A New Approach To Study The Challenges Of E-Learning Advantages And Disadvantage

Mohammed Alkaragole, Sulaiman M. Karim, Saadaldeen Rashid Ahmed

1 IRAQI MINISTRY OF EDUCATION, AL-KARKH FIRST DIRECTORATES. 2 KARABUK UNIVERSITY, COMPUTER ENGINEERING. 3 TIKRIT UNIVERSITY, COMPUTER SCIENCE.

Email: 1 Mohammed.alkaragole@gmail.com, 2 sulaiman_musaria@yahoo.com, 3 Saadaljanabi78@gmail.com

Abstract. In this paper, we have a tendency to tried to explore the challenges in implication of E-Learning and its benefits and downsides. There are some problems for not implementing E-Learning as there's some lacking's of understanding concerning E-Learning to the teacher additionally the scholars and have some issue that bothers to just accept E-Learning instead of ancient learning surroundings, and our main target is to seek out the explanation behind at purpose there's some opinion concerning the challenges for not implementing E-Learning. E-Learning surroundings is principally an online bases platform in academic section mistreatment technology, the aspects of E-Learning surroundings is mistreatment digital tools in courses of study, it presents the ways, tasks, and therefore the connections between a course structure and introduced new assessment in numerous stages. The aim of this study is to spot what challenges or that factors are the explanation for not applying E-Learning normally additionally it'll concentrate on what challenges lecturers and what issues students suffers in learning through virtual policy separately by this it'll helps to seek out the views of E-Learning of lecturers and student each. currently a day's E-Learning is known in academic sites. E-Learning surroundings is principally student targeted. It’s essentially virtual resource area, wherever student should get to be liable for their own work, because it may be a self-format study therefore student should have privy to their responsibility. I attempted to explore the challenges in implication E-Learning, because the outcome is affirmative there are some problems for not implementing E-Learning as there's some lacking's of understanding concerning E-Learning to the teacher additionally the scholars and have some issue that bothers to just accept E-Learning instead of ancient learning surroundings, and my main target is to seek out the explanation behind at purpose there's some opinion concerning the challenges for not implementing E-Learning.

Keywords: E-learning, Speech Recognition, Virtual classroom environment, web platform, Digital Learning, Higher Education.

1. Introduction

The field of E-Learning has changed quite a bit in the past couple of decades. While it was difficult in the past to gather enough data on some characteristic to create reliable automatic classification [1]. Todays processing speed and memory capacities as well as the global connectivity via the Internet allow for large databases available for use. However, often only the part of the data which will be used to predict the characteristic is abundant [2], which is called features. The other part, in case of a discrete characteristic going by the name class label, has to be assigned by an annotator. This can result in errors and be costly, especially if the annotator is a human. Using the field of speech
recognition as an example, spoken words are easy to obtain from videos, radio or phone calls, what words exactly were spoken is usually unknown [3]. To be able to use the data, a human would have to listen to it and write the correct words down, which is a boring and time-consuming, but necessary, work. Due to this, a class of algorithms called active learners has formed to help minimize the work necessary. As a rule of thumb, the more annotated data is used for the classification, the better it will perform, i.e. less misclassifications occur. But it is important which part of the data has been selected, as some instances, the combination of features and class label, are more informative as others. Thus, annotating the useful instances first and the rest later or not at all minimizes the impairment of the classifier’s performance for equal effort [4]. In order to stop the annotation when it is no longer sensible or wanted, we need some way of estimating the current performance. Traditionally, this is done one of three ways: either some part of the annotated data is set aside for so-called holdout testing, which is unwanted since it wastes precious annotated data, some kind of partitioning to create artificial test sets is done, or the classifier’s performance in the past in conjunction with a suitable function model is used to extrapolate to the current situation. The second option struggles with systematic deviation of their estimations from the true performance, while the third heavily depends on the model chosen as well as a number of already present estimates [5].

