CASPIAN

Trusting Machine Learning in Anti-Money Laundering: A Risk-Based Approach

Joe Faith[1], Bashar Awwad Shiekh Hasan[1], and Amir Enshaie[1,2]
1: Caspian Learning Ltd, UK
2: Translational and Clinical Research Institute, Faculty of Medical Sciences, Newcastle University, UK
contact: joe.faith@caspian.co.uk
v1.2, 06/01/2020, for external review

Contents

Executive Summary

Machine Learning (ML) is increasingly used by financial institutions for the detection and investigation of financial crime, and Anti-Money Laundering activity (AML) in particular, but how can we can trust these systems? This paper takes a Risk-Based Approach (RBA), as mandated by international regulators. It identifies common risks and possible mitigations. It presents criteria for judging the maturity of a Machine Learning (ML) system, and the degree of human oversight required, similar to the levels of driving automation for autonomous vehicles.

We find that the **minimum** requirement for an ML to make decisions on live cases is that those decisions can be explained and justified, **and** that significant performance risks are identified and managed. Without these, there can be no effective accountability for the decisions that it makes.

This framework is based on our experience deploying ML systems in financial and clinical domains, but the overall strategy may be useful for other safety-critical and highly-regulated applications.

Machine Learning in Anti-Money Laundering

It is estimated that 6.7% of global GDP, worth US$5.8tn, is due to criminal activity, from counterfeit goods to the trafficking of humans, drugs, and arms. Three quarters of these funds are then laundered through the international financial system so they can be used by criminals to make legitimate purchases, or to fund terrorist activity (FCN 2019). There are now stringent requirements on financial institutions in major markets to investigate and detect suspicious activity, and large fines have been levied where compliance with these requirements has not been met (BIS 2016). This places a large burden on financial institutions to ensure regulatory compliance. Approximately 1 million Suspicious Activity Reports (SAR's) are generated every year globally, one every 30 seconds, and approximately 10% of all staff in major financial institutions are engaged in compliance activity. Increasingly, banks are adopting solutions that use machine learning, in whole or in part, to manage this process. The Institute of International Finance found that, as of 2018, 35% of 59 financial institutions surveyed were using machine learning in AML, and a further 34% were experimenting with pilot projects (IIF 2018).

Machine Learning is primarily used in two classes of Anti Money Laundering (AML) applications.

1. **Customer Due Diligence**: investigating new, and existing, customers to understand if they represent a risk of money laundering. These background checks are usually done before the customer is onboarded, and then repeated periodically, using a mixture of public and private sources of information (proof of identity, company registers, previous bank statements, reports of criminal activity, etc). One of the most important questions in CDD is ascertaining the 'source of wealth', whether the funds being deposited in the institution can be accounted for through legitimate routes such as inheritance or legitimate business activity.
2. **Transaction Monitoring**: is a customer's account being used for legitimate purposes, or is it now being used to launder money? This typically involves screening all transactions looking for

suspicious activity (such as frequent large credits matched with large debits), or activity that does not fit the profile of the customer (e.g. a student with unusually large funds).

There are many challenges with applying machine learning techniques to Anti Money Laundering investigations, including:

1. **Lack of ground truth:** criminals don't want to be found. In most cases financial institutions do not have objective ground truth of whether a customer or transaction in their training data are proven criminal activity or not, instead they have to assess levels of risk based on best available expert assessment.
2. **Skewed class distributions**: only a small minority of transactions or customers are suspicious, and the asymmetry affects how generalisation performance is measured. (If 99% of all transactions are innocent, then a system that rules that *all* transactions are innocent will be right 99% of the time).
3. **High cost of labelled data:** human expertise in anti-money laundering is expensive, so labelled training sets are typically small.
4. **Regulation and compliance:** financial institutions are subject to strict regulatory oversight, so evidence and audit trails for decisions are important.
5. **Risk escalations:** Machine learners are just one part of an overall chain of responsibility, up to a, for example, Money Laundering Reporting Office who is required under UK law to provide oversight and responsibility for the Anti Money Laundering processes at all levels.

But the focus in this paper is the issue of ***Trust***.

Why is Trust a Problem for Machine Learning?

