International Journal of Population Data Science

Journal Website: www.ijpds.org

IJPDS International Journal of Population Data Science

Swansea University Prifysgol Abertawe

Machine learning models in trusted research environments – understanding operational risks

Felix Ritchie[1,*], Amy Tilbrook[2], Christian Cole[3], Emily Jefferson[3], Susan Krueger[4], Esma Mansouri-Benssassi[5], Simon Rogers[6], and Jim Smith[7]

Submission History

Submitted:	30/04/2023
Accepted:	30/10/2023
Published:	14/12/2023

[1]Bristol Business School, University of the West of England, Coldharbour Lane, Bristol BS16 1QY
[2]University of Edinburgh, South Bridge, Edinburgh EH8 9YL
[3]Division of Population Health and Genomics, Ninewells Hospital and Medical School, Dundee DD1 9SY
[4]Health Informatics Centre, Ninewells Hospital and Medical School, Dundee DD1 9SY
[5]AffectiveHalo Ltd, Tom Morris Drive, St Andrews.KY16 8HS
[6]NHS National Services Scotland, Gyle Square, 1 South Gyle Crescent, Edinburgh EH12 9EB
[7]School of Computer Science and Creative Technologies, University of the West of England, Coldharbour Lane, Bristol BS16 1QY

Abstract

Introduction

Trusted research environments (TREs) provide secure access to very sensitive data for research. All TREs operate manual checks on outputs to ensure there is no residual disclosure risk. Machine learning (ML) models require very large amount of data; if this data is personal, the TRE is a well-established data management solution. However, ML models present novel disclosure risks, in both type and scale.

Objectives

As part of a series on ML disclosure risk in TREs, this article is intended to introduce TRE managers to the conceptual problems and work being done to address them.

Methods

We demonstrate how ML models present a qualitatively different type of disclosure risk, compared to traditional statistical outputs. These arise from both the nature and the scale of ML modelling.

Results

We show that there are a large number of unresolved issues, although there is progress in many areas. We show where areas of uncertainty remain, as well as remedial responses available to TREs.

Conclusions

At this stage, disclosure checking of ML models is very much a specialist activity. However, TRE managers need a basic awareness of the potential risk in ML models to enable them to make sensible decisions on using TREs for ML model development.

Keywords

confidentiality; output checking; machine learning; artificial intelligence; data enclave; trusted research environment

*Corresponding Author:
Email Address: felix.ritchie@uwe.ac.uk (Felix Ritchie)

https://doi.org/10.23889/ijpds.v8i1.2165

Introduction

This paper is part of a series investigating the risks of Machine Learning (ML) model development and release from Trusted Research Environments (TREs). This paper focuses on operational risks for TRE managers, and potential methods to combat them.

ML models are growing in popularity, particularly in health where they can play an important role supporting clinical and operational practice. These models are trained to, for example, recognise early-stage carcinomas or predict demand for a service to improve resource scheduling.

ML is "a subset of Artificial Intelligence, that automatically learns patterns from datasets. It can be used to help humans better understand complex data, or make predictions based upon new, unseen data" [1]. Unlike traditional statistical models, where the estimation method is specified by the researcher, ML models are provided with an approach to learning and goals and left to work out the method. The models repeatedly interrogate the data, often in multiple stages and possibly with multiple learning approaches. The resulting model (the reason the ML process ends up with a model configured in a particular way) may not be understandable even to the model designer.

ML models are 'trained' by providing them with a large number of examples, and by allowing the model to identify for itself the relationships that matter. This training data (also known as source, record-level or micro data in traditional statistical models) comprises detailed and often sensitive training datasets – such as electronic health records or confidential business data – and can cover millions of individual records. Where models are trained using large amounts of such confidential data, with methodology that is not easily explained even by the creator, concerns over data management arise.

Trusted research environments (TREs, also called safe havens or secure data environments) provide a ready-made solution to confidential data management. The secure facilities, designed to allow researchers to work on highly sensitive and confidential de-identified data, are often part of a high-performance computing environment. These environments offer obvious advantages for researchers developing machine learning (ML) models. They are designed to allow largely unrestricted freedom to work with and manipulate the data, with limitations only coming into play when the models are released into the public domain. A researcher can explore different specifications and learning models, in collaboration with other researchers, confident in the knowledge that the TRE 'sandpit' manages the risk of data leakage.

TRE processes typically follow common frameworks such as the 5 Safes (safe projects, people, settings, data and outputs) [2] and include requirements for; completing application forms; requesting specific data relevant for the research question; completing relevant training; using specific facilities and having the results of analysis ("outputs") checked by TRE staff for confidentiality risks before they are allowed out into the public domain for dissemination. While details vary, application forms, security, training and checking of outputs are all fairly standard and well understood [3]. A growing literature based on 20^{th} Century TRE development [4] means both researchers and TRE staff are aware of what constitutes a safe project, a safe person, a safe setting, a safe data request and a safe output. This gives TREs, researchers, data controllers and the public confidence that data is being used safely [5].

