I TRIED A BUNCH OF THINGS: THE DANGERS OF UNEXPECTED OVERFITTING IN CLASSIFICATION OF BRAIN DATA

MAHAN HOSSEINI[1*], MICHAEL POWELL[2*], JOHN COLLINS[3], CHLOE CALLAHAN-FLINTOFT[4],WILLIAM JONES[1], HOWARD BOWMAN[1,5], AND BRAD WYBLE[6]

1. Computing Department, University of Kent

2. Manada Technology LLC

3. Physics Department, Penn State University

4. Army Research Lab, Aberdeen Proving Grounds

5. School of Psychology, University of Birmingham

6. Psychology Department, Penn State University

*Co-first authors

None of the authors have competing interests with regards to this work.

Figures to be printed in Black and White, but online in color.

See OSF page for code sample and data:
https://osf.io/qkvhd/?view_only=bb01fb14e61c405a936765f1524b36b9

ABSTRACT

Machine learning has enhanced the abilities of neuroscientists to interpret information collected through EEG, fMRI, and MEG data. With these powerful techniques comes the danger of *overfitting of hyperparameters* which can render results invalid. We refer to this problem as *'over-hyping'* and show that it is pernicious despite commonly used precautions. Over-hyping occurs when analysis decisions are made after observing analysis outcomes and can produce results that are partially or even completely spurious. It is commonly assumed that cross-validation is an effective protection against overfitting or overhyping, but this is not actually true. In this article,

we show that spurious result can be obtained on random data by modifying hyperparameters in seemingly innocuous ways, despite the use of cross-validation. We recommend a number of techniques for limiting over-hyping, such as lock boxes, blind analyses, pre-registrations, and nested cross-validation. These techniques, are common in other fields that use machine learning, including computer science and physics. Adopting similar safeguards is critical for ensuring the robustness of machine-learning techniques in the neurosciences.

Keywords: Overfitting; over-hyping; machine learning; classification; analysis, EEG

INTRODUCTION

Computers have revolutionized approaches to data analysis in psychology and neuroscience, effectively allowing one to interpret not only the neural correlates of cognitive processes, but also the information content that is represented in the brain through the use of machine learning. However, with these new and powerful tools come new dangers. Machine learning algorithms allow a pattern classifier to weave many subtle threads of information together to detect subtle patterns, e.g. to determine from MEG data whether someone is currently viewing a building or an animal (Cichy, Pantavis & Oliva 2014). However, these pattern classifiers are essentially black boxes to their human operators, as they create complex mappings between features and outputs that exceed one's ability to comprehend. This lack of interpretability can be especially pernicious when combined with the dangers of overfitting, which is a problem inherent to all fitting algorithms, see Table 1 and (Poldrack et al 2020). Specifically, interpretability enables the plausibility with which a classification or prediction is arrived at to be assessed against prior understanding and theory. Consequently, when using "black-box" machine learning (i.e. algorithms where the internal parameters are essentially uninterpretable by humans), one can unintentionally create a classifier that does very well on a specific data set, but poorly on other data sets (i.e. we say the classifer has been *overfit* to the training data; see Table 1), with no ready way to critique or judge the plausibility of the solution found by the algorithm.

The issue of overfitting is related to another topic that is frequently discussed in the scientific literature, which is *researcher degrees of freedom* (e.g. Simmons, Nelson & Simonsohn 2011). This term reflects the fact that choices made during analysis can erroneously inflate findings of statistical significance by eliminating options that produce non-significant or otherwise unwanted results. A parallel issue exists in machine learning, but with additional layers of complexity that can obscure the influence of choices made by the researcher on the analysis outcome. For example, techniques such as *cross-validation* (i.e. tools for reducing overfitting, see Table 1) are often thought to insulate the analysis from the statistical inflation provided by degrees of freedom in the analysis, but it will be shown here that this is not the case.

Issues associated with analysis overfitting are by no means new to science: High-energy physics has had a number of high-profile false discoveries, some of which were the result of overfitting an analysis to a particular data set. Related difficulties have been argued to have arisen during the search for gravitational-waves (Creswell et al, 2017; New Scientist, 2018). Indeed, because of several high-profile false discoveries, high-energy physics has already gone through a replicability crisis, and has had to rearrange its methods to deal with the consequences. A classic case, which was a big wake-up call for the field, was the so-called split-A2 from Chikovani et al. (1967). Had this effect been genuine, it would have engendered a theoretical revolution, but when more data became available, the effect disappeared; see Harrison (2002) for a recent view. It appeared that inappropriate selection of data was the culprit. For accounts of some of these in the light of current experimental practice, see Harrison (2002) and Dorigo (2015).

The similarities between data analysis in high-energy physics and modern neuroscience are striking: both fields have enormous quantities of data that need to be reduced to discover signals of interest. As such, it is useful and common to apply cuts to the data, i.e. to restrict analysis to certain regions of interest (ROI), as is common to the analysis of fMRI and EEG data. Because the purpose of the cuts is to enhance a signal of interest, there is a danger that the choice of a cut made on the basis of the data being analyzed (and on the basis of the desired result) may create apparent signal where none actually exists, much like in aforementioned case from physics. Furthermore, when making measurements in high-energy physics and neuroscience, complicated apparatuses are often used, and analyses typically contain an extremely sophisticated set of software algorithms. Optimization (i.e. making choices to increase effectiveness), and debugging of

complex analysis pipelines for both neuroscience and physics data sets require many decisions that are often necessary to, and yet present grave dangers to the generalizability of the results, such that the results will not replicate on a separate data set.

To prevent such cases, the high-energy physics community has adopted several conventions and methods in the analysis and interpretation of data. For example, *blind analysis* refers to a technique in which analysis optimization occurs without consulting the dependent variable of interest (e.g. Klein & Roodman 2005). Since the optimization algorithm is blind to the result of interest, researcher degrees of freedom will be unable to artificially inflate estimates of statistical significance. Unlike physics, while related issues have been discussed in the literature (Kriegeskorte et al 2009; Button 2019; Brooks et al 2017; Bowman et al. In Press), the neuroscience field has not yet fully responded to the dangers of over-hyping when complex analyses are used, which increases the potential of false findings and presents a major barrier to the replicability of the literature. At the end of this paper, we will discuss several preventative solutions, including blind analysis.

As mentioned above overfitting is the optimization of an analysis such that performance improves on the data being evaluated but remains constant or degrades on other similar data. This 'other' data can be referred to as *out-of-sample*, meaning that it is outside of the data that was used to train and evaluate the classifier. In other words, if one were to develop a machine learning approach on one data set and then apply the same algorithm to a second set of data drawn from the same distribution, performance might be much worse than on the original set of data even though one might expect the results to be highly similar. This is a severe problem, because models that cannot generalize to out-of-sample data have little to say about brain function in general: Their results are valid only on the data set used to configure the classifier, are tuned to the specific pattern of noise in the data, and are unlikely to be replicated on any other set. One of the earlier and more startling examples of overfitting was performed by Freedman (1983), where he showed—with high statistical significance—that a regression model could be used to find a strong relationship between independent random variables drawn from a standard normal distribution (which have no real relationship whatsoever).

To better understand the principles of this conundrum in machine learning, we rely on a commonly used distinction between *parameters* and *hyperparameters*. In the context of machine learning, we use the term parameter to refer to aspects of the analysis that are directly driven by the data through a training algorithm. For example when training a support-vector-machine (or SVM, a commonly used classifier in machine learning), the training algorithm uses the data to adjust a set of parameters which allow that classifier to learn how specific patterns of brain activity predict specific dependent variables. Hyperparameters, on the other hand, refer to aspects of an analysis that are configured (often by manual selection) to improve the outcome of the training process (see Table 1). In neuroscience hyperparameters will include, but are not necessarily limited to the following: artifact rejection criteria, feature selection (i.e. electrodes or ROIs in the brain), frequency filter settings, control parameters of classifiers (e.g. choice of kernels, setting of regularisation parameters), and even choice of classifier (e.g. SVM vs. random forests vs naïve Bayes). These are settings and choices that could, at least in principle, apply across a class of data sets.

In this context, we propose the term *over-hyping* as a specific case of (typically unintentional) overfitting through adjustment of analysis hyperparameters to improve the results for a specific data set after which point the same results cannot be obtained on another data set with the same hyperparameters. We suggest that over-hyping is a fairly widespread and poorly understood problem in the neurosciences, particularly because the field utilizes relatively expensive and time consuming data collection practices (unlike the field of machine-vision, for example).We feel that a better understanding of the error introduced through over-hyping is crucial, since this error is easy to commit yet difficult to detect. Furthermore, while there has been a lot of discussion of problems of circularity and inflated effects in neuroscience analyses (e.g. Kriegeskorte, Simmons, Bellgowan & Baker 2009; Vul, Harris, Winkielman & Pashler 2009; Eklund, Nichols, Anderson & Knutsson 2015; Brooks, Zoumpoulaki & Bowman, 2017; Bowman et al. In Press), machine learning algorithms are so effective that they provide dangers above and beyond those that have been discussed. Optimization of hyperparameters is a common and necessary practice in the machine learning literature (Bouthillier & Varoquaux 2020) and it is difficult to determine how the data were treated during the optimization process. Importantly, as will be demonstrated below, *the technique of cross-validation, often employed as a safeguard against overfitting, is not entirely*

effective at ensuring generalizability. We suspect that the incidence of accidental overfitting errors in the literature could be substantial already, and may increase as machine learning methods increase in popularity.