Software systems have been routinely trusted in safety-critical applications for decades, from aircraft and spacecraft control systems to managing nuclear power plants. Building software systems that we can trust is not a solved problem, but it is a well understood one. So why do machine learning techniques introduce novel challenges?

The strength, and weakness, of ML is that it can be applied in applications where we cannot define a formal specification for the solution but can only give examples of correct behaviours. This presents two challenges: **correctness** and **accountability**.

Correctness is a problem because we cannot guarantee that the system will do the right thing (make the correct decisions) in future circumstances. The system will be trained on a set of cases for which the correct behaviour is known, but when in production it will be required to generalise to unseen cases. The most basic strategy for testing generalisation performance is cross-validation, or out-of-sample testing: using some of the training data as a test set. But, as we shall see, this is a very low bar. Other processes are required to ensure the decisions made on live cases can be trusted.

The second, less obvious, problem is that financial institutions, and not artificial intelligences, are accountable for the decisions that they make, and accountability requires that they can *explain* and

justify those decisions. If automated systems are built to explicit requirements, then humans can be held accountable for both their correctness and implementation. But machine learners such as 'deep' neural networks are often black boxes, which by their nature make it difficult to understand how individual decision are made. As a result, the explainability of such systems has become a major research topic (see Samek et al, 2019 for a summary, and further discussion below). But even machine learners that generate explicit rules, such as decision trees, that require no further explanation, still face the problem of *justifying* that those rules can be trusted, that they are the correct rules to rely on.

Underlying both these issues is a general principle that AI should not increase the risk assumed by the organization as compared to a human-based process. This, in turn, requires that innocent humans should not be harmed by the introduction of artificial intelligence, either through intent or neglect. Automated systems are often introduced by financial systems to reduce the overhead of existing human-based processes, but the quality of those processes and the decisions they produce should not suffer as a result. Trust, in this context, means trust that the institution is doing the right thing by its customers and other stakeholders such as regulators, as well as trusting that the machine learner is producing the decisions that the organisation considers correct.

The Cost of Failure

The failure to build ML systems we can trust has human, financial, and institutional costs:

- **Human:** We will not know if the ML is effectively identifying criminal activity, or whether it is harming innocent parties.
- **Financial:** Resources will be wasted on ineffective solutions. A lack of trust also means that ML processes are often shadowed or replicated or reviewed by human processes, preventing us realising the full benefits of automation.
- **Institutional:** Regulators have repeatedly sanctioned financial institutions for systemic and human failures in their AML processes; and institutions are also accountable for failures in the ML systems they employ.

Why Use a Risk-Based Approach?

Most approaches to trust and machine learning have focussed on individual technical solutions (see, for example, IEEE-TPS and Marcus 2019), and although each of these may be valuable and important, they only address individual issues. They can only provide the assurances we need when put in the context of a complete risk management process.

Both the Bank of International Settlements (2016) and the G7 Financial Action Task Force on Money Laundering (FATF 2014) mandate that a Risk-Based Approach (RBA) should be used for Anti Money Laundering and Counter-Terrorist Financing. As the FATF guidelines state, in their rationale for taking a Risk Based Approach

> The 2012 Recommendations consider the RBA to be an 'essential foundation' of a country's AML/CTF framework. This is an over-arching requirement applicable to all

> relevant FATF Recommendations. ... The application of a RBA is therefore not optional, but a prerequisite for the effective implementation of the FATF Standards (FATF 2014)

If RBA is mandated for all AML/CTF then it is also necessary for the machine learning processes they increasingly depend on.

A RBA is based around three activities: identifying risks, assessing risks, and mitigating risks. In the next section we identify some of the major individual risks associated with using machine learning in AML, along with possible mitigations. We then use this to build a maturity model for judging when we can trust a machine learning system, and what safeguards it requires. The risks we identify, priorities, and mitigations, are based on applications in Anti Money Laundering, but we expect that the strategy and model will apply to other domains.

