Communications of the
Association for Information Systems

Accepted Manuscript

How to Conduct Rigorous Supervised Machine Learning in Information Systems Research: The Supervised Machine Learning Reportcard

Niklas Kühl
Karlsruhe Institute of Technology (KIT) / IBM
niklas.kuehl@kit.edu

Robin Hirt
Karlsruhe Institute of Technology (KIT) / prenode

Lucas Baier
Karlsruhe Institute of Technology (KIT)

Björn Schmitz
Karlsruhe Institute of Technology (KIT) / IBM

Gerhard Satzger
Karlsruhe Institute of Technology (KIT) / IBM

Please cite this article as: Kühl, Niklas; Hirt, Robin; Baier, Lucas; Schmitz, Björn; Satzger, Gerhard: How to Conduct Rigorous Supervised Machine Learning in Information Systems Research: The Supervised Machine Learning Reportcard, *Communications of the Association for Information Systems* (forthcoming), In Press.

Communications of the Association for Information Systems

Research Paper ISSN: 1529-3181

How to Conduct Rigorous Supervised Machine Learning in Information Systems Research: The Supervised Machine Learning Reportcard

Niklas Kühl
Karlsruhe Institute of Technology (KIT) / IBM
niklas.kuehl@kit.edu

Robin Hirt
Karlsruhe Institute of Technology (KIT) / prenode

Lucas Baier
Karlsruhe Institute of Technology (KIT)

Björn Schmitz
Karlsruhe Institute of Technology (KIT) / IBM

Gerhard Satzger
Karlsruhe Institute of Technology (KIT) / IBM

Abstract:

Within the last decade, the application of supervised machine learning (SML) has become increasingly popular in the field of information systems (IS) research. Although the choices among different data preprocessing techniques, as well as different algorithms and their individual implementations, are fundamental building blocks of SML results, their documentation—and therefore reproducibility—is inconsistent across published IS research papers.

This may be quite understandable, since the goals and motivations for SML applications vary and since the field has been rapidly evolving within IS. For the IS research community, however, this poses a big challenge, because even with full access to the data neither a complete evaluation of the SML approaches nor a replication of the research results is possible.

Therefore, this article aims to provide the IS community with guidelines for comprehensively and rigorously conducting, as well as documenting, SML research: First, we review the literature concerning steps and SML process frameworks to extract relevant problem characteristics and relevant choices to be made in the application of SML. Second, we integrate these into a comprehensive "Supervised Machine Learning Reportcard (SMLR)" as an artifact to be used in future SML endeavors. Third, we apply this reportcard to a set of 121 relevant articles published in renowned IS outlets between 2010 and 2018 and demonstrate how and where the documentation of current IS research articles can be improved. Thus, this work should contribute to a more complete and rigorous application and documentation of SML approaches, thereby enabling a deeper evaluation and reproducibility / replication of results in IS research.

Keywords: Supervised machine learning, Research documentation, Research replication, Methodological framework

[Department statements, if appropriate, will be added by the editors. Teaching cases and panel reports will have a statement, which is also added by the editors.]

[Note: this page has no footnotes.]

This manuscript underwent [editorial/peer] review. It was received xx/xx/20xx and was with the authors for XX months for XX revisions. [firstname lastname] served as Associate Editor.] **or** The Associate Editor chose to remain anonymous.]

1 Introduction

Replication of published research is an important endeavor in the academic world. Replication studies repeat previously conducted studies with the goal to investigate whether the findings are reliable—and to what extent they can be generalized. Over the last decade, a lack of these methodologically important supplements have constituted the so-called "replication crisis" —reflecting that many scientific studies and their results are in fact difficult or even impossible to replicate. So far, this replication crisis has particularly been proclaimed in the fields of medicine and psychology (Schooler, 2014; Tackett et al., 2019).

While Information Systems (IS) research has started to actively incentivizing replication studies (Olbrich et al., 2017; Weinhardt et al., 2019), the rise of methods from Machine Learning in IS entail new challenges in replication (Coiera et al., 2018; Hutson, 2018). Especially supervised machine learning (SML) is gaining increasing popularity in the field: Between 2010 and 2018, 35 contributions published in *Management Information Systems Quarterly* (*MISQ*), *Information Systems Research (ISR)* and *Journal of Management Information Systems (JMIS)* apply SML in their research. In addition, the number of publications in typical IS conferences (*European Conference on Information Systems (ECIS), International Conference on Information Systems (ICIS)*) that rely on SML as a key method is also steadily growing over time.

While SML is enjoying widespread popularity and promises considerable potential in IS research, there is room for improvement when it comes to rigorously applying these technologies: Many IS research articles lack a thorough documentation of the SML process and the results obtained, which makes it challenging or virtually impossible to reproduce or replicate their results. Naturally, researchers may prefer discussing the implications of SML results instead of stringently documenting the SML process itself. This, however, will contribute to spread the replication crisis described above also in the IS research community, as it is neither possible to follow or replicate the precise choices of the research nor to judge whether its results are indeed meaningful. We set out to address this problem, and develop and test a documentation standard ultimately enabling frequent replication of SML studies in IS. To this end, we first review the literature to identify the typical problem characteristics and choices to be made in SML endeavors. On this basis, we develop a "Supervised Machine Learning Reportcard (SMLR)" to provide guidelines for comprehensively and rigorously conducting and documenting SML research. We review the literature concerning extant steps and SML process frameworks and integrate them into a comprehensive reportcard. Finally, we review 121 relevant articles, which were published from 2010 to 2018 in renowned IS outlets, such as *Management Information Systems Quarterly* (*MISQ*), *Information Systems Research* (*ISR*) and *Journal of Management Information Systems (JMIS)* and the proceedings of the *International Conference on Information Systems (ICIS)* and the *European Conference on Information Systems (ECIS).* We use this broad sample to analyze how and where the SML documentation of current articles could be improved. This article therefore contributes to a complete and rigorous application and documentation of SML research, which promotes meaningful and reproducible results.

The remainder of this article is structured as follows: We introduce the fundamentals and positioning in the upcoming Section 2. Then, we derive and describe the problem characteristics and key choices of each SML endeavor in Section 3, followed by the introduction of the Supervised Machine Learning Reportcard (SMLR) addressing them. In Section 4, we apply this reportcard in an empirical study to relevant IS articles and analyze their precision when it comes to SML application and documentation. In Section 5, we conclude with recommendations, a summary and limitations of the study.

2 Fundamentals and Positioning

When it comes to their type of learning, machine learning techniques can be classified as either supervised or unsupervised ones[1] (Mohri et al., 2013). In fact, most real-word applications of machine learning are of supervised nature (Jordan & Mitchell, 2015), whereby SML aims to predict the (discrete or continuous) value of an element by using a data set of observations in which this element is already known and labelled with the correct value (Rätsch, 2004). Precisely, we define supervised machine learning as follows—based on Mohri et al. (2013, p.5): *Supervised machine learning is the concept of learning a function mapping an input to an output based on labelled training data, i.e. a sample of input-output pairs.* For discrete target values, the problem is called a classification problem, for example, when determining product returns in e-commerce

[1] Other sources, for example, Fu (2003), also consider reinforcement learning as a third type. However, there is no academic consensus on this definitory classification.

(Heilig et al., 2016). In contrast, predictions of continuous variables, such as forecasts of electricity prices (Feuerriegel & Fehrer, 2016), are subsumed as regression problems. Here, the output of the SML algorithm is not a class, but a numerical value that specifies the predicted attribute.

An SML endeavor, i.e. the application of SML methods to a problem, may serve different purposes and its specific design heavily depends on the particular problem. Shmueli & Koppius (2011) differentiate these purposes in either *explaining* or *predicting* a phenomenon. Regarding the first, statistical models can support explanatory-oriented research for testing causal hypotheses. For instance, if a researcher aims at *explaining* patterns in the data with a linear regression, individual model results (like the loading of the regression coefficients, the coefficient of determination R^2, or p-values) might already fully warrant applying the model; there is no further need to evaluate its predictive power on an unseen test or validation set for possible deployment within information systems artifacts (Gong et al. (2017); Z. Li et al. (2016); Martens & Provost (2014)).

On the other hand, *predictive* models can be used to anticipate unseen or future observations. In order to do so, researchers need to analyze SML's potential to solve an empirical prediction problem. Thus, they need to show its effectiveness in their field studies by reporting on the predictive qualities of a trained model. Researchers might compare an SML endeavor to different benchmarks and, consequently, not only show its basic functionality, but also the efficiency of leveraging SML for a certain, possibly productive task (Pant & Srinivasan, 2010). For instance, they may analyze whether a machine can perform a task better than a human (H. Han et al., 2015). Depending on the scope, this step may even require to implement a predictive model and embed it into a software tool, for example, to continuously make predictions (Oroszi & Ruhland, 2010). The focus of our work is on SML applications for *predictive* purposes.

When discussing replicability or reproducibility of SML studies for predictive purposes, we need to distinguish different possible levels of documentation. The spectrum of reproducibility originally developed by Peng (2011) for the field of computer science, is well applicable to our IS SML endeavors. On that basis, Figure 1 denotes the range of options that increasingly allow reproduction of results: While mere results in a publication do not support any reproducibility, the exposure of method details, code and/or data will help to do so. He argues for the publication of "linked and executable code and data" along with the core article as a gold standard to assure reproducibility.