CROSS-VALIDATION DOES NOT PREVENT OVER-HYPING WHEN RE-USED ON THE SAME DATA SET

In the neuroscience literature and also in machine learning more generally, a method that is typically employed to prevent overfitting is cross-validation, in which data are repeatedly partitioned into two non-overlapping subsets. In each iteration, classifiers are trained on one set and tested on the other and the results of multiple iterations are averaged together.

There are many varieties of cross-validation, such as K-fold, in which the data are divided into K equal subsets (or "folds") and the train/testing process is repeated once for each of the subsets. In each repetition, the designated subset is used for testing while the remaining subsets are combined together to form a training set. Thus for a 10 fold cross-validation scheme, ten separate classifiers are trained, each trained on 90% of the data, and tested on 10%. The results are then computed as the average accuracy of the 10 classifiers on the test set. The accuracy scores from the training sets are not used, as these scores are likely to reflect some amount of overfitting.

Other approaches to cross-validation are similar. Stratified sampling can be used to ensure that each subset of the data has an equal proportion of samples from each class of data (e.g. hit vs miss trials) before the K folds are defined. Leave-One-Out methods break up the data into subsets such that each subset corresponds to one group of trials (e.g. one subject) and the classifier is trained for subsets excluding each such group in turn. Thus for a data set with 20 subjects, twenty classifiers would be trained, one excluding the data from each subject in turn and then tested on the excluded subject (but see Varoquaux et al. 2017 for a discussion of the increased likelihood for unstable accuracy estimates from leave-one-out techniques)

Regardless of which specific form of cross-validation is used, the principle of cross validation is that because the training and testing sets are disjoint in each iteration, the average performance on the test sets can be taken as an unbiased estimate of classifier performance on out-of-sample data.

However, this is only true as long as one important restriction is obeyed: After performing cross-validation, decisions regarding the analysis pipeline must *not* be made to obtain higher performance *on that same data*. Reusing the same data to optimize analysis parameters can induce over-hyping, *even if cross-validation is used at each iteration*.

The reason that over-hyping can occur despite cross-validation is that all data sets are composed of a combination of signal and noise. The signal is the portion of the data containing the useful information that one would like the machine learning classifier to discover, while the noise includes other sources of variability. However, when an analysis is optimized on a given data set after viewing the results, the choice of hyperparameters can be influenced by how the noise affected the classification accuracy. In other words, some of the unwanted noise "leaks" into the hyperparameter configuration. Consequently, while the optimization improves classification accuracy on this data set, performance may remain constant or even worsen on a completely distinct set of data, because (in a statistical sense) its noise is not shared with the data driving the optimization (Figure 1). In other words, the analysis would not replicate at the same level of significance on a distinct dataset even if the sampling conditions and the analysis were identical.

The possibility that cross-validation does not prevent over-hyping, is well known in the machine learning and machine vision communities (Domingos 2012), which are taking increasing care to avoid the problem. For example, machine-learning competitions on websites such as Kaggle.com provide contestants with sample data on which to optimize their models. However the final evaluation of the contestants is performed on a different data set that is either held as confidential by the sponsoring organization, or is released only a few days before the end of the competition (i.e. the "held out" set or "private set"). Contestants who access the data more often than the rules

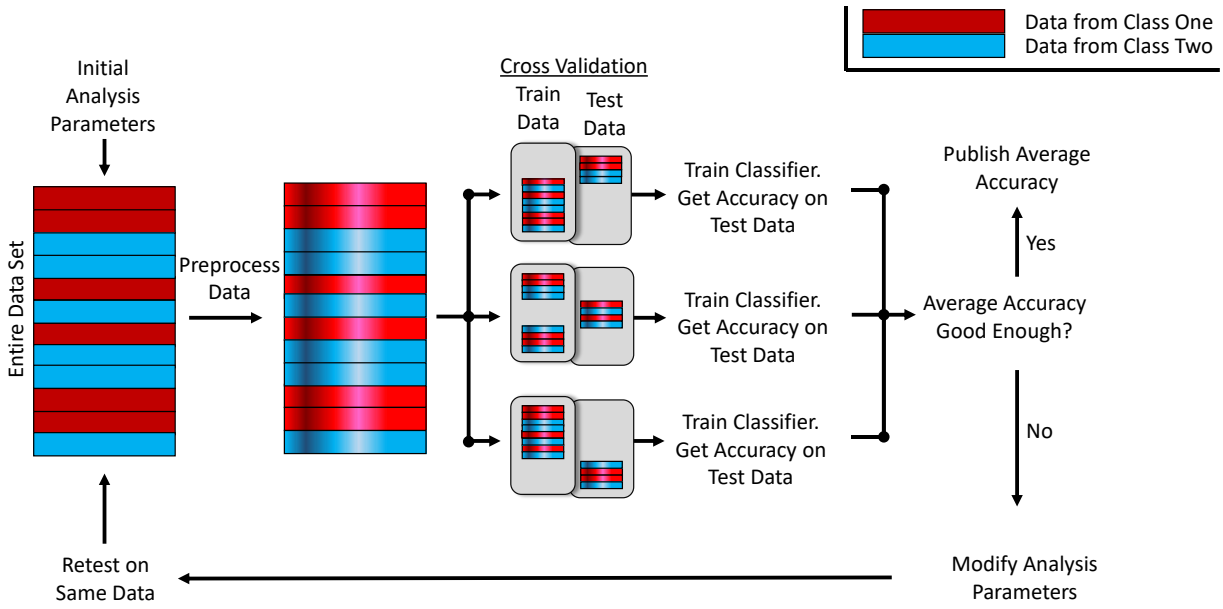

Figure 1. An example of how over-hyping can be induced by modifying hyperparameters after evaluating a system through cross-validation. The feedback loop allowing hyperparameters to be adjusted after viewing the results provides a route for analysis decisions to be made in response to the noise in the data set, despite the separation of data into training/testing sets.

permit are disqualified, their organizations can be barred from future competitions and in one recent high-profile case a lead scientist was fired (Markoff 2015).

In writing this paper, we share the experience of our colleagues in the physics and computer science disciplines so as to encourage more rigorous standards of machine learning before a replicability crisis in neuroscience machine learning unfolds. It is not our intent to call out specific examples of bad practice in the literature, although in our informal survey of the neuroscience classification literature it was rarely the case that appropriate precautions had been documented

(i.e. some variety of preregistration, blind analysis optimization, nested cross-validation or lock box, which will be described below). Without these precautions it is impossible to determine from a paper's methods whether overfitting of hyperparameters occurred. This is concerning because overhyping is unlike the problem of double-dipping(Kriegeskorte, Simmons, Belgowan & Baker 2009; Button 2019), which is more clearly discernible from the methods. Double-dipping refers to the practice of selecting a subset of data based on particular values in the data (i.e. picking a highly active set of voxels), and then running a statistical analysis on that same subset. The greater difficulty in identifying cases of over-hyping is that it would have occurred during the optimization of the analysis, and a description of the analyses performed during optimization of the analysis typically omitted from the methods. Another issue that we observe in the literature is inconsistent terminology, which makes it harder to understand exactly what was done (e.g. Ng (1997) and Varoquaux et al. (2017) use incompatible definitions of 'test set'). To help clarify terminology, we offer a table describing common terms and descriptions of what they are typically taken to mean (Table 1). We suggest a new term, the *Lock box*, which refers to a set of data that is held-out from the optimization process for verification and should not be consulted until the method's hyperparameters have been completely determined. The term *hold-out* data set is sometimes taken to mean this, but that term is also used inconsistently and is easy to misinterpret as a test-set in its most common usage. The term *lock box* more clearly indicates the importance of holding the data in an inaccessible reserve. More will be said about this below. Next, we provide clear examples of over-hyping despite use of cross-validation using a sample of EEG data recorded from our own lab. We use real data instead of simulated data, to ensure that the noise reflects the genuine variability typically found in similar datasets.

The first example shown here is a one-shot hyperparameter adjustment, in which 40 variations of machine classification are tested using cross-validation on a set of randomly scrambled data (i.e. data in which there is no signal). By taking the most favorable result from these 40 variations from each of a large number of iterations (1000 simulations in total) we evaluate how often a spurious result can be obtained by making a single hyperparameter choice despite cross-validation. The one-shot hyperparameter adjustment was often able to reveal a spurious classification effect using conventional temporal-generalization analyses that are currently favored by the EEG classification community (e.g. King & Dehaene 2014, Cichy et al., 2014). The illusory effect obtained by over-hyping, though small, would provide erroneous evidence of target discrimination in the EEG data

over long periods of time, which is commonly taken as evidence that a neural correlate of working memory has been measured. A comparison to a lock box data set (i.e. data that were not consulted during analysis optimization) reveals that there is no reliable classification of target presence by this chosen set of hyperparameters, which is the expected outcome from a randomly shuffling of labels on the data set.