Risk Management Culture

The goal of risk management is not to eliminate all risks – or all incorrect decisions, in the case of an ML system – because this is not possible. Bear in mind that no *human*-based process will be perfect. As the FATF guidance states, 'the RBA is not a “zero failure” approach; there may be occasions where an institution has taken all reasonable measures to identify and mitigate AML/CFT risks, but it is still used for Money Laundering or Terrorist Financing purposes' (FATF 2014). Rather, the goal is to ensure that there is a process in place to confirm that everything possible is done to identify, prioritise, and mitigate those risks at a level that is acceptable to the organisation and regulators.

Making 'zero failure' the only goal discourages honest and thorough testing, and it discourages openness about possible risks – both of which are necessary to ensure risks are identified and addressed. Rather, it encourages a culture of secrecy and denial both within organisations and between vendors and customers. Vendors should be judged on how thorough and open they are about their risk management processes.

Risks and Mitigations

Machine Learning systems can make incorrect decisions for many reasons. This list is based on our experience in building and deploying machine learning-based systems in for detecting and investigating Money Laundering threats that are used in production by Tier 1 financial institutions, and in using ML to develop prognostic and predictive tools for the treatment of childhood leukaemia, but the overall strategy and model may be useful for applying machine learning in other safety-critical and highly-regulated domains.

It is not intended to be an exhaustive list of ML best practice, but to serve as an illustration and starting point for a risk management process. *The process is more important than the particular list of risks.*

1. Poorly Labelled Training Data

Garbage in, Garbage out! If the labels in the training and testing data is incorrect then the machine learner will produce incorrect decisions, and, worse, the institution will not know it is producing incorrect decisions. Anti-Money Laundering is a complex, specialised, domain and human expertise is rare and expensive, so the monotonous task of labelling often large amounts of training data is sometimes relegated to less-skilled subject matter experts (SMEs).

Mitigations

1. **Independent labelling:** All training data should be labelled by multiple independent analysts who are subject matter experts in their field (SMEs), as judged by the institution.
2. **Data access**: SMEs should have access to all background information on the case available to the institution -- and not just the data shown to the machine learner. They should also make a note of how they reached their decision, especially if other data sources were consulted, so that it can be determined if the solution needs to include these data sources in order to make reliable judgements.
3. **Manage consensus:** Simple majority consensus is a good starting point but does not work where the number of labels is greater than the number of labellers (Brodley and Friedl, 1999). Disagreements between SMEs should be escalated to more senior authorities for resolution.
4. **Monitor performance:** Consensus management can be improved by monitoring the performance of individual SMEs, identifying those who most frequently diverge from the consensus. Weight decisions by the degree of trust in the labellers.
5. **Use probabilistic solutions**: they are available for identifying incorrect labels, and improving generalisation performance in the face of label noise – see Northcutt et al 2020 for a concise survey. They may be useful in some domains where training data is plentiful and trust requirements are lower, but we do not believe they would be suitable for high risk, low volume, domains.

2. Reproducibility and Traceability

If the financial institution cannot recreate the conditions – including the training data and state of the model – that lead to a decision, then it cannot provide full justification of a decision or the evidence that lead to it, or reproduce the decision if it is contested. This means that the processes, and the organisations and individuals responsible for them, cannot be held accountable and cannot be regulated.

Mitigations

1. **Link models:** All decisions should be linked to a model version, and all model versions should be linked to an archive of the training data, model algorithm, training regime, and any prior parameters used in training the model (including any randomisation seeds). This version linkage should include all external libraries used by the learner.

2. **Test your processes:** Specifically those processes involved in recovering and reproducing a decision from archived code and data. Unless and until you have tested the recovery process then assume that the data is lost.

3. Opacity

If humans cannot understand how an ML derives a decision from input and training data, then they are effectively taken out of the loop. In particular:

1. Higher-level analysts (from L3 analysts up to the Money Laundering Reporting Officer) cannot use the outputs of the ML in their wider and deeper investigation of the case. They must recreate the work of the ML in order to understand which factors were important and which require further validation.
2. Decisions cannot be explained or justified to regulators or auditors.
3. Human monitoring processes cannot judge the quality or validity of decisions that they might want to contest.
4. Human expert understanding is hard to use in improvingmodels because the reasons for any errors are unclear.
5. Humans cannot provide an effective fail-safeby being extra-eyes to validate decisions.
6. Biases are hard to detect (see below)

Mitigations

The decisions of the ML should be explained in a form, and level of detail, such that a non-technical SME would agree that the decisions were justified based purely on that explanation. Ideally this should include both qualitative, plain language explanations, backed by quantitative evidence. See Molnar (2018) for an introduction to the issues of explaining decisions in ML models. If the ML is not 'natively' explainable – i.e. if additional software and is required to explain its decision – then an audit trail should be available that shows how that explanation is derived from the state of the model given the inputs for that case.