1.1. Aim of Study

The aim of this study is to spot what challenges or that factors square measure the rationale for not applying E-Learning generally additionally it'll specialise in what challenges lecturers and what issues students suffers in learning through virtual policy severally by this it'll helps to seek out the views of E-Learning of lecturers and student each. Moreover, this study can verify what square measure the directions lecturers and students counsel for implementing E-Learning. we'll investigate the properties of ways combining partitioning in addition as info regarding the educational method. For this, subsets square measure created from the information used for coaching. These square measures then seen as individual, smaller coaching sets and build the premise of classifiers themselves, with the remaining information serving as a take a look at set. This ends up in variety of performance estimations for coaching sets of varied sizes, which might be seen as tagged instances for a unique learning downside and square measure accustomed learn a model predicting the performance development of the first classifier. Applying the model to this classifier state represented by its coaching set size then ends up in a performance prediction.

2. BACKGROUND

In recent studies concerning E-Learning, they struggle to research the present scenario of this net platform. For this a number of the man of science provide their opinion concerning E-Learning within the English Departments in King Khalid University within the Kingdom of Saudi Arabia(KSA) they realize that some college boy realize E-learning is healthier and simpler as on-line quizzes is a lot of useful then assignments and a few of the scholar get Diamond State driven as several of them misuse the platform doing plagiarism [6]. Internationally E-Learning is one among the most important topics of analysis in pedagogy. In Arabian Higher instructional establishments normally and in King Khalid University (KKU) particularly is creating substantial strides. the rein case of E-learning lecturers have to be compelled to be aware and additionally verifyi to assist to search out out necessary information from mistreatment technology .Also it's aforementioned in the alternative facet as E-Learning is use for date assortment from completely different tools ;here observation, questionnaires, interviews it's undoubtedly produce an enormous assessment in instructional sector. Yes, E learning is incredibly useful in several stages in our trendy life and additionally in higher education; the talk isn't clear however concerning its disadvantage and advantage [9]. the advantages of E-Learning square measure plain and there square measure such a big amount of papers concerning its advantage and varied aspects of E-Learning, [7] states that there square measure several positive aspects of E-learning, that it is simply monitored than ancient schoolroom.

3. METHODOLOGY

The methods evaluated in this this research work build on both approaches. Generalizing the extrapolating approach, estimating the performance of a classifier based on individual estimates is a
type of learning itself, falling into the category of regression since performance is not discreet. But
instead of using the classifier’s journey, the combined estimator simulates different training sets that
could have been an earlier state of the classifier,
i.e. the subsets of the current training set. To help illustrating this method, the following scenario is
assumed: a dataset composed of both labeled and unlabeled data
\[ D = L \cup U \] with \( L = \{ (-x_i, y_i) : i \in \{1, \ldots, n\} \} \) and \( U = \{ (-x_i, :) : i \in \{n + 1, n + 2, \ldots\} \} \), \( y \in \{0, 1\} \),
serves as the basis for an active learner, which selects a training set \( X \subseteq L \) of size \( k \leq n \). This set is
used to train a classifier \( c_X: \sim x \mapsto y \). We now would like to know the accuracy \( acc_c(X, D) \) the
classifier has on the entire dataset.

### 3.1. Performance estimation on training sub-sets

In order to obtain the performance estimates for the classifier trained on training sub-sets, leave-p-out
(LPO) cross-validation with \( p \in \{1, \ldots, k - 1\} \) is used. As \( p \) is the number of omitted and thus test
instances, the training subsets. \( S_j^k \) \( X \) are of size \( jS_j^k \) \( j = j \) and \( i = f1; \ldots; \) \( kg \), resulting in \( k \) \( p \) accuracy
estimates for each subset size. The corresponding test sets are \( T_j^k = X nS_j^k \) \( p \). While LPO is a
pessimistic estimator.