In the second example, we show a more extreme case of overhyping. Hyperparameters were iteratively optimized to eliminate some features of the data set through a genetic algorithm using cross-validation at each step. This process is analogous to recursive feature elimination (RFE), a commonly used technique in analysis optimization. Performance was compared to lock box data that were set aside and not used in the genetic algorithm's fitness function. Performance was shown to improve on the data on which the classifiers were optimized, but not on the lock box data. Note that highly robust over-hyping was obtained, despite the use of cross-validation. The obtained results, presented below, demonstrate that classifiers can easily be over-hyped to obtain performance that will not generalize to set-aside or out-of-sample data.

METHODS

EEG METHODS

The simulations presented below were performed on EEG data, which was collected from rapid serial visual presentation (RSVP; Experiment 3 of Callahan-Flintoft, Chen and Wyble 2018; see supplemental for comprehensive methods). Subjects viewed a series of changing letters, presented bilaterally, updating at intervals of 150ms, and were tasked with reporting the one or two digits that would appear on each trial. For this analysis, we selected the trials containing either a single digit, or two digits presented in sequence separated by 600msand attempted to classify for each trial, whether one or two digits had been presented. However, the trial labels were randomly shuffled within subjects to obscure any actual effect of this manipulation. During each trial, EEG was recorded at 32 electrode sites and according to the standard 10-20 system. It was further bandpass filtered from 0.05 – 100 Hz, originally sampled at 500 Hz and down-sampled offline to 125 Hz for the present analysis. For further details on pre-processing and artifact rejection, see the EEG recordings section of experiment one in the original paper (Callahan-Flintoft et al., 2018). The original study excluded one subject due to an insufficient

number of trials after artifact rejection. We decided to exclude an additional subject, choosing the one with the least number of trials, in order to be able to split the data into two equal parts for hyperparameter optimization and lock box, detailed below. The final number of subjects was 24. Finally, the original study divided the data based on the visual hemifield in which the target stimulus was presented and only included trials in which correct responses were provided. We collapsed the data across hemifields and included all trials regardless of accuracy to increase the number of available trials per subject. Using all trials in this way is an experimenter degree of freedom (i.e. a hyperparameter) that was adopted without looking at the analysis results and thus could not have contributed to over-hyping. The complete methods from the original paper are provided in the supplemental.

SIMULATION 1. OVERHYPING DUE TO KERNEL SELECTION DESPITE CROSS-VALIDATION

The first analysis measures the property of temporal generalisation within an EEG signal, which indicates whether a classifier trained at one point in time relative to stimulus onset is able to classify trial categories at other time points. Such analyses have been used to examine whether memory representations are stable over time in working memory research (e.g. Dehaene & King 2014).

We ran a series of 1000 independent executions (which we will refer to as iterations below) to measure whether and how often a spurious effect could be obtained if one tested a set of 40 different classifiers on independent random shuffles of a data set. In effect, this is similar to 1000 scientists trying to perform over-hyping on 1000 randomly shuffled copies of the same data set. Each of the 1000 scientists uses cross-validation on 40 different kinds of classifiers and then chooses their best result from the 40.

It needs to be stressed: all analyses were exclusively performed on null-data. Hence, any systematic improvements above chance performance must be due to over-hyping. Also, the dataset was randomly split into two equal parts of 12 subjects. One set was for hyperparameter OPtimization (OP) and the other was the lock box (LB) set. The data were reshuffled into new OP and LB sets at the beginning of every iteration to ensure that any effects were not subject-specific. Our temporal generalisation analyses used functions of the MVPA-Light toolbox (Treder, 2020).

For each of the 1000 iterations randomized OP & LB data sets were created. 40 configurations of classifiers (i.e. 40 different hyperparameter configurations) were used in each iteration to generate temporal generalisation maps to determine which configuration had produced the most desired outcome classifying the random OP data set. The 40 configurations were derived from 4 different classifiers: support vector machines (SVM) with three different kernels (linear, polynomial (order of 2); radial basis function (RBF)) and a linear discriminant analysis (LDA). Additionally, the extent of regularization was varied through 10 choices for each classifier. For SVMs, the C parameter took values of 0.0001, 0.0007, 0.0059, 0.0464, 0.03593, 2.7825, 21.5443, 166.81, 1291.5496 and 10000. The choice of C values was inspired by (and equal to) the search space of MVPA-Light's default regularization search for SVMs. For LDA, candidate lambdas were 1, 0.88, 0.77, 0.66, 0.55, 0.44, 0.33, 0.22, 0.11 and 0. Temporal generalisation analyses were performed using 5-fold cross-validation.

These 40 candidate configurations competed in each of the 1000 iterations of the analysis, which we call *OP Competition* as it represents a competition between hyperparameters to decide the best available. This OP competition was decided using a measure we call classification mass (*C-Mass*), which was computed on group-average temporal generalization maps (i.e. averages of 5-fold cross-validated single-subject maps). C-Mass reflects the average AUC value across the entire temporal generalization map. For each of the 1000 iterations, the hyperparameter configuration that led to maximum C-Mass (i.e. highest map-average AUC value) when classifying the OP data set was selected as the respective winner of the OP competition for that iteration. These winning configurations were then used to assess the degree of over-hyping by comparing them to the LB set.

There are a number of plausible ways to formulate a C-Mass index. An alternative set of of simulations is presented in the appendix. The alternative measured the extent of above- as well as below-chance AUC across the entire temporal generalization map to acknowledge the fact that below-chance classification in the context of EEG data can be meaningful (we provide a brief discussion of this in the appendix, too). Both versions of the C-Mass analysis reveal essentially similar results and we use the above-chance variant in the main body because above-chance classification is the more canonical approach.

We selected one of the 1000 iterations to demonstrate how manual selection could produce *what appears to be* a theoretically meaningful result in a temporal generalization map for data that was randomly shuffled and subjected to cross validation (Figure 2). Evaluating the efficacy of different hyperparameter configurations on the same dataset can be considered an analog of an exploratory analysis in which an analyst runs a series of cross-validated pilot analyses and stops on finding one that is theoretically suitable. In the working memory literature, it is considered important that a classifier is able to decode the condition label after the stimulus has disappeared. In our manually selected case, the winning OP map can be regarded theoretically suitable as it appears to exhibit this property whereby the accuracy remains well above chance for a substantial period of time after the target onset (see the supplemental for a randomly selected set of 9 additional iterations). However, this effect is demonstrably spurious since the trial labels were all randomly shuffled. To demonstrate that this observed pattern is due to overhyping and not a general property of our analysis, we also present the results of the same analysis configuration for the companion LB set as well as of an alternative classifier configuration for the same OP set. It is clear that the pattern observed in the winning OP map does not generalize either to another data set drawn from the same population using the same kernel configuration (the LB set) or to a reanalysis of the same data set with a different configuration (the losing OP set). This is an instance of overhyping (overfitting due to selection of hyperparameters), because the hyperparameters determined with the OP set fitted the noise best compared to the other candidate hyperparameters. As the noise differed in the LB data set, classification performance was overall at a lower level and the pattern of more successful classification at later time points was also disrupted, causing our permutation test, introduced next, to generate significant AUC clusters for the winning OP, but not the LB map.

We adopted a cluster-extent permutation test for our temporal generalisation maps, which was based on functions of the ADAM toolbox (Fahrenfort, van Driel, van Gaal & Olivers, 2018). We performed a first-level Wilcoxon signed rank test (non-parametric alternative to a t-test, preferred as distributions of AUCs do not meet parametric assumptions) at each pixel of the temporal generalisation map across single-subject maps, which resulted in a map of p-values. Neighbouring AUCs found to be significant for this test formed clusters and these clusters' sizes were subsequently tested against a permutation-distribution of maximum cluster-sizes under the null. Clusters were determined statistically significant if only 5% of permuted maximum cluster-

sizes exceeded their size (i.e. alpha of 0.05). For a more detailed introduction of this test, see the supplementary material.

SIMULATION 1. RESULTS

The results of these simulations revealed a systematic improvement in C-Mass by selection of the winning analysis. To illustrate that these effects are systematic, a comparison of the OP competition winners against their respective LB counterparts shows how the C-Mass

distributions are shifted by the selection process, despite the use of cross-validation (Figure 3).