4. Hidden Stratification

Aggregate generalisation performance figures may be misleading. Even if the overall performance is good, the ML may perform poorly on important subsets, or 'strata' – especially where they are rare or harder to diagnose. At worse, the system could perform above threshold across the whole train-test sets but be consistently wrong on the highest risk examples. (See Oakden Rayner, 2019, for an example of this issue from the medical diagnosis domain.)

This problem is exacerbated by:

- Strata that may be hidden, and not obvious from the data. For example, transaction alerts reported to a transaction investigation system are typically categorised by alert type (unusually large transaction, new recipient for this account, high-risk geo, etc.), but there may also be

important subsets of cases that aren't captured by this taxonomy (such as new customer, or high value account).
- The fact that the most important, highest risk, strata may be rare, and so will have fewer training examples.
- The most important strata potentially being the hardest to diagnose.

Mitigations

1. **Identify major strata:** include'hidden' ones that are not identified as distinct labelled classes – and highlight those that are high risk, small, or known to be hard to diagnose. Consider using unsupervised clustering to find hidden strata. When investigating transaction alerts, for example, measure performance on the explicit strata, such as type of alert (exceptionally large payment, change in pattern of transactions, transaction with a counterparty in a high-risk geo). But also use experts to identify hidden strata that may be important, such as types of customer or types of account. This is known as schema completion in the medical literature (Oakden Rayner, 2019).
2. **Track performance and confidence**: Monitor the performance on each strata, and show the confidence levels for each class, especially where the number of cases is small. If there are low-confidence or low-performance strata, then highlight this in any outputs or decisions for those strata.
3. **Gather data**: Proactively gather more training data for minority, high-risk, or low accuracy classes. Use simulations to generate training examples for minority classes or hidden strata if there is insufficient available.

5. Data Drift

Customer behaviour changes over time. This could be due to seasonal factors (e.g. major holidays) or longer-term shifts (e.g. less use of cash for smaller transactions). The characteristics of the input data will drift over time which means that the performance of the model cannot be guaranteed. A system that was very effective at spotting risk in cash transactions may not be so effective once most transactions are cashless, or during the Christmas party season. The institution needs to know about possible issues before they affect performance of the live system.

Mitigations

1. **Monitor the distribution of features and labels** in the input data and define alerts when they drift outside defined limits. (See Davis *et al* 2017 for an example of how this may be done in the medical domain.)
2. **Monitor the performance of the model on live cases**, based on independent human QA judgements and define alert thresholds. Monitor overall performance, and per strata (see 'Hidden Stratification')
3. **Provide training mechanisms**: for incrementally training the model on fresher cases, whilst including back-testing on recent cases before deploying the revised model. Ensure the audit /

reproducibility processes can cope with models that have been incrementally trained on multiple training data sets.

4. **Stress-test models**: Test the robustness of models to drift by using simulated data based on the observed distribution, and use the results to set drift alert thresholds.

6. Anomalous Inputs

New, unseen, behaviours or cases may appear, even while the overall behaviour of customers remains constant. If these are not spotted, and are treated as 'normal' cases, then the decision of the ML cannot be relied on in these cases. These anomalous behaviours may also be leading indicators of new high-risk behaviours that the institution should be aware of.

Mitigation

Monitor live cases for anomalies using unsupervised anomaly-detection methods. The most anomalous cases should be caught and highlighted, so that decisions can be manually reviewed, and lessons learnt.