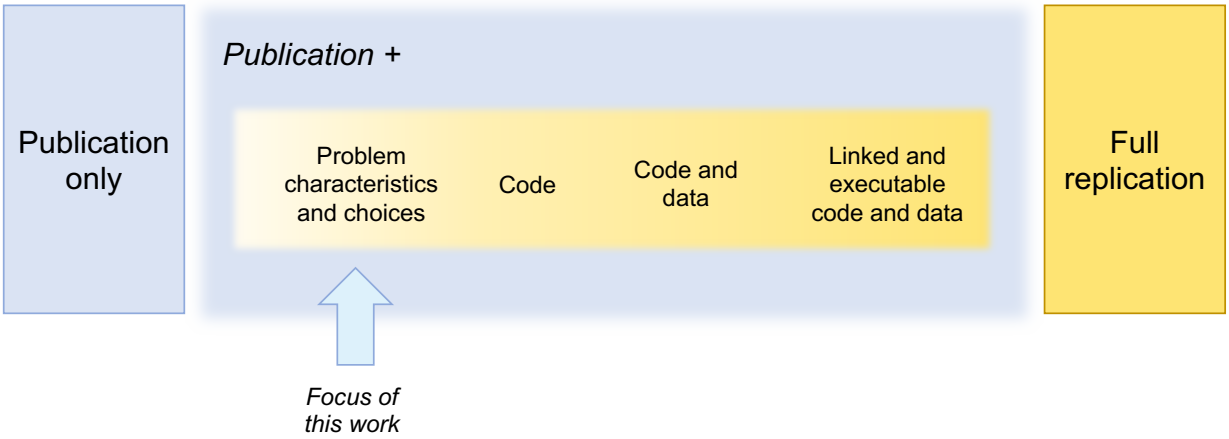

Figure 1. The spectrum of reproducibility; extended figure based on Peng (2011)

However, typical IS studies cannot comply with a publication of code and/or data due to confidentiality issues (Gimpel et al., 2018; Sharp & Babb, 2018; Timmerman & Bronselaer, 2019), at least if not publicly available data sources are used. For the work at hand, we will, therefore, primarily focus on the documentation of the problem characteristics and choices of applying SML—but still stress the importance of providing code and data whenever possible.

When it comes to process models that support SML for predictive tasks, a variety of different possibilities exist—the most common being Knowledge Discovery in Databases (KDD) (Fayyad et al., 1996), Cross-Industry Standard Process for Data Mining (CRISP-DM) (Chapman et al., 2000) and Microsoft Team Data Science Process (Microsoft, 2020). Although these process models are extremely popular, they are very broad and do not go deep enough to derive measurable criteria for SML endeavors. As they are designed for more general data mining and machine learning purposes, they are (by design) not detailed and lack

helpfulness and transparency for our purpose. The same shortcoming of high-level abstraction applies to other, less popular process models (Anand & Büchner, 1998; Brodley & Smyth, 1995; Cabena et al., 1998; Cios et al., 2000; Witten et al., 2011). Since these process models are highly generic and can be applied to any kind of data analysis projects—and not SML exclusively—they only focus on a limited part of the overall choices and problem characteristics (Kurgan & Musilek, 2006). Furthermore, they do not include precise guidelines for the performance estimation and deployment of an SML endeavor, which are especially important in IS (Shmueli & Koppius, 2011). A process model is also not suitable for communicating results in a scientific publication.

In this article, we therefore derive problem characteristics and key choices as part of the SMLR; every SML endeavor needs to consider and document them to enable readers and reviewers to fully grasp and judge the individual project—also for replication studies of machine learning in IS research (Hutson, 2018; Olorisade et al., 2017; Voets et al., 2018). Similarly to the proposed reportcard for IS research, related "checklists" were proposed in other disciplines—with the idea to append them when submitting a manuscript to a conference or journal. A number of articles originate from the field of medicine and aim to educate physicians the application of machine learning (Mongan et al., 2020; Pineau, 2020; Qiao, 2019; Winkler-Schwartz et al., 2019). While these articles share some problem characteristics and choices with IS research, their main goal is to map them to the specific needs of a clinical audience. In the field of CS, three main articles are important: Pineau (2020) proposes a short checklist to foster reproducibility in general machine learning endeavors. He emphasizes precise descriptions in the areas of models, theory, data, code and results, e.g., to include clear README files. In the area of Natural Language Processing (NLP), Dodge et al. (2019) stress aspects of result reporting and especially hyperparameter tuning. To allow for more realistic results, they propose that researchers utilize their novel technique of *expected validation performance*. Furthermore, they elaborate on the documentation of the used hardware. While hardware is an important metric in CS to estimate runtimes and complexities of machine learning models (Dodge et al., 2019; Pineau, 2020), these aspects play a minor role in the reproducibility of the more application-oriented IS—and will be neglected in the remainder of this work. Mitchell et al. (2019) present a "model card" with a focus on fairness and ethics of machine learning models, as they conclude fairness and bias topics are not (yet) integrated into the minds of data scientists. Apart from CS and with a strong focus on the industrial sector, Studer et al. (2020) propose an adapted version of CRISP-DM for the application of machine learning in the automotive sector with a checklist on specific quality assessment measures. In contrast to these related checklists, our proposed SMLR a) focusses on the holistic SML process from problem statement to productive deployment, b) details the necessary problem characteristics of specifically SML (and not ML in general) and c) presents the findings with an IS audience in mind. Where appropriate, we will highlight where insights from other articles influenced the design of our presented SMLR.

3 Towards Rigorous Supervised Machine Learning Documentation

The results of the literature review confirm that so far no process model systematically captures all the problem characteristics to be reported and choices to be made in SML projects in the field of IS. Thus, we set out to collect and merge the necessary problem characteristics and key choices from various sources: We gather individual parts of the entire process from relevant literature and augment other parts based on logical reasoning and best practices gained from the execution of typical SML projects.

3.1 Problem Characteristics and Key Choices of Supervised Machine Learning

For the subsequent analysis, we further divide an SML endeavor into the following three main steps: model initiation, model performance estimation, and, if applicable, model deployment (Hirt et al., 2017)—as illustrated in Figure 2. In the model initiation step, the objectives for the endeavor are formulated and the matching data set is gathered, prepared, and characterized. Having initiated a model, its performance will be estimated by training and testing models on a data set D in which the target to be predicted is known. First, models learn patterns in the data from a training subset $D_{Tr} \subseteq D$ and then apply it towards a test set $D_{Te} = D / D_{Tr}$ of the data, which was not used for training. Cross-validation approaches are applied to perform this with various alternative D_{Tr}/D_{Te} splits.

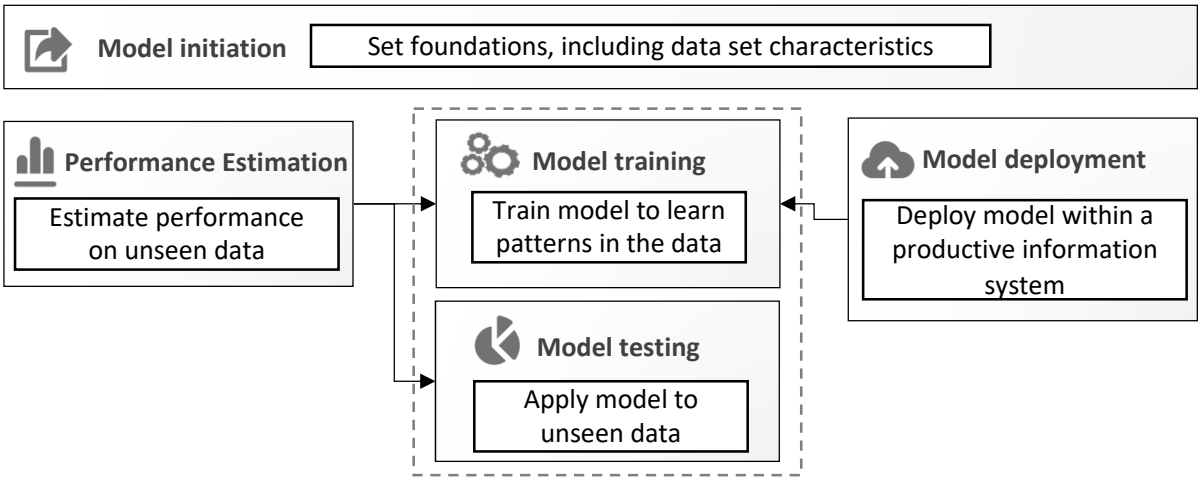

Figure 2. Overview of Supervised Machine Learning Steps

When conducting SML endeavors, it is important to specify problem characteristics (e.g., class distribution) and elaborate on the choices made (e.g., performance measure). Additionally, it is necessary to state these key insights when publishing the results, because only with this context information can the reader judge the endeavor's rigor and meaningfulness. For instance, if the author does not specify if hyperparameter optimization was used in the SML process, it is difficult to verify whether the models' performance could be further improved or if the author has simply accepted the performance of the first best tuple of hyperparameters (Dodge et al., 2019).