The top panel illustrates that the average C-Mass is greater for the winning OP than the set of

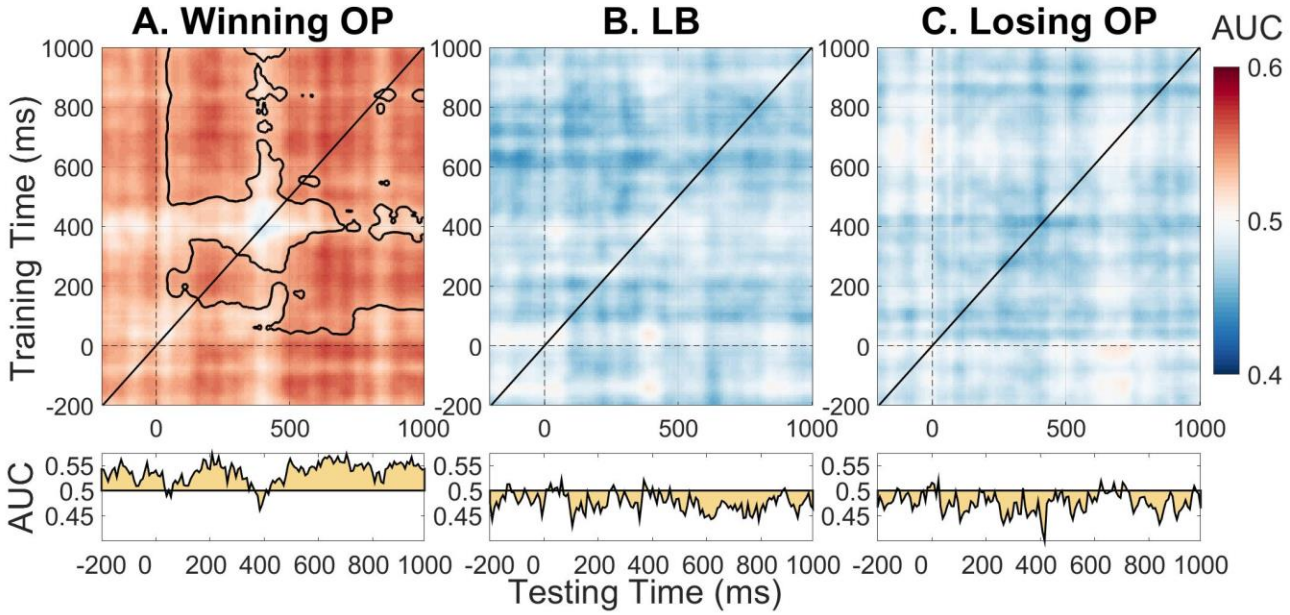

Figure 2. How overhyping manifests in temporal generalisation maps. Maps of a winning optimization set (Winning OP), its corresponding Lock Box (LB) and the worst optimization set (Losing OP), which implemented hyperparameters that led to *minimal* C-Mass, (panel C) are plotted with their main diagonal AUC vectors below. Beige areas in AUC time-series plots show divergence from chance-level classification (i.e. AUC of 0.5) in main diagonals. Classification performance was at a higher level for the winning OP compared to both other analyses. A family-wise error correction cluster-extent test was performed (Nichols & Holmes 2002) for winning OP & LB maps and only showed statistically significant AUC clusters for the OP map. Maps and cluster-boundaries (i.e. matrices determining statistical significance) were 2D-smoothed separately using a boxcar of 40 ms width. This was only done to facilitate visualization and did not affect any analyses, which were all computed prior to smoothing. As all three analyses decoded null-data, any differences in classification performance must be due to the effectiveness of classifiers' hyperparameters (in this case an LDA classifier with a lambda of 1 for winning OP & LB). This is a demonstration of overhyping because these hyperparameters fitted the noise of the OP dataset best, which however differed in the LB dataset and thus led to decreased classification performance for the LB. This map triplet was manually chosen. An additional 9 triplets can be found in the supplementary material.

LB's.

The second panel illustrates that classification performance on a randomly selected set of hyperparameters, as opposed to the winning set from the parameter-optimization phase, is approximately equal to performance on the LBs dataset when using the winning hyperparameter set, as it should be if the difference between OP and LB classification is due entirely to noise. This result demonstrates how a Lock Box provides an unbiased estimate of performance, as the resulting C-Mass is free of any overhyping effects.

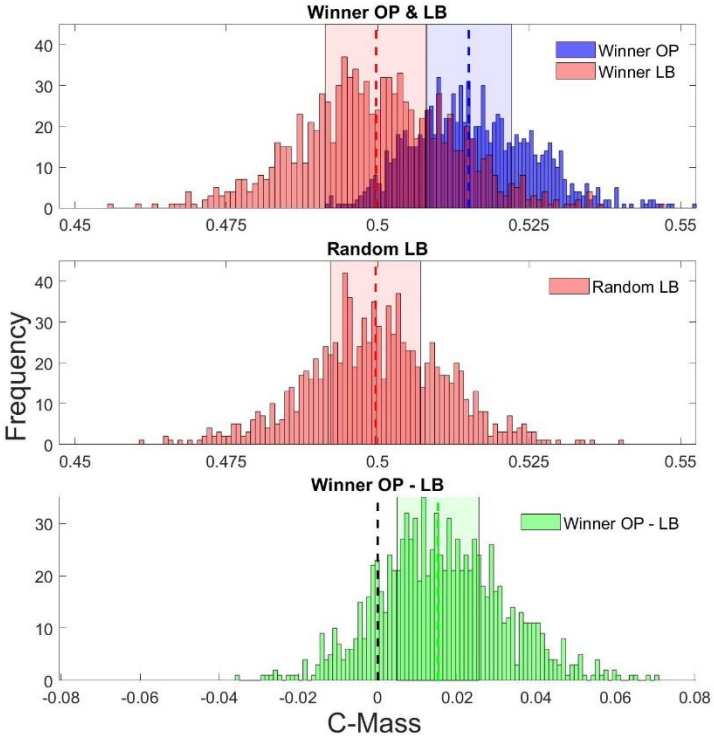

Figure 3. C-Mass distributions of OP (blue) and LB (red) maps (top two panels), as well as their within-iteration difference (bottom panel). The top panel shows C-Mass results for the OP & LB maps that incorporated winning hyperparameters from the parameter-optimization phase, the middle panel shows the distribution of LB C-Mass after choosing hyperparameters randomly. The coloured vertical lines indicate distributions' median value and the rectangles surrounding these lines indicate the interquartile ranges. The black vertical line in the bottom panel indicates a OP – LB difference of zero (i.e. no overhyping).

The focus of this analysis lies in the top panel of Figure 3: the distributions of OP & LB C-Mass for winning hyperparameters of the OP competition clearly demonstrate overhyping of classification results. If overhyping was absent, these distributions should sit on top of one another. However, the POOP C-Mass distribution has a higher mean (0.516), median (0.515) and smaller variance (0.0001) compared to the LB C-Mass distribution (mean: 0.5, median:0.5, variance:0.0002). The bottom panel of Figure 2 illustrates how the *within-iteration* differences in C-Mass between OP and LB were distributed. This distribution should be centred around zero if no overhyping was

observed (i.e. a given set of hyperparameters leading to similar success in decoding between-class differences for OP as well as LB null-data). The observed mean (0.016) and median (0.015) of the difference distribution was positive, implying higher C-Mass in OP maps. Permutation tests, which were based on randomly determining the direction of subtraction between OP & LB C-Mass to generate a distribution of OP-LB C-Mass differences under the null, confirmed that both values were significantly different from zero ($p < .001$), which provides evidence for overhyping. However, p-values obtained from simulation analyses should be interpreted with caution, as we discuss in more detail in the supplementary material.

We further assessed how vulnerable the different classifiers were to overhyping. Across all classifiers, the median difference in C-Mass between winning OP and LB was positive and significantly different from zero after performing the permutation test introduced above (linear SVM: median = 0.015, n = 329; polynomial SVM: median =0.013, n =189; RBF SVM: median = 0.013, n =132; LDA: median =0.018, n =350). We investigated whether overhyping was more pronounced for certain classifiers by conducting a Kruskal-Wallis test (due to non-normality of C-Mass values), which revealed a significant difference among the four classifier types (χ^2 (3,996) =20.07, $p < .001$). Post-hoc pair-wise tests of mean rank-differences between classifiers provided evidence that over-hyping was significantly larger after LDA classification compared to all three SVM classifiers. The differences between SVM classifiers were all non-significant (we provide detailed results of this analysis in the supplementary material).

Finally, we present an exploratory analysis in the supplementary material, which suggests that temporal generalization with simple classifiers (e.g. having linear classification kernels) generates less stable (i.e. more variable) C-Mass values. In our simulations, this led to such models winning and losing (the latter implying minimal C-Mass across all hyperparameter configurations in a given iteration) the OP competition about twice as often as more complex models.

SIMULATION 2. OVERHYPING BY FEATURE SELECTION DESPITE CROSS-VALIDATION

In addition to kernel parameters, analysis optimization can involve feature selection, in which portions of the data set are excluded from the pipeline on the grounds that they contain irrelevant information that can reduce classifier accuracy (e.g. Deshpande et al. 2010). This method is widely used and is included as Recursive Feature Elimination (RFE) in scikit-learn (Pedregos et al. 2011). Here, we show that when cross-validation is the only protection against over-hyping, this method will induce spurious findings of significant classification accuracy on randomly shuffled data when feature selection is based on classification accuracy.