7. Confounders

Confounders are factors in the data that create spurious associations that can mislead machine learners into producing unreliable predictions. For example, cheques are associated with property transactions, but this is because property transactions are large. Large transactions are typically transacted by cheque, and property trades are the most common very large transactions in retail banking. If you control for size of transaction, then property trades are no more likely to be transacted by cheque than other types of deal. If you do not control for this factor, then the ML model will over-assume that all cheques are property transactions (which are typically low risk) and discount higher-risk possibilities.

Mitigation

Identify, and provide controls for, likely confounders through gaining an expert understanding of the likely causal factors (see Pearl, 2018, for an accessible introduction to this issue).

8. Discriminatory Bias

Spurious correlations used by naive machine learning models can introduce unexpected biases that treat certain groups unfairly. For example, a health algorithm that uses health costs as a proxy for health needs was found to lead to racial bias against African American patients (Obermyer 2019). Within the AML domain, students are often targeted by money launderers recruiting mules, and a greater than average proportion of the Chinese population in the UK are students. So, a naive machine learning algorithm may falsely predict that Chinese nationals in the UK are a money laundering risk.

In other cases, these correlations may reflect a genuine causal relationship. For example, the UK Financial Conduct Authority (FCA) and European Banking Authority (EBA) both recommend that country is considered as a factor when assessing the risk of a financial transaction. A large amount of money

being transferred with a Russian entity is inherently riskier than one with a German entity, for example, because of the different regulatory framework and political context of the two entities. Whether this association is causal and reliable, or spurious and discriminatory, can be decided statistically, but only if other possible confounding factors are identified. It may be that the association between Russian entities and risk may be due to a higher proportion of, say, politically exposed persons in the Russian sample.

Mitigations

Biases are typically due to spurious associations between subject groups and risky behaviours that are actually due to some other confounding factor. As above, the solutions are:

1. **Use expert understanding:** Detect and control for confounding factors by incorporating expert understanding of causal factors, and
2. **Explain judgements:** Ensure that all judgements are explained, so that unfounded biases are surfaced and can be addressed (see maturity model below)
3. **Manage consensus:** biases may be due to human biases of the SMEs expressed within the training data. Managing SME consensus (as above) will help address this risk.

9. Input Hacking

Bad actors can shape their behaviour to avoid known alerts. Examples in AML include segmenting large transactions to get below the $100k threshold used in many transaction alerting systems; randomising 'sliced' payments to obscure patterns of repeated transaction; or adding noise to machine-generated false accounts to obscure a common source. (See Goodfellow et al, 2017 for more examples).

Mitigations

1. **Use expert knowledge**:to identify possible hacking strategies and test the performance of ML using white hat simulations.
2. **Use 'distillation':** to automatically identify small perturbations of inputs that significantly change risk decisions (Papernot et al 2016)

Learned vs Engineered Features

There has been much debate in the machine learning community between the use of learned versus engineered features -- also presented as a debate between deep and shallow learning. 'Deep' or 'pure' learners, such as those using neural networks with many intermediate layers, are presented with raw unmediated data. When learning how to play a video game, for example, they will be given the value of every pixel on the screen from which they may learn salient features, such as the position of targets and obstacles (Mnih et al 2015). Shallow or hybrid learners, on the other hand, are given inputs that are carefully engineered to represent what the domain expert believes to be causally relevant features. In the video games example, they will be given the position of those obstacles and targets explicitly.

Deep learners can generate solutions with super-human levels of performance, exploiting opportunities and information too fleeting for humans to detect. Pure deep learning models may have higher 'headline' or aggregate performance than hybrid models exploiting engineered features, but there are trade-offs. From a risk management perspective, models that exploit engineered features have the following advantages:

Explainability

In a machine learner that uses engineered features it is possible to quantify the contribution of each human-understandable input with a final decision (see Ribeiro 2016, and Lundberg 2017). In contrast, it is difficult, and in some cases impossible, to retrospectively explain the decisions of pure learning models in human-accessible terms. We can visualise the input field of intermediate neurons (Bau 2017) and identify the 'building blocks' of interpretation (Olah 2018) but they may not necessarily be accessible enough to a human to allow domain experts, or regulatory authorities, to judge if the decision were justified. (Olah et al 2018 provide interactive graphics at https://distill.pub/2018/building-blocks/ that show both the power and limitations to this approach to understanding decisions made by deep learners: we can understand the 'logic' of the network in deciding cases where the correct decisions are obvious to us, but it would not be possible to use this method to adjudicate on cases where the correct decision is unknown or disputed.)