Generally, TREs are able to control the projects and data, and environments are necessarily being redesigned to allow ML and other development software to be used, but a gap remains in considering output risks, and how they relate to people: When researchers come to release their ML model from the secure environment where they have been developed, what risks might they (inadvertently) raise to the TRE or data controller? (Note: ML models do pose some additional 'project' risks, in terms of their intended use; this is outside the scope of this paper, but was explored extensively in the study).

TREs were developed to support traditional statistical analyses (such as estimation, graphical analysis, or cross-tabulations) on confidential data [4]. These analyses have a small but non-negligible risk of disclosing confidential information about the data – for example, by a table disclosing that a single female accountant from Cornwall has died of a new Covid variant. The uniqueness of this set of circumstances makes it possible for others to identify the individual, and potentially learn something new about them without having access to the original data. TREs have well-developed processes for managing this residual risk of disclosure [6], and good practice is largely well-established and uncontroversial (such as [7, 8]).

However, ML models present a number of challenges to the traditional output-checking processes of TREs. The scale of risk changes: with potentially millions of parameters, it is no longer possible to check manually for disclosures. The predictive potential changes: ML models are usually designed to give useful (and so accurate) predictions, compared to traditional statistical models which value a broad understanding of relationships. New risks appear: ML models provide different incentives for 'attackers' which do not appear in traditional TRE analysis. Finally, ML models are challenging the conception of what disclosure means.

While there have been a number of papers identifying the risks ([9] is a short but comprehensive summary), at present there are few guidelines on how to manage these risks. This makes TRE managers understandably cautious about supporting ML modelling [10]. We focus on structured data sources (such as digital consumer data or medical records), rather than the large-language models (such as ChatGPT) currently the focus of media attention, as the latter are less appropriate for TRE development.

This paper is part of a series, initially funded by UK Research and Infrastructure (GRAIMatter [11]) to investigate the risks of releasing ML models developed within a TRE infrastructure. The first output from this project [1] describes the technical aspects of the problem. The aim of this paper is to provide an initial scoping analysis of the operational aspects of this problem that TRE managers face. At this stage, we do not have solutions, but simply aim to identify the core issues.

Methods

This paper synthesizes initial evidence and risks from the evolving literature on ML models into a format for TRE managers. Initially funded by UKRI, the GRAIMatter team investigated several areas including:

- A literature search on ML models, breaches, TREs and risks
- Generating ML models and subjecting them to attacks to assess disclosure risk is possible once a model is released, including models fitted to synthetic data.
- Identifying the aspects of model design to avoid disclosure risks.
- Evaluating tools to semi-automate the process of risk assessing ML model output.
- Investigation into the legal and ethical issues applicable to ML model release from TREs, the obligations on TREs and researchers and data controllers.
- A series of public engagement workshops to evaluate public perceptions of ML and risks.

Building on this work, this paper looks at the risks from the point of view of TRE managers on how to think practically about the theoretical risks outlined in the existing literature.

Results

The starting position: checking traditional outputs, and 'safe statistics'

As described in previous literature [1] ML models do not fit existing models of output disclosure protection – or Statistical Disclosure Control (SDC). Traditional output SDC has been explored for a variety of outputs ([7, 8]). A key principle is the concept of 'safe/unsafe statistics' [12, 13]. 'Safe statistics' present negligible inherent disclosure risk, where it is extremely unlikely an individual can be identified just from looking at statistical models or results. These can be released with minimal administrative checks; examples include linear and logistic regression models, statistical test values, or concentration indexes. 'Unsafe statistics' present a non-negligible disclosure risk, and so each requested release must be checked for risk; examples include frequency tables, percentiles, or Kaplan-Meier graphs.

The classification of estimated coefficients in regression models is the most relevant to ML model disclosure. These are 'safe statistics' [13]: in practical situations, the coefficients do not reveal individual records, and there is no differencing risk (comparing two outputs differing by one observation to ascertain the value of the omitted observation) as adding individuals to a population result in the recalculation of every parameter. The qualifier 'in practical situations' is important [13] as it shows that it is possible to mistakenly reconstruct a table as a regression, in which case the regression coefficients represent the mean values in a multi-dimensional table. This is only likely to occur if the regression only has one or two binary variables, which is easily spotted by an output checker. It is possible to deliberately falsify results [13], but there is no incentive for most users to do so, as they can directly see the source data. Users of services which permit them to send in code and receive results without seeing the data ('remote job servers') do have a theoretical incentive to hide individual observations in regressions; again, this is not likely to be a practical risk as the necessary transformations are easy to detect and have no genuine statistical use, and so any check on the code would clearly show unlawful activity.

The safe/unsafe distinction is important as safe statistics require by their nature, far less concern and checks within the TRE and prima facie, ML models would seem to fit into the 'safe statistics' class: they estimate summary parameters through complex processes, those parameters are not necessarily associated with any single record but are representative of all interactions, and the addition of a new record before re-estimation should lead to all the parameters being re-estimated (and hence preventing disclosure by differencing). However, the features of ML modelling means that the analogy has important differences.