We ran a series of 16 independent executions to measure how effectively one could overhype a data set using feature selection via a genetic algorithm approach for feature selection. This simulation is similar to 16 different scientists trying to perform over-hyping on 16 randomly shuffled copies of the same data set. Each of those 16 scientists uses cross validation for several hundred iterations, progressively improving the analysis hyperparameters at each iteration. The raw data are the same as were used in the first analysis and are again randomly shuffled to remove differences between conditions (in a statistical sense). A simpler classifier is used which determines on each trial whether one or two targets had been presented based on the output of a spectral analysis. In this analysis, selection of features occurs by weighting different frequency components with channels collapsed.

A fast Fourier transform (FFT) of the 64 data points (comprising 256 ms) from each EEG channel after the first target onset were extracted from each trial. The log of the absolute value of the FFT was computed, and spectra across all channels were summed, resulting in 64 frequency values per trial. The classifier attempted to determine whether subjects had seen one or two targets within a given trial based on these 64 frequency values that represented the scalp-wide power spectrum from the 256ms time period after target onset. As above, the trial labels were randomly shuffled prior to the analysis to remove the correspondence between data and conditions.

A support vector machine (SVM) was used to classify the post-processed EEG data and the over-hyping was accomplished with a custom genetic algorithm that adjusted weights for the 64 frequency bands available to the classifier. The SVM was MATLAB's fitcsvm, with an RBF kernel and kernelscale set at 25. No additional classifiers or kernel settings were attempted for this analysis.

To demonstrate that cross-validation is inadequate protection against overhyping, the analysis was repeated for 16 iterations. For each iteration, 15% of the data were set aside in a Lock Box (LB) to test for over-hyping. Since the data was randomized, it was expected that performance on this outer test set should be at 50% (chance level), while performance on the 85% of the trials that formed the Hyperparameter Optimization (OP) set would be elevated above chance by the last generation of the genetic algorithm. Over-hyping on the 16 OP sets was performed using a genetic algorithm coupled with cross-validation. For each generation of the genetic algorithm, 10 candidate feature-weight vectors were each evaluated against a shared set of 10 random partitions of the OP dataset, with 85% of trials in each partition used to train the SVM and 15% used for testing. At the start of the optimization procedure, the 10 candidate weight vectors were randomly constructed with 64 values ranging from 0.95 to 1.05. During training and testing, these vectors were multiplied by the power spectra for each trial before being provided to the SVM.

Within each OP iteration, for each of the 10 candidate feature weight vectors, the SVM performance in terms of AUC on the 10 random partitions was averaged to compute performance for each candidate. The best candidate was selected and then repeatedly mutated by adding 64 random numbers (range [-.05 .05]) to create 10 new candidates for the next generation of the genetic algorithm. This process was repeated for 400 generations to optimize the analysis.

To measure over-hyping, after each generation, the best feature weight vector was also used in a classification of the LB set for each of the 16 iterations and the results were not used to inform the evolution of the feature-weight vector. This is a strong violation of the principle of using a lock box but it is done here as a demonstration. In practice accessing a lock box multiple times can itself result in overfitting, particularly if the results are used to influence analysis choices or stopping criteria.

To measure the statistical significance of the model's classification on the hyperparameter optimization set, a permutation test was run after the final generation of the genetic algorithm. First, the analysis result was computed as the mean AUC across the ten OP partitions using the final generation of feature-weights. Then, all condition labels for the trials (i.e. the target-type) were randomly shuffled 1000 times, a number chosen to balance the computational costs of running 1000 separate analyses. After each such shuffling, for each of the ten partitions, the SVM

classifier was retrained with the best final weight vector and the AUC was measured. These AUC values were shuffled to create a null-hypothesis distribution of 1000 values.The p-value was then computed as the fraction of the null-hypothesis distribution that was larger than the non-permutated classification result (i.e. the proportion of shufflings that produced a mean AUC greater than the mean AUC on the unshuffled data).

This entire procedure was repeated independently for 16 iterations times to demonstrate the robustness of over-hyping. In each case, the data were randomly repartitioned into a hyperparameter optimization set and a lock box, and the genetic algorithm was used to optimize weights for the hyperparameter optimization..

SIMULATION 2. RESULTS

The results of overhyping by feature selection are illustrated in Figure 4, which shows that performance improves on the hyperparameter optimization set without corresponding changes on the lock box set. As the labels were randomly shuffled, any performance above chance (AUC of 0.5) in a statistical sense, would indicate overhyping. All 16 iterations of the OP set had significantly elevated performance by the final generation of the feature-selection genetic algorithm. One of the LB sets was significant.

DISCUSSION

This paper demonstrates the ease with which over-hyping can be induced when using machine learning algorithms despite the use of cross-validation. The approaches used here are analogous to optimization procedures that have been used in EEG/MEG classification such as exploring various kernel options or discarding channels and frequency bands to improve classification performance. Similar problems may exist with other hyperparameters, e.g. choosing time windows, or different ways of filtering out artifacts. Moreover, the same concerns apply to any kind of large neural data set. For example, in the case of using multi-voxel pattern-analysis (MVPA) on fMRI data, optimization through selection of any analysis step in the pipeline during consultation with the data could lead to the same kinds of over-hyping that we demonstrate here.

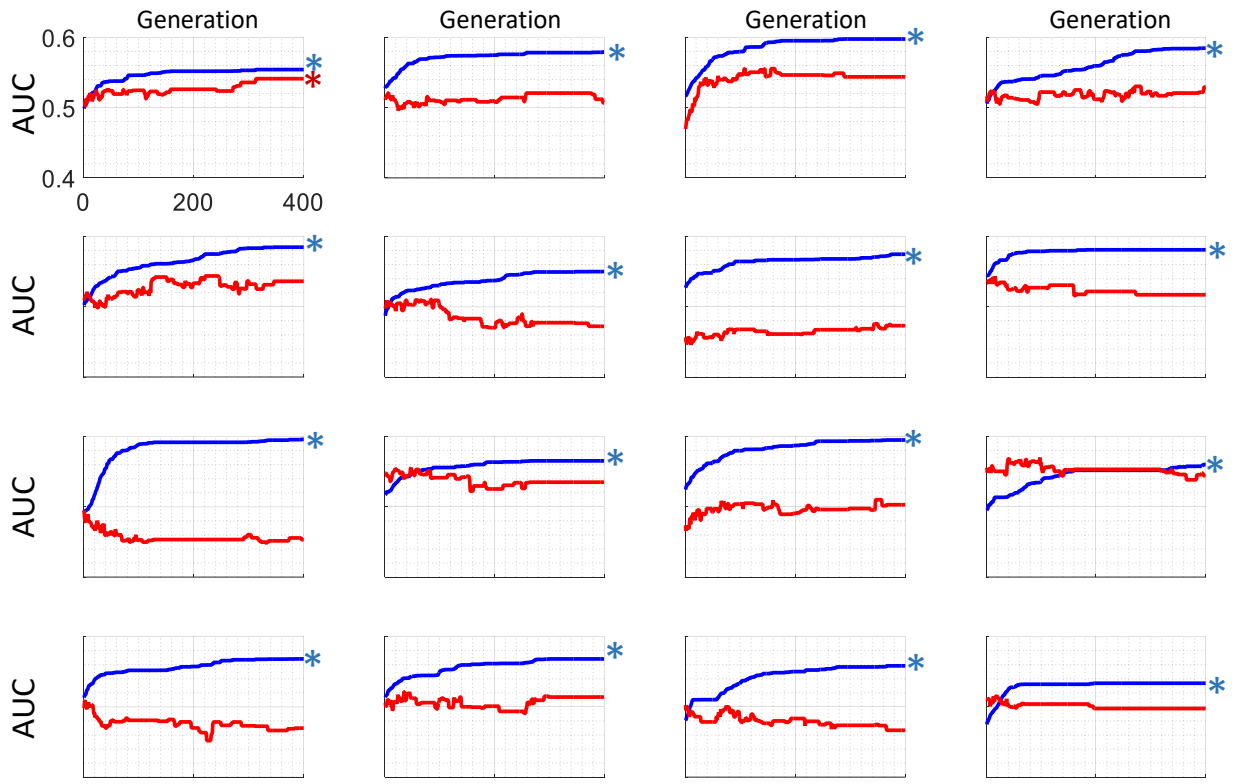

Figure 4. To demonstrate that models can be over-hyped using feature selection, a genetic algorithm was used to iteratively select features to optimize performance on a randomly shuffled EEG data set, thus performance should not deviate from chance. The optimization procedure was run for 16 iterations, with 400 generations in each. The blue trace indicates accuracy from a cross-validation test on the hyperparameter optimization set, while the red shows performance on a lockbox set. The asterisks indicate when the results of the final generation differed significantly from chance at an alpha level of .05. All of the OP sets were significantly different from chance, while only one of the LB sets was.

These results should not be taken to indict cross-validation as a poor methodological choice: It is considered to be state-of-the-art by many in the machine vision and machine learning communities for good theoretical and practical reasons (Arlot & Celisse 2010). However, our result does clearly indicate that cross-validation does not permit heedless analysis optimization.