Efficiency

Exploiting existing expert understanding of the domain to engineer input features reduces the volume of training data required and the cost of labelling. This can increase return on investment and accelerate time-to-value.

Reliability

Providing the machine learner with human-salient inputs that are known (or at least suspected) to be relevant to risk makes it possible to eliminate the effect of confounders. This, in turn, helps to address the impact of hidden stratification, and detect and address discriminatory bias.

Hybrid Solutions

Our approach is pragmatic, favouring solutions that combine learned and engineered features in order to best manage risk. We believe that it is possible that pure deep learning models can be made safe and explainable, but they require substantial extra engineering safeguards and investment compared to those using engineered features. In some cases, simple business rules are the most effective and appropriate solution, where the cost to reach a similar level of performance using machine learning would be prohibitive.

Trust Maturity Model

How good is good enough? What degree of trust is required before a system is used on live cases? We identify four levels of trust – from experimental to autonomous – that determine whether a system can be used in live applications and what level of human oversight is required. The levels are defined in terms of the maturity of the system along the major axes of Accountability and Correctness and specific examples of technical requirements that would meet those levels are given. The lists of requirements are not intended to be exhaustive, but to illustrate how we can use an RBA to make principled decisions about the requirements of a ML-based system necessary to use it in production and decide the appropriate level of human oversight.

We find that there are two minimum requirements for a ML system to make decisions on live cases affecting humans:

1. Those decisions can be explained and justified in a way accessible to humans.
2. Significant risks that may cause the wrong decision to be made have been identified.

Without these, the institution cannot be held accountable for those decisions. This requirement applies even if those decisions are subject to human review, for without the ability to explain the decisions or knowledge of the risks that may cause the decision to be incorrect, then the human cannot independently and effectively verify its correctness.

Trust Level	Maturity: Accountability	Maturity: Correctness	Application and Oversight	Example Requirements
Experimental	The ML is a black box; its decisions are not explained.	Aggregate performance is comparable with humans, but there is a *significant* and *unknown* risk that the ML will return incorrect decisions in specific cases.	Not suitable for use on live cases since the risk of error or bias on new cases is unknown, and there is no effective way to review its decisions.	• An ML model trained on an expert-labelled train-test set, with aggregate performance (RoC, F1 or similar) comparable to human experts. • No risk management process.
Accountable	The decisions of the ML can be explained and audited.	Aggregate performance is comparable with humans, but there is a *significant* and *unknown* risk that the ML will return incorrect decisions in specific cases.	May be used for making or contributing to decisions affecting humans, but **every** decision affecting a human should be reviewed by a domain expert who is aware of the main risks.	• Major risks are identified and prioritised and communicated to human reviewers. • All decisions are explained in a form, and level of detail, such that a non-technical SME (i.e. one not familiar with the ML technology) would agree that the decision was justified and fair, based purely on that explanation. • All decisions are traceable to a model version; all model versions are linked to a training set (including algorithm parameters); and all labels in that set are traceable to a known human expert.

Trusted (active oversight)	The decisions of the ML can be explained and audited.	The risk that the model/algorithm will return incorrect and/or biased decisions in all specific cases is known.	May be used for making or contributing to decisions affecting humans, but all decisions in high risk or high impact strata are reviewed by a human expert.	• Major risks are mitigated where possible. • Hidden strata are identified from expert human understanding of the domain, and high impact strata identified. • Generalisation performance on each stratum is measured, and high-risk strata identified. • Performance of SMEs is monitored and consensus between them is managed.
Autonomous (passive oversight)	The decisions of the ML can be explained and audited.	The risk of incorrect or biased decisions is automatically monitored and alerted if it exceeds thresholds.	Humans are no longer in the decision-making loop, unless requested by the system. Humans review decisions primarily to identify new risks, based on risk management process.	• The list of known risks is periodically reviewed and mitigated as required. • Known high risk cases are alerted for human review. • Anomalous cases are alerted for human review. • Data drift is monitored with alert thresholds based on human review.