As previously explained, the goal of the endeavor needs to be precisely defined: It should show the purpose and the targeted application (Mongan et al., 2020). The necessary activities of initiation and performance estimation are linked to the first two, while model deployment is also important when implementation is the goal. Model performance estimation aims to estimate a model's performance on unseen data based on a set D_{Te} of data for which the feature to be predicted is known. This is a typical SML step across all disciplines which leverage it, for example, medicine (Shipp et al., 2002) or physics (Rupp et al., 2013). However, when conducting an SML endeavor in IS, not only performance estimation is an inherent step, but also model deployment. This implementation within a productive software tool continuously exposes the model to new, incoming data (Shmueli & Koppius, 2011). While model performance estimation builds on both training and testing activities, model deployment only leverages the training to create a deployable model. For instance, within a model performance estimation not all data can be used for model training, as a certain share needs to be saved for validation and/or testing purposes. For model deployment, however, it is important to use as much data as is available—because more data enables the model to achieve better performances (Banko & Brill, 2001). Therefore, after estimating the model performance, the final model is built by using all available data D in the model deployment phase.

3.1.1 Model Initiation

When conducting SML, a model needs to be defined. A model can be considered as a tuple of parameters that describe which algorithm is used, how its parameters are initiated, and what the general process is like. These basic assumptions and surrounding conditions are defined in the model initiation. They serve as the basis for the subsequent model building, for model evaluation (as part of performance estimation), as well as for model deployment.

First, it is important to state the problem which the SML endeavor aims to address (Qiao, 2019). This requires specifying a target value and the SML problem type—for instance, binary / multi-class classification or regression problems. It should be clear from the start what the problem type is ("What should be solved") (Chapman et al., 2000). Next, the different aspects of the data used and its characteristics are important to estimate the complexity of the task and also to enable meaningful judgement of the final results at a later point. This starts with the data gathering and precise definitions on how it is performed (Oquendo et al., 2012; Winkler-Schwartz et al., 2019). SML requires a target value, which can either be collected together with the data or it can be separately labelled (automatically or manually) after the collection. In any event, it needs to be explained if and how the labelling takes place. If the volume of the data is too large to be

analyzed, it is possible to conduct a *sampling*[2], which pulls a representative subset of the larger data set (Dhar et al., 2014). Especially in recent years, the process of sampling has not only been relevant to retrieve a representative data set, but also a fair one without any biases (Barocas et al., 2017). With a data set to analyze, additional problem characteristics and key choices need to be specified. The data distribution is of major importance, since it ultimately determines the interpretation of the results (He & Ma, 2013). For instance, in a binary classification on a data set with a minority class distribution of 10%, an accuracy of 90% is easily achievable by simply predicting all observations as belonging to the majority class. This is, furthermore, also a question of the performance metric, which we address at a later point. Irrespective of the performance metric, however, the number of classes and their shares need to be specifically mentioned for every classification problem (e.g., as a table). The same applies to regression problems (e.g., a representation as a boxplot) to enable the reader to understand the basic problem. Furthermore, it is important if and which data preprocessing methods are applied—for any type of data. For instance, in the specific case of natural language processing (NLP), the possibilities of transforming unstructured text data into structured, machine-digestible formats are manifold (Manning & Schütze, 2000). We, therefore, need to specify which transformation techniques are applied and why they are applied for a specific problem. Apart from the preprocessing, statements about the data quality are of interest. Data quality covers many aspects, including correctness ("is it true?"), accuracy ("how precise?"), completeness ("is it complete?") and relevance ("is it related to the initial problem?") (R. Y. Wang et al., 1993). Sparsity and noise are two examples of data quality characteristics—and there are a number of different complexity measures available to assess them (Ho & Basu, 2002).

3.1.2 Model Training and Testing

Training and testing are essential parts of each machine learning endeavor. However, the purpose of these activities needs to be clearly defined: We particularly distinguish between estimating the model's performance on unseen data (Section 3.3) and deploying a model within a software tool (Section 3.4).

In the model training phase, the sampling of data, which occurs prior to training a model, can have a significant impact on the performance (Chawla, 2005). Popular sampling techniques for dealing with uneven class sizes are undersampling, oversampling or Synthetic Minority Over-sampling Technique (SMOTE). *Undersampling* is applied when the number of random sample instances taken from the majority of observations is limited to match the size of the minority data set used for training purposes (Rahman & Davis, 2013). In contrast, *oversampling* randomly duplicates instances from the minority class so that researchers can work with more instances than originally available (Rahman & Davis, 2013). *SMOTE* creates new additional synthetic instances to match the number of training set elements in the majority class (Chawla et al., 2002).

The core of the model training phase consists of selecting an algorithm, as well as its parameters, which creates another set of choices. For instance, popular machine learning frameworks like the python-based "scikit-learn" (Pedregosa & Varoquaux, 2011) and the Java-based "WEKA" (Hall et al., 2009) feature more than sixty, respectively, thirty supervised learning algorithm implementations. SML algorithms can be classified in different ways (Caruana & Niculescu-Mizil, 2006; Hastie et al., 2009; Kotsiantis, 2007). Aggarwal and Zhai (2012) divide supervised algorithms into the major classes of linear algorithms (e.g., Support Vector Machines or regressions), decision trees, pattern (rule-)based algorithms, probabilistic and Naive Bayes algorithms, and meta-algorithms. Each of these classes has its advantages and disadvantages—in general, as well as in relation to the specific data and problem they are applied to. While we cannot go into the details of each class, Kotsiantis (2007) provides more details on the particular selection criteria.

When it comes to model testing, it is important to early define one or multiple performance metrics, which serve as the central criteria to estimate alternative models' performance and to finally evaluate the success of the SML endeavor. Common metrics used for classification tasks are, for instance, accuracy, precision, sensitivity, specificity, recall, F-measure or AUC (Powers, 2011). Metrics for regression tasks, on the other hand, include mean squared error (MSE), R^2, correlation coefficient (CC), normalized root mean squared error (NRMSE), signal-noise ratio (SNR), coefficient of determination (COD), as well as global deviation (GD) (Spuler et al., 2015). When it comes to choosing one or multiple metrics, it is again important to

[2] It should be noted that when it comes to machine learning, the term sampling can be used in three different scenarios with different objectives: It can be used to pull representative data as part of data gathering (as described above), it can be used in the distribution of data for a fold as part of the cross-validation (stratified sampling), or it can be used to counterbalance a minority class as part of the model training set (e.g., oversampling).

consider the nature of the problem, as well as the data set. For instance, although recall is a valuable metric to present the fraction of relevant observations among the retrieved observations, it is not meaningful on its own, since it can easily be brought to 100% by simply predicting all observations as belonging to the positive class. The inherent tradeoff between precision and recall is designed into the set of F-metrics (Goutte & Gaussier, 2005). In the case of regression, R^2 and explained variance are popular choices. Additionally, for both regression and classification, the plotting of a learning curve can be meaningful, because it can show the training and test set errors for each fold of the cross-validation and the respective amounts of data, which helps estimate the bias-variance tradeoff (Blanc, 2016).

3.1.3 Performance Estimation

Based on the performance estimation it is possible to draw conclusions on how the trained model performs on unseen data. In order to do so, it leverages the previously described steps of training and testing. The important step to conduct is splitting the data set to allow for these two activities. There are two different options when it comes to data splitting, namely percentage split and cross-validation (Abdullah et al., 2011). A simple split into a (larger) training set and a (smaller) test set is called a percentage split. The machine learning model is trained on the training set and then applied to the test set for evaluation. In IS research, data is often precious with a limited amount of available observations. Therefore, the prediction performance on the test set may vary significantly in the case of a percentage split, because, depending on which instances are present in the training set, it may or may not be trained as "well" (James et al., 2013). Generally, the error resulting from this prediction can be divided into bias, variance, and irreducible error (Friedman, 1997). In order to counteract the random effect of choosing data for the sets, a k-fold cross-validation can be implemented. Here, the original data is divided into k folds of equal size. The model is trained with (k-1) folds (training set) and applied on the remaining fold, called validation set or local test set. This process is repeated k times with each of the k folds. The aggregated performances from the individual iterations are averaged and represent a more meaningful performance assessment than a single percentage split (Golub et al., 1979). For both cases, percentage split and cross-validation, stratified sampling allows for maintaining the original data set's distribution within the training and test set (Neyman, 1934), which reduces the randomness associated with allocating the two subsets.

If the goal is to simply demonstrate the capabilities of one machine learning model, one-time splits, such as percentage or k-fold, can be sufficient. If, however, the plan is to try out different models, optimize parameters, and estimate the error of a model on unseen data, additional steps should be undertaken. If any optimization takes place, it is important to test the model on completely unseen data—that is, data, which has never been used in any training or optimization iteration (Cawley & Talbot, 2010). A so-called hold-out set or global test set should never be used to change models or the choice of them, but preferably only to evaluate them once (Tušar et al., 2017). In order to address this, the nested cross-validation first splits the data into training/validation set and a hold-out set. Then, cross-validation with parameter optimization can be applied within an inner cross-validation, thereby making it possible to select and evaluate—but not again optimize—the best performing models within the outer cross-validation. To summarize, when it comes to model performance estimation, separating the data into multiple sets is of importance and depends on the use case:

- Training set refers to the data set on which the model is trained.
- Validation set or local test set refers to the data set on which the model is optimized. It must, however, not be used to evaluate the model's performance, otherwise the model tends to overfit. A validation set is crucially important if parameter optimization is performed.
- Hold-out set or global test refers to the data set according to which the model is evaluated, but according to which it is never optimized.