Importantly, the problem of over-hyping becomes more severe as the sample size reduces. This reflects the fact that error bars are larger when samples are small (Lorca-Puls et al, 2018), a phenomenon that has been compelling demonstrated in machine learning applied to neuroimaging data (Flint, et al, 2019; Varoquaux, 2018). This mirrors the law of large numbers in classical statistics, which states that there is increased error in estimates as samples get smaller (Dekking et

al, 2005). The combination of large error bars and over-hyping means that applications of machine learning in neuroimaging are likely to be especially vulnerable to the file-drawer effect (Lorca-Puls et al, 2018), which reflects the fact that only analyses that generate significant effects get published, leading to potentially very severe inflation of published accuracies and effect-sizes.

There are several ways in which over-hyping can be protected against, above and beyond standard forms of cross-validation. We suggest that, in order to increase generalizability and replicability, journals publishing data from classification analyses encourage the use of one of the approaches listed below.

THE PRE-REGISTRATION APPROACH

In cases where there is a clearly defined analysis plan that exists before efforts are made to analyze the data, a really good approach to minimizing over-hyping is pre-registration. Pre-registration (Nosek, Ebersole, DeHaven, & Mellor 2018). involves submitting a complete analysis plan to an external server that is accessible to the journal's readership. This practice encourages the practitioner to specify all hyperparameters at the onset of an analysis and provides a time stamp indicating that they have done so. This is helpful because cross-validation does succeed in providing an unbiased estimate of out-of-sample performance when classification results are not used to iteratively optimize performance. Therefore, it is safe to pre-register or otherwise rigidly specify a classification analysis before attempting it. The pre-registration would provide evidence that the hyperparameters were finalized prior to attempting the analysis using previously established methods. The advantage of this approach is that all of the data can be used in the final estimate of performance. The disadvantage is that hyperparameter optimization is not permitted, which limits the effectiveness of the analysis. The Registered Report (Chambers, Forstmann, & Pruszynski, 2017) is another publication format that can guard against over-hyping in a similar way as pre-registration. In this context, an analysis plan is developed in consultation with a reviewing team before the data are analyzed and the article is published regardless of the outcome. This approach removes any opportunity to overhype provided that no modifications to the analysis are performed.

THE LOCK BOX APPROACH.

Using a metaphorical data lock box makes it possible to determine whether over-hyping has occurred. This entails setting aside an amount of data at the beginning of an analysis and not accessing that data until the analysis protocol is clearly defined, which includes all stages of pre-processing, artifact correction/rejection, channel or voxel selection, kernel parameter choice, and the selection of all other hyperparameters. A close variation of this technique is already standard practice in machine learning competitions. When submitting a candidate for such a competition, the ultimate performance of the algorithm is evaluated on a separate set of data that is reserved until the final stage of the test. The workflow of using a lock box is shown in Figure 5.

We suggest that, moving forward, when machine classification approaches to data analysis in neuroscience must be developed without clear default choices for hyperparameters or existing software, that such approaches should incorporate a lock box approach, in which data are set aside at the beginning of the development of an analysis and not assessed until the paper is ready for submission (or equivalently, new data are collected at the end of analysis optimization). At this point, the data in the lock box should be accessed just one time to generate an unbiased estimate of the algorithm's performance. This result is likely to be less favorable than the data that were being used during optimization and should be published alongside the results from any other analyses. At the same time, reviewers would need to be more willing to accept results that seem less positive than they historically have, since our current understanding of generalized machine learning accuracy is likely to be biased by current practices.

If it turns out that the results from the lock box test are unsatisfactory, a new analysis might be attempted, but if so, the lock box should be re-loaded with new data, either collected from a new sample or from additional data that were set aside at the beginning of the analysis (but not from a repartitioning of the same data that had originally been used in the lock box).

A possible alternative is to access the lock box multiple times during optimization, but to apply a correction to any resultant statistics as a function of the number of times the lock box data was evaluated. A method for accessing a lock box multiple times while limiting overfitting was suggested by Dwork (2015). This method called for simultaneously evaluating a given model on both the hyperparameter optimization set and on the lock box, and then only revealing the

performance on the lock box to the operator if that performance was significantly different than that of the model on the hyperparameter optimization set. Furthermore, the performance on the lock box set would be presented only after being summed with a Laplacian noise variable. By following this method, the maximum error rate when generalizing to out-of-sample data can be limited by only observing the performance on the lock box a set number of times (and halting hyperparameter optimization once that limit is reached). While this is an innovative method for limiting overfitting, it only sets the maximum error rate when generalizing – To get the true error rate, a second lock box would have to be used.

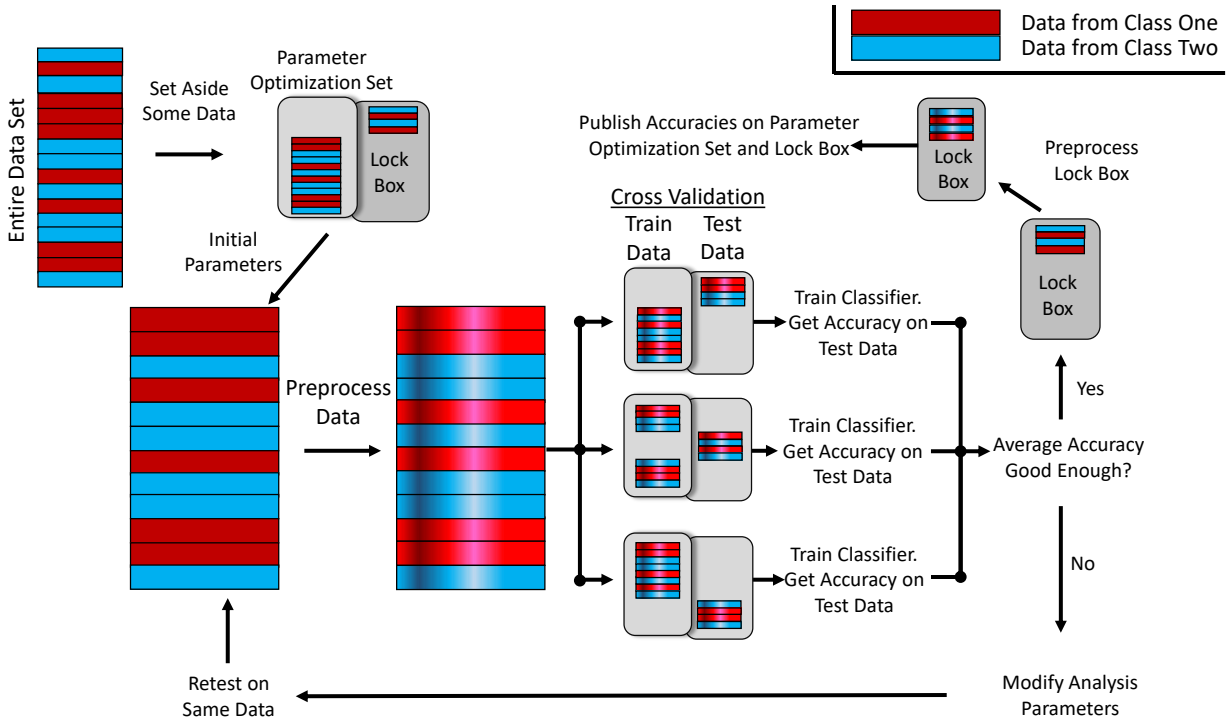

Figure 5. Here the workflow of using a lock box is demonstrated in illustrative form. Data is first divided into a hyperparameter optimization set and a lock box. The model can be repeatedly tested and hyperparameters can be iteratively modified on the hyperparameter optimization set. After all hyperparameter optimization and the analysis workflow is determined, the model can be tested against the lock box data. By doing this, an unbiased estimate of overfitting can be obtained, and an objective measure of how well this system will generalize is achieved.

Note that this lock box approach is evaluative. It does not prevent over-hyping, but allows one to test whether it has occurred. However, the performance of the algorithm on the lock box is guaranteed to be a non- over-hyped result if the technique was correctly used.

NESTED CROSS-VALIDATION

Another way to respond to the problem of overfitting hyperparameters is to use a generalization of cross-validation, called *nested cross-validation* (Cawley & Talbot 2010; Stone 1974). Nested CV helps to ensure that results are not specific to a given analysis configuration by showing that the results generalize to out-of-sample data. In this approach, inner cross-validations are run within an outer cross-validation procedure, with a different portion of the data serving as outer "hold-out set" on each outer iteration. Importantly, for each outer iteration, an unbiased assessment of accuracy can be obtained by testing on this outer hold-out set. That is, the best parameters and hyperparameters determined on each inner cross-validation, can be assessed out-of-sample on the corresponding outer hold-out set.