Inherent risks in the models

Perceptions of risk in the purpose of models

ML models differ from statistical models in a crucial way: their primary purpose is to provide accurate, usable predictions. The particular is important, and some types of ML models may generate a very large number of parameters to ensure predictive accuracy. In contrast, statistical models are primarily designed to highlight the relationships between entities; the general is important. For statistical models, the important question is "are tall people better at basketball than short people"; for ML models, the question is "how good at basketball is this person likely to be?" - the individual predictions matter, not the model. This focus feels inherently riskier to a TRE manager.

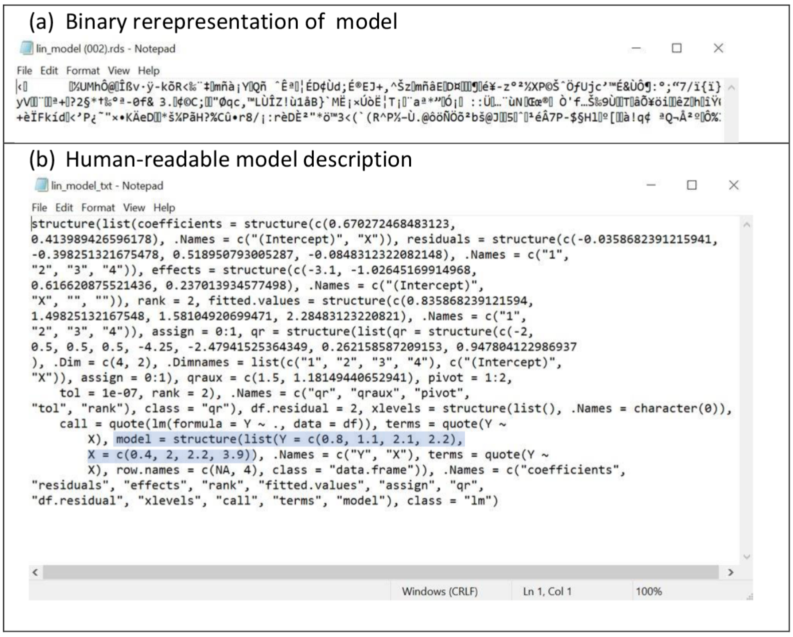
Sometimes, statistical models are used for prediction; and ML models can be used to identify relationships of interest between factors. Both approaches caution against 'overfitting' (estimating so many parameters that the model predicts the training data very well, but has little value for interpretation or application to new data), and both employ techniques to avoid this. But in general, statisticians are looking for parsimonious models with human-interpretable coefficients, and not to target individuals. This perceived difference in the function of models can cause concern.

As well as the perceived riskiness, there are three specific technical problems.

Quantity and format

A very large statistical model might estimate a hundred or more coefficients; but an ML model may generate millions of parameters. This creates the first practical output checking problem: the quantity and format. A full model specification is often not human-readable, unlike statistical outputs. Moreover, the large number of parameters means that the model is likely to be presented as a binary file. Figure 1 shows the description of the same model (a simple linear

Figure 1: Binary description versus the same description in human-readable form



(a) Binary rerepresentation of model

lin_model (002).rds - Notepad
File Edit Format View Help

```
‹      %UMhÔ@Îßv·ÿ-kõR<‰¨‡mñà¡YQñ ˆÊª¦ÉD¢Ùd;É®EJ+,^ŠzmñâEDD¤¶é¥-z°²½XP©Š˜ÖƒUjc’™É&ÙÔ¶:°;“7/ï{ï}
yV¨ª+?2§*†‰°ª-0f& 3.¢©C;"Øqc,™LÙÎZ!ù1åB}`MË¡×ÚòË¦T¡¨aª*”Ó¡ ::Ü…¨ùNŒœ® Ò'f…Š‰9ÙTâÕ¥öiêZhîŸ
+èÏFkíd<’P¿˜"×•KÄeD*š¼PãH?%Cû•r8/¡:rèDȲ"*ö™3<(`(R^P½-Ù.@ôöÑÖõ²bš@J5ˆ¹éÂ7P-$§H1º[à!q¢ ªQ¬Å²ºÔ%:
```

