping criterion for the number of boosting iterations. To select hyper parameters, we use a method combining a random and an iterative local grid search procedure (Bergstra *et al.*, 2011) that we have developed. Our final predictive model takes the form:

$$f_{\text{final}}(x_i) = \frac{1}{10}\sum_{k=1}^{10} f_{\text{LO fold } k}(\boldsymbol{x}_i) \tag{2}$$

where $f_{\text{LO fold } k}$ ( ) is the model fit when the k-th fold is left out (using the remaining 90 per cent of the training data) when training the model. Taking a pure average of these 10 CV-fitted models is essentially a bagging average (Hastie *et al.*, 2009, Ch 8.7) (with 90 per cent subsampling of the data instead of bootstrapping). This is a more natural choice than learning an ensemble model for combining the predictions from the CV-fitted models into a single predictive model, as the CV-fitted models have exactly the same specifications, just fitted on partly different data. The model described in equation (2) was also found to give better predictive performance than retraining the model with the full training set using the best hyperparameters from the tuning process.

Alternative modelling frameworks were also considered, such as elastic net (Zou and Hastie, 2005) and Random Forest (Hastie *et al.*, 2009, Ch. 15), but none of them were found to be competitive with XGBoost. Combining the XGBoost model with GLMnet, Random Forest or other XGBoost configurations into a single final model using an ensemble technique was also rejected because of the increased model complexity.

3.2 Are three or four classes better than two?

The model described above merged the different types of non-reported transactions into a single class. As additional model configurations, we consider two variants of multiclass models. The first has four classes (A)-(D) (Figure 1). The second has three classes: (A) and (D) constitutes two separate classes, while (B) and (C) are merged into a single class. A multiclass model with K classes can be fitted with XGBoost by combining $K - 1$ binary models, each of them modelling the difference between a reference class and one of the $K - 1$ other classes. This is carried out analogously to how multinomial logistic regression extends regular logistic regression. The multiclass model is introduced to give more flexibility to the model, as the key discriminators may not be the same when attempting to distinguish between, e.g. (A) and (D), as between (C) and (D). That being said, our aim when applying the predictive model is still to distinguish between reported and non-reported transactions, and not between the sub-classes. Therefore, this model is compared directly with the binary version in terms of its predictive performance as a discriminator between the reported transactions (D) and the non-reported transactions (A) + (B) + (C).

4. Results

4.1 Performance metrics

We use three different criteria for measuring the performance of the predictions from the various models on the test set: The Brier score, the AUC and our own invention, the PPP.

Let $p_i = f(\boldsymbol{x}_i)$ and y_i be the i-th prediction and observed true response in the test set. The Brier score (Brier, 1950) then takes the form:

$$\mathrm{BS} = \frac{1}{n_{\mathrm{test}}} \sum_{i=1}^{n_{\mathrm{test}}} (p_i - y_i)^2, \qquad (3)$$

That is, the mean squared error of the predicted probabilities compared to the true response. This is a proper scoring rule (Gneiting and Raftery, 2007), and lower values indicate a better model. It penalises for lack of refinement, meaning that a prediction p_i is far from the true response y_i relative to other predictions. It also penalises for lack of calibration. A calibrated predictor has the property that in the long run, for all observations with a prediction $p_i = p$, a proportion corresponding to p of them have $y_i = 1$, i.e. the predictions are true probabilities.

The area under the [receiver operating characteristic (ROC)] curve, or simply AUC, is another measure of the quality of the predictions (Fawcett, 2006). The ROC curve shows which true positive rate (TPR) corresponds to which false positive rate (FPR) when assigning Class 1 to all predictions above a threshold τ, while moving τ from 1 to 0. The AUC is the area under this curve between 0 and 1. It takes the value 0.5 for completely random predictions and 1 for perfect predictions. As the AUC is based solely on the ranking of the predictions, it penalises only for refinement relative to other predictions and is ignorant to calibration of the predictions. Thus, this is a better measure when the predicted probabilities themselves are not of interest.