Nested cross-validation can be thought of as a repeated lock box approach, in which a new box (the hold-out set) is set aside and locked for each iteration of the inner cross-validation loop (Figure 6). Then, an overall accuracy (and indeed dispersion of accuracies) can be obtained by averaging across the accuracies determined from the hold-out sets of each outer iteration. This will typically be a more reliable measure of accuracy than that obtained from any individual outer iteration (i.e. the lock box approach). However, it is critical that the outer folds are not cherry-picked to find the best solutions, since this would constitute over-hyping. It is also important that the algorithm not be re-run in its entirety with different parameters after viewing the results, since this again would result in over-hyping.

An issue for nested cross-validation relative to the lockbox is that the average accuracy obtained at the end of the procedure will be the result of multiple configurations of hyperparameters, and thus it may be especially difficult to understand the link between the data and the accuracy. For example, in analysis of fMRI data where the region of interest is one of the hyperparameters, different iterations of the outer loop may converge on different regions of the brain. It would therefore be difficult to gain insight into what brain areas are driving the classification. We give more details of nested cross-validation and a simplified example in the Supplementary Material.

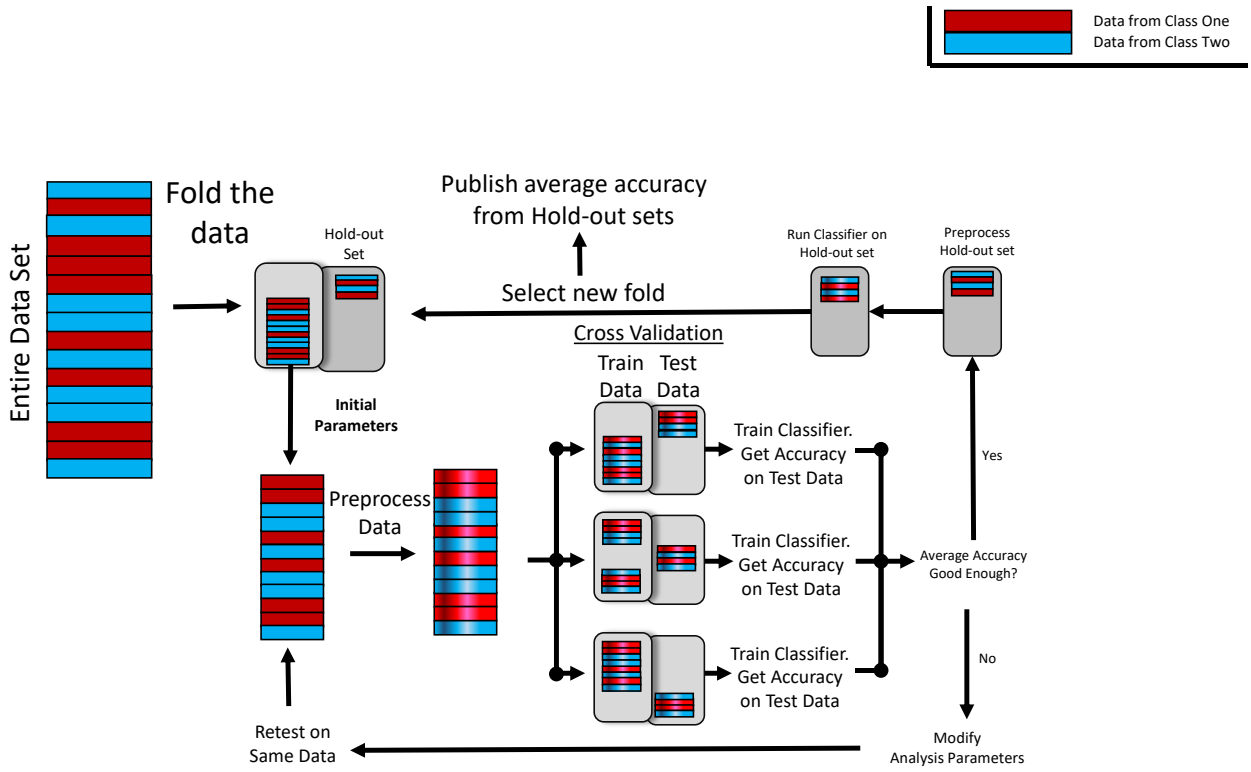

Figure 6. Here the workflow of nested cross-validation is demonstrated in illustrative form. The data set is folded into multiple combinations of hold-out set and inner optimization set. Each of these folds is essentially similar to the lock box approach described above and can be optimized. The final accuracy would be the average accuracy computed across all of the hold-out sets.

THE BLIND ANALYSIS APPROACH

Blind analysis can be an appropriate tool for preventing over-hyping when testing a well-defined hypothesis. In other words, the analysis protocol is develop using real data, but with the labels of each trial or subject obscured so that the analysis optimization process is unable to produce over-hyping. An alternative is to use an orthogonal contrast, where classification is done on unaltered data but using a condition that is orthogonal to the classification one will ultimately use (Brooks et al. 2017; Bowman et al. In Press). Some examples of using blind analysis include scrambling all condition labels and then artificially adding ‘target signals’ to some trials. The hyperparameters of the model can then be optimized to detect the signal present in the modified data. Once the

hyperparameters are locked in, the blind can be lifted (e.g. conditions unscrambled and modifications removed), and the true results can be calculated. The advantage of this approach is that all of the data can be used during the optimization phase, and the final evaluation of performance can be done across all of the data instead of just the outer-box set. Note that blind-analysis is a way to minimize over-hyping. If used in conjunction with a lock box, one can both minimize and diagnose overfitting. The disadvantage of the blind analysis is that it obscures accuracy on the key predicted variable, and this may prevent the development of an effective analysis plan depending on the type of data one uses, in which case a lock box is a good solution.

WHICH APPROACH TO USE

Each of these approaches is ideal for particular use cases. The simplest decision point hinges on whether the analysis plan is already established, in which case pre-registration is clearly the best choice. Blind analysis is suitable when hyperparameters need tuning to accommodate unanticipated variability in the data that is orthogonal to the predictor (e.g. finding the time window or location of a brain signal of interest). Nested cross-validation is well suited to a case in which an automated algorithm can be used to tune hyperparameters, and the precise values of those hyperparameters are not of interest. Finally, the Lock box, particularly when it is very large, is best suited to a case in which the values of tunable hyperparameters are of particular interest or the process of tuning them is done partially by hand, rather than by automation.

Regardless of which approach one takes, it seems crucial that more transparency should be applied to documenting how data is treated through the entire process of developing a pipeline. For example pilot tests of an analysis can lead to overhyping if they inform the search range of hyperparameter optimization prior to partitioning data into different sets. In such cases, being transparent can highlight the points where leakage of information into the (hyperparameter dependent) pipeline may have occurred.

SAFE VERSUS EFFECTIVE USE OF MACHINE LEARNING

Optimal use of machine learning in neuroscience requires that it be used both safely (i.e. without over-hyping such that the results can be trusted) and effectively (i.e. the classifier is

appropriately tuned to discriminating signal). In the terminology of machine learning, *safe* largely corresponds to *minimizing variance*, while *effective* largely corresponds to *reducing bias* Geman, Bienenstock, & Doursat (1992). The methods described above help to ensure safety, but do not necessarily provide effective solutions, since the avoidance of over-hyping is often obtained by limiting the amount of analysis optimization that is allowed. When data is easy to obtain, this limitation is not as severe, since analysis chains can be repeatedly adjusted, and tested against new data. However data in the neurosciences is often expensive and time consuming to collect. Unfortunately, this means that one often has to choose between analyses that are highly optimized but over-hyped, or weakly optimized and not over-hyped. The best path forward is to make use of expertise when it is available, such that good decisions are made up front, and ideally even pre-registered prior to viewing the results of analysis on critical data.

CONCLUSION

The biggest danger of data science is that the methods are powerful enough to find apparent signal in noise, and thus the likelihood of data over-hyping is substantial. Furthermore, analysis pipelines are complex, which makes it difficult to clearly understand the possibilities for leakage between optimization and evaluation stages that can lead to over-hyping. Our results illustrate how easily this can occur despite the use of cross-validation. Moreover, it can be difficult to detect over-hyping without having an abundance of data, which can be costly to collect. However, as reproducibility is a cornerstone of scientific research, it is vital that methods of assessing and assuring generalizability be used. By setting aside an amount of data that is not accessed until the absolute completion of model modification (i.e. a lock box), one can obtain an unbiased estimate of the generalizability of one's system and of how much over-hyping has occurred. Alternatively, blind analysis methods, good faith pre-registrations of the analysis parameters and nested cross-validation reduce the possibility of overfitting. Conversely, using any method that allows one to check performance on the same data repeatedly without independent data that has not been consulted can induce over-hyping, inflating false positive rates and damaging replicability. Devoting more attention to these dangers at this point, when machine learning approaches in neuroscience are relatively nascent, will allow us to improve the state of science before inappropriate methods become standardized.

ACKNOWLEDGEMENTS:

This work was performed with the support of NSF grant 1734220 to B. W.

REFERENCES:

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics surveys*, *4*, 40-79.