(b) Human-readable model description

lin_model_txt - Notepad
File Edit Format View Help

```
structure(list(coefficients = structure(c(0.670272468483123,
0.413989426596178), .Names = c("(Intercept)", "X")), residuals = structure(c(-0.0358682391215941,
-0.398251321675478, 0.518950793005287, -0.0848312322082148), .Names = c("1",
"2", "3", "4")), effects = structure(c(-3.1, -1.02645169914968,
0.616620875521436, 0.237013934577498), .Names = c("(Intercept)",
"X", "", "")), rank = 2, fitted.values = structure(c(0.835868239121594,
1.49825132167548, 1.58104920699471, 2.28483123220821), .Names = c("1",
"2", "3", "4")), assign = 0:1, qr = structure(list(qr = structure(c(-2,
0.5, 0.5, 0.5, -4.25, -2.47941525364349, 0.262158587209153, 0.947804122986937
), .Dim = c(4, 2), .Dimnames = list(c("1", "2", "3", "4"), c("(Intercept)",
"X")), assign = 0:1), qraux = c(1.5, 1.18149440652941), pivot = 1:2,
    tol = 1e-07, rank = 2), .Names = c("qr", "qraux", "pivot",
"tol", "rank"), class = "qr"), df.residual = 2, xlevels = structure(list(), .Names = character(0)),
    call = quote(lm(formula = Y ~ ., data = df)), terms = quote(Y ~
        X), model = structure(list(Y = c(0.8, 1.1, 2.1, 2.2),
        X = c(0.4, 2, 2.2, 3.9)), .Names = c("Y", "X"), terms = quote(Y ~
        X), row.names = c(NA, 4), class = "data.frame")), .Names = c("coefficients",
"residuals", "effects", "rank", "fitted.values", "assign", "qr",
"df.residual", "xlevels", "call", "terms", "model"), class = "lm")
```

regression, in the format that an ML model would characterise it) presented in binary and human-readable form.

Even in the latter case, the code is not easily readable when compared to say, the regression and some descriptive statistics that a traditional analysis would have generated. This implies that an ML model output requires computerised assessment.

Data included in the model

The second, related, issue is that ML models may well include detailed information about the training data as part of the model description - for example, in Support Vector Machines (SVMs: [14]). As [15] notes, "… nearest neighbour classifiers and SVMs explicitly store some training data points in [the output vector]". The shaded text in Figure 1 illustrates this: the description of a simple ML model contains the original training data in the form of a two-variable, four observation model.

Overfitting

The third issue concerns the likelihood of being able to identify individual cases when so many parameters are released. The large number of parameters in the model increases the likelihood that some lead directly to identification. This is particularly the case for the membership attack (trying to see if a known individual is in the data: see below) where 'edge values' (exceptional characteristics which the model tries to accommodate) can be used to target individuals. This is less of an issue for standard statistical analysis where fewer free parameters mean that there is a greater degree of uncertainty, and a large number of simple statistical tests are available to identify influential points. With the much greater number of parameters in ML models, it seems reasonable that there is a higher probability of individuals being compromised.

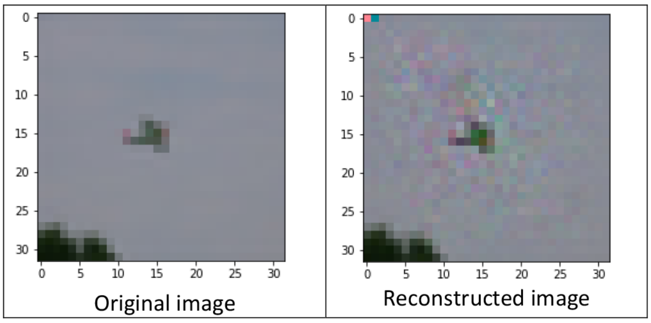
If ML models were seen as standard estimation models but with more parameters, the response to this third problem might be 'even so, so what'? ML predictions might be more detailed, but there remains uncertainty. Even in 'edge cases', it is very unlikely that an individual record could be matched exactly: the model will always have some regularity in it (and hence error in individual cases). If not, it is useless for modelling on new data (assuming it is not deliberately parameterised to identify an individual; see below). So, if the model is known not to be an exact representation of any individual, why isn't treating it as a standard regression appropriate? This requires a re-evaluation of what is meant by 'disclosure'.

The meaning of disclosure in ML model attacks

Traditional SDC describes how someone might be identified, what could be learned about them, and what risks that might bring [16]. In numerical data structured as rows and columns representing records and variables, disclosure results from, for example, a single person being identified in a frequency table as under 16 and having diabetes. In addition, magnitude tables (such as means or totals) have rules applied to them; these are intended to ensure that, even if the respondent is identified, the value supplied by the respondent can only be estimated to a degree of uncertainty. The literature focuses on the reconstruction of a single data point; for example, [13] shows how to calculate estimated maximum/minimum prediction errors for within-sample and out-of-sample dependent variables, respectively, in a linear regression. There is no discussion of whether approximately reconstructing several data points for the same subject counts as 'disclosure'.

There is no clear understanding of the disclosure potential of the sorts of data used for ML modelling. Traditional SDC literature (such as [16]) typically only considers probability of disclosure in the training data, not outputs; the literature on output disclosure risk treats re-identification as a binary

Figure 2: Reconstructed image (taken from [18], with permission)



possibility (although the literature on training such as [17] emphasises probability). In practice, TRE staff and attackers can make educated guesses on the likelihood that this combination of data identifies a single person (knowledge of how many people in the population have diabetes, are male and are under 16), and staff can risk assess outputs accordingly. However, as the GRAIMATTER project showed [11], the range of potential metrics makes such an assessment extremely difficult.

