Knowl Inf Syst (2018) 57:245–285
https://doi.org/10.1007/s10115-017-1144-z

SURVEY PAPER

Machine learning techniques for anti-money laundering (AML) solutions in suspicious transaction detection: a review

Zhiyuan Chen[1] · Le Dinh Van Khoa[1] · Ee Na Teoh[2] · Amril Nazir[2] · Ettikan Kandasamy Karuppiah[2] · Kim Sim Lam[1]

Received: 31 October 2016 / Revised: 18 September 2017 / Accepted: 27 December 2017 /
Published online: 10 February 2018

Abstract Money laundering has been affecting the global economy for many years. Large sums of money are laundered every year, posing a threat to the global economy and its security. Money laundering encompasses illegal activities that are used to make illegally acquired funds appear legal and legitimate. This paper aims to provide a comprehensive survey of machine learning algorithms and methods applied to detect suspicious transactions. In particular, solutions of anti-money laundering typologies, link analysis, behavioural modelling, risk scoring, anomaly detection, and geographic capability have been identified and analysed. Key steps of data preparation, data transformation, and data analytics techniques have been discussed; existing machine learning algorithms and methods described in the literature have been categorised, summarised, and compared. Finally, what techniques were lacking or under-addressed in the existing research has been elaborated with the purpose of pinpointing future research directions.

Keywords Anti-money laundering · Data mining methods and algorithms · Supervised learning · Unsupervised learning · Anti-money laundering typologies · Link analysis · Behavioural modelling · Risk scoring · Anomaly detection · Geographic capability

1 Introduction

Financial crime detection has become an important priority for banks. Fraud is increasing dramatically with the expansion of modern technology and global communication, resulting in substantial losses to the businesses [1]. Recently, HSBC was fined by the US authorities a large sum of $1.9bn (£1.2bn) in a settlement over money laundering, the largest paid in such

✉ Zhiyuan Chen
Zhiyuan.Chen@nottingham.edu.my

1 Faculty of Science, School of Computer Science, University of Nottingham, Malaysia Campus, Jalan Broga, 43500 Semenyih, Selangor Darul Ehsan, Malaysia

2 Accelerative Technology Lab of ICT Division at MIMOS Bhd, 57000 Kuala Lumpur, Malaysia

a case [2,3]. It was criticised for having poor money laundering controls. HSBC is not the only bank penalised for its lack of money laundering controls. In 2009, Switzerland's Credit Suisse Group was fined $536 million and the UK's Lloyds Banking Group PLC was fined $350 million for money laundering allegations [4,5]. The ING Bank Group was also fined a substantial penalty of $619 million for enabling launderers to illegally moving billions of dollars through the US banking system [6].

More recently, in 2014 Standard Bank PLC was fined 7.6 m for failures in its anti-money laundering controls [7]. Tracey McDermott, director of enforcement and financial crime in the UK, made a statement: 'One of the Financial Services Authority's (FSA) objectives is to protect and enhance the integrity of the financial system'. Banks are in the front line of the fight against money laundering. If they accept business from high-risk customers, they must have effective systems, controls, and practices in place to manage the risk [8].

There is a need to have effective anti-money laundering (AML) systems, controls, and practices in place to manage the risk of money laundering activities for banks. They will now have to take serious measures against any possible risk of money laundering activities. Such controls must be audited against any future allegations that can be costly to the banks.

In the past few decades, an operational model that is based on embedded rules was very popular among the community where the rules were simple and easy to code, and normally they were developed by consultants and domain experts who implement their working experience into the automated decision process. For example, a system starts with 100 scenarios and 100 rules to handle money laundering problem. As time goes by, more rules need to be created as more exceptions arise; but the integration between new rules and existing rules will impair the system performance when dealing with the dynamically changing data. Meanwhile, it is extraordinarily difficult to evaluate the performance of all rules and to identify all exceptions.

Structured and constrained data will ensure the performance of the rule-based system. However, high volume dataset with different types of structured or semi-structured data and unstructured data seems to be a fatal problem to rule-based system too. Therefore, machine learning is more emphasised by the programming community nowadays. The most perceptible function with machine learning is its ability to create generalisations on the data during the training phase. Also, it will learn the frontiers between decision boundaries automatically. The machine learning algorithm uses the information gained from training phase can effectively interpolate or extrapolate on new scenarios without being taught ever of a possibility. In contrast, with the rule-based system, expert knowledge is mandatory in order to create new rules.

Money laundering activities could occur anywhere because there are various ways to launder money. In this survey, we focus on AML detection solutions for suspicious transactions in financial institution, especially retail or commercial banks. AML solutions, being part of the overall fraud control, automate and help to reduce the manual work of a screening/checking process. Currently, most banks rely on rule-based systems to filter out any suspicious transactions based on pre-generated static rules. It is impossible to be certain of the legitimacy and intention behind an application or transaction. Given this reality, the best cost effective option is to determine the possible evidence of abnormal activities from the available data using machine learning techniques.

Numerous research communities, especially those from developed countries, are still using analytical engines and softwares, which are driven by financial institution predefined rules [9,10]. More recently, researchers have begun to venture on the feasibility and the practicality of artificial intelligence and machine learning techniques such as fuzzy logic, support vector

machines (SVM), clustering, outlier detection, neural networks, genetic algorithms, Bayesian network, and sequence matching in order to better evaluate suspicious transactions [11–16].

There are many criticisms of artificial intelligence-based AML research too. The lack of publicly available real data to perform experiments and the detection of suspicious transactions only relies on discovered transaction are main concerns. The other disadvantage is the cost incurred in the maintenance and repair where programs need to be updated to adapt for the changing requirements. The effectiveness of a machine learning algorithm is largely influenced by the unique characteristics of financial money laundering data. It is, therefore, crucial to understand the strengths and the limitations of each algorithm when applied to money laundering problem. This requires a global view of many research efforts. Most efforts focus on how a single algorithm or a few combined algorithms are applied. There is a lack of comprehensive survey that summarises and compares various machine learning algorithms in the field of anti-money laundering solutions.

To address these issues, this paper surveys the most recently published literature of machine learning techniques for AML solutions in suspicious transaction detection area. Innovative algorithms and methods have been reviewed. These techniques have covered wide area of machine learning, from supervised to unsupervised and semi-supervised. Besides, they also investigate the money laundering pattern in various financial circumstances. For example, Liu and Zhang [14] have proposed a suspicious activity recognition method based on scan statistics to identify suspicious transaction for financial institutions and Cahill et al. [11] have proposed a methodology that was developed on tracking calling behaviours on an account over time and scoring calls to detect fraud. Also, Gao and Ye [12] have proposed a framework for DM-based AML research which was based on suspicious data preparations and rare transactional pattern recognition. Furthermore, Yue et al. [16] have proposed a generic FFD framework for understanding and classifying different combinations of data mining algorithms and financial fraud detection techniques. Although lot of techniques have been proposed, little attention concentrated on comparing the effectiveness in different phases of AML procedures.

The rest of this paper is organised as follows. Section 2 describes the most important preprocessing and analysis steps for an AML solution. Section 3 presents existing detection tools for AML solutions that apply machine learning algorithms and statistical measurements. In Sect. 4, commercial softwares that provide anti-money laundering solutions are discussed, and we conclude our survey paper in Sect. 5.

2 AML pre/processing phases

A successful detection system can recognise trends or patterns of suspicious behaviour by generating the alarm which indicates the likelihood of a fraud. Once the score is above a certain threshold, the case will be further investigated [17,18]. There are two phases to monitor suspicious behaviours. It starts before the account is opened, and then, it switches to monitoring the activity of the new account. A full detection process is available in criminal networks as robust and resilient organisation [13].

Challenges for analysing such data can be categorised into five groups, namely volume and complexity of data, class imbalance, concept drift, class overlap, and class mislabelling [15]. An effective AML solution should also provide a comprehensive and concrete reasoning

in suspicious transaction reporting. A useful report comes from an effective identification, monitoring, and managing of the risks of money laundering activities. In achieving the success, AML solution tools that are used to identify suspicious transactions should have the following important features: data quality, detection accuracy, scalability, and reaction time.

Data quality refers to the ability of the system to ensure data accuracy, correctness, completeness, and relevance. Data quality assurance and detection accuracy are tightly coupled since detection accuracy cannot be obtained without a considerable amount of data quality assurance in investigating and describing the various categories of desirable attributes (or dimensions) of data [17]. Detection accuracy refers to the capability of the system to reduce the number of false positives (i.e. low errors incurred by false alerts) while maintaining a high detection rate [19]. Scalability refers to the capability of handling the immense volume of financial data. It is often the norm for a large-sized bank to deal with several millions of transactions on a daily basis. Hence, when the volume of data increases, scalability is important to ensure that the system works effectively [20].

Finally, reaction time refers to the capability of the system to rapidly process the large volume of financial data in terms of the millions of transactions within a certain time period. A state-of-the-art AML solution should have the capability to process tens of millions of transactions for alarms in a few h, min or less, allowing analysts to conduct their investigations in a timely manner. In the next subsections, important phases in AML pre/processing are discussed.

2.1 Data preparation

The raw data collected from banking transactions should be processed to represent transaction behaviours of banking customers. Further, by examining investment activities across different time intervals, additional insights can be obtained to detect abnormal behaviours. The raw data (semi-structured or unstructured) must be processed into a form that is accurate, correct, and meaningful before being run through machine learning algorithms. However, data transformation involves a considerable amount of effort in data pre-processing [17]. Due to a large amount of transaction data (e.g. 5 million transaction activities daily), data processing constitutes a large amount of work where it takes most of the total time in the money laundering detection process.

While it has been accepted, pre-processing for AML solution is a very challenging and time-consuming task. There are a few existing publications which have addressed this issue. For example, Sudjianto et al. [15] proposed three engineered features/attributes (peer comparison score in terms of transaction volume, transaction speed score, and individual expected level of activity score) to train various machines learning algorithms (tree based, logistic regression, neural network and Bayesian belief networks) and give performance comparison; Zhang et al. [21] spend a large amount of their time pre-processing text documents and extracting features from them.

Furthermore, different AML analytic tools will need different input data format, for example, BeyondCore [22] requires data input as table, view, or comma separated (.csv) flat file of rows and columns, SAS [23] needs to create a SAS Data File or a SAS Data View, INETCO Analytics [24] can only take an appropriate form for things such as ATM analytics and modelling (e.g. queuing, transaction sequence, cash balances), omni-channel and digital banking analysis, and card analytics. Next, various possible ways of the data preparation stage in the field of AML solutions are discussed.

2.1.1 Prepare data structure

Raw transaction data from financial institutions can come from multiple data sources such as a relational database, Oracle, Microsoft SQL, and PostgreSQL. Even the data structure is in a table format, the complexity level is still very high. For example, individual personal information is held in one table while transaction information is held in another a table. It is often the norm for a bank to hold several millions of customer accounts; each account may have thousands of past transactions. These can add to hundreds of millions of raw transactions. With more than one table, the rows to be combined are O(nm) (in which n represents the number of rows and m represents the number of columns) computational complexity, whereby each table must be examined for matching rows. With such complexity, the joining operation consumes huge computational power since records from multiple tables need to be streamlined based on primary keys (i.e. customer number and account numbers), and any mismatched records and duplications must be identified and eliminated [25].

If the raw data stored in a dimensional data warehouse or transactional database format, BeyondCore [22] will need to use record identifiers or primary keys to join fields from multiple tables to create a single unified, flattened view. If multiple input files as instream input data to SAS [23], DATA step problem will read the records in each file and creates the ALL_ERRORS SAS data set. For INETCO Analytics [24], raw data are firstly run through a PCI compliant security model to eliminate sensitive information, and tokenize individual card numbers. Hence, preparation of the raw data can take a considerable amount of time.

2.1.2 Create individual/peer profile

Another useful step of data preparation for AML solutions is to consolidate raw transactions based on an individual or specific peer group feature [26]. By using a unique customer identification and/or specific peer group identification, it is possible to obtain a global view of an individuals behaviour over time. Similarly, the data pre-processing steps either involve splitting the raw transactions into multiple tables, whereby each intermediate table is represented by one customer, or the raw transactions need to be queried each time to create a specific customer profile. Either way, the complexity is significantly high for millions of transaction records.

2.1.3 Consolidate raw transactions

There is a need to consolidate raw transactions across time horizons in order to capture a global view of individual transaction activities. To achieve this, individual account information needs to be merged based on specific time horizons, e.g. day, week, and month. For example, the total sending amount of an individual for a week can be obtained by summing up all daily sending transactions from day 1 to day 7. From a database perspective, this consolidation can be performed by table joining operations on customer identification and transaction date attributes from the relevant tables [27,28]. The output can then be stored and included in the customer profile. Similarly, given the vast amount of transactions, these operations are computationally extensive.

2.1.4 Create new, meaningful attributes

The process of transforming raw transactions into new and more relevant information is a continuous process. Based on the specific problem domain, new and meaningful attributes

might need to be created and included in the algorithm training process. For example, it may be helpful to have the average/mean and standard deviation of the daily, weekly, and monthly incoming transaction amounts. The largest credit transaction amounts and transaction frequency on a daily, weekly, and monthly basis may be useful as well. Such additional attributes can increase the detection accuracy if it gives new information, which is not captured with/by other existing attributes [18].

In the majority of the research works published in the AML field, the chosen attributes used to train the machine learning systems are carefully hand-picked by the domain experts. Most literature often cites information on what attributes/features are used for a particular machine learning training, but no detailed information is provided on the feature selection process. An automated feature selection process is important due to the nature of banking transaction data that have diverse dimensions.

On the other hand, feature extraction is a machine learning technique that creates new attributes from functions of the original attributes. Feature extraction involves converting existing attributes into a new attribute that can accurately represent a large set of data. For example, a bank transaction in AML solution consists of a large number of attributes (e.g. over 50 attributes). In such a case, feature extraction is useful to reduce the dimensionality of the data by converting the existing attributes into a new set of attributes while retaining an accurate representation of the data. This is also useful to reduce data noise since noisy data can have a detrimental effect on the machine learning performance accuracy [29].