Bouthillier, X., Varoquaux, G. (2020) Survey of machine-learning experimental methods at NeurIPS2019 and ICLR2020. [Research Report] Inria Saclay Ile de France. 2020. ffhal-02447823f

Bowman, H., Brooks, J.L., Hajilou, O., Zoumpoulaki, A. and Litvak, V. (in press) "Breaking the Circularity in Circular Analyses: Simulations and Formal Treatment of the Flattened Average Approach" *PloS Computational Biology*.

Brooks, J. L., Zoumpoulaki, A., & Bowman, H. (2017). Data-driven region-of-interest selection without inflating Type I error rate. *Psychophysiology*, *54*(1), 100-113.

Button, K. S. (2019). Double-dipping revisited. Nature neuroscience, 22(5), 688-690.

Cawley, G. C., & Talbot, N. L. (2010). On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, *11*, 2079-2107.

Chambers, C. D., Forstmann, B., & Pruszynski, J. A. (2017). Registered reports at the European Journal of Neuroscience: consolidating and extending peer-reviewed study pre-registration. European Journal of Neuroscience, 45(5), 627-628.

Chikovani, G., Focacci, M. N., Kienzle, W., Lechanoine, C., Levrat, B., Maglić, B., ... & Grieder, P. (1967). Evidence for a two-peak structure in the A 2 meson. *Physics Letters B*, *25*(1), 44-47.

Cichy, R. M., Pantazis, D., & Oliva, A. (2014). Resolving human object recognition in space and time. Nature neuroscience, 17(3), 455-462.

Creswell, J., Von Hausegger, S., Jackson, A. D., Liu, H., & Naselsky, P. (2017). On the time lags of the LIGO signals. Journal of Cosmology and Astroparticle Physics, 2017(08), 013.

Dekking, F. M., Kraaikamp, C., Lopuhaä, H. P., & Meester, L. E. (2005). A Modern Introduction to Probability and Statistics: Understanding why and how. Springer Science & Business Media.

Deshpande, G., Li, Z., Santhanam, P., Coles, C. D., Lynch, M. E., Hamann, S., & Hu, X. (2010). Recursive cluster elimination based support vector machine for disease state prediction using resting state functional and effective brain connectivity. PloS one, 5(12), e14277.

Domingos, P. (2012). A few useful things to know about machine learning. Communications of the ACM, 55(10), 78-87.

Dorigo, T. (2015). "Extraordinary claims: the 0.000029% solution", EPJ Web of Conferences 95, 02003.

Dwork, C., Feldman, V., Hardt, M., Pitassi, T., Reingold, O., & Roth, A. (2015). The reusable holdout: Preserving validity in adaptive data analysis. Science, 349(6248), 636-638.

Eklund, A., Nichols, T., Andersson, M., & Knutsson, H. (2015, April). Empirically investigating the statistical validity of SPM, FSL and AFNI for single subject fMRI analysis. In *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on* (pp. 1376-1380). IEEE.

Fahrenfort, J. J., van Driel, J., van Gaal, S., & Olivers, C. N. L. (2018). From ERPs to MVPA Using the Amsterdam Decoding and Modeling Toolbox (ADAM). Frontiers in Neuroscience. Retrieved from https://www.frontiersin.org/article/10.3389/fnins.2018.00368

Flint, C., Cearns, M., Opel, N., Redlich, R., Mehler, D., Emden, D., ... & Krug, A. (2019). Systematic Overestimation of Machine Learning Performance in Neuroimaging Studies of Depression. arXiv preprint arXiv:1912.06686.

Freedman, D. A.(1983). A note on screening regression equations. *the American Statistician*, 37(2), 152-155.

Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. Neural computation, 4(1), 1-58.

Harrison, P.F. (2002). “Blind Analysis”, J. Phys. G: Nucl. Part. Phys. 28, 2679-2691.

Klein, J. R., & Roodman, A. (2005). Blind analysis in nuclear and particle physics. *Annu. Rev. Nucl. Part. Sci.*, *55*, 141-163.

King, Jean-Rémi, Alexandre Gramfort, Aaron Schurger, Lionel Naccache, and Stanislas Dehaene. "Two distinct dynamic modes subtend the detection of unexpected sounds." PloS one 9, no. 1 (2014): e85791.

King, J. R., & Dehaene, S. (2014). Characterizing the dynamics of mental representations: the temporal generalization method. *Trends in cognitive sciences*, 18(4), 203-210.

Kriegeskorte, N., Simmons, W. K., Bellgowan, P. S., & Baker, C. I. (2009). Circular analysis in systems neuroscience: the dangers of double dipping. *Nature neuroscience*, *12*(5), 535-540.

Kilner, J. M. (2013). Bias in a common EEG and MEG statistical analysis and how to avoid it. Clinical Neurophysiology, 124(10), 2062–3. http://doi.org/10.1016/j.clinph.2013.03.024.

Lorca-Puls, D. L., Gajardo-Vidal, A., White, J., Seghier, M. L., Leff, A. P., Green, D. W., ... & Price, C. J. (2018). The impact of sample size on the reproducibility of voxel-based lesion-deficit mappings. Neuropsychologia, 115, 101-111.

Markoff (2015) Baidu Fires Researcher Tied to Contest Disqualification [Web log post], retrieved from http://bits.blogs.nytimes.com/2015/06/11/baidu-fires-researcher-tied-to-contest-disqualification/

New Scientist, October 2018: https://www.newscientist.com/article/mg24032022-600-exclusive-grave-doubts-over-ligos-discovery-of-gravitational-waves/

Ng, A. Y. (1997, July). Preventing" overfitting" of cross-validation data. In ICML (Vol. 97, pp. 245-253).

Nichols, T. E., & Holmes, A. P. (2002). Nonparametric permutation tests for functional neuroimaging: a primer with examples. *Human brain mapping*, 15(1), 1-25.

Nosek, B. A., Ebersole, C. R., DeHaven, A. C., & Mellor, D. T. (2018). The preregistration revolution. Proceedings of the National Academy of Sciences, 115(11), 2600-2606.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel,P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau,M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *JMLR*, 12:2825–2830, 2011.

Pernet, C. R., Latinus, M., Nichols, T. E., & Rousselet, G. A. (2015). Cluster-based computational methods for mass univariate analyses of event-related brain potentials/fields: A simulation study. *Journal of neuroscience methods*, *250*, 85-93.Poldrack, R. A., Huckins, G., & Varoquaux, G. (2020). Establishment of best practices for evidence for prediction: a review. JAMA psychiatry, 77(5), 534-540.

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology: Undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science*, 22(11), 1359-1366.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. Journal of the royal statistical society. Series B (Methodological), 111-147.

Treder, M. S. (2020). MVPA-Light: A Classification and Regression Toolbox for Multi-Dimensional Data. Frontiers in Neuroscience. Retrieved from https://www.frontiersin.org/article/10.3389/fnins.2020.00289

Varoquaux, G., Raamana, P. R., Engemann, D. A., Hoyos-Idrobo, A., Schwartz, Y., & Thirion, B. (2017). Assessing and tuning brain decoders: cross-validation, caveats, and guidelines. NeuroImage, 145, 166-179.

Varoquaux, G. (2018). Cross-validation failure: small sample sizes lead to large error bars. Neuroimage, 180, 68-77.

Vul, E., Harris, C., Winkielman, P., & Pashler, H. (2009). Puzzlingly high correlations in fMRI studies of emotion, personality, and social cognition. *Perspectives on psychological science*, *4*(3), 274-290.

TABLES

Table 1. The terminology used in this and other papers are defined in this table.

Term	Definition
Machine Learning	Machine learning is the use of semi-automated fitting algorithms to discern patterns in data. Typically, machine learning algorithms are trained with labeled data from two or more classes, and are then used to predict which class a new and unlabeled data item belongs to. Examples of machine learning algorithms are support vector machine (SVM) classifiers, random forest models, and naive Bayes classifiers.
Cross-Validation	A technique commonly used to evaluate classification performance that repeatedly divides the data into two subsets (each division is a ***fold***), one of which is used to train a classifier, the other being used to test it. Performance is taken as the average across all folds. See the supplemental for a more thorough description.
Training and Testing sets	These terms generally refer to the two subsets of data used during cross-validation. However, the term test-set is sometimes used to refer to data that has been set-aside for later evaluation. We advise against that usage for the sake of consistency.
Nested Cross-Validation	Nested cross-validation is a generalization of cross-validation in which the data are now partitioned into N outer sets/ folds. Each of these folds provides an outer hold-out set, and an inner set, on one outer cycle. On each such outer cycle, cross-validation is performed on the inner set. The benefit of the nested approach is that is provides a reliable assessment of

	overfitting of hyperparameters. See the supplemental for a more thorough description.
Lock box	We introduce the term lock box to mean a subset of data that are removed from the analysis pipeline at the very start of optimization and not accessed until all hyperparameter adjustments and training have been completed.
Hyperparameter	Hyperparameters are a kind of parameter whose values are adjusted either by hand or by algorithms to improve model performance (e.g. weights of electrodes, regions of interest, SVM model kernel functions, classification model types). They are distinct from other parameters, whose values are set during classifier training (e.g. the linear function that results from training a least squares model or the classification function that results from training a Support Vector Machine classifier).