We also add 90 per cent confidence intervals (CI) to both the AUC and the Brier scores, computed using Gaussian approximations based on asymptotic theory (DeLong *et al.*, 1988). We consider scores with non-overlapping CI as significantly different.

Both AUC and the Brier scores are commonly used for measuring the performance of a predictive binary model. Unfortunately, we cannot compute such metrics for the alert/case system in use today, as it is a highly manual process without available scores or rankings. We therefore include a third measure which can be compared directly to the current system. This measure is equal to the proportion of all predictions classified as positive when adjusting the classification threshold such that the TPR is at a certain level (say 0.95). This corresponds to the proportion of transactions that needs to be controlled to find 95 per cent of the reported transactions, when sequentially controlling transactions, starting with those having the highest predicted probabilities. Thus, lower values indicate a more efficient classifier. We refer to this "proportion of positive predictions" measure as PPP(TPR = γ) below. The value of PPP(TPR = γ) is directly comparable to γ times the proportion of all transactions in the test set that are manually controlled in the current system:

$$\#(B + C + D)/n_{test} \tag{4}$$

As this measure is not a proper scoring rule, and is rather sensitive to the predictions for which $Y_i = 1$ (especially if γ is close to 1), we use it with care.

Although we consider three performance metrics, the ranking-based AUC is the most important one, as its properties match those which are desirable in the current system. This is also why we use AUC as a stopping criterion (see Section 3.1).

4.2 The value of different classes of legitimate transactions

To properly validate the effect of including the different types of legitimate transactions in the training process of our predictive model, we compare models where both levels of legitimate transactions are used to models which leave out, respectively, the normal transactions and the alerts/cases not leading to reporting. We evaluate the performance using the Brier score and AUC for these different models using both 1) all transactions in the test set and 2) only using the alerted transactions in the test set. The latter is included in order to show that the gain of including normal transactions in the training is only visible when evaluating on a realistic test set where normal transactions are actually present. For the PPP measure, we include results using TPRs of both 0.8 and 0.95 with the full test set. To make the performance study completely fair, the different models need to have the same number of observations available in the training set. Thus, when training the model with all variable types, only half of the normal transactions (A) and half of the non-reported alerts/cases (B) and (C) are included in the training. These are selected at random. This ensures that all models use 13,782 transactions for training, and can be compared under equal terms.

The first message from the binary model comparison shown in Table I is that excluding the transactions stemming from the non-reported alerts/cases while training (column 2), gives a significant performance decrease for all performance measures, compared to the model using all transaction types (column 1). This is natural as non-reported alerts/cases are typically more similar to reported transactions ($Y_i = 1$) than normal transactions ($Y_i = 0$), causing the model to incorrectly assign high probabilities to transactions stemming from the

Table I.
Training scenario
(model type)

Evaluation metric [test data]	All types (binary)	No non-reported alerts/cases (binary)	No normal transactions (binary)	All types (Multiresponse 3)	All types (Multiresponse 4)
AUC [all]	0.907 [0.893, 0.921]	0.852 [0.836, 0.869]	0.872 [0.849,0.895]	0.910 [0.896, 0.924]	0.910 [0.895, 0.924]
AUC [only alerts]	0.822 [0.795, 0.848]	0.706 [0.675, 0.738]	0.819 [0.791,0.847]	0.826 [0.799, 0.854]	0.825 [0.798, 0.853]
Brier [all]	0.025 [0.022, 0.028]	0.340 [0.330, 0.351]	0.024 [0.021,0.028]	0.025 [0.022, 0.027]	0.025 [0.022, 0.027]
Brier [only alerts]	0.047 [0.044, 0.051]	0.655 [0.646, 0.665]	0.047 [0.043,0.051]	0.047 [0.044, 0.051]	0.047 [0.044, 0.051]
PPP(TPR = 0.95) [all]	0.315	0.373	0.452	0.305	0.330
PPP(TPR = 0.8) [all]	0.203	0.260	0.268	0.195	0.196