Discussion

Criminal activity is directly harming billions of people, our political systems, and our environment on a global scale; and money laundering is its oxygen. Crime is less attractive if the criminal cannot enjoy the proceeds of it. Successfully combatting money laundering in the context of ever more complex financial systems will require the widespread adoption of machine learning technology, and building trust is vital in ensuring that their benefits are realised. Trust is not an optional extra.

The international regulatory framework mandates a risk-based approach to AML/CTF processes, and so they must apply to ML-based solutions too. In this paper we outline what such an approach looks like, identify specific risks and mitigations, and how it can be used to prioritise specific requirements when building and deploying ML-based systems.

We find that the **minimum** requirement for an ML to make decisions on live cases is that those decisions can be explained and justified, **and** that significant risks that may cause the wrong decision to be made have been identified. Without these, there can be no effective accountability and oversight of the decisions that it makes – no matter the apparent performance level.

References

1. Bank for International Settlements (2016), *Guidelines for the Sound Management of Risks related to Money Laundering and Financing of Terrorism* (www.bis.org/bcbs/publ/d353.pdf).
2. Bau, D., *et al* (2017). Network dissection: Quantifying interpretability of deep visual representations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
3. Brodley, C.E. and Friedl, M.A. (1999). Identifying Mislabeled Training Data, *Journal of Artificial Intelligence Research*, 11, 131-167e.
4. Curtis G. Northcutt, Lu Jiang, Isaac L. Chuang (2020). Confident Learning: Estimating Uncertainty in Dataset Labels, *AI Statistics*, 2020.
5. Davis SE *et al* (2017), Calibration drift in regression and machine learning models for acute kidney injury. *Journal American Medical Information Association*. 24:1052–61.
6. Financial Action Task Force on Money Laundering (2014*). Risk-Based Approach Guidance for the Banking Sector* (www.fatf-gafi.org/publications/fatfrecommendations/documents/risk-based-approach-banking-sector.html).
7. Financial Conduct Authority (2019*). Financial Crime Guide: A firm's guide to countering financial crime risks (FCG).* (www.handbook.fca.org.uk/handbook/FCG.pdf).
8. Financial Crime News (2019). *Global Threat Assessment*, (thefinancialcrimenews.com/).
9. Goodfellow, I. *et al*, 2017. *Attacking Machine Learning with Adversarial Examples*, (openai.com/blog/adversarial-example-research/).
10. IEEE-TPS 2019: *The First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems, and Applications*, 2019.
11. Institute of International Finance, (2018*). Machine Learning in Anti-Money Laundering* (www.iif.com/Publications/ID/1421/Machine-Learning-in-Anti-Money-Laundering).

12. Lundberg, S. M., & Lee, S.-I. (2017). *Consistent feature attribution for tree ensembles* (arxiv.org/abs/1706. 06060).
13. Marcus, Gary, (2019). *Rebooting AI, Building Artificial Intelligence we can Trust,* Pantheon.
14. Mnih, V., *et al* (2015). *Human-level control through deep reinforcement learning*. Nature, 518(7540), 529.
15. Molnar, C., *et al*. (2018). *Interpretable machine learning: A guide for making black box models explainable*. Christoph Molnar, Leanpub.
16. Oakden-Rayner, L. *et al* (2019). *Hidden Stratification Causes Clinically Meaningful Failures in Machine Learning for Medical Imaging*, (arxiv.org/abs/1909.12475).
17. Obermeyer, Z. *et al* (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science,* 25 Oct 2019 : 447-453.
18. Olah, *et al.*, "The Building Blocks of Interpretability", Distill, 2018.
19. Papernot, N, *et al* (2016). Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks, *37th IEEE Symposium on Security & Privacy*, IEEE 2016.
20. Pearl, J., Mackenzie D., (2018). *The Book of Why: The New Science of Cause and Effect,* Random Books.
21. Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*. 1: 81–106.
22. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144).
23. Samek, W. *et al* (2019). *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer.
24. Shapley, L. S. (1953). A value for n-person games*. Contributions to the Theory of Games*, 2 (28), 307–317.