2.1.5 Handling missing values

Missing values is a common issue when extracting data from the transactional banking database. The cause could be various, such as due to omission, irrelevance or inapplicability in a specific context. In particular, some customers are not willing to provide/share their annual income, occupation, or even valid addresses to a financial institution due to privacy concern. However, most machine learning algorithms are very sensitive to missing values as it can adversely affect the algorithm performance. Missing values can be handled by either removing all transactions that contain missing values or replacing the missing values using filtering algorithms, such as mean substitution, cold deck imputation, and hot deck imputation methods [30,31].

2.2 Data analytics

Once the data preparation steps are completed, the data analytics operations can then be carried out. The operations involve using the cleansed data to make a prediction with the aim to improve the alert quality and accuracy. The overall goal is to significantly lower false positives while maintaining accurate alerts, which reduces the staffing hours required for investigations. Two features that are considered to be currently lacking in the existing solutions are discussed in the following two sections.

2.2.1 AML system with human interaction

In order to maintain the accuracy of the detection rate and to reduce the number of false positives, a state-of-the-art AML system must not only automate an analysis of the transactions, but also should have the ability to handle discoveries of new abnormal patterns from unseen transactions and to predict unseen cases without diminishing the accuracy of the detection performance. One way of achieving this is to allow human expert interaction with

the detection system so that the machine learning model can be adjusted to adapt to different behaviours of crimes.

2.2.2 Process and analyse large data volumes

Current AML solutions heavily rely on relational databases to store and manage data retrieval and data pre-processing capabilities. However, conventional relational databases have severe performance bottlenecks when dealing with large volumes of datasets (i.e. millions of customer transactions on daily basis). Recent big data technologies such as Hadoop [32] can be cost effective to manage and provide the capability of ingesting and analysing large volumes of data. It is also an ideal environment for extracting and transforming huge volumes of data as it provides a scalable, reliable, and distributed processing environment.

3 Machine learning algorithms used

Researchers have been studying and implementing machine learning algorithms to automatically learn and recognise patterns in large amounts of data. There are a great variety of machine learning techniques within the available literature that make comparisons difficult. The machine learning algorithms for AML solutions in suspicious transaction detection are generally categorised into two main groups in literature: supervised and unsupervised techniques.

Supervised techniques refer to algorithms that learn from a set of labelled examples, known as the training set [33,34]. For example, the label can be either Normal or Suspicious, or represents as a binary variable. Based on the training examples, a supervised learning model is built to classify new unseen data (known as the testing set) into different label categories.

Alternatively, unsupervised techniques consist of algorithms that try to separate data examples (without label information) into different groups. These groups, known as clusters, hold their unique characteristics or patterns and define by similar member instances. Although sometimes unsupervised learning technique also contains training set, the data label will be omitted during learning process. The label will be used to evaluate the final decision when clusters have been formed [33,34].

It should be noted that current commercial AML solution system consists of multiple detection tools to identify suspicious transactions [26]. Since tremendous approaches have been implemented for AML cases, this paper will not try to cover all of them. Discussion of these AML solutions for this study is focused on six common aspects, which are AML typologies, link analysis, cross-channel support geographic capability, behavioural modelling, risk scoring, and anomaly detection. These aspects are generated from the know your customer guideline (KYC) which is a standard procedure for financial institution against AML [35,36].

The intuitive idea of KYC is to reinforce the due diligence principles. According to financial institutes rules, the flow of money must be fully tracked. As a consequence, information includes customers identity, occupation, frequent financing activities, credit report, tax number and annual income must be recorded to identify the origin of money. Due to various financial services an institute provides to customers, each type of transaction activities requires specific information on it. Thus, the AML techniques should be designed to handle different type of transactions in various scenarios. Common AML procedures define important factors to captured fully transaction information in specific perspectives.

The AML typologies concentrate on generating a model that defines the AML cases captured from the past. As following the KYC guidelines, the institute must provide a thoroughly

understanding on the beneficial owner and parties related to the transactions. In particular, this composed of the source, the purpose of funds, the appropriateness and reasonableness of the business activity. The link analysis is useful to extract these information while behavioural modelling can capture the pattern of transactions in the context of business. In the case of risk scoring, the KYC module estimates risk scores for customers based on financial institution predefined static factors including geography, nationality, business and occupation type, or income and source of funds. The risk score is compared with the adjustable thresholds, set by financial institution, to place customers into specific categories. This allows the financial institution to apply a risk based approach to support their KYC and AML Policy.

Appropriate solutions that are commonly implemented in the six aspects will be examined in this survey. In the following sections, AML solutions proposed by various authors will be discussed. Some proposed methods may consist of only one detection tool, and others may have a combination of different tools with the aim to improve the detection performance. In this paper, we have categorised all these techniques into six aspects for a better understanding of AML solutions.

3.1 AML typologies

AML typologies refer to the ability of an algorithm to detect and prevent similar money laundering cases that had occurred before. In this section, the approaches proposed by various authors are reviewed. The importance of AML typologies focused on the detection rate (DR) in suspicious money laundering cases that have been previously detected. DR refers to the number of suspicious transactions detected by the system divided by the number of suspicious instances presented in the test set. Typically, AML typologies method requires a set of training data (i.e. labelled data) that contains information of previously identified suspicious transactions and normal transactions. Various algorithms are discussed in this section on how authors use the labelled dataset to train a learning model and test the model using unseen data (i.e. unlabelled data) in order to measure the performance of the proposed algorithms.

3.1.1 Fuzzy rules

Anti-money laundering software tools and systems in the market are normally rule based that generate alerts using some sets of pre-defined rules and thresholds. A standard fuzzy logic control system is comprised by three sections fuzzifier, inference engine, and defuzzifier. A set of knowledges based defined by fuzzy conditional rules is designed by domain expert to determine process. The fuzzy set theory was presented by Zadeh to identify the problem of vague information [37] which is commonly appear in AML field. In fuzzification phase, raw input will be converted by antecedent part in membership functions before sending data to inference engine. Suppose a fuzzy set A is defined on the universe of discourse $[a, b]$, the fuzzification result of $x \in [a, b]$ denotes as $\mu_A(x)$.

The fuzzy rule based calculates the rule strength to direct flow of input variables known as consequence of the rule to determine output branch. Hence, it represents the knowledge and experience of expert domain that map input from fuzzy controller to the output. According to [38], there are two major types of fuzzy rules known as Mamdani fuzzy rules [39] and Takagi–Sugeno (TS) [40] fuzzy rules. In order to calculate the combined output from the rule antecedent, fuzzy inference methods are required. The Mamdani and TS fuzzy rules applied different fuzzy inference methods. Among various options, there are four popular solutions for Mamdani approach in fuzzy control and modelling. Supposed $\mu_A(x)$ is the membership

function of fuzzy set A and μ is the combined membership in the sequences of rule, the four fuzzy inferences, Mamdani minimum inference R_{M}, Larsen product inference R_{L}, Drastic product inference R_{DP}, Bounded product inference R_{BP} are represented, respectively, as

$$R_{\mathrm{M}}: \min(\mu, \mu_A(x)), \quad \forall x \tag{1}$$

$$R_{\mathrm{L}}: (\mu \times \mu_A(x)), \quad \forall x \tag{2}$$

$$R_{\mathrm{DP}}: \begin{cases} \mu, & \text{for} \quad \mu_A(x) = 1 \\ \mu, & \text{for} \quad \mu = 1 \\ 0, & \text{for} \quad \mu < 1; \mu_A(x) < 1 \end{cases} \tag{3}$$

$$R_{\mathrm{BP}}: \max(\mu + \mu_A(x) - 1, 0) \tag{4}$$

Denote the output variable of fuzzy control as z, and the general defuzzifier function was defined as

$$z = \frac{\sum_{k=1}^{N} \mu_k^{\alpha} \beta_k}{\sum_{k=1}^{N} \mu_k^{\alpha}} \tag{5}$$

whereas α represents the design parameter; μ_k denotes the membership values of rules. β indicates the values at which the fuzzy sets output z are nonzero. Common defuzzy method which computes the centroid, mean, or optima will convert the output from inference engine to meaningful result.

Chen and Mathe [41] made a formal attempt to utilise fuzzy logic as a statistical method over traditional pre-defined rule-based systems. The proposed idea is to translate an AML scenario into fuzzy rules to generalise the results of fuzzy inference as a suspicious score. The suspicious score takes the amount received and a match score as input and generates a suspicious score as the output. Match score was defined as the real number within interval [0, 1] represented the degree of suspicious. The amount of money received and the amount of money withdrawn within a time window would be the control variables for rule function. From these three parameters, five rules were generated that categories match score into three levels zero, high, and moderate. The fuzzy inference engine will evaluate interpolations of scores from each subregions using nine fuzzy rules.

Log-likelihood (LLH) function was proposed to estimate the membership function. Particularly, the LLH of observed vector $x = (x_0, \ldots, x_j, \ldots, x_n)$ in the problem space of normal distribution with mean μ and standard deviation σ_0 will be estimated as follows:

$$L(\mu, \sigma_0^2) = \left(2\pi\sigma_0^2\right)^{-0.5} e^{\frac{-\left(x_j - \mu_0\right)^2}{2\sigma_0^2}} \tag{6}$$

Fuzzy rules are able to detect complex money laundering patterns only if the rules can be generated automatically. Further, as only the transaction amount, frequency and time are used in building the rules, the solution is practical because some financial institutions do not have access to numerous other databases that keep the relevant documents for checking. The method could be effective because the different financial institution can generate their own sets of fuzzy rules to identify similar patterns in the future.

However, the issue with fuzzy rules is that it requires the domain experts intervention since the fuzzy rules cannot be generated automatically. As in experiment [41], the large value received set as $10,000 while the immediate withdraw limited in 1–3 days. From this setting, 73 out of 710 records surpassed the suspicious score 0.8. Besides, the author did not mention the accuracy by comparing with other approaches, or providing false positive rate (FPR) and true positive rate (TPR).

Meanwhile, there are always trade-off issues for fuzzy rule-based system, such as between the classification accuracy and the average rule length, between interpretability and accuracy. Researchers [42] designed computational experiments to examine such trade-off by using error rates on test data and training data. Results showed that the relation between the error rate on test data and the average rule length was problem dependent. Also overfitting due to the increase in the average rule length was observed. In 2007, same group of researchers [43] conducted another sets of experiments to study the interpretability–accuracy trade-off in fuzzy rule-based classifiers using a multi-objective fuzzy genetics-based machine learning (GBML) algorithm.

Scalability issue in fuzzy rule-based learning has been discussed in [44]. According to this study, increasing the complexity of fuzzy models affects both its online and off-line performances. The complexity of a model may be increased in two manners: by increasing the number of rules in a dimension or increasing the number of dimensions of the fuzzy associative memory (FAM). The run-time efficiency decreases with an increase in the FAM dimensionality. Moreover, the applicability of FAM learning algorithms is limited by the amount of training data required. In order to achieve high accuracy, the rules need to be developed based on trial and error (e.g. misclassification rates), and the domain experts constant feedbacks are required in order to maintain the accuracy of the detection system for future cases.

3.1.2 Frequent rule

Luo [45] proposed to use frequent pattern (FP) algorithm to identify a frequent pattern in banking transactions to detect suspicious transaction patterns between customers accounts in anti-money laundering solution. The proposed method consists of two stages. The first stage refers to the discovery of frequent rules and constructing a FP tree from the training set. The author proposed to use the Hoeffding bound to estimate the support of rules. As the streaming of banking transaction data can be extremely large, which causes hardware and software to perform poorly, the author proposed to group transactions into several time windows and constructs the FP tree according to each time window cumulatively. The second stage consists of building a classifier model using the rules found in the first stage. If all the rules matching the new transaction have the same class label, then the transaction is assigned to that class directly. Otherwise, a group of class labels according to different rules will be applied to the new transaction [41].

Simulated datasets were used to conduct experiments with the proposed algorithm. The results showed that higher precision can be achieved by adding more transaction examples on the training dataset. However, there is issue with using FP rules. The amount of transaction must be discretised into a categorical variable in order to produce a generic tree. Further, constructing rules and trees based on transaction amount could result in a very large tree structure. In order to reduce the complexity and memory resource, alternative FP approaches such as FP Growth, FP Close can be implemented. FP Growth performs a divide-and-conquer process as scanning the DB structure once to partition into smaller sets. As this technique does not generate candidate item set, the shrinking scale is not expensive as traditional FP tree [46].

Later improvement in FP Growth is presented by Grahne, Zhu known as FP Close [47] and FP Max [48]. While FP Close verifies whether the explored itemset is closed to the pool of found subset, the FP Max evaluates the explored itemset. FP Max determines if the item belongs to the subset of maximal found itemset. Besides, in the real banking world, the

number of actually suspicious transactions could be too little to build a FP tree where each convicted suspicious transaction is unique, highly complicated, and infrequent.

3.1.3 Support vector machine (SVM)

SVM is a learning approach with a statistical evaluation to solve classification and regression problems introduced by Vapnik [49]. The primary objective of SVM is to construct decision boundary with the largest distance to sample called maximum margin separator. The constructed hyperplane that separates data is mathematically defined by the equation

$$w^{\mathrm{T}}x + b = 0 \tag{7}$$

which is the dot product of the observed vector x with the transpose of weighted coefficient matrix w, whereas $w \in F$ feature space and $b \in R$. Consequently, the two classes will have all data point of 1 in one side and -1 in the opposite. In order to prevent the overfitting problem when dealing with noisy data, slack variable ξ is introduced to create the soft margin. The idea was expressed as minimising the empirical risk measurement (ERM) with respect to the parameters of w, b, ξ_i:

$$E(w) = CF\left(\sum_{i=1}^{l} \xi_i + \xi_i^*\right) + \frac{1}{2}\|w\|^2, \tag{8}$$

whereas ξ_i and ξ_i^* are the positive slack variables.