Notes: Table I. Performance scores on the test set when training the model on all transactions (A-D), excluding normal transactions (A) and excluding non-reported alerts/cases (B + C) in the training. The two rightmost columns contain performance scores on the test set with a multiresponse with three or four outcomes. All models are trained on 13,782 transactions. The Brier and AUC scores are evaluated on test sets with either only alert data ($n_{\text{test, only alerts}} = 2{,}557$) or all data (alert data and normal transactions, $n_{\text{test, all}} = 4{,}967$). The Brier and AUC scores are given with 90 per cent CI in brackets non-reported alerts/cases. Relative to the reported transactions, this has a greater negative impact on the predicted probabilities for the non-reported alerts/cases (too high probabilities), than their ranking. As a consequence, the Brier score is affected more by this than the other criteria.

The second message is that not using normal transactions when training (column 3) gives no decrease in performance when considering only alerted transactions in the test set. There is, however, clearly a decrease in performance when all transactions are considered (in terms of AUC and PPP). The Brier score is more or less unaffected by the exclusion of the normal transactions, quite likely since the degree of over/underfitting is optimised with respect to AUC (the stopping criterion), which is ignorant to calibration. Note that the performance for "only alerts" in the test set does not decline when including normal transactions, demonstrating that including the normal transactions does not confuse the rest of the predictive model. To compare the efficiency of the predictive models with the current system, we rely on the PPP scores. As noted above, these scores are directly comparable to the proportions of the transactions being manually investigated by the bank in the current system (times γ). In our test set these scores are 0.489 and 0.412 ($\gamma \cdot 0.515$), for $\gamma = 0.95$ and $\gamma = 0.8$, respectively. 0.515 is here found equation (4). Thus, using all transaction classes in the training of the binary model, we get a reduction in the number of transactions we need to consider of $(0.95 - 0.515 - 0.315)/(0.95 - 0.515) = 36$ per cent and $(0.8 - 0.515 - 0.203)/(0.8 - 0.515) = 51$ per cent when requiring detection of, respectively, 95 per cent and 80 per cent of the reported transactions. This is a major improvement compared to the current rule based system.

4.3 Multiclass model

As mentioned in Section 3.2, we fit both three and four class models, which handle the non-reported transaction types as separate classes. As our interest is still in separating non-reported and reported transactions, we use the performance measures for the binary model when comparing the multiclass models to the binary ones. The results are found in Table I in the two rightmost columns and should be compared with the "All types (binary)" column.

There is a tiny performance increase in terms of AUC for both multiclass models. Although this performance increase is far from significant, it could stem from the multiclass models concentrating on the distinguishing features of one class at a time, which is beneficial for the performance. Note, however, that when using the four class model, the PPP with TPR = 0.95 increases. This is perhaps a result of the predictive model becoming too busy attempting to distinguish between the different subclasses, while the PPP measure is solely concerned with distinguishing the reported cases from the rest.

5. Discussion and conclusion

We constructed and properly validated a machine learning model for prioritising which transactions should be further investigated by AML investigators. We demonstrated that the common approach of ignoring non-reported alerts/cases in the training of the model can lead to far from optimal results. The unfortunate habit of ignoring non-reported alerts/cases could be because of the fact that these alerts/cases are simply not stored, as they are not considered important in the day-to-day activities. On the other hand, it is required by law to report cases, and normal transactions are always available. As shown in our study, the ideal data set comprises all data; transactions related to reported cases, non-reported alerts/cases and normal transactions. We carried out the time validated analysis using a comprehensive, real data set from Norway. Our results should therefore closely resemble those expected in a production setting.

In a real-life setting, the procedure to detect suspicious transactions would be run through all transactions for a certain time period. The small number of reported cases would make this highly inefficient and too time-consuming for the present study. Instead, we therefore applied our trained model on a test set of the same type as the training set, where there are as many normal transactions (A) as there are non-reported alerted transactions (B) + (C). This test set is less imbalanced than the set of all transactions within a time period, but it is still heavily imbalanced (with 147 reported versus 4,820 non-reported transactions). With a higher proportion of normal transactions in the test set, we would most likely see slightly lower Brier and PPP scores, and slightly higher AUC scores. The reason for this is that a higher proportion of the non-reported transactions is easier to distinguish from the reported ones. In a production setting, we therefore expect slightly better results than those reported in Table I.