| Symbol | Description |
|--------|-------------|
| \( D \) | Whole dataset |
| \( U \) | Unlabelled part of \( D \) |
| \( L \) | Labelled part of \( D \) |
| \( X \) | Current training set |
| \( k \) | Size of \( X \): \( k = jX_j \) |
| \( ~x_i \) | Instance of \( X \): \( \sim x_i \) \( 2 X \) \( \sim i = f1; \ldots; \) \( kg \) |
| \( y_i \) | Class label of instance \( \sim x_i \) |
| \( c_X \) | Classifier trained with set \( X \) |
| \( acc_c(A; B) \) | True accuracy of classifier \( c \) trained with set \( A \) on the instances \( B \) |
| \( c(A; B) \) | Estimate of \( acc_c(A; B) \) |
| \( acc S_i \) | Selected subsets of \( X \) from capped sub-sampling |
| \( S \) | Set of paths \((n\)-tuples of size \( k \) \( 1)\) |
| \( \sim S \) | Test sets \( X n S \) for estimation |
| \( I; I; I \) | Set of tuples containing a subset estimate and the subset’s size; input of the fitting algorithm |
| \( Y_m \) | Function fitted using \( Y_m \) depending on \( j \) as the classifier’s training set size |
| \( f^m(j) \) | Holdout test sets |
| \( X^k \) | Set containing all training sets of size \( k \) |
| \( K \) | Kernel for KDE |
| \( h \) | Bandwidth for KDE |
When applied to a classifier with training set size $k$, i.e. it systematically places the accuracy lower than its true value, it is unbiased for a classifier with a training set of size $k_p$ [2]. This leaves us with $2k$ $2$ estimates in total, each of them unbiased, but at the cost of exponential time complexity; some sort of selection may be reasonable to reduce it.

Another option with regard to the estimation is bootstrapping. It offers a little more variety, as different types like naive, leave-one-out and .632 are available. Also, individual estimates are more expensive to compute, as multiple bootstrap samples are first created. Also, there is no real equivalent to LPO for bootstrapping, meaning that the estimation looks slightly different: for leave-one-out (LOO) bootstrapping, an estimate for a subset $S_j$ and a corresponding bootstrap sample is actually the average of $j$ estimates, holding out each of the $j$ instances once and testing the resulting classifier on the hold-out instance. That is how bootstrapping was defined originally, anyway; however, the instances in $T_i^j$ not used for training would be left out. As they are not part of the training, they may as well be used as a test set. Problematic is that .632 bootstrap relies on a ratio of expected pessimism and expected optimism of LOO’s estimate and the classifier’s training error respectively. This is based on the probability of one instance not occurring in $n$ draws with replacement, i.e. how likely it is that a bootstrap sample does not contain a given instance. Since the instances in $X S_j^i$ cannot possibly occur in the bootstrap sample, which is drawn from $S_j^i$, the bootstrap estimate should be less pessimistic, and the ratio may not hold anymore.

### 3.1.1. Sub-sampling of fitting points

Regardless of which estimation technique is used, the complexity still scales exponentially with the training set size $k$. To reduce the amount of estimates for the model fitting process, some sort of selection has to occur. In this section, three sub-sampling strategies are explored. However, it is to be kept in mind that reducing the information available naturally has some drawbacks, including an expected higher variance and, if not done properly, an added bias. Also, the sampling may influence the fitting itself, potentially inflicting additional penalties to the robustness.

A simple, yet effective method which will be called capped sub-sampling is to cap the number of estimates. Possible options are to either impose a fixed, hard cap for all training set sizes $k$, or to use a polynomial dependent on it, e.g. $k^2$. A potential pitfall is the selection of the remaining subsets: selecting either randomly over all possible subsets or from pools for each subset size $k_p$ with sizes proportional to $k$ prevents unintentional importance assignment to subset sizes. The computation of this reduced set $S^- \subseteq S$ is illustrated in 1, the corresponding test sets $T_j^j = X \backslash S^-j$.

```
1: $X = \{(x_1, y_1), ..., (x_k, y_k)\}$ . Current training set
2: numOfPaths ← ...
3:
4: $S \leftarrow powerset(X)$
5:
6: $\tilde{S} \leftarrow$ ...
7: for $i \leftarrow 1$ to NUM_OF_PATHS do
8:   currPath $\leftarrow nTuple(k)$
9:   for $j \leftarrow 1$ to $k \leftarrow 1$ do
10:      currPath $\leftarrow drawRandom(S_j)$ . Get a random subset of size $j$
11:   end for
12:   currPath $\leftarrow i$ . The same path may get drawn multiple times,
13:   $\tilde{S} \leftarrow \tilde{S} \cup currPath$ . but it still has to be in the set
14: end for
```