C indicates the trade-off between error and regularisation. As C comes to significantly large value, the ERM would be minimised, though the model becomes more complex with expensively computational cost, whereas lower value would result in a simpler classifier. The first term represents the empirical risk or cost function and can be defined as the insensitive loss function. The regularisation term defines the fitness and is subject to the constraints

$$\begin{cases} y_i - f(x_i, w) \leq \varepsilon + \xi_i \\ y_i - f(x_i, w) \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{9}$$

In a binary classification problem as fraud transaction detection, as minimising ERM hypothesis, the SVM constructs the hyperplane $\langle w, x\rangle + b = 0$ that separates fraud and normal classes. The notation $\langle w, x\rangle$ is the weighted sum of input attributes, evaluated by the dot product of coefficient(weighted) vector w and attribute vector x.

When handling with high-dimensional problems, SVM can generalise well by applying a proper kernel which measures the similarity. A kernel is represented as

$$K(x, x_i) = \langle \phi(x), \phi(x_i)\rangle \tag{10}$$

in which ϕ is the transformation function that maps the input vector from vector space X into a higher-dimensional space L. Depending on the objective function, common kernels such as linear and Gaussian will be effective to solve linear or nonlinear problem, respectively. In particular, Gaussian kernel and linear kernel of empirical data with variance σ and degree of freedom d were defined as follows, respectively

$$K(x, x_i) = e^{\frac{\|x - x_i\|^2}{2\sigma^2}} \tag{11}$$

$$K(x, x_i) = (x, x_i + k)^d \tag{12}$$

As replacing with kernel function and applying Lagrange techniques for the dot production calculation, former decision function becomes

$$F(x) = \operatorname{sgn}\left(\sum_{i=1}^{n} \alpha_i K(x, x_i) - \rho\right) \tag{13}$$

As generalise well in high-dimensional space problem, SVM is considered as an attractive solution for AML. The SVM gains its benefit from transforming original nonlinear problem space to properly linear problem in a higher-dimensional space. Specially, as dealing with rich features problem like AML, the simplicity of classifier can well generalise [50].

Due to the heavy dependence on training dataset, standard SVM only performs well on a medium and small dataset. When handling with real money laundering data, the performance of normal SVM does not impress as other techniques. In addition, with real data, the case of suspicious transaction rarely record when compared with normal ones. SVM is popular in high generalisation capability and its ability to handle high-dimensional input. The accuracy of SVM can be improved by many ways, such as by training on auxiliary data sources [51]; by using aggressive feature selection [52]; by transforming kernels [53]; and by ensemble predictive model [54]. It does not suffer from the local minima problem, and it produces stable and reproducible results.

However, SVMs suffer from slow training, especially with nonlinear kernels and with large input data size. Recently, researchers proposed a model implemented with a clustering method called the clustering-based SVM (CB-SVM) which on a limited resource will increase the performance of support vector machines over large data sets [55]. In the work of [56], they presented an improved multi-class cutting plane algorithm that extends the new SVM structural formulation [57] to improve multi-class linear training time. Different from traditional SVM, one-class SVM is an unsupervised learning approach. One-class SVM attempts to explore a searching kernel that can group at most as possible all normal items. As a result, the approach minimises the anomalies as much as its capability. The approach characteristic allows one-class SVM to be suitable for AML problem. Hence, traditional SVM needs to be modified for adapting with such large and imbalanced data.

Liu and Yu [58] proposed of using the improved RBF kernel function based on the heterogeneous value different metric (HVDM) distance of the SVM algorithm discussed in Tang and Yin [59] (see Sect. 3.5) in order to identify suspicious transactions. The proposed method aimed to solve the problem of heterogeneous dataset in ML. Common distance functions in form of Minkowskys formula such as Euclidean and Manhattan has the weakness of overpowering attributes. Consequently, normalisation process that divides the distance of each attribute by the standard deviation must be applied. The alternative HVDM distance takes heterogeneous data as input instead of normalising. In addition, the alternative kernel formula RBF involved as the replacement for linear kernel trick when dealing with heterogeneous value. By mapping input to real number Hilbert space, the transformed linear space allows the inner product to be computed.

A grid evaluation metric is used to obtain the optimal parameters by setting penalty parameter C and control factor ensemble g a range of value with their offset. During experiments, cross-validation has been applied to avoid the overfitting problem. In the experiments, researchers managed to detect around 65% of total suspicious transactions, at which almost 80% is the real suspicious.

The authors performed two experiments based on real transaction record that is acquired from Wuhan Branch of Agriculture Bank in south central China which comprises 5000 accounts with 1.2 million records. The aim of the first experiment is to get customer's inner

behaviours using SVM algorithm. The second experiment focuses on improving the parameter of SVM model by using the cross-validation method. By using K-cross-validation, the author obtained the optimal parameters for training SVM. Experimental results showed that cross-validation method is more effectively than randomly selected parameters in term of detection rate and false positive rate.

3.1.4 Radial basis function (RBF)

The RBF technique adopts neural network architecture which constructs nodes in different layers together for cognitive learning. The strength of connection among nodes between 2 adjacent layers is defined through a set of weights. The weights value of one node depends on the distance to connected nodes from the previous layer and is evaluated by nonlinear activation function like Gaussian. The primary goal of a network is to determine the centre and width of RBF and adjust the weight. The process of learning parameter is similar to semi-supervised technique and divides into 2 phases.

The first phase applies unsupervised learning, such as k-means, approach to determine the parameters in an activation function. The second phase maintains the learned parameters and implements simple linear classification for weight tuning. By using only one hidden layer, the time performance is reduced greatly as compared with multi-layer perceptron. This structure also has an advantage as two sets of parameters can be measured independently. However, RBF also has the same disadvantage with SVM in dealing with irrelevant attributes. In particular, SVM with Gaussian kernel, which kernel function is the centroid of training instances, is a type of RBF.

An approach by Lv et al. [60] is proposed to use radial basis function network (RBFN) to detect suspicious transactions from the normal transactions. Compared to the traditional RFBN, the author proposed of using APC III clustering algorithm [61] to speed up the learning process, i.e. optimising the parameter of radial basis function in the hidden layer of the network. Furthermore, the author introduced recursive least square algorithm (RLS) to improve the convergence speed of the algorithm in updating the weights between the hidden layer and the output layer.

The APC is a distance-based clustering algorithm that concentrates on defining the radius, R_0 using minimum distance among the number of training patterns.

The process of APC clustering

1. Initialise the number of cluster, number of samples in cluster, and centre of cluster.
2. Screen through each training samples for further process
3. Calculate the distance $d_j i$ between training sample x_j to each ith cluster.
4. By comparing with radius value ($d_j i \leq R_0$), the algorithm updates the clustering patterns, and its centre.
5. The list of the cluster is updated in case of a difference from the centre.

In order to clarify the accuracy, the false positive rate as well as the detection rate has been selected as the performance measurement to compare with the SVM and outlier analysis. For the neural networks learning the purpose, the input layer consists of the following attributes: frequency deposit, the frequency of withdrawals, and transaction amount. In order to simplify the network structure, the number of nodes in the hidden layers also takes the same amount of nodes as input layers. Subsequently, the output data will contain only two nodes indicating normal or suspicious transaction. In order to prevent bias in learning and reduce computation complexity, all input values are normalised into the range between 0 and 1.

The experiment results indicated the improvement in sensitivity as using RBF technique instead of SVM and outlier analysis. However, as the detection rate was too high as above 80% while the number of suspicious instances was only half of dataset, the techniques fell to overfitting cases and detected all transactions as fraud. A general issue with using neural networks and SVM algorithm in money laundering detection is the difficulty in explaining how a transaction is considered suspicious by the system. Besides, money laundering is a sophisticated and complex problem that requires more inputs and components such as link analysis between behaviour and money flow. Therefore, the patterns that can be trained and detected by using the frequency and transaction amount only are not comprehensive enough to understand the AML problem.

3.1.5 CLOPE

Cao and Do [62] proposed a method of applying the clustering approach-based CLOPE algorithm to detect 3 common money laundering cases, which are (1) transferring money in a circular pattern, (2) distributing small amounts of money to many recipients, and (3) gathering money from several sources. The proposed CLOPE algorithm exploited the height-to-width ratio to increase intra-cluster overlapping among transaction items [63]. As performed on a large database with multiple attributes, this approach only defined a better cluster in a short period of time.

The performance of CLOPE algorithm was measured by using a real bank dataset comprising of only 12 transaction records. Such tiny test set implies statistical uncertainty and subsequently, therefore, makes it difficult to draw any meaningful conclusions about the algorithm performance. Since suspicious transaction information is not available, additional 25 suspicious transaction records were simulated. The following attributes were used for clustering: the transaction account, the sum of money sent, the sum of money received, the frequency of sending money, the frequency of receiving money, the frequency of the accounts sending money to a specific account, the frequency that a particular account receives from a specific account, and the sum of money received minus the sum of money sent.

The experimental results showed that CLOPE could detect all 25 simulated suspicious records with only 6 clusters when compared to the k-means algorithm which required 11 clusters. However, there was no mechanism to determine which clusters belonged to the money laundering category. Despite detecting all 25 simulated records in 6 clusters, the approach has a drawback as it could not determine whether any of the CLOPE clusters belonged to the money laundering category. This drawback makes the approach impractical. It is virtually impossible to determine the correct clusters to analyse for money laundering cases without user intervention. Furthermore, the CLOPE algorithm has a limitation in which it only accepts nominal variables. Continuous variables such as transaction amounts should be discretised and assigned with a meaningful label. The end user is also required to specify the number of bins for the data discretisation task. This is impractical because the number of discretisation bins may change based on the different datasets.

3.1.6 Semi-supervised and hybrid approaches

There are only a few papers that presented the semi-supervised approach in AML. A combination of clustering and multi-layer perceptron (MLP) was proposed by Le-Khac et al. [28]. The authors proposed a simple centre-based clustering technique to detect suspicious cases of money laundering. This technique is based on the frequency and amount of transactions that are used as the input of a MLPs training process.

In their experiments, the frequency and value (amount) of the transactions were aggregated by time intervals, i.e. daily, weekly, and monthly. Two parameters of the model have also been introduced, namely the proportion between the redemption value and the subscription value conditional on time (D1); and the proportion between a specific redemption value and the total value of the investors shares conditional on time (D2). The 10 million transactions from 10,000 customers were extracted from CE bank. The transactions were divided into 2 groups: individual and corporate. Suspicious transactions were synthetically generated due to the lack of suspicious transactions. The preliminary results showed that proposed approach was efficient. However, it was not clear how much reduction in false positives was obtained. Nonetheless, the number of features and the number of training patterns were small. This drawback could have affected the overall precision.

Moreover, the same authors from [28] also proposed another framework that implemented both k-means clustering and artificial neural network to detect money laundering in an investment bank [64]. However, it [28] used 5–10% of the population as a training set and the reminder as a testing set. This low training/test split ratio may lead to poor performance of the model. The approach consolidated the transaction records to daily, weekly, and monthly transactions. K-means clustering was then performed on each of these transactions to detect suspicious clusters. Each transaction in the suspicious clusters was labelled as suspicious, while and the remaining transactions in the other clusters were labelled as normal. These labelled transactions data were then used to train the neural network model. However, due to the lack of suspicious transactions, the authors proposed the use of a genetic algorithm to produce more synthetic transactions based on the suspicious transactions detected after the clustering process to train the model.

Finally, new incoming transactions were tested using the trained model to detect suspicious money laundering cases. The drawback of this approach is that it requires expert knowledge when identifying suspicious clusters during the transaction labelling process. Moreover, defining the number of clusters to be used in the clustering process is also an issue and would require a heuristic approach as mentioned by the authors in this study.

Liu et al. [65] introduced a core decision tree method to discover money laundering pattern and rules. A subset of training data was used to construct clusters using the k-means algorithm. The produced centroids are used as the core to build the initial decision tree using the BIRCH algorithm [66]. Next, the remaining training data are used to improve the tree with BIRCH algorithm. The general issue in using both k-means and BIRCH is how to identify the number of clusters to generate. The author described that centroids produced in k-means would make up the core or root node for the decision tree, thus obtaining the optimal cluster number was crucial. Besides, building the tree using BIRCH requires two parameters, namely branching factor B and threshold T. Both parameters will affect the height and size of the tree.

However, the method to obtain k, B, and T is not mentioned in the paper; thus, it can be difficult to apply the algorithm in identifying abnormal transactions. Furthermore, the author [65] indicated that the same operation (i.e. transaction) executed repeatedly in a few days could suggest a suspicious behaviour. However, legitimate transactions could also be conducted repeatedly in a few days and no experimental result has shown to prove the proposed algorithm.

3.1.7 Deep learning

Paula et al. [67] proposed a deep learning method for anomaly detection as support fraud investigation in anti-money laundering. The authors constructed a model which used cross-industry standard process for data mining (CRISP-DM) as a reference model. The CRISP-

DM consists of several phases, namely business understanding, data understanding, data preparation, modelling, evaluation, and deployment.

During data understanding and data preparation phases, eight attributes were identified that proved sufficient to characterise fraudulent exports based on the experience of the author and empirical studies conducted. The attributes were distributed in ten different dimensions, particularly registration, foreign trade export, tax collection, financial transactions, tax withheld at source, employees, electronic, supplementary, inspection, and others. Oxdatas H_2O software 3 which is connected to *R* by H_2O R package was used by the authors for data modelling. In total, 811,990 records were processed corresponding to companies operating directly or indirectly in exports in the Brazilian market in 2014. One of the proposed models was Deep Learning AutoEncoder which was compared to PCA method in this experiment. Two models were evaluated by mean squared error (MSE) of the predictions that vary from real data.