We used two months of history when summarising the transaction history of every party. This was chosen in cooperation with AML experts. However, our modelling procedure can be carried out using a longer or shorter transaction history, or using multiple history lengths (for example the past week, month and year). As this will increase the number of explanatory variables substantially, investigation of the effect of multiple history lengths was therefore deemed out of scope for the current paper.

Our approach can be extended further in many ways. One approach is to include information on *who* the funds are transferred to, i.e. how the cash flows through the financial network around every account and party (Savage *et al.*, 2016; Colladon and Remondi, 2017). Such data were unfortunately not available for the present study. As the AML process is stepwise, we also envision that it can be beneficial with a hierarchical model (Gelman and Hill, 2006) with different levels of suspiciousness as an extension of the (flat) multiclass model.

Our approach is preferable to the current rule-based approaches that many banks rely on. Instead of keeping the rules up-to-date at all times, our model has relatively few assumptions about the money laundering patterns and can continuously adapt and learn from new data.

Although the XGBoost framework is efficient, computing the set of explanatory variables themselves can be computationally demanding because of the sheer number of transactions that go through a bank. The most straightforward method for updating each of the transaction variables would require a daily scan of the two-month transaction history of every DNB customer. Instead, by temporarily storing the values of the explanatory variables for each day in the past two months, only a scan through the transactions history for the current day, and some clever bookkeeping, would be sufficient for maximums/minimums, means and other combinations of statistical moments (Pebay, 2008).

Finally, our model predicts the probability that a *transaction* should be reported. With minimal effort, the same approach could instead be used to predict the probability that a *customer* should be reported.

References

Alexandre, C. and Balsa, J. (2015), "Client profiling for an anti-money laundering system", arXiv preprint arXiv:1510.00878.

Bergstra, J.S., Bardenet, R., Bengio, Y. and Kégl, B. (2011), "Algorithms for hyper-parameter optimization", *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pp. 2546-2554.

Bolton, R.J. and Hand, D.J. (2002), "Statistical fraud detection: a review (with discussion)", *Statistical Science*, Vol. 17, pp. 235-249.

Brier, G.W. (1950), "Verification of forecasts expressed in terms of probability", *Monthly Weather Review*, Vol. 78 No. 1, pp. 1-3.

Chen, T. and Guestrin, C. (2016), "Xgboost: a scalable tree boosting system", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *ACM*, pp. 785-794.

Colladon, A.F. and Remondi, E. (2017), "Using social network analysis to prevent money laundering", *Expert Systems with Applications*, Vol. 67, pp. 49-58.

DeLong, E.R., DeLong, D.M. and Clarke-Pearson, D.L. (1988), "Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach", *Biometrics*, Vol. 44 No. 3, pp. 837-845.

Demetis, D.S. (2018), "Fighting money laundering with technology: a case study of bank x in the UK", *Decision Support Systems*, Vol. 105, pp. 96-107.

Deng, X., Joseph, V.R., Sudjianto, A. and Wu, C.J. (2009), "Active learning through sequential design, with applications to detection of money laun- dering", *Journal of the American Statistical Association*, Vol. 104 No. 487, pp. 969-981.

Fawcett, T. (2006), "An introduction to ROC analysis", *Pattern Recognition Letters*, Vol. 27 No. 8, pp. 861-874.

Gao, Z. and Ye, M. (2007), "A framework for data mining-based anti-money laundering research", *Journal of Money Laundering Control*, Vol. 10 No. 2, pp. 170-179.

Garavaglia, S. and Sharma, A. (1998), "A smart guide to dummy variables: four applications and a macro", *Proceedings of the Northeast SAS Users Group Conference*, p. 43.

Gelman, A. and Hill, J. (2006), *Data Analysis Using Regression and Multi-Level/Hierarchical Models*, Cambridge university press, Cambridge.

Gneiting, T. and Raftery, A.E. (2007), "Strictly proper scoring rules, prediction, and estimation", *Journal of the American Statistical Association*, Vol. 102 No. 477, pp. 359-378.