Algorithm 1: Pseudocode for capped sub-sampling
A related approach exports the computational cost to the fitting process. Instead of selecting multiple subsets $S_j$ per size once and using them as a basis for one model, this sub-sampling creates multiple models and selects only one subset $S_j$ per $j$. Formally, we have a tuple $S^i = (s_1, ..., s_{k-1})$ with $i = 1, ..., r$ and $s_j \in S_j$, which will be called a path. Their respective test sets are $T^i = (X \setminus s_1, ..., X \setminus s_{k-1})$. The $S_j$ can be drawn with or without replacement, although the latter may lead to a lower variance, as seeing the same constellation multiple times does not add information, whereas a different one does. The parameter $r$ is up to choosing, with an upper limit of $k-1$ as the number of combinations for drawing with replacement; the algorithm for this is depicted in 2 and goes by the name of path sub-sampling. This would result in a higher complexity than exponential, namely $O(k!)$. However, accounting for all possible subset combinations may not be necessary, as there are far less unique combinations of accuracy estimations. This is due to the number of test instances available for a given training subset. For example, a classifier trained on a set of size one tested against a set of size three will have four potential test outcomes: either one, two, three or none instances were correctly classified, resulting in an estimated accuracy of 1, 2, 1 and 0, respectively. As a growth in size of the training set in turn causes a reduction in size of the test set, the amount of potential accuracies $\text{acc ce}(S_ji ; T_ji)$ shrinks from $k$ to 2 for $j = \{1, ..., k-1\}$. Thus, the number of unique combinations would shrink to $k!$.

Since the actual realization of the accuracies is not yet known, they would have to be computed first. This means that this approach is inept to reduce the number of needed subsets, but the idea of using multiple models instead of only one is still valid; it may increase the spread of the final estimate $\text{acc ce}(X, D)$ due to the reduced information available for each model, but also allows to derive a distribution for the accuracy. Following this, there are other possible restrictions with regard to the selection of the paths $S^i$. As it is, randomly taking subsets of each size allows for instances present in $S_j$ to not be selected for the larger sets $S_{j+1}, S_{j+2}, ...$. However, a classifier trained by an active learner does not usually discard previously selected instances. Thus, it seems logical to restrict subsets of larger size to be supersets of their predecessors $S_j \subseteq S_{j+1}$: This again reduces the possible combinations to $k!$ as the sampling from $X$ is now done without replacement. This time, however, the estimates for every subset are not needed because each path is equally likely, meaning that this sub-sampling is applicable even without precomputing the estimates. This assumption only holds in general for random sampling; other active learners may show preferences for some instances which, in their eyes, improves the classifier’s performance the most. Due to its restriction of the possible paths this method will be referenced as path-superset sub-sampling.

3.2. Combining sub-estimates with curve fitting
The function model used has to be capable of modeling the learning process and also largely determines the spectrum of algorithms available. For a linear model or one that can be linearized, e.g. the 2-parameter exponential law, the computation of the parameters which minimize the squared error is well known and trivial. More complex algorithms are necessary for functions which cannot be linearized, like the 3-parameter exponential law. Then, iterative methods have to be used, like the Levenberg-Marquardt algorithm. It works by iteratively adapting the parameters following its approximated gradient, with the goal to find a minimum for the least squares error function. While they are able to handle a larger number of functions, they also need to be provided with various tuning parameters. In the case of Levenberg-Marquardt, initial values and maximal change per parameter as well as the partial derivatives w.r.t. the parameters must be given. Also, convergence is not guaranteed; unlike linear least squares, where minimizing parameters exist for at least two data points with different x components, poorly chosen initial parameters or too few iterations may lead to divergence. Potentially even more detrimental are local minima, as a divergence may be recognized. Here, the derivative of the error function is zero, indicating a minimal, but different ones with lower absolute error values exist. However, the algorithm has no way of detecting this; the only options to avoid such a, quite literal, pitfall is to try the fitting with multiple initial parameters, hoping to get lucky, or to exploit previous knowledge about the data. Potential function models were examined in section, especially in. A good candidate for least squares fitting seems to be the 3-parameter exponential law in

\[ f_E(x) = a + b \cdot e^{c \cdot x} \]