The authors claimed that the result showed a performance to reduce dimensionalities about 20 times faster using deep learning AutoEncoder compared with PCA. The major difficulty in the use of unsupervised techniques is the evaluation of the results against the business objectives, and the evaluation of third party experts is subjective. Deep learning is designed to deal with complex and huge scaled datasets; however, the amount of computation power needed to train and test deep networks is also high. There is also a trend to bring deep learning to low-power, embedded devices.

Research work in [68] presented a novel hashing based technique to drastically reduce the amount of computation needed to train and test deep networks. The new algorithm for deep learning reduced the overall computational cost of forward and backpropagation by operating on significantly fewer (sparse) nodes. It used only 5% of the total multiplications, while keeping on average within 1% of the accuracy of the original model.

3.2 Link analysis

In this section, we briefly discuss the algorithms, which focused on identifying the links or connections between entities such as bank customers accounts. Link analysis is typically used to analyse the nature of the relationship between bank accounts. Based on the transaction activities carried out by each customers accounts, links are constructed using algorithms proposed by various authors. When links have been constructed, the behaviour of a particular group can be analysed as a whole.

3.2.1 System supports money laundering detection

Dreewski et al. [69] proposed a system supporting money laundering detection that consists of 3 main components: clustering, mining for frequent patterns in clusters, and data visualisation. The clustering component refers to the construction of graphs that represents the flow of money and captures only suspicious money transfers between groups of accounts. The second component consists of algorithms (i.e. FP Growth, FP Close, and FP Max) that mine frequent money laundering patterns that have been captured in the clustering component. The author explains that mining the patterns could increase the chances of enabling the clustering component to find suspicious transaction patterns.

The proposed system only supports suspicious cases that happen between groups of people. A view with clusters and a view with frequent patterns were used to visualise the result data. The cluster was presented as a list of graph nodes while the frequent patterns with two elements in the form source bank accounts were meaningful as they allow the identification

of distributive boxes and collective boxes, respectively. Besides, building the graphs in the clustering component will require the source and destination of a particular banking transaction and it can be difficult to pinpoint the exact destination if the money transfer occurs across different banks and countries. Also, the time taken for the realisation of money transfer was considered where the duration of time of the cluster was not greater than a given size of the time window. However, the system can be practical when it comes to establishing links between suspected cases that have already happened within a bank in order to verify the validity of the cases.

3.2.2 Social network analysis (SNA) algorithm

Dreewski et al. [69] also proposed a complete social network analysis (SNA) algorithm known as money laundering detection system (MLDS) to support the detection of money laundering in [84]. MLDS consists of few components in addition to the system in [69] i.e. profile generation for company/organisation module and social network analysis module that builds social networks that assign roles to nodes, where each node represents a customer account.

In analysing social network between accounts, six basic measures were calculated in order to assign roles to each account. Each measure had its own default interval for each role where the intervals were established based on the characteristics of roles. The measures include closeness, betweeness, page rank, degree, authorativeness, and hubness. Each of these measures calculated the relationships between accounts and their connections on the basis of transaction activities conducted by them. Arquilla and Ronfeldt [13] proposed 11 roles of offenders in a criminal organisation, including organisers, insulators, communicators, guardians, extenders, monitors, crossovers, soldiers, recruits, outsiders, and occasional.

The above set of roles was standard and can be found in every criminal organisation with specific characteristics. Once the roles of each account had been assigned, a social network graph can be built and visualised. The author in [70] stated that the nodes with the same roles in data sets coming from different domains allowed them to confirm the correctness of the assigned roles. The experiment indicated that the performance of the role finding algorithm was worse than the algorithm analysing connections between the roles. Also, the analysis of the connection between the roles was much more scalable than the role finding analysis.

A recent study of Colladon and Remondi has developed the SNA to integrate with the system of factoring company [71]. The network for each transaction designed from 4 specific relational graphs. Firstly, an economic sector network evaluated the operational risk through a set of rule based defined by company experts. These rules took into account 15 attributes, which represented the production sales, and the public procurement. The second graph measured risk level of different geographic regions. The risk score for one transaction was the weighted sum of partial risks from its location percentile. A number of transactions and transaction frequency had been studied in the third graph. Lastly, the tacit link network investigated the potential harmful relationship in the companies sharing the same owners or representatives.

Result visualisation from the experiments indicated that suspicious profile was less central to its link graph and usually operates with a higher amount of transactions. From experiments observation, the authors claimed to increase the probability of handling suspicious subject when using a larger in-degree centrality on a geographic region with a lower network constraint for a business sector.

However, the study still has some limitations. The author cannot guarantee the performance to be effective when applying in large dataset from financial institutions instead of

factoring businesses. In addition, the irrelevant attributes that are easily obtained from factoring business rather than financial institutions may affect the detection rate. Finally, different techniques can be applied to classify high-risk profile as a replacement to avoid the strong connection with past event.

3.3 Behavioural modelling

In this section, we briefly discuss on algorithms, which focused on identifying the behaviours of bank customers based on their transaction activities. This tool aims to construct a behavioural model that attempts to combine information from various domains such as political, social, economic, and cultural domains on money launderers [72].

3.3.1 Expectation–maximisation (EM) algorithm

Chitra and Subashini [73] suggested to use the expectation–maximisation (EM) algorithm as the clustering model in discovering fraud. It was suggested that EM cluster could be used in grouping the data into similar clusters.

The strategy was to build a model that maximises the probability density function that obeys the Gaussian distribution based on the past transaction behaviour of each bank customer. The model built was then used to compare against the new transaction behaviour. Hence, the data can be broken into related components in such a way that changes in patterns and orders become visible. The EM algorithm requires a pre-defined number of clusters in order to estimate and maximise each data point for each cluster created. For example, if two clusters are specified, each cluster is represented as a Gaussian distribution with different function parameters. However, the authors did not provide any experimental results on the performance of the algorithm on either synthetic or real datasets. Therefore, the feasibility of such an approach has not been validated. Besides, the approach assumed that every customer transaction obeys the Gaussian distribution, which in fact may not be true. More experiment and analysis need to be performed before making assumptions of the underlying distribution, especially when money laundering activities are highly complicated and difficult to predict as it involves multiple variables.

Chen et al. [27] have pointed out that there is a lack of studies conducted in employing expectation–maximisation (EM) for anti-money laundering (AML). The authors have exploited the advantages of using EM for suspicious transaction detection. The experiment was conducted by using data obtained through local X bank in Malaysia which consists of 30 million transactions recorded throughout the 2-year period and were consolidated into three different time intervals: daily, weekly, and monthly. The authors mentioned that real money laundering cases were mostly detected from daily and weekly consolidated records based on historical performance. Each record held approximately 70 attributes including a class label and other information such as total debit amount, total credit account, the frequency of debit, and frequency of credit.

In order to design the experiment, 2700 instances were extracted for daily records; 1486 instances for weekly records; and 689 instances for monthly records. A total of 43 instances were labelled as anomalies for each of the categories. For the validation purpose, 2035 transaction containing 31 suspicious were extracted for training phase and the rest for the testing phase. Weekly and monthly experiments data were designed with (number of training instances/suspicious) 1086/30 and 505/31.

The results showed that EM managed to target all anomaly instances in daily and weekly cases. In addition, EM managed to reduce almost half of a number of false positive results

while remaining true positive detection. If k-means separated data into 60 and 40%, the performance of EM is more promising as grouping the data with 70 and 30% ratio. The searching space of suspicious transactions was reduced to be below 10% as the number of clusters increase. Also, EM proved to perform effectively in extremely imbalance data effectively. The authors concluded that EM was demonstrated as a potential methodology to support clustering process in AML with low false positive detection rate and gaining confident to capture all suspicious transaction.

3.3.2 Empirical mode decomposition (EMD)

Zhu [74] proposed a time series decomposition method, empirical mode decomposition (EMD), based on the concept of peer group comparison for behaviour comparison. The method extracted the fluctuation features first and then compared them with those of its industry peers, thus relieving the false suspicion and discovering new unusualness. The algorithm aimed to represent and extract the fluctuation features of the object under detection, and describes and compares the discriminations between two objects under detection. The author claimed that EMD is chosen due to its superiority in analysing stochastic time series when compared to the discrete Fourier decomposition (DFD) and wavelet decomposition methods. The approach was validated with foreign exchange transaction datasets, and the experiments have shown that the EMD approach efficiently reduces the number of false positives. However, the experiments were only validated with foreign exchange datasets. Datasets were not available to validate the approach on real money laundering data. Therefore, there are no preliminary results to determine the detection rate and false positive rate in actual real money laundering data.

3.4 Risk scoring

Statistical models and business rules are defined to automatically rank all transactions and customers by a potential risk. The risk factor could help in making a decision on whether to report a particular transaction as suspicious or normal. The risk scoring tool can be applied to assign risks to customers and transactions. The former is important to identify high-risk customers, whereas the latter is used as a detection tool for a suspicious transaction.

3.4.1 Decision tree

The algorithm processes derived from a chain of divide and conquer as a final decision is obtained after traversing through a sequence of testing. A complete solution with decision tree structure was described in work of Quinlan on decision tree induction ID3 and C4.5 [75] and other well-known study of classification and regression trees (CARTs) [76]. The tree was constructed in a top-down order. Each node corresponds to a specific linear test function defined by input attributes as partitions, like transaction frequency, account's income. In learning theory, it considered that a node defines a partition of the problem space. For the case of a classifier, the output or leaf value indicates the risk value such as low or high. Given the two plane $LEFT$, $RIGHT$ with input vector x defines p-dimensional problem space R

$$LEFT(j, s) = \left\{x \in R^p : x_j \leq s\right\} \tag{14}$$

$$RIGHT(j, s) = \left\{x \in R^p : x_j > s\right\}, \tag{15}$$

the decision for a candidate x to expand the tree to be left or right with feature j and threshold s with the comparison to the target values y is to minimise the measurement

$$\min_{j,s}\left[\min_{C_1}\sum_{x_i\in R_1(j,s)}(y_i-c_1)^2+\min_{C_2}\sum_{x_i\in R_2(j,s)}(y_i-c_2)^2\right] \tag{16}$$

subject to $C_1, C_2 \in \mathbb{R}$;

C_1, C_2 denotes the expected mean values of left and right branch, respectively. The algorithm then applies the induction recursively by considering the node as new root to generate subtrees The decision to determine which selected subtree can be expressed as minimising error function. Depending on the setting of criteria and type of problem, common formula such as mean squared error

$$\sum_{m=1}^{|T|}\sum_{i:x_i\in R_m}\left(y_i-\hat{y}R_m\right)^2+\alpha|T| \tag{17}$$

or mean absolute error

$$\sum_{m=1}^{|T|}\sum_{i:x_i\in R_m}|y_i-\hat{y}R_m|+\alpha|T| \tag{18}$$

The tree traverse to each class is observed and counted. In the simplest case of binary classification, popular node impurity approach like Gini index and entropy are applied [77].

Since constructing a sequence of rules, the classifier can adapt training sample very well by discovering a critical pattern. Decision tree and its variation are preferred due to their simplicity and interpret ability. Unlike other learning approaches, the reason for issuing decision can be interpreted directly for human understanding. According to banking process of marking a transaction or customer profile is suspicious, the experts should provide convincing reason follow by a sequence of rules. In the case of AML, a clear explanation has a lot of benefits when releasing a report as it can check if the system follows financial procedures. In case of exporting report to clarify analysis, DT dominants other black box solution such as SVM, MLP since the rules are derived directly from generated conditions.

However, decision tree easily leads to an overfitting issue if the tree has a complicated structure while the learning pattern does not have any improvement. As a result, pruning technique is required to eliminate node which has little learning progression, and thus simplify the tree. The scalability issue of decision tree has been discussed in [78], the decision tree construction for very large, real-world databases can become inefficient due to swapping of the training samples in and out of main and cache memories, while for relatively small data sets it is not a problem.

Sudhakar and Reddy [79] have proposed a two-step loan credibility prediction system that helps organisations in making the correct decisions to approve or reject the loan request of the customers with the application of decision tree induction algorithm. The authors have pointed out that credit risk management is critical for successful bank lending. The construction of the model requires five main phases including problem understanding, data understanding, data filtering, system modelling, and system evaluation. Decision tree induction data mining technique is used to generate useful and relevant variables and make the final decision in the model. Moreover, the variables are classified as independent and dependent variables where as an independent variable, for example, Age and Occupation, is the condition or characteristic that affects one or more variables: its size, number, length or whatever exists

independently and is not affected. A dependent variable changes as a result of changes to the independent variables, for example, Credit (Approved or Not).

J48 builds decision trees from a set of labelled training data and splits the data into smaller subsets since each attribute of the data can affect the final decision. J48 examines the normalised information gain(difference in entropy) that results from choosing an attribute from splitting the data. The highest normalised information gain is used. Then, the algorithm recurs on smaller subsets until all instances in a subset belong to same class. A leaf node is then created to the decision tree telling to choose that class. The main point of this model is it accurately classifies the loan application using traditional credit scoring and improved behaviour scoring.

Wang and Yang [80] presented money laundering risk evaluation method using the decision tree ID3 to rank customer risk. The decision tree method was used to create the rules of money laundering risks based on customer profiles. For the pre-processing step, the authors conducted experiments with following attributes: industry type, business location, business size, and the bank product. In order to evaluate the suspicious transactions, each attribute categorised into low, medium, and high risk. In particular, the risk levels were assigned based on the industrial type (e.g. manufacturing, chemical, domestic trading, medicine, IT, foreign trading, retail, advertisement, automobile sales). Each industrial type was assigned with a priority from low to high. The proposed method was only used to evaluate the risks of current and future customers where those at high risk would be put under strict monitoring.