3.1.4 Model Deployment

The final model deployment phase aims at generating, implementing, and distributing a previously built supervised machine learning model within a software tool. Data contains information and is valuable—therefore, using the complete data set is meaningful for the final machine learning as depicted in Figure 3 (Gama et al., 2004). It would incorporate parameters, which were typically previously selected from the performance estimation. These parameters also help in understanding the robustness of the model (i.e., its tendency for overfitting). For instance, analyzing the optimal parameters of the cross-validation's inner folds might reveal that a specific parameter combination occurs multiple times, or, if the model is very stable, all the time. This combination of parameters might then be directly used for the final training. Alternatively, an

additional cross-validation with the complete data set can be utilized to choose the parameters for final training.

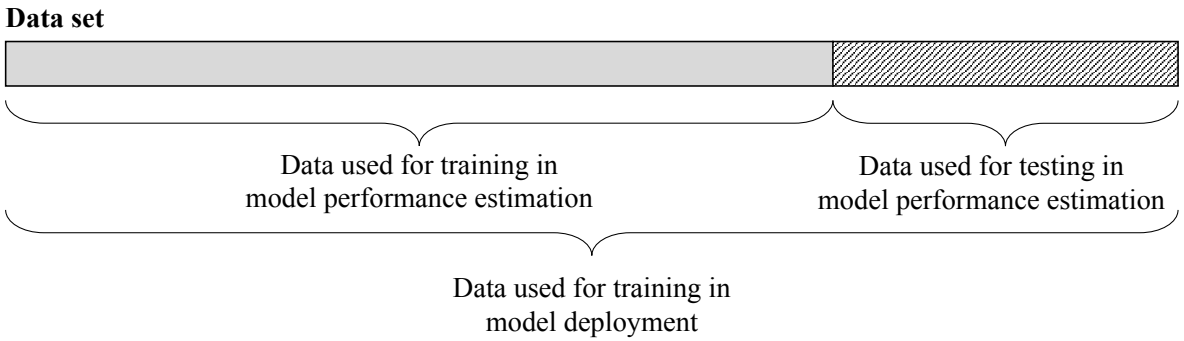

Figure 3. Data Sets for Training, Testing, and Final Deployment

Then, an export of the final model, also called serialization (Zaharia et al., 2018), is needed to save the state of the model and the used preprocessing pipeline for further usage. Having concluded the serialization phase, the serialized object can be built into a workflow, such as a connected web service, to predict the target value of new, incoming data. Hereby, data is sent to the serialized object to be preprocessed and classified by the model. It is important to consider the validity of this final model, for example, how robust is it to changes in the data (Gama et al., 2004) and/or whether its performance is continuously maintained (Feurer et al., 2015). Since the model building data might be topical at that point in time, the data might change in the future. It is important to address this, preferably directly by continuously updating the model automatically, or, at least, by (qualitatively) estimating the performance for future changes (Baier et al., 2019). For instance, in the case of sensor data in a production line, the predictive model might still be valid for a long time—as long as the produced goods remain the same. However, if elements of the production line change or new goods are produced, the model needs to be updated. In sum: It is important to address how the model copes with new, incoming data and, consequently, whether or not the model is continuously improved—and if not, why it is not necessary.

3.2 The Supervised Machine Learning Reportcard (SMLR)

For each step of an SML endeavor that were laid out in Section 3.1, we aim to identify key choices and problem characteristics to systematically capture and document them. In Figure 4, we present the Supervised Machine Learning Reportcard (SMLR), which allocates the identified problem characteristics and key choices alongside these steps. When conducting and describing a supervised machine learning endeavor, they should be addressed and defined.

During the model initiation phase, the problem statement itself is a key characteristic, which classifies the supervised machine learning problem as being either a binary, a multiclass or a regression problem. Since every supervised approach requires data, a detailed description of the data gathering process, as well as the construction of a ground truth data set, should be provided. In order to better understand the data itself, data distribution should be described, as well as the overall data quality, that is, for example, the sparsity and noise of the data. Depending on the distribution of classes, sampling of data points might be necessary and needs to be described by the authors (e.g., type of sampling). Lastly, data preprocessing (Kotsiantis, 2007), as well as feature engineering and vectorizing (Domingos, 2012), not only have a major influence on the overall performance of the trained model, but also bear the risk for major methodological mistakes, such as data leakage. It is important to consider different methods, as well as reasons for their usage.

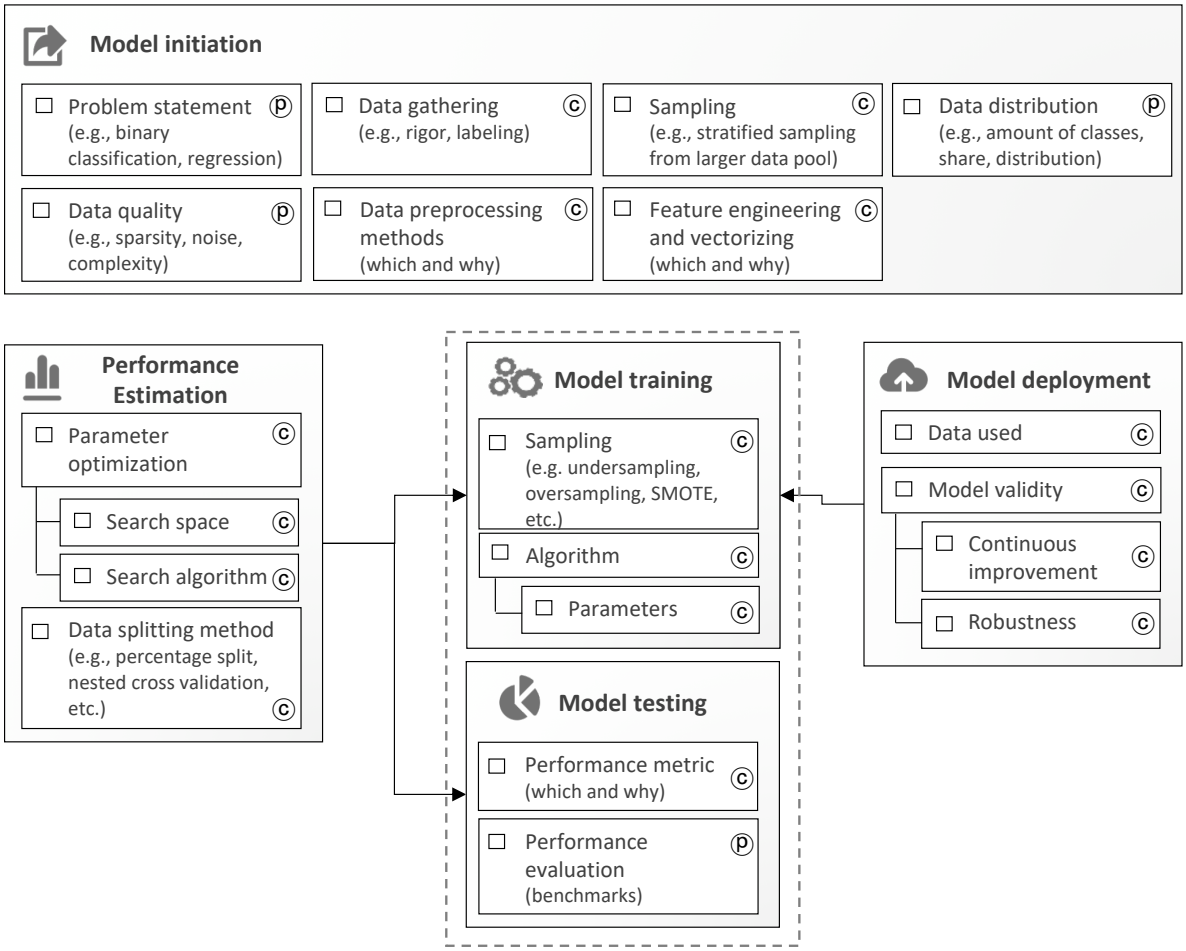

Figure 4. Overview of Supervised Machine Learning Steps and Corresponding Problem Characteristics and Key Choices

In the error estimation phase, the model's performance on unseen data should be estimated. Thus, information about the algorithm, the parameter search space, and the search algorithm (e.g., grid search, random search), as well as the data splitting method (e.g., percentage split, cross validation) needs to be specified. In the proposed reportcard, we list model training and testing as two separate units, which require thorough description. During the model training phase, data can be sampled to train a better prediction model. Furthermore, researchers should describe the algorithm that was used, as well as its implementation. This requirement goes beyond simply reporting the name of the approach. Especially for neural networks, researchers need to rigorously document the architecture of their model which for instance includes the type of network layers (e.g. convolutional, recurrent, or fully connected layers) applied and the number of neurons per layer. The choice of a suitable performance metric for a given problem is essential for the success of a supervised machine learning endeavor. Whereas accuracy might represent a model's performance well in a class-balanced scenario, its descriptive capability typically decreases when it comes to highly imbalanced data. Each performance metric has its advantages and disadvantages. It is advisable to either use multiple (e.g. Accuracy + Precision + Recall + AUC) or composed (e.g. F-score) metrics, as single metrics can be easily tuned and do not represent a holistic overview of the qualities of the predictive model. Furthermore, the results need to be contextualized according to a performance evaluation/benchmark. For instance, if the utilized data set has been used in other articles or even data science challenges like Kaggle, the performance results obtained from these works should serve as a benchmark for direct comparison. If such results are not available, obvious benchmarks should be referred to. These could be either naïve models (e.g., a random guess or the prediction of the majority class/mean from the training set) or simpler models

(e.g., a basic linear or logistic regression). By providing this context, the reader can better understand the quality of the obtained performance.