Unfortunately, it falls in the category” non-linearizable” and requires iterative fitting. A similar function class are sigmoid. The evaluation investigates the characteristics of the sigmoid in comparison to While it was not part of the testing in the cited articles, it can be similar in shape thanks to the e exponential part and has the advantage of semantic parameters, that means they communicate the function’s shape without the need to draw it. In this case, y0 indicates the y-intercept, S is the asymptotic threshold, and m communicates the function’s slope. This way, it is easier to find appropriate bounds for the parameters during fitting: clearly, a learning curve has to have both y-intercept and asymptote between 0 and 1 as well as a slope larger or equal to 0. While similar parameters can be found for the exponential function, they are not as precise, leaving more room for potentially wrong guessing, especially for the initial parameters. As overfitting is a real possibility for all kinds of learning, the third function tested is the simple linear model of form

\[ f_L(x) = a + b \cdot x \]

### 3.3. Tuning the model fitting

Usually, data gathered for curve fitting is not uniform, e.g. some data points are more likely to be tainted with error or do not carry much information. An example could be
Figure 2: The shape of the three function models fit on example estimates produced by averaged grouping with $k = 9$ for the seeds dataset.

A similar situation presents itself when using averaged grouping, where the subset estimates $\text{acc}(S_j, T_j)$ for each $j$ are first averaged, then sent to the fitting algorithm. However, this procedure omits how many estimates were originally present, which is quite important information: more estimates mean the resulting average should be less noisy. This is where statistical weights come into play: they inform the fitting algorithm about how important a low error of the fitted function for a given data point is, the difference in shape for an example exponential function is depicted in the weighted curve is flatter since the more extreme points at $x = 1$ and 6 have less influence. Assigning weights proportional to the amount of estimates that went into each $\gamma_{mi}$ may ensure that less noisy data influences the shape of the curve more heavily. Another potential improvement for the fitting process is the addition of data. Of course, the number of subsets is fixed without purchasing more labels. But information not covered by leave-p-out may prove useful: showcased in .632+ bootstrapping uses the no-information rate to identify special overfit cases with data independent of its predictors. It is a heuristic used to approximate the error a classifier without any training would make when being tested on the dataset, the formula can be found in

![Figure 3](image-url)

Figure 3: Difference of function shape for the sigmoid model when using weighting or the no-information rate.

And while performance estimates for each subset size $S_j \{1, ..., k-1\}$ are available, the estimates do not yet include an estimate for $j = 0$, since most classifiers require at least one training instance to work. The no-information rate may be used as the 0th fitting point and usually slightly lifts the beginning of the function curve, filling that gap and help primarily in scenarios with small $k$. For larger training sets, its influence should decrease as more regular estimates are available.

4. RESULTS

All tests are conducted with three different active learners: random sampling, uncertainty sampling and PAL; which one is indicated in the graphics. For their functioning. Although it does not matter due to the dichotomous nature of the used datasets, uncertainty sampling uses the maximum entropy for instance selection. The choice of an adequate classifier is mostly limited by the active learners and the datasets used in the evaluation. From the dataset side, it has to be able to accept continuous features. The output of class assignment probability is necessitated by PAL and uncertainty sampling. Also, PAL requires some sort of density estimation; to keep the comparability of both active learners up, a classifier making use of kernel density estimation (KDE) is reasonable. Considering this, Parzen window lends itself to be the classifier of choice. It is a non-parametric classifier directly build on KDE. The assumption is that the data is distributed according to some probability distribution; basic assumption is the normal distribution. Then, the kernel $K$, which in our case is the probability density function (but it does not have to be), is approximated at a given instance $\sim x$ using the following formula:
As the training set, \( X \) contains the already known instances and \( h \) is the so-called bandwidth, a smoothing factor for the kernel. It has to be estimated, which is usually done by applying a function called Silverman’s rule of thumb, which is dependent on \( n \), the dimensionality of the data, its estimated standard deviation and some statistical properties of the kernel largely irrelevant to this work. The classifier used in the evaluation however simply assumed a standard deviation of 0.1 across all dimensions. If the data in question is multivariate, \( h \) has to be a bit modified; \( K \) is then usually the product of the kernel for each dimension, same goes for the bandwidth. The Parzen window classifier estimates the densities of an instance to be labeled for each class label, each time using only the instances with the same label as \( \sim x \). Then, they are multiplied with the prior class probabilities, i.e. the share each class label has among the labeled instances. Normalization then results in the wanted (estimated) class probabilities, with the largest probability dictating the resulting class, shows the classification of a Parzen window classifier: the color expresses the predicted class label, while the \( z \) coordinate equals the kernel density with either one positive or two negative instances as its base, depending on which one is larger.