The experiment established a 4-level depth tree with 15 determination rules that filtered 12% customer profiles as suspicious. However, such method cannot be a standalone solution as it only investigates the customer profiles, while the most crucial factor in money laundering is the transactions conducted by customers. Besides, it can be difficult to obtain the risk categories for each attribute because a thorough survey has to be done prior to assigning categories scores. There is also a possibility of high false positive cases because once a customer is classified as high risk, all his transactions will be monitored. As spanning to the large and imbalance transaction dataset, the decision tree becomes unstable that generates different pattern.

Rojas et al. [81] presented two decision tree approaches, namely decision trees and decision rules for money laundering detection. From the decision trees group, random tree, random forest, and J48graft were selected, whereas from the decision rules group, JRip, and decision tree were selected. A synthetic data generator, named multi-agent-based simulation (MABS), was used to simulate mobile money transactions using agents in order to test the selected algorithms [82]. The authors discovered that JRip gave the best accuracy with only 0.012 false positive transactions and above 0.9 true positive transactions.

Although the result was satisfactory, the transaction data that were used to run the algorithm were artificially generated data. In practice, the solution could only work on any data if a clean training data is available, but that may not be the case when using real dataset. As the data used are generated synthetically, there is no need for the tedious pre-processing stage.

3.4.2 Sequence matching algorithm

Liu et al. [83] proposed a sequence matching algorithm to identify suspicious transactions. Sequence matching is a method that focuses on a comparison between the sequences in time series databases, and it has been widely adopted in stock analysis. The authors recommended the use of an individual accounts history and the transaction information from other accounts in a peer group for a sequence matching such as transaction time, account number, transaction direction, and transaction amount.

The primary goal was to extract sequences of transactions within a peer group to a daily level. The high-risk sequence was then chosen from the extracted sequence by using a probabilistic model, and such sequences are compared with the historical transaction sequence of each account. The comparison was done using a Euclidean similarity distance, and each high-risk sequence was assigned a similarity score. A pre-defined threshold score was used to extract high-risk sequences as suspicious sequence according to the score assigned.

Experiments were performed where transactions were extracted from real financial data from a Chinese financial institution. Each transaction was labelled as either normal or suspicious. A total of 640 accounts with 120,986 transactions were extracted for the normal category. On the other hand, 64 accounts with 1940 transactions records were extracted for the suspicious category. The authors discovered that the algorithm performance can improve greatly when the threshold decreases from 0.9 to 0.6. When tuning the threshold from 0.9 to 0.6, the false positives increased significantly from 9335 to 27,196.

One of the limitations of sequence matching is the need of a threshold value. Defining the threshold manually would not be ideal because each account or customer may require a different threshold. However, one way of choosing an appropriate threshold is using cross-validation. Cross-validation is a standard tool in analytics and feature to develop and fine-tune data mining models while allowing us to set the accuracy bar for predictions. Moreover, it is often impossible to estimate the number of suspicious sequences since that is unknown in the real world. The algorithm performed is therefore largely dependent on the chosen threshold level. Setting a higher threshold may result in a lower false positive rate and higher detection rate, but more suspicious transactions may be missed. On the other hand, setting a lower threshold may increase the false positive rate.

3.4.3 Modified version of Euclidean adaptive resonance theory (TEART) algorithm

Larik and Haider [84] proposed a modified version of the Euclidean adaptive resonance theory (TEART) algorithm that eliminated the weakness of both distance-based and density-based clustering algorithms when processing large datasets. The distance-based clustering algorithm classified data points far from the mean cluster points as outliers, while the density-based algorithm considers data points within the thin population area as anomalies. The authors indicated that in the AML problem domain, the problem space can be very complex since the outlier patterns may overlap with the normal area population. The algorithm first filtered the data to be noise-free by normalising the datasets. Next, a principal component analysis (PCA) was applied to determine critical attributes that cover most of the variances for a dimensional reduction.

Then, the TEART adopted distance computation from the k-means algorithm and proposed a vigilance parameter, v to control the creation of clusters. By comparing the distance between a data point and all cluster centroids, the smallest distance was taken and compared with v. The data point was added to the cluster if the minimum distance was smaller than v. Otherwise, a new cluster was formed. Furthermore, in order to ensure balanced clusters are formed, clusters with less than 1% of data points were discarded and the process continues until clusters with less than 1% of data points are combined to form a cluster. Next, an anomaly index named AICAF was calculated to measure the deviation of transaction amounts and the frequency amounts for both credit and debit from the established behaviour of the cluster the customer belongs to. Transactions with high deviation values were regarded as suspicious.

The method produces a set of balanced clusters that enables the grouping of similar behaviour patterns and behaviour that slightly deviates from the pattern. This means that a cluster could have a normal and suspicious version of the similar behaviour. The AICAF

index is used to differentiate between the two versions when classifying a transaction. The experiments have shown that the algorithm was more efficient in differentiating customers with significant variations in credit amounts, debit amounts, credit frequency, and debit frequency when compared to a standard k-means algorithm.

One issue concerning the algorithm is deciding the parameter of v. The value of v is what influencing the grouping of data points into the cluster. Smaller v could cause some data points to never be clustered, whereas larger v could cause more than one behaviour patterns grouped into one cluster, affecting the accuracy in classifying suspicious transactions. In addition, as TEART related on a distance-based clustering technique, it requires comparing with likelihood clustering techniques such as EM to verify the performance.

3.4.4 Bitmap index-based decision tree

Jayasree and Balan [85] proposed a Bitmap Index-based decision tree for risk evaluation on financial money laundering. The proposed work was to evaluate the risk factor using indexing scheme to enhance adaptability and scalability rate. The mechanism used the rows and columns in bitmap indexing to store the information. A decision tree in then constructed with mapping of the binary fuzzy form to create the rules in BIDT technique. The focus of indexing in the technique was to provide pointers to the rows in a table containing given key values.

The authors show that the money laundering account is effectively evaluated with bitmaps using cardinality rows and columns where the bitmap index uses the arrays and bitwise logical operation AND for the results. The total number of integers in bitmaps was calculated to obtain population frequencies for risk factor evaluation. Received results were used to build the decision tree where the customer region of transaction and risk occurrence are identified.

The experiment was conducted by using JAVA platform and Statlog German Credit Data from the UCI repository classifies the people by a set of attribute lists for the easier evaluation of the risk factor. BIDT technique is compared with the existing method such as TrustedPals, a smart card-based security framework (SCSF) [86], and multi-layered detection system (MDS) [87]. The experiment evaluation is based on the following factors risk identification time, false positive rate, adaptability rate, and regulatory risk rate. The authors indicated that according to the experiment, BIDT technique has a more promising result in terms of risk identification time, false positive rate, and adaptability than the compared methods.

However, the approach has not been investigated using real financial transactions dataset. In addition, the compared methods were designed to solve specific credit application fraud detection instead of AML. Therefore, the study should be examined further with various AML strategies to verify the performance.

3.5 Anomaly detection

Anomaly detection refers to the ability of an AML solution to identify the deviation of particular transactions from the norm transactional behaviour. In general, the solution should be able to recognise the unusual transactional behaviour of each customer account

3.5.1 Improved version of one-class SVM algorithm

As conventional SVM mostly separates labelled data problem, Schölkopf proposed a extension of SVM that operates with unlabelled data known as one-class SVM [88]. Given the dataset with its probability distribution P, the problem is described as to determine the subset

S within features space, in which a test point from P lies outside of S is bounded by priori specified value between 0 and 1. The attempted solution tries to estimate the function f which is positive on S and negative on the complement. Hence, the minimising function has been slightly modified from the origin

$$\frac{1}{\nu n}\left(\sum_{i=1}^{l}\xi_i - \rho\right) + \frac{1}{2}\|w\|^2, \tag{19}$$

subject to

$$\begin{cases} w \cdot \phi(x_i) \geq p - \xi_i \\ \xi_i \geq 0 \end{cases} \tag{20}$$

Instead of smoothness parameter C, alternative parameter ν is used to control the boundary.

Kernel transformation that is regularised by controlling the length of weighted vector maps data into the feature space H. The expansion coefficients are obtained by solving a quadratic programming problem.

Tang and Yin [59] proposed of using an improved version of one-class SVM algorithm to detect unusual customer behaviour based on transactions activities. The author described that SVM was used because of its less sensitivity to high-dimensional data that are common when it comes to financial transactions. One-class SVM attempted pre-learning and converts input space into high-dimensional space with a nonlinear transformation. The inner product kernel function implemented RBF approach measured by HVDM distance.

Obtained result helped to strike the optimal classification surface in new space. The kernel contributes by mapping the input data to the feature space. The algorithm also suggested an improved kernel function to perform SVM anomaly detection in case the feature space met all conditions for a calculation. The author performed experiments based on sum and frequency of accumulated transactions and business fluctuation. A trial-and-error method was used when adjusting the parameters for the algorithm.

The author showed that highest detection rate is only 69.13% with 5.4% of false positive rate based on experiments performed on a real dataset. Looking at the performance results, although the false positive rate is very low, the algorithms accuracy could be improved by using more attributes such as the mean and standard deviation of transaction amount and frequency or other variables that can contribute to the detection of unusual behaviour. Besides, using one-class SVM requires the users to input the percentage of outliers from the set of data and the number could be different depending on the dataset.

Furthermore, the output of the algorithm also highly dependent on the dataset used and the practicality of the solution is questionable, i.e. How to apply the solution on a daily transaction, and is it feasible to compare between transactions on a daily basis instead of comparing the daily transactions with a specific dataset?

3.5.2 Link discovery based on correlation analysis (LDCA)

The use of large amounts of unstructured text and Web data such as free text documents, Web pages, emails, and SMS messages is common in adversarial domains, but still unexplored in fraud detection literature [16]. Zhang et al. [21] proposed a clustering algorithm based on Link Discovery of Correlation Analysis (LDCA). The authors approached the matter by extracting n suspicious individuals, who have relation to suspicious cases collected by the investigator. The author further mapped the transactions performed by those individuals in an $n + 2$-dimensional Euclidean space, representing time and transactions.

Consequently, the clustering problem is reduced by discretising the timeline into different time instances, resulting in each transaction being viewed as a node in a one-dimensional timeline space. The problem was further simplified by either accumulating monetary amounts or the transaction frequency in each timeline instance. Subsequently, the k-means algorithm was used to perform histogram segmentations where each k cluster represented a segmented histogram. The suspicious cases were identified by observing abnormal hills in a histogram and correlation between histograms of different individuals.

The proposed method is able to compare individual transactions with their peers using transactions without external knowledge such as business size or occupation. However, the duration of each time instance and the number of clusters have to be appropriately defined in order to capture abnormal transactions. Besides, simply applying the approach in a financial institution with a large number of transactions could produce too many irregularities in the histograms. Thus, the solution can be used to fully focus on suspected money laundering cases where it has been detected by other tools (e.g. rule-based system, risk factors).

3.5.3 Minimum spanning tree (MST) algorithm

Wang and Dong [89] proposed to use minimum spanning tree (MST) clustering algorithm to detect suspicious transactions. The author improves the MST algorithm by dividing the data into three subsets namely upper, lower, and middle datasets. The algorithm worked by randomly selecting a data point as the root of the tree. A data point is placed into the upper dataset in case the dimensional value of the data point is lower or equal to the root. If the data point had dimensional value greater than or equal to the root, it would be mapped to the lower dataset. Middle dataset contained all data points that did not fit into either upper or lower dataset. The next step was to build the tree where all data points in the upper dataset would be the roots left subtree, all elements in the lower dataset will be the roots right subtree, and middle subtree. Each of the subtrees roots was chosen by the shortest distance to the main trees root.

Finally, the nodes in all subtrees used the MST algorithm to construct the tree. In order to obtain k clusters, the longest $k-1$ edges were removed so that the MST split into k subtrees. In identifying the anomalous clusters, the author proposed a new dissimilarity metric function. The metric was able to effectively reflect the differences between two points and govern the differences between two points with smaller values. Thus, the metric provided robustness to noise data. The calculated dissimilarity values are sorted for each cluster to discover anomalies.

The author performed experiments using a real dataset with different values of k (i.e. 5, 10, 15, and 20) on 65,001 records and 60 artificial money laundering records. The results showed that when k is 15, the approach was able to group all anomalous records in 4 clusters with an average of 72% proportion of anomaly points. This proves that the algorithm is able to identify all suspicious transactions without prior knowledge and assign a dissimilarity score to each record, thus providing a direction for unsupervised learning in this domain as well as a guide for decision making for the compliance officers. However, the number of false positives (i.e. the legitimate records in the 4 clusters) is not presented.

Nevertheless, the approach needs to be improved further to avoid following drawbacks. Obtaining the optimal number of cluster, k, would require few trial-and-error cycles. Hence, it is not feasible in applying to the real AML system. Identifying anomalous clusters requires domain expert at the time of reporting. Artificial anomalous data are created and used in the experiment makes the results questionable in identifying anomalies on real unseen dataset.

3.5.4 Suspicious activity reporting using dynamic Bayesian networks (SARDBN)

Raza and Haider [90] proposed a combination of clustering and Bayesian networks called suspicious activity reporting using dynamic Bayesian networks (SARDBN) to identify anomalies in transactions. The flow of the approach consisted of three major phases.

Firstly, the process of clustering to group customers was based on average monthly credit and debit transaction amount, average monthly credit and debit transaction frequency, and the time between the consecutive transactions. The clustering phase focused on grouping customer behaviour based on their transaction activities using the fuzzy *c*-means algorithm.

In the next process, SARDBN determined the formation of dynamic Bayesian network (DBN) on each individual clusters. Each DBN was constructed using three variables namely transaction amount, transaction mode, and period of transactions on three-time slices. As the transaction amount was a numerical variable, discretisation on this attribute was performed individually using the *k*-bins method on each *c*-means cluster. A sequence of transactions and their dependencies was used as the input to form the structure that is transformed into a dynamic connection DBN with different time slices. As the transaction entered DBN, new posterior probability distribution for each variable was produced to provide predictive reasoning.