ML models of course can be used on these types of structured quantitative data, but are also likely to be used on new types of data such as genomics or images. Currently the disclosure risk of a model which identifies an image is not understood [9]. Figure 2 demonstrates the reproduction of an image from a model [18], trained on a public dataset.

Clearly the reconstructed image is not the same as the original, but is it close enough to breach expectations of confidentiality, if the data were confidential? Theoretical re-identifications degrade quickly in practice, but while small perturbations in images might fool an ML model, they can still be defeated by human inspection [9]. 'Adversarial attacks' make (to humans) imperceptible modifications to images which nevertheless can lead the ML models to come to different conclusions [19].

As humans we are very effective at recognising faces or voices, for example, even from imperfect copies. Similarly, a scan of an unusual tumour may be recognisable to those involved in the clinical treatment even if the tumour is not reproduced exactly. Data holders are well aware that the *perception* of disclosure can be as damaging as an actual disclosure. The risk of *approximate* disclosure in ML models is therefore already recognised in practice at TREs, but guidelines on where the line exists between disclosive and non-disclosive models are not yet in place.

Finally, there is also a distinction between the statistical and computing communities in terms of what counts as acceptable risks. The latter tends to think of actual risk, such as the relative likelihood of true and false positives. This arises from a perspective that views attackers as data experts. These were the metrics that the GRAIMATTER project focused on.

In contrast, the statistical and TRE community tends to focus on perceptions, whether true or false that no one should be able to *think* they have identified an individual, irrespective of whether they have or not. This reflects a perspective considering perceptions of the non-specialist and is a common view of the data providers who must not lose public trust in data being kept confidential.

This is not just a theoretical argument. Consider, the case of 'class disclosure', which arises when something can be said about a group of individuals: "all the students showed traces of THC in their systems"; "none of the patients in the higher income brackets contracted the disease". In traditional SDC, the disclosiveness of such statements is known to be very sensitive to context [17], but at least the disclosive classes can be easily identified. In contrast, an ML decision tree may have a large number of single leaves, representing a class disclosure; but when there are many disclosures, generated by a large number of small variations, does any one matter? This is an unresolved issue.

Ultimately the issue is that disclosure, even in traditional models of SDC, is poorly defined – traditional SDC guidance is very much of the 'we know it when we see it' kind. However, for ML models, we can no longer 'see' the risk in the same way, and are forced to make judgements based upon mathematical assessments which do not have clear (or possibly even interpretable) standards.

Novel people risks

Two attack methods [1] widely used in the literature to reidentify individuals from the outputs of ML models are:

- Model inversion: using the parameters of the model to try to re-create training data. Essentially, the attacker creates a noisy potential dataset and then repeatedly re-estimates it to find out which training data could have generated the model parameters, like bootstrapping in reverse.

- Membership attack: using the parameters to test whether a known individual is in the training dataset. This uses the fact that an individual in the training dataset would influence the model outcomes

In theory, these both also exist for traditional estimated models but, as noted above, the practical risk is negligible. However, assessing the risks is further complicated by the attack modes possible for ML models. [1, 9] describe two attack modes:

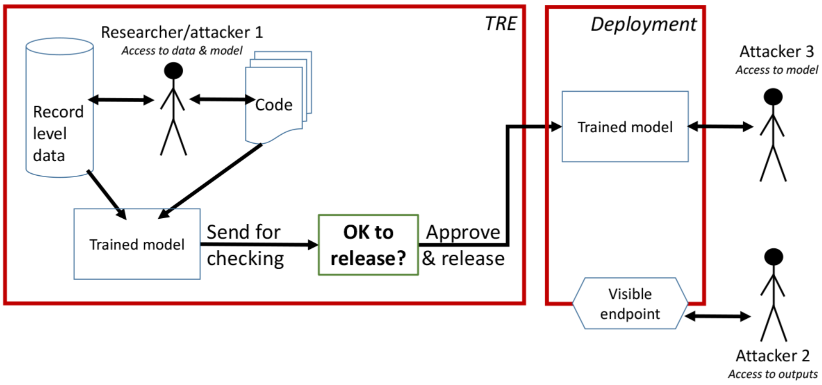
- 'Blind'– trying to re-identify individuals with access only to predictions (that is, send input to an interface and get back results; also called 'black box' attack)

- 'Sighted' – having full access to the model parameters and architecture (that is, being able to inspect the model's inner workings; also called 'white box' attack).

Both types of attack and the attack modes occur on the outputs of ML models once they have left the TRE environment, as shown by Attacker 2 and Attacker 3 in Figure 3.

Most TREs have training for users which includes how to create outputs that will get released safely and quickly. Despite training, and generally being trustworthy [1], there is a theoretical risk of users bypassing processes to release potentially risky results: attacker 1 in Figure 3. Risks from outputs come from the possibility of a user concealing raw

Figure 3: Summary of TRE output scenarios and attack possibilities



data within an output – either by replacing output results with raw data points, or by hiding them elsewhere in the output file.