Grint, R. O'Driscoll, C. and Patton, S. (2017), "New technologies and anti-money laundering compliance report", available at: www.fca.org.uk/publications/research/new-technologies-and-anti-money-laundering-compliance-report

Hastie, T., Tibshirani, R. and Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer, Berlin.

Liu, X., Zhang, P. and Zeng, D. (2008), "Sequence matching for suspicious activity detection in anti-money laundering", Proceedings of the IEEE ISI 2008 PAISI, PACCF, and SOCO international workshops on Intelligence and Security Informatics, *Springer*, pp. 50-61.

Lopez-Rojas, E.A. and Axelsson, S. (2012), *Money Laundering Detection Using Synthetic Data, the 27th Annual Workshop of the Swedish Artificial Intelligence Society (SAIS), 14-15 May 2012, Link«oping University Electronic Press*, *Örebro*, pp. 33-40.

Mitchell, R. and Frank, E. (2017), "Accelerating the XGBoost algorithm using GPU computing", *PeerJ Computer Science*, Vol. 3, p. e127.

Ngai, E., Hu, Y., Wong, Y., Chen, Y. and Sun, X. (2011), "The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature", *Decision Support Systems*, Vol. 50 No. 3, pp. 559-569.

Økokrim (2016), *Annual Report. The National Authority for Investigation and Prosecution of Economic and Environmental Crime*, available at: www.okokrim.no/getfile.php/3881783.2528.7nannji7ut7tj7/2016_AnnualReport_okokrim.pdf

Pebay, P.P. (2008), "Formulas for robust, one-pass parallel computation of covariances and arbitrary-order statistical moments", Technical Report SAND2008-6212, Sandia National Laboratories.

Reuter, P. and Truman, E.M. (2004), *Chasing Dirty Money: The Fight against Money Laundering*, Peterson Institute for International Economics, Washington, DC.

Riani, M., Corbellini, A. and Atkinson, A.C. (2018), "The use of prior information in very robust regression for fraud detection", *International Statistical Review*, Vol. 86 No. 2, pp. 205-218.

Savage, D. Wang, Q. Chou, P. Zhang, X. and Yu, X. (2016), "Detection of money laundering groups using supervised learning in networks", arXiv preprint arXiv:1608.00708.

Schott, P.A. (2006), *Reference Guide to anti-Money Laundering and Combating the Financing of Terrorism*, The World Bank, Washington, DC.

Sudjianto, A., Nair, S., Yuan, M., Zhang, A., Kern, D. and Cela-Díaz, F. (2010), "Statistical methods for fighting financial crimes", *Technometrics*, Vol. 52 No. 1, pp. 5-19.

The Norwegian Money Laundering Act, Chapter 3 (2009), "The Norwegian money laundering act, chapter 3", In Norwegian, available at: https://lovdata.no/dokument/NL/lov/2009-03-06-11#KAPITTEL_3 (accessed 15 January 2018).

US Congress (1995), "Office of Technology Assessment, Information Tech- neologies for Control of Money Laundering", Technical report, OTA-ITC- 630, Government Printing Office, Washington, DC.

United Nations Office on Drugs and Crime (2011). "Estimating illicit financial flows resulting from drug trafficking and other transnational organized crimes", available at: www.unodc.org/documents/data-and-analysis/Studies/Illicit_financial_flows_2011_web.pdf

Walker, J. (1999), "How big is global money laundering?", *Journal of Money Laundering Control*, Vol. 3 No. 1, pp. 25-37.

Whitrow, C., Hand, D.J., Juszczak, P., Weston, D. and Adams, N.M. (2009), "Transaction aggregation as a strategy for credit card fraud detection", *Data Mining and Knowledge Discovery*, Vol. 18 No. 1, pp. 30-55.

Zou, H. and Hastie, T. (2005), "Regularization and variable selection via the elastic net", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Vol. 67 No. 2, pp. 301-320.

Corresponding author
Martin Jullum can be contacted at: jullum@nr.no