The result of the recent transaction was used to detect anomalies for incoming transactions. The datasets were sent through the network to determine which cluster, a data instance belongs to, and to predict the amount as well as the mode. A ranking system evaluates the posterior probabilities to assign their prediction order. A formula has been developed to calculate the anomaly index using rank and entropy (AIRE). Total AIRE is the weighted sum of all partial AIRE values. The purpose of this measurement is to compare the AIRE with a predefined threshold to classify a normal or suspicious transaction. While ranking describes the level of suspicion, entropy determines the conclusiveness of the model to report the anomaly.

The performance of the algorithm was influenced by the defined threshold. Experiments were conducted with approximately 100,000 customers incurring 8.2 million transactions for a 1-year period. The training dataset comprised of transactions from January until October (10-month data) and was used for clustering and learning process in dynamic Bayesian network. The test data included transactions during November and December (2-month data) and were used to analyse the model for prediction and anomaly detection. The method was able to classify 95% of the correctly predicted TxnAmount using second-order DBN for cluster 2. Based on the optimal threshold, 917 anomalies were detected.

The common issue of using clustering method in suspicious transaction detection is how to obtain the optimal number of clusters to produce, and to find the best training examples for clustering. Using too many training examples to obtain different customer behaviour will potentially lead to a poor performance. The reason is the probability of data with two or more different labels that could be in the same cluster. Therefore, the problem arises during the detection phase while the cluster contains two or more behaviours. In the other hand, the use of too little training examples could prevent the formation of representative clusters. Other issue involve the outliers that could form many small clusters without useful information. However, the authors overcome this problem by using fuzzy *c*-means (i.e. a soft clustering) algorithm that allows every data point to belong to any cluster, thus giving space to domain experts for adjustments.

3.6 Geographic capability

The geographic capability tool refers to the ability to identify money laundering activities across different countries and language differences. Hence, the geographic capability tool would require strong ties between financial institutions of different countries in order to share information that links to financial terrorists. Consequently, a system that explores this aspect strongly concentrates on currency exchange rate and geographic location as the main attributes.

One example of such support system was presented by Yang et al. [91] where an AML service system for a union bank is constructed to detect money laundering on online payment. The logical framework of the approach composed of five sequential layers: database layer, basic data resource base layer, data analysis layer, application service layer, and the interface layer. The database layer collected transaction information and history database. The basic data resource layer contained knowledge base, case base, and other informative data that enabled the discovery of money laundering cases. The information collected is then transformed into useful applications through the real-time multi-agent based system in the data analysis layer. In this layer, data cleaning was performed by the real-time data monitoring agent and then output the result to several agents that include neural network agent, expert system agent, data mining agent to analyse. The detection component was in the application service layer where relevant information from new incoming transactions was extracted and displayed to users to request for a decision. When money laundering is discovered based on the transaction, the result is sent to the union bank. The standard interface layer works as an external interface that transmits transaction information from different financial institutions to the union bank through the Internet. The difficulties are related to the requirement for cross-system integration since transactions of this type usually involve in multi-geographic regions with different currency. In addition, the system of union bank in [91] has to integrate with many submodule services besides other commercial bank such as police bureau, prosecutorial, and foreign exchange administration to support the domestic environment.

3.7 Multi-aspect combination

In some commercial solutions, cross-channel support is also considered as one of the most important tools. Cross-channel support refers to the detection of money laundering based on different payment type such as credit card payment, bank transfers, and insurance payment. Zhang et al. [21] have proposed a clustering algorithm using link discovery based on correlation analysis (LDCA) that supports cross-channel detection by potentially mining texts across different data sources.

Tang [92] proposed a new cross-outlier detection model based on distance definition incorporated with the financial transaction data features. The authors stated outlier detection is a key element for intelligent financial surveillance system with two different categories of detection procedures. The first compared every transaction against its account history, and the other compared against a peer group to determine if the behaviour is unusual. Also, peer group analysis concept was largely dependent on a cross-datasets outlier detection model. The authors have utilised an approximation algorithm accompanied with the model which is provided to optimise the computation of the deviation from tested data point to the reference dataset.

An experiment was conducted based on a real financial transaction record dataset with 5000 accounts, 12 million records over 7 months from the year 2003–2004. The results of the pre-defined rule, SVM, and cross-dataset approach were compared and analysed. Using the

pre-rule-based system in the bank, there were 18,661 false positive cases and 2 detected cases. The result of SVM showed that there were 1338 false positive cases and 9 detected cases. For cross-datasets, there were only 219 false positive cases and 52 detected cases. Eighty synthetic accounts were included in the dataset, and cross-dataset approach has detection ratio of 65% with the lowest amount of false positive cases. Although the experiments result was impressive as comparing with the others, the approach needs to be studied carefully under real fraud dataset. In addition, different parameters experiments for pre-defined rules and SVM have not been investigated when compare with the proposed technique. However, the peer group detection technique has been successfully adopted in other financial fraud detection such as stock fraud [93] and plastic card fraud [94]. In addition, the performance of peer comparison also depends on the clustering phase, which divides dataset into specific groups before estimating is identical rate.

3.8 Summary

Most of the reviewed papers prefer the AML typologies, the ability of an algorithm to detect previously identified suspicious transaction. Furthermore, the performance of the reviewed algorithms can be measured by various formulas that weigh the correct detection of normal and suspicious transactions.

The second most popular tool is the anomaly detection algorithm. Many researchers have found different ways to identify anomalies in transactions. However, it should be noted that the anomaly detection tool only identifies transactions that deviate from the norm. Such anomalies could be suspicious, but it can also be caused by some external factors that are up to the experts to investigate. For example, the bank customers could be receiving bonuses, celebrating festival seasons that caused fluctuations in their banking activities.

Table 1 illustrates a variation in machine learning algorithms, which have been explored and examined in the field of anti-money laundering (AML) from the existing literature.

Foremost, the method and algorithm detection accuracy is the most crucial element when determining the suitability of an approach. Unfortunately, detection accuracy cannot be directly measured because experimental results are largely influenced by the type of datasets and the size of datasets used for the evaluation. Also, since the majority of the money laundering cases were artificially created in most research studies during the training phase, it is not possible to analyse the detection accuracy between the various methods. Therefore, only conclusions based on the observations made by the authors for a particular experimental evaluation can be reported.

Based on the findings from existing literature reviews, it can be observed that most methods use similar attributes such as the amount received, the amount withdrawn, and the debit/credit transaction frequency within certain time windows, e.g. daily, weekly, and monthly. These include researches from [28,41,58–60,62,84,90]. The authors in [21,45,62,70,80] used additional attributes such as risk value, the individuals salary information, and the senders/receivers individual account history as part of their methods.

Most methods use real datasets when validating their approaches during the testing phase. This is ideal since real datasets determine the applicability of the method when applied in a real financial environment. However, it was observed that for most methods, the real datasets only comprised of normal transaction activities, whereas money laundering activities were artificially introduced. In fact, only for methods by [21,41,45,60,80,83,84,90] were datasets that contained real money laundering transaction cases. Most methods used a mixture of real normal transactions and artificially generated suspicious transactions. Due to the class imbalance issue, the proportion between normal and suspicious transactions was very

Table 1 Classification of machine learning algorithms in anti-money laundering (AML) field

Detection aspect	Learning class	Techniques	References
AML typologies	Supervised classification	Fuzzy logic, fuzzy inference system	[41]
		Support vector machine with heterogeneous distance HVDM estimation	[58,59]
		Neural network with RBF function and recursive least square training algorithm	[60]
	Unsupervised clustering	CLOPE clustering with sLOPE	[62]
	Semi-supervised	Neural network on cluster	[28]
		K-means, neural network, genetic algorithm	[64]
		BIRCH-balanced iterative reducing and clustering using hierarchies, K-means, decision tree	[65]
Link analysis	Semi-supervised	Graph clustering, frequent pattern FP Close, FP Max, sequence miner, visualisation	[69]
		Clustering, K-means, graph construction	[70]
	Other	Link and correlation analysis, K-means	[21]
Behavioural modelling	Supervised classification	Decision tree, random tree, random forest, J48graft, JRip	[81,82]
	Unsupervised clustering	EM—expectation–maximisation	[27,73]
	Other	EMD empirical mode decomposition on financial time series	[74]
Risk scoring	Supervised classification	Decision tree	[80]
		Sequence matching, nearest neighbour	[83]
	Unsupervised clustering	TEART Euclidean adaptive resonance theory	[84]
Anomaly detection	Supervised classification	Support vector machine	[92]
	Unsupervised clustering	Minimum spanning tree clustering	[89]
	Semi-supervised	Fuzzy c-means, dynamic Bayesian networks	[90]
Geographic capability	Semi-supervised	Neural network, genetic algorithm, case base reasoning	[91]

high (i.e. 99:1 ratio). The suspicious transactions were artificially generated by sampling techniques, cost-sensitive learning, and/or kernel-based and active learning methods.

The size of the experiment datasets will indicate the ability of the method to handle and process large volumes of transactions. From the findings of this study, it was observed that the majority of the methods were only validated on a small dataset. The small dataset refers to a transaction size of below 10,000 transactions. Only three methods were validated on medium datasets (less than 100,000 transactions) and large datasets (more than 1 million transactions). As a consequence of using small datasets, the time performance in the above techniques cannot be effectively evaluated and compared since the results are obtained very

fast. Furthermore, the datasets at above solutions are not consistent in terms of size and number of attributes. These methods need to be re-evaluated to determine whether they are able to scale effectively to a large volume of transaction data.

Tables 2 and 3 provide the comparisons of the AML detection categories in terms of their capabilities and features; Table 4 provides the comparisons of the AML detection categories in terms of scalability, recall/precision, efficiency, and applicability. Most of the methods concentrate on explaining the implementation and did not mention to handle missing values. The study of [84] provides feature selection/extraction by applying the PCA technique to the raw data. The principal component analysis (PCA) is a statistical technique that uses orthogonal transformation to combine similar/correlated attributes and create a new set of values of linearly correlated attributes. The main advantages of PCA are (1) lack of redundancy of data given the orthogonal components; (2) reduced complexity in images grouping with the use of PCA; (3) smaller database representation; and (4) reduction in noise. Later research of [67] suggests autoencode as a replacement for PCA.

It was further observed that most of the methods did not support reinforcement learning, which keeps system continuing to learn from new input data. Instead, the system was retrained from beginning when updates new data. As a result, the learning process requires more time and is not effective to handle large dataset. The approach in [60] had awareness of this issue and thus provided the ability to adapt unseen transaction data. Its novel and inventive approach to dynamically fine-tune the clustering radius when new data transactions are introduced allows the system to operate simultaneously with new training. This provides the ability for the neural networks to control the number of normal and suspicious clusters, and these clusters can be easily differentiated by their radius distance.

In general, for rule-based approaches, decision tree and its alternative versions are selected as the common solutions. Various discussed solutions have been developed from the DT structure such as [81] with random forest, Adaboost, or [85] inspired bitmap-based. Besides, for classification techniques, SVM and neural network are selected more often for enhancing performance or comparing with the proposed approaches. Different kernels are also examined during the experiments. C-SVM and one-class SVM are the two most used solutions while multi-layer network with backpropagation using gradient descent is a favourite choice for neural network. The trend of improving the performance of SVM and NN in discussed solution refers to the change of algorithm kernel such as RBF. Other solution also changed learning method from gradient descent to recursive least square [60]. For clustering technique, K-means has been mentioned mostly due to its simplicity and pure nature.

However, the contribution of K-means in the above studies is usually to compare with other techniques, or to reduce the learning pattern for later classification step at semi-supervised learning. Most of enhancing K-means approaches exploit various distance measurement methods instead of traditional Euclidean or Manhattan such as TEART [84]. Other techniques, which concentrate on evaluating the likelihood or matching probability rather than distance such as EM or CLOPE, produced more promising result and usually can be applied as the standalone solutions rather than combining with one classification techniques. Current state of the art in classification, deep learning was also examined as a promising solution for AML [67].