On traditional statistical outputs, this is a feasible but high effort/high risk/low reward strategy. The incentives and possibilities for hiding specific results or whole datasets are virtually nil [13], and generally covered by simple administrative checks done by TRE staff. There is no evidence to support malicious misuse of this type in TREs, and so attacker 1 is not seen as relevant to SDC.

Traditional SDC focuses on 'attacker 2', the 'black box' attacker, where 'outputs' mean model parameters, details of the data and research method, and possibly even the source code. Given this assumption, there is little meaningful distinction between attacker 2 and attacker 3 for traditional models.

For ML models, this balance is altered, with attacker 1 becoming a source of concern. The data types in training datasets (images, genomic data, multimedia, unstructured text) have value in the whole record, not a part of it. It is infeasible that an entire image could be recalled perfectly, so if that image has value, hiding the image in the model may be both possible and worthwhile – as long as it is not discovered. Hence attacker 1 is a meaningful (non-negligible) threat, even if 'black box' attacks are likely to be more common in practice.

Without better knowledge of ML models and outputs, TRE staff can't be sure nefarious modes of hiding data within models haven't been done (in theory, it is possible for staff to spot this in the code, but is likely to be easy to hide). For a .doc or .xls file there are basic if not foolproof checks for simple hiding techniques. For ML models, the possibilities of hiding (particularly steganographic techniques) have been demonstrated by data scientists (for example [19]), but there are no clear ideas how a TRE manager should set about checking to see if this has happened.

Even assuming no malicious misuse, risks remain for accidental disclosure. As ML models are a relatively new thing to be used in TREs, users may not be aware what they are requesting to be removed. The traditional model involves a user requesting TRE staff to check and release an .xls, .doc, .pdf, or .html file. Raw code/statistical output files are also checked but may be discouraged. Users are generally encouraged to do as much work as possible within the secure environment of the TRE, and to provide as much explanation as possible with their outputs so staff can check it is safe.

As noted above, some algorithms (e.g. support vector machines) by default in practice include some of the training set items - i.e. parts of the raw data the model was trained on, so it can function on new data. Obviously, this shouldn't leave the secure environment of the TRE, but currently (a) researchers are not aware this is not releasable, and (b) output checkers are not always able to identify data separately from model parameters?

Overall, ML modelling both increases the likelihood of mistaken release, and the value of malicious data hiding.

Discussion

For TREs to be able to facilitate the uses of new types of data and ML models, risk assessment, feasible control, and operational guidelines are needed. There is the possibility to learn from traditional SDC in all these areas, but several factors need to be understood before concrete guidelines can be put in place. Below are presented some developments in progress.

Risk measurement

This is an active area of ML research not confined to TREs, but at present these risk measures focus on theoretical risk; that is, they describe the risk and how it may come about [20], but there is little evidence of how meaningful it is in operational research environments. Development is needed to identify (a) practical risk and (b) the necessary conditions for the risk to occur. The GRAIMATTER project [11] has begun real-life trials of its recommendations. However, as noted in the results above, we do not have agreement on what constitutes as an 'acceptable' risk, with large differences between those concerned with actual risk and those concerned with perceptions.

Feasible controls

Operational controls can be statistical or non-statistical. As an example of statistical rule, Ritchie [12] argued that at least one coefficient should be withheld from regression results to prevent all theoretical risk scenarios occurring; the choice of withheld coefficient was irrelevant. However, the same author later argued [13] that this was an unnecessary condition, as the practical likelihood of the problematic situations occurring was negligible. This emphasises the requirement above for having risk measurement for ML models on both theoretical and practical risks.

For ML models, possible statistical controls could be: limiting the amount of output; reducing the amount of model information provided; or ensuring that no data are included in model descriptions. It is possible that some approaches to increase explainability (such as LIME) could be used to increase confidence of ML models. These approaches turn the complex output into something already familiar and understood – such as a statistical type output. As regressions are one of the most basic types of ML model, could all models be reduced to something so explainable?

These may be quite blunt controls, but the TRE experience has been that blunt controls can be very effective: by reducing checking time for the bulk of outputs, they allow more time to be spent on reviewing time-consuming but more valuable exceptions. It may be possible to develop more finely-tuned statistical controls; for example, tools that could assess and respond to edge-case risks. It seems likely that any such statistical controls will need to be automated, given the complexity of ML models.

The major uncertainty with statistical controls is the range of models that need to be considered for ML techniques. There have been no fundamentally new statistical models developed during the 21st century, the lifetime of the current generation of TREs. In contrast, new ML architectures and methods are being developed and applied continuously. It may be that the same statistical controls can be applied to all ML models; it may be that ML models can be organised into classes where the same rules apply; or it may be that statistical controls have to be specific to the model architecture. At this stage, there is no literature to explain; nor any sense on how, for example, the life cycle of the model (train, use, re-train, re-use…) affects any statistical controls.

Operational controls could include: limited ML modelling to only 'highly trusted' researchers; requiring users to demonstrate that the ML model is developed from running a particular piece of code; or using audit models where data controllers or RDC staff can audit the model pipeline from data, features, model creation and submitting models and code for independent review before release.