The performance of an estimated model can be used to show the effectiveness of a model. If it needs to be implemented for predictive modeling as part of the model deployment phase, the model is put into practice to solve the initial problem. In this scenario, the algorithm, as well as the previously identified parameters and sampling method should be used for model training. Furthermore, the data, which was used for training the final model, should be described. Since models can only represent a hypothesis based on training data, its validity decreases as the corresponding real-world situation changes. In order to address these changes, researchers should address the model validity and possible continuous improvement techniques, as well as the model's application to unseen data (robustness).

To ensure the completeness of our approach, we compare the characteristics and choices included in the reportcard with two widely used process models for data science projects, namely CRISP-DM (Chapman et al., 2000) and Microsoft Team Data Science Process (Microsoft, 2020). This analysis reveals that the reportcard in fact covers all important aspects of a machine learning endeavor. We can only determine a gap between the reportcard and the two process models regarding the documentation of requirements from the field as well as details on the business assessment. However, those two aspects usually do not apply to the academic context. A detailed comparison with CRISP-DM and with Microsoft Team Data Science Process can be found in the appendix in Table 4 and Table 5.

The first and foremost aim of this work in general and the SMLR in specific is to generate awareness for the identified problem characteristics and key choices when conducting SML. However, if applicable, it can be also utilized as a framework to document these precise choices. To demonstrate a possible application, we depict a typical machine learning challenge—using the Iris data set (Fisher, 1936)—and report on the results in Table 1.

Table 1. Exemplary Reportcard based on the Iris Data Set. Bold Writing indicates a Problem Characteristic or Choice from the Reportcard.

Problem statement	Predict iris flower class based on the four attributes four attributes *Petal Length, Petal Width, Sepal Length, Sepal width*		
Data gathering	Pre-defined data set by scikit-learn package for Python (Pedregosa & Varoquaux, 2011), originating from Fisher (1936)		
Data distribution	Three flower classes *setosa*, *versicolor*, *virginica* with 50 instances each; 150 instances in total		
Sampling	No sampling		
Data quality	No missing values		
Data preprocessing methods	No preprocessing		
Feature engineering and vectorizing	No additional features apart from *Petal Length, Petal Width, Sepal Length, Sepal width*, no		
Performance estimation			
Parameter Optimization	Yes		
	Search Space	RBF kernel	$\gamma \in \{0.001;0.0001\}$ $C \in \{1;10;100;1000\}$
		linear kernel	$C \in \{1;10;100;1000\}$
	Search Algorithm	Grid Search	
Data split	Nested cross-validation, 3 outer folds, 5 inner folds		
Algorithm	Support Vector Classifier		
Sampling	No sampling		
Performance metric	F_1-score as a compromise between precision and recall		
Performance evaluation	Average F_1-score performance on outer folds: 0.9778, which is a nearly perfect score		
Model deployment			
Data used	Full data set (150 instances)		
Model validity	**Continuous Improvement**	No continuous improvement	
	Robustness	No statement about the suitability possible	
Sampling	No sampling		
Algorithm	Support Vector Classifier		
	Parameters	RBF kernel	$\gamma = 0.001$ $C = 1000$

4 Empirical Study

With the SMLR at hand, we review renowned articles from IS literature to identify the strengths and possible improvements on the basis of the presented key choices and problem characteristics.

4.1 Methodology and Data Set

For our study, we aim at covering a broad range of high standard, high quality publications in IS. The JOURQUAL3 rating, which conducted a total of 64,113 journal and conference evaluations from 1,100 professors (VHB, 2012, 2019) serves as our basis. We focus on the top three journals and top two conference proceedings in the IS community (Hennig-Thurau et al., 2004), namely *Management Information Systems Quarterly* (*MISQ*), *Information Systems Research* (*ISR*), *Journal of Management Information Systems (JMIS)*, of as well as the proceedings of, respectively, the *International Conference on Information Systems (ICIS)* and the *European Conference on Information Systems (ECIS)*.

Table 2. Number of Screened and Relevant Articles for each Outlet from 2010 to 2018

	MISQ	ISR	JMIS	ICIS	ECIS	∑
Screened articles	288	463	390	3,118	2,257	6,516
Relevant articles	7	13	15	43	43	121
Binary classification	1	8	8	19	24	60
Multi-class classification	5	2	7	15	10	39
Regression	1	3	0	9	9	22

In order to obtain a meaningful number of articles for our study, we cover the time range from 2010 to 2018. In total, we download and screen 6,516 articles. Among those papers, we identify those articles where the application of SML plays a major role. Naturally, there are “borderline cases” where SML is only applied on a side note and documented within a few sentences or small paragraph—while the overall goal of the research article is of different nature and SML is not at the core of the project. To name a few examples: Huang, Boh, and Goh (2017) apply SML for an automated sentiment labeling, Walden et al. (2018) utilize SML for an aspect of their experiment analysis and Ivanov and Sharman (2018) merely apply SML in the appendix for a robustness check. For our study of rigor SML application, we exclude these cases as SML was not the designated *main* method for the respective articles. However, we want to stress that our proposed SMLR would be a meaningful addition to the documentation of these small applications, too: While researchers would not need to go into detail in the body of the text, they could just append the filled reportcard at the end of the article for the interested reader and replicant (see Table C1 in the appendix).

In a first step, we identify 121 full-research and research-in-progress articles, which describe an application of SML as detailed in Table 2. It is interesting to note how the importance of SML in IS developed over the years. In 2010, only six articles were published which applied SML in their research; in 2018, their number peaked with 30 research articles. More details on the chronological development in the distinct outlets are presented in Figure 5.

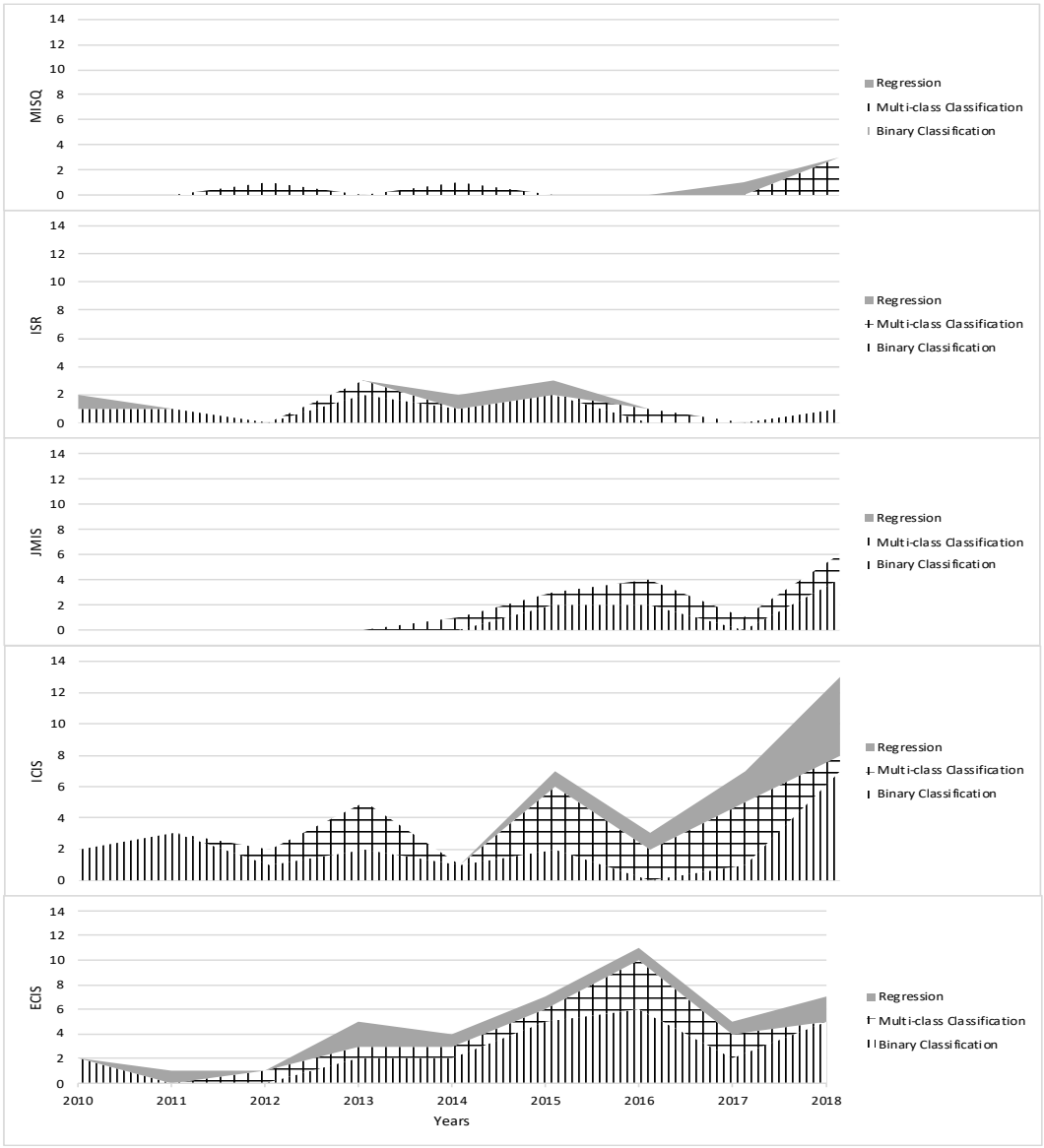

Figure 5. Amount of Supervised Machine Learning Articles in the Outlets of MISQ, ISR, JMIS, ICIS, and ECIS from 2010 to 2018