4.1. Datasets
The datasets used should both be realistic and cover most uses; a method that does well on specifically designed test sets but fails in the real world is only interesting as a proof-of-concept. Secondly, PAL was only defined for dichotomous data. Although an extension on multiple-class problems should be possible, I did not want to tamper with the formulas, instead restricting the datasets to binary-class problems. Considering these constraints the selection contains the following datasets:

\[
\hat{f}_h(x, X) = \frac{1}{h \cdot |X|} \sum_{\substack{x \in X \ni X \neq x}} K(\frac{x - x}{h})
\]

Figure 4: The estimated kernel density for a grid with one positive and two negative instances; lower \( Z \) value indicates lower certainty for the class assignment.
A real-world dataset using eight features associated with predict the age of abalones. Originally, the number of rings (and thus ages) of a specimen were predicted. To create a binary set, all ages below 10 form one class, while the rest forms the other.

**Figure 5:** Visualizations of the datasets checke1, 2dData, seeds and a downsized version of abalone.

Figure 6 shows the two-dimensional visualization, while Figure 5 illustrates the rough learning curve of a Parzen window classifier with random sampling of each dataset.

**Figure 6:** Visualizations of the datasets checke1, 2dData, seeds and a downsized version of abalone.
Due to the sheer amount of data, this section contains only the most expressive graphs. To help keeping an overview, the evaluation is structured into the following parts: first, we take a look at the mean error for the traditional and unweighted estimators using the exponential function model. After that follows a survey on whether the different models for fitting improve the bias or not, as well as what influence statistical weighting has. Then comes the comparison of the mean squared error for some selected estimators as well as the Kullback-Leibler divergence, for those it is available for. An analysis of the computation time necessary for each method finalizes the evaluation.

![Graphs showing mean errors for different active learners and datasets using the exponential model. The darker colored bars mark the errors of later learning stages.](image)

**Figure 7:** Average mean errors for the different active learners and datasets using the exponential model. The darker colored bars mark the errors of later learning stages.

The error spread is quite a bit lower on the priority list than the estimation bias. To keep the evaluation within a reasonable frame, Each bar holds the error of one estimator, while its subsections indicate the share each learning stage brings to the total mean. Similarly to the estimation bias, both non-random active learners tend to come with an increase in spread.
Figure 8: Average squared errors for the different active learners and datasets with the respective share of each learning stage.

Figure 9: Left: Average computation times for the estimators. Right: Histogram of computation time for PathSuper.

The figure 8 and 9 shows the average squared errors for the different active learners and Left: Average computation times for the estimators and this is to be expected, as a high error implies a high squared
error. For the most part, .632+ BS has the lowest spread, while averaged, averagedBS and 5-Fold CV mark the upper end. With random sampling, the spread drops for larger k for all estimators, as it is to be expected: more training instances mean more subsets, which increases the amount of accuracy estimates available for the model fitting. This is not the case for uncertainty sampling, as the additional estimates are obtained from more instances at a (likely) noisy decision boundary.

5. CONCLUSION
In this research paper, the properties of four performance estimators utilizing estimation on training subsets, model fitting and subsequent extrapolation to approximate the accuracy of a classifier in the context of active learning. For this, the methods path and pathSuper, which simulating the classifier’s training process, as well as averaged and averagedBS, which use leave-p-out cross-validation, were evaluated on different datasets and active learners with regard to their estimation bias, spread and computation time. Each of these estimators was tested with an exponential, sigmoid, and linear function model as basis for the extrapolation, broken down by learning stage. Additionally, the effect of enhancements to the fitting process for some selected estimators were studied. To compare them to state-of-the-art methods, 5-fold cross-validation and .632+ bootstrapping also participated in the evaluation.