Generally operational controls are more burdensome. They also require TRE owners to take a view on what is an acceptable risk; for example, the TRE policy might state 'we only let the most trusted researchers do ML modelling, as we think that accidental release is unlikely but we are concerned about malicious misuse'.

However, there is one advantage that ML models do have over traditional models; their rarity. A researcher carrying out statistical analysis may generate a very large number of models as they explore different specifications and evaluate them. In contrast, in ML, the evaluation is carried out in the modelling process itself, which is designed to produce the definitive model with minimal human intervention. Hence, an ML model being released outside the TRE is likely to be a relatively rare event. This makes it feasible for TREs to have very detailed review processes.

Guidelines

As noted above, there are a number of potential controls, which could be classed into different 'levels':

- high level: which models are allowed
- human level: ensuring researchers are aware of the risks and mitigation
- algorithmic level: differentially private training
- lowest level: identified bad parameter ranges for models

The difficulty is turning these into practical measures.

Ideally, risk assessment and controls need to be presented in the form of guidelines which have meaning for both TRE managers and researchers. There is great value in exploring the theoretical risks to explain the worst-case scenario – but TREs also need guidance on practical risks and controls. Most TREs do not have the resources to specialise in types of output, and so output checkers need to have general knowledge of likely outputs. Increasingly, self-checking of outputs by researchers (with review by TRE staff) is seen as the most cost-effective strategy [6]. However, GRAIMATTER [11] notes that the large number of disclosure risks mean that even the most well-disposed and well-trained researcher can miss important factors, and recommends double-checking by an expert in machine learning trained to spot risks. A follow-up project (sponsored by the same funders as GRAIMATTER) is specifically tasked with trying to make the GRAIMATTER recommendations more interpretable. The initial findings from that project are expected in November 2023.

What next?

The eXplainable AI (XAI) movement has risen in response to the inherent mystery of how ML models work, which should be welcomed by the TRE community. Researchers, TRE staff and data providers all need to understand enough to demonstrate, that the people and/or entities the data represents are not identifiable. Without understanding the model it is hard to be confident that when disclosed from the TRE it will not cause the sensitive data to be revealed.

There is a growing TRE community worldwide; data providers, users, funders, TRE operators and publishers all have stakes in making the best use of data. At present there is not a single party who has the knowledge to design how to best facilitate ML use within TREs. Knowledge of ML models and uses can only come from users, development of controls from technical and governance experts, requirements for publishing and understanding of feasibility from funders and publishers – the whole community will need to buy in and contribute to develop a shared understanding of the risks, controls and guidelines. New models, data types and technology will require further guidance as they are developed. Just as TREs have had to develop ways to include more and different software, output checks will also need to evolve.

The above is part of the current development of making better use of large datasets in a safe way – something that has been in the making for twenty years or more. The GRAIMATTER has brought together the communities who understand ML use and TREs and disclosure risk to further this understanding.

Conclusion

The growing use of new data types, and the ML models to analyse them create big uncertainties. Without understanding what disclosure could even theoretically happen, or even what it means in this context, TREs either run the risk of releasing raw or disclosive data from the safe environment, or not being fit for purpose and closing doors to this kind of analysis. Once ML models are out in the world, uses of the algorithm are uncontrolled. Overall, there is not enough knowledge within TREs on what ML models can do, who might use them once outside the environment, and how they could be used to reidentify individuals within the data. It is likely that this is preventing a productive partnership between TREs and ML modellers [10].

To allow ML models to fully exploit TRE infrastructure, urgent research is needed into two technical areas:

- Understanding the types of ML models likely to be used within TREs
- Identifying the risks of the most common types of ML models, and the people most likely to use them

These are ongoing by the GRAIMATTER research team. However, that project highlighted some more nebulous problems needing development, and which are perhaps more pressing to TREs:

- Identifying disclosure means, and how risks should be described and assessed.
- Designing controls and guidelines to reduce the *practical* risks posed by these new analyses.
- Helping non-specialist TRE managers and data owners make their own decisions about what parameters for approving models are reasonable.

The follow-up project SACRO https://dareuk.org.uk/driver-project-sacro/ is now turning its hand to these more fundamental questions; we will be reporting in 2024.

Acknowledgments

The GRAIMATTER project was funded by the UK Medical Research Council as a DARE Phase 1 Sprint Project https://dareuk.org.uk/sprint-exemplar-project-graimatter/. The DARE project report [11] was the main project output. This post-project article was produced by members of the GRAIMATTER team most closely connected to this workstream. In addition, Amy Tilbrook contributed significantly to the writing of this document but did not receive any funding through DARE or MRC for her contribution. Susan Krueger also contributed to the background analysis and writing of this document but did not receive any GRAIMATTER funding. She does however received funding for the MRC PICTURES project developing AI applications of image data, which continues to help the research team tests recommendations in a genuine environment.