Next, we thoroughly examine all 121 articles across the entire time frame regarding the reportcard steps with their problem characteristics and key choices previously defined in Section 3. We distinguish between binary classification, multi-class classification, and regression problems (Chollet, 2018). The majority of SML-based articles (60) solves binary classification problems (e.g., Oh and Sheng 2011; Pant and Srinivasan 2010; Amrit, Wijnhoven, and Beckers 2015), followed by 39 articles with multi-class (e.g., Dorner and Alpers 2017; T. Wang et al. 2013; Geva and Oestreicher-Singer 2013) and 22 articles with regression problems (e.g., Riekert et al. 2017; Feuerriegel, Riedlinger, and Neumann 2014; Ding, Li, and Chatterjee 2015).

Next, we describe our findings with regard to the different steps of model initiation, performance estimation, and model deployment, which we have defined in Section 3. These findings are summarized in Table 3 and will be discussed in the following. Tables A1 and B1 in the appendices show the individual analyses for journals and conferences. It is important to note that we assess all publications according to the same, objective criteria. We do not consider whether each of the indicators is meaningful for the individual

publication; for example, it might not be necessary for a study on the feasibility of SML for a certain business challenge to deal with the necessary steps for deployment.

Table 3. Overview of Supervised Machine Learning Reportcard Steps and their Documentation

Step	Indicator		Described in articles			Positive Example
Model initiation	Problem statement		100.00%	(121/121)		Abbasi et al. 2012
	Data gathering		87.60%	(106/121)		Lin et al. 2017
	Data distribution		74.38%	(90/121)		Stange and Funk 2016
	Sampling		14.88%	(18/121)		Dhar et al. 2014
	Data quality		67.77%	(82/121)		Hopf et al. 2017
	Data preprocessing methods		76.03%	(92/121)		Johnson, Safadi, and Faraj 2015
	Feature engineering and vectorizing		74.38%	(90/121)		Martens and Provost 2014
Performance estimation	Parameter Optimization	Search Space	12.40%	(15/121)		Riekert et al. 2017
		Search Algorithm	13.22%	(16/121)		Zhou 2017
	Data split		95.87%	(116/121)		Urbanke, Uhlig, and Kranz 2017
	Algorithm		100.00%	(121/121)		Oh and Sheng 2011
	Sampling		6.61%	(8/121)		T. Wang et al. 2013
	Performance metric (reasoned)		48.76%	(59/121)		Fang et al. 2013
	Performance evaluation		48.76%	(59/121)		X. Han, Wang, and Huang 2017
Model deployment	Data used		1.65%	(2/121)		Abbasi et al. 2018
	Model validity	Continuous Improvement	1.65%	(2/121)		Seebach, Pahlke, and Beck 2011
		Robustness	15.70%	(19/121)		Koroleva and José Bolufé Röhler 2012

4.2 Model Initiation

Describing the data characteristics is a fundamental part of understanding the model that is built on top of it. At first, it is necessary to name the data source and/or the data collection process. In 12% (15/121) of all the reviewed articles, neither the data's origin, nor the source from where the authors have gathered it, is clearly stated. The quality of data determines the quality of the model; however, 32% (39/121) of the screened articles do not provide any information on data quality. 26% (31/121) of the articles do not describe the statistical distribution of the applied data set. This relates both to the distribution of the target variable and to the information about the attributes, which is used for prediction. When this information is lacking, it is impossible to judge the final model's performance for a given metric. Furthermore, if the distribution is unknown to the reader, the performance values can be meaningless—for example, a reported accuracy of 99% with a 1% minority class can already be achieved by simply assigning all instances to the majority class. A good example of a sound data description is provided in Bretschneider and Peters (2016) who refer to the total number of messages and the number of harassment messages (target variable), which is included in their dataset. Data preprocessing and the engineering of features are also essential choices during an SML endeavor. However, 13% (16/121) of the reviewed articles do not include any information about the preprocessing or feature engineering activities that are chosen. Yet, this information is very valuable to any researcher or practitioner who wants to build a predictive model in the same domain. For instance, if we do not know how quality issues, such as incomplete data, have been handled, the results may be flawed. Furthermore, it is impossible for other researchers to re-create results if the data's preprocessing techniques are omitted, because various different possibilities for preprocessing exist. Stange and Funk (2015) thoroughly explain how they transform real-time advertising data before feeding this data into the model training phase. Thereby, they enable others to benefit from their knowledge.

The performance assessment of a model highly depends on the chosen performance metric. This is therefore, a critically important decision for every SML endeavor. Due to the importance of this step, it is vital to specify the performance metric and the reason why it has been chosen. Nevertheless, only 49% (59/121) of all the reviewed articles actually give the reason for the choice of their evaluation metric. For

instance, Riekert et al. (2016) state that they apply accuracy as evaluation metric—however, they do not explain why this is the best suited (and meaningful) metric for their underlying problem.

4.3 Performance Estimation

In only 19% (23/121) of the reviewed articles authors mention the parameters, which they use in the model training phase. Model's performance can vary significantly depending on the chosen parameters and therefore the parameter space has to be thoroughly defined and described[3]. In fact, 96% (116/121) of all the reviewed articles include information about how they split the dataset into a training set and a test set (e.g., Lash and Zhao 2016; Urbanke, Uhlig, and Kranz 2017; Chatterjee et al. 2018). If authors do not disclose this information, the reader cannot judge whether induced results are truly rigorous, because it might even imply that they did not split their data at all. If model training and model testing are performed on the same dataset, the measured performance is misleading and unrealistically high (James et al., 2013).

In order to comprehensively understand a trained model's performance, it is important to compare it to previously built models—or other approaches that strive to solve the same problem. Thus, if any previous research or algorithm deals with the same problem or data set, the performance of the developed model should always be compared to the previous model. If there is no previous research, performance should be compared to other metrics, for example, random guesses (L. Li et al., 2013) or standard SML algorithms. Kozlovskiy et al. (2016) provide a good example by comparing their model's performance to a random guess. Only 49% (59/121) of the reviewed articles actually introduce a performance comparison (e.g., Cui, Wong, and Wan (2012); Geva and Oestreicher-Singer (2013); X. Han, Wang, and Huang (2017)). The remaining articles merely introduce the results of the predictive models without any comparison, in which case a reader can hardly judge the actual quality of the presented model.

4.4 Model Deployment

The articles in our study show the least reportcard compliance when it comes to the model deployment phase. As we pointed out earlier, the deployment phase is not a mandatory/necessary phase for each SML endeavor in IS research. In certain cases, authors may only want to prove the feasibility of an approach, which includes the application of SML. If a project focuses on this, it is not necessary to build a deployable solution and describe how this is best achieved. Nevertheless, only 26% (31/121) out of all the reviewed articles in our study at least describe the thoughts about a possible model deployment and the corresponding implications. This is only a small share of all screened articles, although IS, as a research discipline, should have a strong focus on producing final, implementable results and implications for practice (Gholami et al., 2016) . On the other hand, we also found some evidence for solutions, which were deployed (e.g. Schwaiger et al. 2017), including explanations on how the authors built their tool and which choices are necessary for deployment in an industry setting. However, even the examples which discuss the model deployment phase do not emphasize which data can be used for the final, to-be-deployed model.

Another consideration is model validity in general and model updates in particular (Baier et al., 2019; Studer et al., 2020). An SML model is built upon data. One assumes that underlying concepts in this data are extracted to fulfill a given task. If an SML model is then deployed, these concepts should not change over time; otherwise the model has to adapt to such “concept drifts” (Gama et al., 2004). If, for example, a model that classifies user-written texts on a social media platform according the age of their authors is not updated from time to time, the prediction quality will decrease—language (i.e., phrases used by certain age groups) will change over time. Thus, we claim that the preservation of model validity needs to be properly addressed.

5 Conclusion

Supervised Machine Learning (SML) has become a popular method to solve problems in Information Systems (IS) research and other disciplines. Although SML offers many possibilities for proving effectiveness, efficiency, and application in the problem spaces of predictive modeling, it is important to conduct this research in a rigorous and comprehensive manner. Only by doing so, IS researchers enable their peers to understand and reproduce the conducted research. In this article, we have developed a Supervised Machine Learning Reportcard (SMLR) capturing important key choices and problem characteristics, which need to be considered in every SML endeavor. We elaborate on them and their

[3] However, this does not apply to, for example, linear regression, since no parameter choice is required.

importance. In an empirical study, we use this reportcard to analyze whether recent articles published in renowned IS outlets already apply the necessary scrutiny in SML descriptions—and we identify several shortcomings in the documentation of SML. For instance, not all the reviewed articles justify the chosen performance metrics and only a minority of them uses benchmarks to help the reader understand the evaluation of the models.