Most of the proposed techniques were compared with their ground truth using confusion matrix. The detection rate together with true positive rate and false positive rate was calculated frequently as proof of improvement. Some references also composed AUROC graph for intuitive demonstration of binary classifier. Recall to the dataset distribution in references which most of real dataset consist of above 90% transactions as normal. This imbalance dataset can mislead the performance to high accuracy as most of the output will easily result

Table 2 Comparisons of selective methods in terms of data processing features

Detection aspect	References	Data processing features			Data analytics supporting large data (samples)	Result analytics AUROC
		Combine different sources	Support automated feature selection and extraction	Support automatic parameter tuning		
AML typologies	[41]	✓	×	×	_	×
	[60]	✓	×	×	_	×
	[58]	✓	×	✓	✓	✓
	[62,65]	✓	×	×	×	✓
	[45,64]	✓	×	×	✓	×
	[28]	✓	×	×	✓	_
Link analysis	[69,70]	✓	×	×	×	×
Behavioural modelling	[41]	✓	×	×	×	_
	[81]	✓	×	×	_	×
Risk scoring	[80]	✓	×	×	_	_
	[83]	✓	×	×	_	✓
	[84]	✓	✓	✓	_	×
Anomaly detection	[59]	✓	×	×	_	✓
	[92]	✓	×	×	_	_
	[89]	✓	×	×	×	×
	[90]	✓	×	×	✓	_
Geographic capability	[91]	✓	_	_	_	×

Table 3 Comparisons of selective methods in terms of attributes used, datasets used, and summary of results

Detection aspect	References	Using real data set Number of attributes		Experiment size (training/testing) Number of suspicious (training/testing) Adding synthetic data (training/testing)			Detection rate TPR/FPR	
AML typologies	[41]	✓	3	_/710	_/_	X/X	0.1	_
	[60]	✓	3	270/90	70/40	X/✓	0.87	1/0.76
	[58]	✓	3	32,000/_	1930/_	✓/✓	0.63	0.79/0.06
	[64]	✓	6	≈3000/_	8/_	X/X	_	1/_
	[45]	Δ	10	100 mils/_	317/_	✓/✓	_	_/_
	[28]	✓	_	≈200 k/1.8 mils	_/10	X/X	0.005	1/_
Link analysis	[69]	X	5	_/_	_/_	✓/✓	_	_/_
Behavioural modelling	[27]	✓	70	2035/665	31/14	X/X	0.2	0.93/0.016
	[74]	✓	4	_/_	_/_	✓/✓	_	_/_
	[81]	X	_	_/_	6006/_	✓/✓	0.012	_/_
Risk scoring	[80]	✓	4	160,000/_	≈18,000/_	X/X	_	_/_
	[83]	✓	6	120,986/_	1940/_	X/X	_	0.954/0.558
	[84]	✓	_	_/_	_/_	_/_	_	_/_
Anomaly detection	[59]	✓	_	32,000/_	30/_	X/_	0.678	_/0.034
	[92]	✓	_	_/_	_/_	_/_	0.65	_/_
	[89]	✓	29	_/65,001	_/60	_/✓	_	0.72/_
	[90]	✓	_	_/_	_/_	_/_	_	_/_
Geographic capability	[91]	✓	_	_/_	_/_	_/_	_	_/_

Table 4 Comparisons of selective methods in terms scalability, recall and precision rate, efficiency, and applicability

Detection aspect	References	Scalability dealing with any amount of data	High recall/ precision rate	Efficiency fast training speed	Applicability parameter tuning needed
AML typologies	[41]	✓	✓	✓	✓
	[45]	✓	✓	_	Δ
	[58]	✓	✓	X	✓
	[60]	X	✓	✓	Δ
	[62]	✓	X	✓	Δ
	[28]	X	X	✓	Δ
	[64]	✓	_	✓	✓
	[65]	✓	X	_	✓
Link analysis	[69,70]	✓	X	✓	✓
Behavioural modelling	[73]	_	_	_	_
	[74]	✓	_	✓	Δ
	[81]	✓	_	_	_
Risk scoring	[80]	✓	_	_	✓
	[83]	✓	✓	_	✓
	[84]	✓	X	_	✓
Anomaly detection	[59]	✓	✓	_	✓
	[92]	✓	✓	✓	Δ
	[89]	✓	X	✓	✓
	[90]	✓	_	_	✓
Geographic capability	[91]	✓	_	_	_

in normal class. When apply AUROC with certain threshold, trade-off between specificity and sensitivity can be observed. Among various techniques, the best approach can identify conveniently, which has its curve closer to one and further from the threshold line. Nonetheless, the approach needs to be validated further since it was only tested with artificial and small datasets.

It is crucial to ensure that the method works effectively on real financial datasets with large volumes. Research is needed to examine more efficient methods for AML solutions. Hence, the opportunity to explore a state-of-the-art solution is still opened and this is a promising research area.

4 Commercial solutions

There are a number of commercial softwares for AML, such as NICE Actimize, SAS, bankers toolbox, Lexis-Nexis and Logica ISL, just to name a few. The suspicious activity monitor

features in NICE Actimize has extensive library define various models in different AML circumstance like banking, securities, insurance industries. In addition, it also provides advanced analytics through data visualisation and exports specific reports. SAS confronts AML and counterterrorist financing regulation with risk scoring and rule-based approaches. The solution also records behavioural activities for later assessment with peer-based analytics to improve accuracy.

In the past, most commercial solutions employ rule-based solutions to filter potentially suspicious transactions based on pre-defined rules and thresholds. Alerts will be issued if the rules and threshold criteria are met. There has been a recent development in using artificial intelligence to fight money laundering. Searchspace was one of the earliest companies to develop an AML solution without using fixed rules, but rather relied on statistical and machine learning techniques. The CEO of Searchspace described how Searchspaces commercial solution employs support vector machines (SVM) with probabilistic thresholds to detect anomaly transactions [95].

The approach is to discover anomalies from large dimensionality: customers, accounts, products, geography, and time. A multi-dimensional adaptive probabilistic matrix is generated for each bank customers. Based on the probabilistic matrix and some pre-defined probabilistic thresholds, the system generates suspicious alerts. The system also uses peer groups in the matrix to establish unusualness outside an individuals behavioural norm. Due to the success of the system, Actimize acquired Searchspace for $73.5 million in August 2004 [96]. NICE Actimize claims it can successfully detect 73% money laundering activities with a ratio of 1:15 false positives (detecting one genuine money laundering [2] transaction out of 15 generated alerts) [97]. Other solutions that use artificial intelligence techniques also made similar claims; for instance, Logica ISL claims its solution can reduce the false positive hit rate by over 50% [98].

The current commercial solutions are capable of providing users the tools to detect and identify suspicious transactions cases. Such strategies are good at detecting suspicious transactions based on the characteristics of various money laundering cases; however, the learning models are constructed in the way that it is generalised to avoid overfitting a particular scenario that could result in high false positive rates. Hence, other than correctly identifying suspicious transactions, a competent AML solution should also consider the output of every tool in the solution to make decisions. The trend is to use the Bayesian belief network (BBN) and deep learning that replicate the human reasoning in establishing inference in making a decision. This way, a set of transactions will be marked as anomalies if it is suspicious within the context of a particular customer [36].

Anti-money laundering system has been widely used to prevent the funding of terrorist organisations. However, researchers still have little attention on the law confliction when developing the detection tools. Most of these common problems involve in confidentiality and discrimination. As recall from previous section, geographic capability is also considered as one important aspect to build up a detection system. If the policies and controls at the time of opening account are not handled well, it may lead to the case of over bias learning. As a result, the trend of specific locations, which are always marked as suspicious, may occur. When the system combines information to obtain customer profile, it may also result in some groups of people not allowed to perform normal financial actions. And then, the decision can also affect ordinary individuals.

Hong Kong monetary authority (HKMA) has used the AIs (in this case banks licensed by the HKMA) to filter 'authorised institutions', and it was too sensitive. For instance, there have been some individuals from certain ethnic minorities that had encountered difficulties in applying for bank accounts with certain AIs. The reason was AIs believed that this has

been linked to the anti-money laundering and countering the financing of terrorism. As from the response from HKMA, they indicated: 'AIs should adopt a balanced and common sense approach with regard to customers connected with jurisdictions which do not or insufficiently apply the financial action task force (FATF) recommendations. While extra care may well be justified in such cases, it is not a requirement that AIs should refuse to do any business with such customers or automatically classify them as high risk and subject them to enhanced customer due diligence (CDD) process. Rather, AIs should weigh all the circumstances of the particular situation and assess whether there is a higher than normal risk of money laundering'.

However, there are still difficulties in providing guidance based on common sense whether it is the responsibility of the banks to provide automated decision management software based on business rules software. Thus, the later AML system may refer to the above suggestions to avoid conflicts. Checklists provide a simple CDD risk rating, with an outcome driven by a numeric result.

5 Conclusion and future work

The importance of having effective systems, controls, and practices in place to manage the risk of money laundering activities for banks requires effective ways to detect money laundering activities. Anti-money laundering (AML) solution system, being part of the overall fraud control, automates and helps reduce the manual parts of a screening/checking process. Effective ways to AML solutions are necessary, but highly challenging. In particular, the rising volumes of customer transactions and the increased automated interaction with customers have made AML compliance more difficult. This paper provides a comprehensive and most recent survey of published research results in AML solutions.

The overall objective of the research in this area is to review a state-of-the-art AML solution that offers data quality assurance, high detection accuracy, scalability, and fast performance in suspicious transaction detection. In achieving these goals, a number of important features that the solution should be capable of were identified. These include efficient data preparation, data transformation, and data analytics techniques. Almost published anti-money laundering techniques were explored, and an in-depth examination was conducted in terms of the attributes/variables or type of dataset.

Based on the findings, it is worth to notice that current methods and algorithms in literature have little attention to the data quality assurance. At first, raw data obtained from financial institutions often result in extremely large volumes. In addition, as recall from Table 3, the datasets for AML are terribly imbalanced as measured in the number of transactions. Consequently, an effective data refinement process, which eliminates similar records or generate synthetic data properly, can boost the learning process more effectively such as SMOTE [99] or multiple resampling method [100]. On the other hand, studies in [28,64,65,90,91] suggest handling the challenge of large data size by applying semi-supervised techniques. After the clustering process, the entire input data will be categorised into smaller groups.

Learning data can be reduced by randomly selecting representatives from the trivial normal group instead of using the whole data. As a consequence, the performance is enhanced significantly when the specific learning model is applied for a specific cluster. In addition, most of the studies select attributes for ML technique based on their own business flow. The features selection techniques have not been examined thoroughly to support the performance.

Since the financial operations may vary from time to time, the need of reinforcement learning that keeps on training should also be put into consideration.

Acknowledgements This work was supported by a 3rd Called Collaboration with Public Universities and Agencies grant from the University of Nottingham, Malaysia Campus with Project No. UNHT0001.

References

1. Kou Y, Lu C-TT, Sirwongwattana S, Huang YP, Sinvongwattana S (2004) Survey of fraud detection techniques. In: 2004 IEEE international conference on networking sensing and control, vol 2(3), pp 749–754
2. Huang JY (2015) Effectiveness of US anti-money laundering regulations and HSBC case study. J Money Laund Control 18(4):525–532
3. Mollenkamp C, Wolf B (n.d.) HSBC to pay record $1.9 billion US fine in money laundering case. Accessed on 29 Dec 2016 (Online). http://uk.reuters.com/article/2012/12/11/uk-hsbc-probe-idUKBRE8BA05K20121211
4. Claudio G, John E (n.d.) Iranian dealings lead to a fine for credit suisse. Accessed on 29 Dec 2016 (Online). http://www.nytimes.com/2009/12/16/business/16bank.html?_r=0
5. Harry W (n.d.) Major banks still vulnerable to money laundering, says top regulator. Accessed on 29 Dec 2016 (Online). http://www.telegraph.co.uk/finance/newsbysector/banksandfinance/10153728/Major-banks-still-vulnerable-to-money-laundering-says-top-regulator.html
6. Reed A (n.d.) ING fined a record amount. Accessed on 29 Dec 2016 (Online). http://online.wsj.com/news/articles/SB10001424052750457746251271333363F78
7. Standard Bank Fined Over Lax Anti-Money Laundering Controls (n.d.). Accessed on 29 Dec 2016 (Online). http://www.bbc.com/news/business-25864499
8. Michael K (n.d.) FCA fines standard bank £7.6m for slack anti-money laundering controls. Accessed on 29 Dec 2016 (Online). http://www.ibtimes.co.uk/fca-fines-standard-bank-7-6m-slack-anti-money-laundering-controls-1433478
9. Gao S, Xu D, Wang H, Green P (2009) Knowledge based anti money laundering: a software agent bank application. J Knowl Manag 13(2):63–75
10. Verhage A (2009) Between the hammer and the anvil? the anti-money laundering-complex and its interactions with the compliance industry. Crime Law Soc Change 52(1):9–32
11. Cahill MH, Lambert D, Pinheiro JC, Sun DX (2002) Detecting fraud in the real world. Handbook of massive data sets. Springer, Berlin, pp 911–929
12. Gao Z, Ye M (2007) A framework for data mining-based anti-money laundering research. J Money Laund Control 10(2):170–179
13. Arquilla J, Ronfeldt D (2002) Networks and netwars. In: The future of terror, crime and militancy, pp 80–82
14. Liu X, Zhang P (2010) A scan statistics based suspicious transactions detection model for anti-money laundering (AML) in financial institutions. In: Proceedings—2010 international conference on multi-media communications, Mediacom, pp 210–213
15. Sudjianto A, Nair S, Yuan M, Zhang A, Kern D, Cela-Díaz F, Cela F (2010) Statistical methods for fighting financial crimes. Technometrics 52(1):5–19
16. Yue D, Wu X, Wang Y, Li Y, Chu CH (2007) A review of data mining-based financial fraud detection research. 2007 international conference on wireless communications, networking and mobile computing, WiCOM 2007, pp 5514–5517
17. Han J (2005) Data mining: concepts and techniques. Morgan Kaufmann Publishers Inc., San Francisco
18. Murphy KP (2012) Machine learning: a probabilistic perspective. The MIT Press, Cambridge
19. Fawcett T, Provost F (1997) Adaptive fraud detection. Data Min Knowl Disc 1(3):291–316
20. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. ACM Comput Surv 41(3):15:1–15:58
21. Zhang ZM, Salerno JJ, Yu PS (2003) Applying data mining in investigating money laundering crimes. ACM, New York, pp 24–27
22. Mannes J (n.d.) Another salesforce acquisition with beyondcore enterprise analytics grab. Accessed on 5 Jan 2017 (Online). https://techcrunch.com/2016/08/15/another-salesforce-acquisition-with-beyondcore-enterprise-analytics-grab/
23. Institute S (2008) SAS/STAT(R) 9.1 user's guide: the REG procedure (book excerpt). SAS Institute, Cary