Statement on conflicts of interest

No conflicts of interest are declared.

Ethics statement

GRAIMATTER did receive ethical approval for the public engagement activities. However, no ethical approval was required for the elements of the project described here. Public engagement activities are not discussed, and no new data collection took place.

References

1. Mansouri-Benssassi E., Rogers S., Smith J., Ritchie F., Jefferson E. (2021) Machine Learning Models Disclosure from Trusted Research Environments (TRE), Challenges and Opportunities. CoRR abs/2111.05628.

2. Ritchie, F. (2017) The "Five Safes": A framework for planning, designing and evaluating data access solutions. Paper presented at Data for Policy 2017, London, UK. https://doi.org/10.5281/zenodo.897821

3. Green, E., Ritchie, F., Tava, F., Ashford, W., & Ferrer Breda, P. (2021, July). The present and future of confidential microdata access: Post-workshop report. https://uwe-repository.worktribe.com/output/8175728.

4. Ritchie, F. (2021). Microdata access and privacy: What have we learned over twenty years? Journal of Privacy and Confidentiality, 11(1), 1-8. https://doi.org/10.29012/jpc.766

5. Harkness F., Rijneveld C., Liu Y., Kashef S., and Cowan M. (2022). A UK-wide public dialogue exploring what the public perceive as 'public good' use of data for research and statistics. Economic and Social Research Council. https://osr.statisticsauthority.gov.uk/wp-content/uploads/2022/10/Public_perceptions_of_public_good.pdf.

6. Alves, K., & Ritchie, F. (2020). Runners, repeaters, strangers and aliens: Operationalising efficient output disclosure control. Statistical Journal of the IAOS, 36(4), 1281-1293. https://doi.org/10.3233/SJI-200661

7. Brandt, M., Franconi, L., Guerke, C., Hundepool, A., Lucarelli, M., Mol, J., Ritchie, F., SeriG., & Welpton, R. (2010). Guidelines for the checking of output based on microdata research. https://uwe-repository.worktribe.com/output/983615.

8. SDAP (2018) Handbook on Statistical Disclosure Control for Outputs https://figshare.com/articles/book/SDC_Handbook/9958520.

9. De Cristofaro E. (2021) A critical overview of privacy in machine learning. IEEE Security and Privacy v19:4 https://doi.org/10.1109/MSEC.2021.3076443

10. Kavianpour S., Sutherland J., Mansouri-Benssassi E, Coull N., & Jefferson E. (2022). A Review of Trusted Research Environments to Support Next Generation Capabilities based on Interview Analysis. J. Medical internet Research https://www.jmir.org/2022/9/e33720.

11. Jefferson E., Liley J., Malone M., Reel S., Crespi-Boixader A., Kerasidou X., Tava F., McCarthy A., Preen R., Blanco-Justicia A., Mansouri-Benssassi E., Domingo-Ferrer J., Beggs J., Chuter A., Cole C., Ritchie F., Daly A., Rogers S. and Smith J. (2022) Recommendations for disclosure control of trained Machine Learning (ML) models from Trusted Research Environments (TREs). https://zenodo.org/record/6896214#.Yt5i2HbMKHs.

12. Ritchie F. (2007) Statistical disclosure control in a research environment, mimeo, Office for National Statistics; available as WISERD Data Resources Paper No. 6 https://uwe-repository.worktribe.com/OutputFile/957317.

13. Ritchie F. (2019) Analyzing the disclosure risk of regression coefficients. Transactions on data privacy, 12(2), 145–173. http://www.tdp.cat/issues16/tdp.a303a18.pdf

14. Christiani N., and Shawe-Taylor, J (2000) An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press.

15. Song C., Ristenpart T., and Shmatikov V. (2017) Machine Learning Models that Remember Too Much. CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, October. pp. 587–601 https://doi.org/10.1145/3133956.3134077

16. Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Lenz, R., Longhurst, J., Schulte Nord-holt, E., Seri, G. and De Wolf, P-P. (2010). Handbook on Statistical Disclosure Control. ESSNet SDC. https://cros-legacy.ec.europa.eu/system/files/SDC_Handbook.pdf

17. ONS (2021) Output checker training course handbook version 1.0. Office for National Statistics. Available on request.

18. Krueger, S., Mansouri-Benssassi, E., Ritchie, F., & Smith, J. (2021). Statistical disclosure controls for machine learning models. UNECE/Eurostat Workshop on Statistical Data Confidentiality 2021. https://uwe-repository.worktribe.com/output/8067227.

19. Kaviani, S., Han, K.J., & Sohn, I (2022). Adversarial attacks and defenses on AI in medical imaging informatics: A survey. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2022.116815

20. Jegorova M, Kaul C, Mayor C, O'Neil AQ, Weir A, Murray-Smith R, Tsaftaris SA. (2023) Survey: Leakage and Privacy at Inference Time. IEEE Trans Pattern Anal Mach Intell. Jul;45(7):9090–9108. https://doi.org/10.1109/TPAMI.2022.3229593