The article at hand has two major limitations. First, we only review articles from five journals/proceedings and only consider instances from 2010 to 2018. While the selection is based on an acknowledged ranking (VHB, 2019), other rankings on important outlets obviously exist. As suggested by this ranking, we treat journal and conference publications alike, although journal publications are typically more mature and show longer histories of revisions. On the other hand, conference publications are timelier and a good indicator for upcoming topics and methods of the community. For the interested reader, however, we append differentiated analyses in the appendix. Regardless of rankings and precise outlets, the general message still remains that we can observe a lack of documentation. This lack can have two reasons: Either the identified key choices and problem characteristics were not considered, or they fell victim to shortening, for example, as part of the review process or the compliance to submission guidelines. Our study can, therefore, only analyze whether important key steps are addressed within the articles; our study does not allow for conclusions to be drawn on the actual research conducted.

The proposed SMLR may prove helpful in future SML endeavors and serve as a guideline to more rigorous, comprehensive research in this area. Once implemented, the reportcard will enable a more transparent view on SML articles—and enable their reproducibility in the future.

Appendix A: Results of SMLR Study for Journals

Table A1. Overview of Supervised Machine Learning Reportcard Steps, their Problem Characteristics and Choices as well as their Documentation in the Journal Publications Analyzed

Step	Indicator		Described in articles			Positive Example
Model initiation	Problem statement		100.00%	(35/35)		Abbasi et al. 2012
	Data gathering		88.57%	(31/35)		Lin et al. 2017
	Data distribution		82.86%	(29/35)		T. Wang et al. 2013
	Sampling		37.14%	(13/35)		Samtani et al. 2017
	Data quality		37.14%	(13/35)		Dong, Liao, and Zhang 2018
	Data preprocessing methods		71.43%	(25/35)		Pant and Srinivasan 2013
	Feature engineering and vectorizing		62.86%	(22/35)		Twyman et al. 2015
Performance estimation	Parameter Optimization	Search Space	8.57%	(3/35)		Martens et al. 2016
		Search Algorithm	11.43%	(4/35)		Martens et al. 2016
	Data split		94.29%	(33/35)		Lash and Zhao 2016
	Algorithm		100.00%	(35/35)		W. Li, Chen, and Nunamaker Jr 2016
	Sampling		5.71%	(2/35)		Kitchens et al. 2018
	Performance metric (reasoned)		42.86%	(15/35)		Shi et al. 2017
	Performance evaluation		68.57%	(24/35)		Cui, Wong, and Wan 2012
Model deployment	Data used		2.86%	(1/35)		Abbasi et al. 2018
	Model validity	Continuous Improvement	2.86%	(1/35)		Abbasi et al. 2018
		Robustness	8.57%	(3/35)		Mo, Sarkar, and Menon 2018

Appendix B: Results of SMLR Study for Conferences

Table B1. Overview of Supervised Machine Learning Reportcard Steps, their Problem Characteristics and Choices as well as their Documentation in the Conference Publications Analyzed

Step	Indicator		Described in articles			Positive Example
Model initiation	Problem statement		100.00%	(86/86)		Kowatsch and Maass 2018
	Data gathering		87.21%	(75/86)		Ram 2015
	Data distribution		70.93%	(61/86)		K.-Y. Huang, Nambisan, and Uzuner 2010
	Sampling		5.81%	(5/86)		Stange and Funk 2015
	Data quality		80.23%	(69/86)		Riekert et al. 2017
	Data preprocessing methods		77.91%	(67/86)		Pröllochs, Feuerriegel, and Neumann 2015
	Feature engineering and vectorizing		79.07%	(68/86)		Baumann et al. 2015
Performance estimation	Parameter Optimization	Search Space	13.95%	(12/86)		Tafti and Gal 2018
		Search Algorithm	13.95%	(12/86)		Staudt, Rausch, and Weinhardt 2018
	Data split		96.51%	(83/86)		Chatterjee et al. 2018
	Algorithm		100.00%	(86/86)		Tripathi and Kaur 2018
	Sampling		6.98%	(6/86)		Lüttenberg, Bartelheimer, and Beverungen 2018
	Performance metric (reasoned)		51.16%	(44/86)		Blanc and Setzer 2015
	Performance evaluation		40.70%	(35/86)		Geva and Oestreicher-Singer 2013
Model deployment	Data used		1.16%	(1/86)		Laing and Kühl 2018
	Model validity	Continuous Improvement	1.16%	(1/86)		Seebach, Pahlke, and Beck 2011
		Robustness	18.60%	(16/86)		Goby et al. 2016

Appendix C: Comparison to Data Science Processes

Table 4. Steps of CRISP-DM and equivalent of Reportcard

CRISP DM phases and tasks	Related reportcard choices / characteristics
Business understanding	
Determine business objectives	Model initation – Problem statement
Assess situation	n.a.
Determine data mining goals	Model initation – Problem statement
Produce project plan	n.a.
Data understanding	
Collect initial data	Model initiation – Data gathering
Describe data	Model initiation – Data distribution
Explore data	Model initiation – Data distribution
Verify data quality	Model initiation – Data quality
Data preparation	
Select data	Model initiation – Sampling
Clean data	Model initiation – Data quality
Construct data	Model initiation – Data preprocessing methods
Integrate data	Model initiation – Data gathering
Format data	Model initiation – Feature engineering and vectorizing
Modeling	
Select Modeling Technique	Model training – Algorithm
Generate Test Design	Performance estimation – Data Splitting method
Build Model	Model training – Algorithm/Performance Estimation – Parameter optimization
Assess Model	Model testing – Performance metric
Evaluation	
Evaluate results	Model testing – Performance evaluation (benchmarks)
Review process	n.a.
Determine next steps	n.a.
Deployment	
Plan deployment	Model deployment – Data used
Plan monitoring and maintenance	Model deployment – Model validity (continuous improvement / robustness)
Produce final report	n.a.
Review project	n.a.

Table 5. Lifecycle of Microsoft Team Data Science Process and equivalent of Reportcard

MTDSP stages	Related reportcard choices / characteristics
Business understanding	
Define objectives	Model initiation – Problem statement
Identify data sources	Model initiation – Data gathering
Data acquisition and understanding	
Ingest the data	Model initiation – Data gathering
Explore the data	Model initiation – Data distribution, Model initiation – Data quality
Set up a data pipeline	n.a.
Modeling	
Feature engineering	Model initiation – Data preprocessing methods, Model initiation – Feature engineering and vectorizing
Model training	Model training – Algorithm, Performance estimation – Data Splitting method, Model testing – Performance metric
Suitability for production	Model testing – Performance evaluation (benchmarks)
Deployment	
Operationalize a model	Model deployment
Customer acceptance	
System validation	n.a.
Project hand-off	n.a.

References

Abbasi, A., Albrecht, C., Vance, A., & Hansen, J. (2012). Metafraud: a meta-learning framework for detecting financial fraud. *Mis Quarterly*, 1293–1327.

Abbasi, A., Zhou, Y., Deng, S., & Zhang, P. (2018). Text analytics to support sense-making in social media: A language-action perspective. *MIS Quarterly*, *42*(2).

Abdullah, H., Qasem, A., Mohammed, N., & Emad, M. (2011). A Comparison Study between Data Mining Tools over some Classification Methods. *International Journal of Advanced Computer Science and Applications*.

Aggarwal, C. C., & Zhai, C. (2012). A Survey of Text Classification Algorithms. In *Mining Text Data* (pp. 163–222).

Amrit, C., Wijnhoven, F., & Beckers, D. (2015). Information Waste on The World Wide Web and Combating the Clutter. *23rd European Conference on Information Systems, ECIS 2015*, *2015*, 1–16.

Anand, S. S., & Büchner, A. G. (1998). *Decision support using data mining*. Financial Times Management.

Baier, L., Kühl, N., & Satzger, G. (2019). How to Cope with Change? Preserving Validity of Predictive Services over Time. *Hawaii International Conference on System Sciences (HICSS-52)*.

Banko, M., & Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*, 26–33.

Barocas, S., Hardt, M., & Narayanan, A. (2017). Fairness in machine learning. *NIPS Tutorial*.

Baumann, A., Lessmann, S., Coussement, K., & De Bock, K. W. (2015). Maximize What Matters: Predicting Customer Churn With Decision-Centric Ensemble Selection. *23rd European Conference on Information Systems, ECIS 2015*, 0–16. http://aisel.aisnet.org/ecis2015_cr/15

Blanc, S. M. (2016). *Bias-Variance Aware Integration of Judgmental Forecasts and Statistical Models* [Karlsruher Institut für Technologie (KIT)].

Blanc, S. M., & Setzer, T. (2015). Improving Forecast Accuracy by Guided Manual Overwrite in Forecast Debiasing. *23rd European Conference on Information Systems, ECIS 2015*, *2015*, 0–9.

Bretschneider, U., & Peters, R. (2016). Detecting cyberbullying in online communities. *24th European Conference on Information Systems, ECIS 2016*.