The results show that a distinction between random and non-random active learning has to be made. Some of the estimators tested are definitely viable for random sampling, namely averagedBS with the linear, averaged with the sigmoid and pathSuper with the exponential model. This is especially true for a classifier trained with a set of instances in the size range of 3 to 7. While it is dependent on the dataset, they show lower biases than .632+ BS and 5-fold CV, but slightly elevated estimation spread. For larger training sets, however, the traditional bootstrapping is the reference. For both uncertainty sampling and PAL, none of the estimators, including traditional cross-validation and bootstrapping, are suitable. Depending on how well PAL can abuse the data structure, the estimates are more or less pessimistic. This is caused by holding out instances from the training set; it reduces the classifier’s information too much, since the instance was likely to be the only one in the specific area. A similar problem has to be faced with uncertainty sampling: instances at decision boundaries tend to be noisy and of mixed labels; a classifier predicting a noisy instance’s label is likely to be wrong, which lowers the accuracy estimate. The addition of statistical weights to the model fitting process showed mixed results. Its effectiveness is largely dependent on the function model itself, the estimator and the learning stage. It reduces the bias of averagedBS with the linear model for training set sizes between three and seven by 20%. The bias of the same classifier for larger training sets rises, however. Another addition, an estimate for a completely untrained classifier with the name no-information rate, only increased the bias in the tests. All of the non-traditional estimators require heavy amounts of computing, caused by their exponential complexity. The computation time is mostly independent of the dataset but varies largely due to the iterative nature of the model fitting. For future work, I suggest taking a look at different weights for the fitting and why the function models are affected so differently. It may also be of interest to investigate how many estimates are used for the linear model; as it stands, those of the largest four subsets are the only ones. Less could better reflect the current accuracy gradient, but also cause more instability.

6. REFERENCES
[1] Stansfield, M.; Connolly, T.; Cartelli, A.; Jimoyiannis, A.; Magalhaes, H.; Maillet, K. The Identification of Key Issues in the Development of Sustainable e-Learning and Virtual Campus Initiatives. Electron. J. e-Learn. 2009, 7, 155–164.
[2] Rovai, A.P.; Downey, J.R. Why Some Distance Education Programs Fail while Others Succeed in a Global Environment. Internet High. Educ. 2010, 13, 141–147.
[3] Chipere, N. A framework for developing sustainable e-learning programmes. Open Learn. J. Open Distance e-Learn. 2017, 32, 36–55.
[4] Aparicio, M.; Bacao, F.; Oliveira, T. An e-Learning Theoretical Framework. J. Educ. Technol. Soc. 2016, 19, 292–307.
[5] Garrison, D.R. Blended Learning a Transformative Design Approach. In Encyclopedia of Distance Learning; Rogers, P.L., Ed.; Information Science Reference: Hershey, PA, USA, 2009; pp. 200–204. ISBN 978-1-60566-198-8.

[6] Sangrà, A.; Vlachopoulos, D.; Cabrera, N. Building an inclusive definition of e-learning: An approach to the conceptual framework. Int. Rev. Res. Open Distrib. Learn. 2012, 13, 145–159.

[7] Rodrigues, H.; Almeida, F.; Figueiredo, V.; Lopes, S.L. Tracking e-learning through published papers: A systematic review. Comput. Educ. 2019, 136, 87–98.

[8] Zare, M.; Pahl, C.; Rahnama, H.; Nilashi, M.; Mardani, A.; Ibrahim, O.; Ahmadi, H. Multi-criteria decision making approach in E-learning: A systematic review and classification. Appl. Soft Comput. 2016, 45, 108–128.

[9] Dron, J.; Anderson, T. The future of e-learning. In The Sage Handbook of E-Learning Research; Haythornthwaite, C.A., Ed.; SAGE Reference: Los Angeles, CA, USA, 2016; pp. 537–556. ISBN 978-1-4739-0232-9.

[10] Njenga, J.K.; Fourie, L.C.H. The myths about e-learning in higher education. Br. J. Educ. Technol. 2016, 41, 199–212.