24. Kepes B (n.d.) More vertical analytics solutions—INETCO goes analytical on ATM data. Accessed on 10 August 2017 (Online). https://www.forbes.com/sites/benkepes/2015/01/15/more-vertical-analytics-solutions-inetco-goes-analytical-on-atm-data/#5755cb4f469c
25. Zhang S, Zhang C, Yang Q (2003) Data preparation for data mining. Appl Artif Intell 17(5–6):375–381
26. Schmidt A (2013) Know your customer (technology abstract). Technical report, The Corporate Executive Board Company
27. Chen Z, Van Khoa LD, Nazir A, Teoh EN, Karupiah EK (2014) Exploration of the effectiveness of expectation maximization algorithm for suspicious transaction detection in anti-money laundering. ICOS 2014–2014 IEEE conference on open systems, pp 145–149
28. Le-Khac NA, Markos S, Kechadi MT (2010) Towards a new data mining-based approach for anti-money laundering in an international investment bank. In: Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering, vol 31 LNICST, pp 77–84
29. Donders AR, van der Heijden GJ, Theo S, Karel GM (2006) Review: a gentle introduction to imputation of missing values. J Clin Epidemiol 59(10):1087–1091
30. Brown ML, John FK (2003) Data mining and the impact of missing data. Ind Manag Data Syst 3(71–81):611–621
31. Garfinkel SL (2006) Forensic feature extraction and cross-drive analysis. Dig Investig 3:71–81
32. Shvachko K, Kuang H, Radia S, Chansler R (2010) The hadoop distributed file system. In: Mass storage systems and technologies (MSST), Incline Village
33. Russell SJ, Norvig P (2002) Artificial intelligence: a modern approach, 2nd edn. Prentice Hall, Upper Saddle River
34. Tan P-N, Steinbach M, Kumar V (2005) Introduction to data mining, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston
35. PWC (n.d.) PWC. Accessed on 5 Jan 2017 (Online). http://www.pwc.com/gx/en/financial-services/publications/anti-money-laundering-know-your-customer-quick-reference-guide.jhtml
36. Schmidt A (2013) Anti-money laundering: technology analysis abstract. Technical report
37. Zadeh L (1965) Fuzzy sets. Inf Control 8(3):338–353
38. Babuska R (1998) Fuzzy modeling for control, 1st edn. Kluwer, Norwell
39. Mamdani EH, Assilian S (1975) An experiment in linguistic synthesis with a fuzzy logic controller. Int J Man Mach Stud 7(1):1–13
40. Takagi T, Sugeno M (1985) Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern SMC–15(1):116–132
41. Chen Y-T, Mathe J (2011) Fuzzy computing applications for anti-money laundering and distributed storage system load monitoring. In: World conference on soft computing
42. Ishibuchi H, Nojima Y (2006) Tradeoff between accuracy and rule length in fuzzy rule-based classification systems for high-dimensional problems. In: 11th international conference on information processing and management of uncertainty in knowledge-based systems
43. Ishibuchi H, Nojima Y (2007) Analysis of interpretability–accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning. Int J Approx Reason 44(1):4–31
44. Sudkamp T, Hammell RJ (1998) Scalability in fuzzy rule-based learning. Inf Sci 109(1–4):135–147
45. Luo X (2014) Suspicious transaction detection for anti-money laundering. Int J Secur Appl 8(2):157–166
46. Han J, Hei J, Yin Y (2000) Mining frequent patterns without candidate generation. In: ACM SIGMOD international conference on management of data, Dallas
47. Grahne ZJ (2003) Efficiently using prefix-trees in mining frequent itemsets. In: ICDM 2003 workshop on frequent itemset mining implementations, Melbourne
48. Grahne ZJ (2005) Fast algorithms for frequent itemset mining using fptrees. IEEE Trans Knowl Data Eng 17(10):1347–1362
49. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
50. Bhattacharyya S, Jha S, Tharakunnel K, Westland JC (2011) Data mining for credit card fraud: a comparative study. Decis Support Syst 50(3):602–613
51. Pengcheng W, Dietterich TG (2004) Improving SVM accuracy by training on auxiliary data sources. In: Proceedings of the 21st international conference on machine learning, ICML'04, p 110
52. Gabrilovich E, Markovitch S (2004) Text categorization with many redundant features: using aggressive feature selection to make SVMS competitive with c4.5. In: Proceedings of the 21st international conference on machine learning, ICML'04. ACM, New York, p 41
53. Segata N, Blanzieri E (2011) Operators for transforming kernels into quasi-local kernels that improve SVM accuracy. J Intell Inf Syst 37(2):155–186
54. Chen Z, Olugbenro O, Seng NLC (2016) Equipment failure analysis for oil and gas industry with an ensemble predictive model. In: The 5th international conference on computer science and computational mathematics, ICCSCM 06

55. Gonzalez JL, Marcelin-Jimenez R (2011) Phoenix: a fault-tolerant distributed web storage based on URLs. In: 2011 IEEE 9th international symposium on parallel and distributed processing with applications, pp 282–287
56. Habib Soliman M, Jugal K (2010) Scalable biomedical named entity recognition and investigation of a database and supported svm approach. Int J Bioinform Res Appl 6(2):191–208
57. Joachims T (2006) Training linear SVMS in linear time. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, KDD'06. ACM, New York, pp 217–226
58. Liu K, Yu T (2011) An improved support-vector network model for anti-money laundering. In: Management of e-commerce and e-government (ICMeCG), Hubei
59. Tang J, Yin J (2005) Developing an intelligent data discriminating system of anti-money laundering based on SVM. In: International conference on machine learning and cybernetics, Guangzhou
60. Lv L-T, Ji N, Zhang J-L (2008) A RBF neural network model for anti-money laundering. In: Wavelet analysis and pattern recognition, Hong Kong
61. Hwang Y-S, Bang S-Y (1994) A neural network model APC-III and its application to unconstrained handwritten digit recognition. In: International conference on neural information processing
62. Cao DK, Do P (2002) Applying data mining in money laundering detection for the vietnamese banking industry. In: 4th Asian conference on intelligent information and database systems
63. Yang Y, Guan X, You J (2002) CLOPE: a fast and effective clustering algorithm for transactional data. In: Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining, Alberta
64. Le-Khac N-A, Kechadi M-T (2010) Application of data mining for anti-money laundering detection: a case study. In: Data mining workshops (ICDMW), Sydney
65. Liu R, Qian X-L, Mao S, Zhu S-Z (2011) Research on anti-money laundering based on core decision tree algorithm. In: Chinese control and decision conference (CCDC), Mianyang
66. Zhang T, Ramakrishnan R, Livny M (1996) BIRCH: an efficient data clustering method for very large databases. ACM SIGMOD Record 25(2):103–114
67. Paula EL, Ladeira M, Carvalho RN, Marzagão T (2016) Deep learning anomaly detection as support fraud investigation in Brazilian exports and anti-money laundering. In: IEEE international conference on machine learning and applications (ICMLA), Anaheim
68. Spring R, Shrivastava A (2017) Scalable and sustainable deep learning via randomized hashing. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, KDD'17. ACM, New York, pp 445–454
69. Dreżewski R, Sepielak J, Filipkowski W (2012) System supporting money laundering detection. Dig Investig 9(1):8–21
70. Dreżewski R, Sepielak J, Filipkowski W (2014) The application of social network analysis algorithms in a system supporting money laundering detection. Inf Sci 295:18–32
71. Colladon AF, Remondi E (2017) Using social network analysis to prevent money laundering. Expert Syst Appl 67:49–87
72. Demetis DS (2010) The risk-based approach and a risk-based data-mining application. In: Technology and anti-money laundering: a systems theory and risk-based approach. Edward Elgar Publishing, Cheltenham. https://doi.org/10.4337/9781849806657
73. Chitra K, Subashini B (2013) Data mining techniques and its applications in banking sector. Int J Emerg Technol Adv Eng 3(8):219–226
74. Zhu T (2006) Suspicious financial transaction detection based on empirical mode decomposition method. In: IEEE Asia-Pacific conference on services computing, Guangzhou
75. Quinlan J (1986) Induction of decision trees. Mach Learn 1(1):81–106
76. Breiman L, Friedman J, Olshen R, Stone C (1984) Classification and regression trees. Wadsworth and Brooks, Monterey
77. Loh W-Y (2009) Improving the precision of classification trees. Ann Appl Stat 3(4):1710–1737
78. Kamber M, Winstone L, Gong W, Cheng S, Han J (1997) Generalization and decision tree induction: efficient classification in data mining. In: Proceedings 7th international workshop on research issues in data engineering. High performance database management for large-scale applications, pp 111–120
79. Sudhakar M, Reddy CVK, Pradesh A (2016) Two step credit risk assesment model for retail bank loan applications using decision tree data. Int J Adv Res Comput Eng Technol (IJARCET) 5(3):705–718
80. Wang S-N, Yang J-G (2007) A money laundering risk evaluation method based on decision tree. In: Machine learning and cybernetics, Hong Kong
81. Rojas L, Alonso E, Axelson S (2012) Multi agent based simulation (MABS) of financial transactions for anti money laundering (AML). In: The 17th Nordic conference on secure IT system, volume: short-paper proceedings

82. Rojas L, Alonso E, Axelson S (2012) Money laundering detection using synthetic data. In: The 27th annual workshop of the Swedish artificial intelligence society (SAIS), Karlskrona
83. Liu X, Zhang P, Zeng D (2008) Sequence matching for suspicious activity detection in anti-money laundering. In: Intelligence and security informatics, Taipei, pp 50–61
84. Larik AS, Haider S (2011) Clustering based anomalous transaction reporting. In: Procedia computer science, Pakistan
85. Vikas J, Balan RS (2016) Money laundering regulatory risk evaluation using bitmap index-based decision tree. J Assoc Arab Univ Basic Appl Sci 23:96–102
86. Cortinas R, Freiling FC, Ghajar-Azadanlou M, Lafuente A, Larrea M, Penso LD, Soraluze I (2012) Secure failure detection and consensus in trustedpals. IEEE Trans Dependable Secure Comput 9(4):610–625
87. Phua C, Smith-Miles K, Lee V, Gayler R (2012) Resilient identity crime detection. IEEE Trans Knowl Data Eng 24(3):533–546
88. Schölkopf B, Platt JC, Shawe-Taylor JC, Smola AJ, Williamson RC (2001) Estimating the support of a high-dimensional distribution. Neural Comput 13(7):1443–1471
89. Wang X, Dong G (2009) Research on money laundering detection based on improved minimum spanning tree clustering and its application. In: 2nd international symposium on knowledge acquisition and modeling
90. Raza S, Haider S (2010) Suspicious activity reporting using dynamic Bayesian networks. Procedia Computer Science
91. Yang Q, Feng B, Song P (2007) Study on anti-money laundering service system of online payment based on union-bank mode. In: Wireless communications, networking and mobile computing, Shanghai
92. Tang J (2006) A peer dataset comparison outlier detection model applied to financial surveillance. In: Pattern recognition, Hong Kong
93. Kim Y, Sohn SY (2012) Stock fraud detection using peer group analysis. Expert Syst Appl 39(10):8986–8992
94. Weston DJ, Hand DJ, Adams NM, Whitrow C, Juszczak P (2008) Plastic card fraud detection using peer group analysis. Adv Data Anal Classif 2(1):45–62
95. Kingdon J (2004) AI fights money laundering. Intell Syst 19(3):87–89
96. NiceActimize (2009) Fortent is now part of NICE actimize. Accessed on 5 Jan 2017 (Online). http://www.niceactimize.com/index.aspx?page=fortent
97. NiceActimize (n.d.) Nice actimize: top-5 US Bank using fraud prevention solution from actimize, a NICE Company, detects 73% of wire fraud attempts in real-time and realizes 100% ROI on 7-digit investment within six weeks. Accessed on 5 Jan 2017 (Online). http://www.prnewswire.com/news-releases/top-5-us-bank-using-fraud-prevention-solution-from-actimize-a-nice-company-detects-73-of-wire-fraud-attempts-in-real-time-and-realizes-100-roi-on-7-digit-investment-within-six-weeks-62237467.html
98. Fulton S (n.d.) Logica announces new intelligent self-learning software to increase banks' filtering systems efficiency. Accessed on 5 Jan 2017 (Online). http://www.marketwired.com/press-release/logica-announces-new-intelligent-self-learning-software-increase-banks-filtering-systems-1688140.htm
99. Ramentol E, Caballero Y, Bello R, Herrera F (2012) SMOTE-RSB*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using SMOTE and rough sets theory. Knowl Inf Syst 33(2):245–265
100. Estabrooks A, Jo T, Japkowicz N (2004) A multiple resampling method for learning from imbalanced data sets. Comput Intell 20(1):18–36

Zhiyuan Chen (Ph.D. in Computer Science) is an Assistant Professor with the University of Nottingham, School of Computer Science, in Malaysia and a Principal Consultant with MIMOS at the Accelerative Technology Lab. She received the M.Phil. and a Ph.D. in Computer Science from the University of Nottingham in 2007 and 2011, respectively. Before joining UNMC, she has been a research associate in the UK Horizon Digital Economy Research Institute. Her research interests are in the area of computer science, machine learning, data mining, user modelling, and artificial intelligence.

Le Dinh Van Khoa (M.Sc.) finished Master degree in Computer Science from University of Nottingham in 2014. He participated in AML project at MIMOS Bh.d. as a research assistant. He is currently a Ph.D. candidate in the University of Nottingham, Malaysia Campus

Ee Na Teoh (B.CS.) is a Researcher in the Accelerative Technology Lab of ICT Division at Mimos Bh.d. She is pursuing Master of Computer Science in Tunku Abdul Rahman University (UTAR).

Amril Nazir received the Ph.D. degree in Computer Science from University College London in 2011. He is currently an Assistant Professor at Taif University, Kingdom of Saudi Arabia (KSA). His research interests include high-performance computing, cloud computing, big data processing, and machine learning. Previously he worked as a Senior Researcher in the Accelerative Technology Lab Group at MIMOS Bh.d, a government R&D institute.

Ettikan Kandasamy Karuppiah (Ph.D in the area of Distributed Computing) is the Principal Researcher and Head of Accelerative Technology Lab of ICT Division at MIMOS Bh.d. Current research interest includes distributed computing, big/media data processing, data sciences, many/multi-processors, algorithm research and optimisation, GPGPU/FPGA, AV, QoS, network processing, system architecture, and performance optimisation. Previously he was attached with Panasonic R&D and Intel Communication Group with R&D responsibility in IP, AV, distributed and embedded communications protocols, and network processors (www.ettikan.org).

Kim Sim Lam (B.Sc.) finished B.Sc. degree in Computer Science from The University of Nottingham in 2016. He was working as a research assistant with Dr. Chen in July 2016. Currently an MPhil student with the Department of Electrical and Electronic Engineering, the University of Nottingham, Malaysia Campus.