Brodley, C. E., & Smyth, P. (1995). The process of applying machine learning algorithms. *Proceedings of the ICML-95 Workshop on Applying Machine Learning in Practice*.

Cabena, P., Hadjinian, P., Stadler, R., Verhees, J., & Zanasi, A. (1998). *Discovering data mining: from concept to implementation*. Prentice-Hall, Inc.

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd International Conference on Machine Learning*, *C*(1), 161–168.

Cawley, G. C., & Talbot, N. L. C. (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, *11*, 2079−2107. http://jmlr.csail.mit.edu/papers/v11/cawley10a.html%5Cnhttp://www.jmlr.org/papers/volume11/cawley10a/cawley10a.pdf

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). Crisp-Dm 1.0. *CRISP-DM Consortium*, 76.

Chatterjee, S., Saeedfar, P., Tofangchi, S., & Kolbe, L. (2018). INTELLIGENT ROAD MAINTENANCE : A MACHINE LEARNING APPROACH FOR SURFACE DEFECT DETECTION. *26th European Conference on Information Systems, ECIS 2018*.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chawla, N. V. (2005). *Data Mining and Knowledge Discovery Handbook* (O. Maimon & L. Rokach (eds.); pp. 853–867). Springer US.

Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.

Cios, K. J., Teresinska, A., Konieczna, S., Potocka, J., & Sharma, S. (2000). Diagnosing myocardial perfusion from PECT bull's-eye maps-A knowledge discovery approach. *IEEE Engineering in Medicine and Biology Magazine*, *19*(4), 17–25.

Coiera, E., Ammenwerth, E., Georgiou, A., & Magrabi, F. (2018). Does health informatics have a replication crisis? *Journal of the American Medical Informatics Association*, *25*(8), 963–968.

Cui, G., Wong, M. L., & Wan, X. (2012). Cost-sensitive learning via priority sampling to improve the return on marketing and CRM investment. *Journal of Management Information Systems*, *29*(1), 341–374.

Dhar, V., Geva, T., Oestreicher-Singer, G., & Sundararajan, A. (2014). Prediction in economic networks. *Information Systems Research*, *25*(2), 264–284.

Ding, A. W., Li, S., & Chatterjee, P. (2015). Learning User Real-Time Intent for Optimal Dynamic Webpage Transformation. *Information Systems Research*, *26*(2), 339–359.

Dodge, J., Gururangan, S., Card, D., Schwartz, R., & Smith, N. A. (2019). Show your work: Improved reporting of experimental results. *ArXiv Preprint ArXiv:1909.03004*.

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

Dong, W., Liao, S., & Zhang, Z. (2018). Leveraging financial social media data for corporate fraud detection. *Journal of Management Information Systems*, *35*(2), 461–487.

Dorner, V., & Alpers, G. W. (2017). DETECTING PANIC POTENTIAL IN SOCIAL MEDIA TWEETS. *25th European Conference on Information Systems*.

Fang, X., Hu, P. J. H., Li, Z. L., & Tsai, W. (2013). Predicting adoption probabilities in social networks. *Information Systems Research*, *24*(1), 128–145.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM*, *39*(11), 27–34.

Feuerriegel, S., & Fehrer, R. (2016). Improving Decision Analytics with Deep Learning: the Case of Financial Disclosures. *24th European Conference on Information Systems, ECIS 2016*.

Feuerriegel, S., Riedlinger, S., & Neumann, D. (2014). Predictive Analytics For Electricity Prices Using Feed-Ins from Renewables. *22nd European Conference on Information Systems, ECIS 2014*, 0–14.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. *Advances in Neural Information Processing Systems*, 2962–2970.

Fisher, R. A. (1936). The use of multiple measures in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188.

Friedman, J. H. (1997). On Bias, Variance, 0/1—Loss, and the Curse-of-Dimensionality. *Data Mining and Knowledge Discovery*, *1*(1), 55–77.

Fu, L.-M. (2003). *Neural networks in computer intelligence*. Tata McGraw-Hill Education.

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. *Brazilian Symposium on Artificial Intelligence*, 286–295.

Geva, T., & Oestreicher-Singer, G. (2013). Do Customers Speak Their Minds? Using Forums and Search for Predicting Sales. *ICIS*, 1–17. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2294609

Gholami, R., Watson, R. T., Molla, A., Hasan, H., & Bjørn-Andersen, N. (2016). Information systems solutions for environmental sustainability: How can we do more? *Journal of the Association for Information Systems*, *17*(8), 521.

Gimpel, H., Kleindienst, D., & Waldmann, D. (2018). The disclosure of private data: measuring the privacy paradox in digital services. *Electronic Markets*, *28*(4), 475–490.

Goby, N., Brandt, T., Feuerriegel, S., & Neumann, D. (2016). Business Intelligence for Business Processes: The Case of IT Incident Management. *24th European Conference on Information Systems, ECIS 2016*, 1–15.

Golub, G. H., Heath, M., & Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, *21*(2), 215–223.

Gong, J., Abhishek, V., & Li, B. (2017). Examining the impact of keyword ambiguity on search advertising performance: A topic model approach. *MIS Quarterly*.

Goutte, C., & Gaussier, E. (2005). A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. *European Conference on Information Retrieval*, 345–359.

Hall, M., National, H., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software : An Update. *SIGKDD Explorations*, *11*(1), 10–18.

Han, H., Otto, C., Liu, X., & Jain, A. K. (2015). Demographic estimation from face images: Human vs. machine performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(6), 1148–1161.

Han, X., Wang, L., & Huang, H. (2017). Deep Investment Behavior Profiling by Recurrent Neural Network in P2P Lending. *ICIS 2017 Proceedings*, 1–11. http://aisel.aisnet.org/icis2017/Peer-to-Peer/Presentations/11

Hastie, T., Tibshirani, R., & Friedman, J. (2009). Overview of supervised learning. In *The elements of statistical learning* (pp. 9–41). Springer.

He, H., & Ma, Y. (2013). *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons.

Heilig, L., Hofer, J., Lessmann, S., & Voc, S. (2016). Data-driven product returns prediction: A cloud-based ensemble selection approach. *24th European Conference on Information Systems, ECIS 2016*, Garanti; Palgrave Macmillan; Turkish Electro Techn.

Hennig-Thurau, T., Walsh, G., & Schrader, U. (2004). VHB-JOURQUAL: Ein Ranking von betriebswirtschaftlich-relevanten Zeitschriften auf der Grundlage von Expertenurteilen. *Schmalenbachs Zeitschrift Für Betriebswirtschaftliche Forschung*, *56*(6), 520–545.

Hirt, R., Kühl, N., & Satzger, G. (2017). An end-to-end process model for supervised machine learning classification: from problem to deployment in information systems. *Designing the Digital Transformation: DESRIST 2017 Research in Progress Proceedings of the 12th International Conference on Design Science Research in Information Systems and Technology. Karlsruhe, Germany. 30 May-1 Jun.*

Ho, T. K., & Basu, M. (2002). Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(3), 289–300.

Hopf, K., Sodenkamp, M., Riechel, S., & Staake, T. (2017). Predictive Customer Data Analytics – The Value of Public Statistical Data and the Geographic Model Transferability. *ICIS 2017 Proceedings*, 1–20.

Huang, J., Boh, W. F., & Goh, K. H. (2017). A Temporal study of the effects of online Opinions: Information sources matter. *Journal of Management Information Systems*, *34*(4), 1169–1202.

Huang, K.-Y., Nambisan, P., & Uzuner, Ö. (2010). Informational support or emotional support: Preliminary study of an automated approach to analyze online support community contents. *ICIS-RP*, 1–11. http://aisel.aisnet.org/icis2010_submissions/210/

Hutson, M. (2018). *Missing data hinder replication of artificial intelligence studies*. Science Magazine.

Ivanov, A., & Sharman, R. (2018). Impact of User-Generated Internet Content on Hospital Reputational Dynamics. *Journal of Management Information Systems*, *35*(4), 1277–1300.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning. In *An Introduction to Statistical Learning*.

Johnson, S. L., Safadi, H., & Faraj, S. (2015). The emergence of online community Leadership. *Information Systems Research*, *July*, 35–68.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. In *Science*.

Kitchens, B., Dobolyi, D., Li, J., & Abbasi, A. (2018). Advanced customer analytics: Strategic value through integration of relationship-oriented big data. *Journal Of Management Information Systems*, *35*(2), 540–574.

Koroleva, K., & José Bolufé Röhler, A. (2012). REDUCING INFORMATION OVERLOAD: DESIGN AND EVALUATION OF FILTERING AND RANKING ALGORITHMS FOR SOCIAL NETWORKING SITES. *20th European Conference on Information Systems, ECIS 2012*, 5–2. http://aisel.aisnet.org/ecis2012%0Ahttp://aisel.aisnet.org/ecis2012/12

Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Informatica*, *31*, 249–268.

Kowatsch, T., & Maass, W. (2018). A Data - analytical System to Predict Therapy Success for Obese Children. *International Conference on Information Systems*, 1–17.

Kozlovskiy, I., Sodenkamp, M. A., Hopf, K., & Staake, T. (2016). Energy Informatics for Environmental, Economic and Societal Sustainability: a Case of the Large-Scale Detection of Households with Old heating Systems. *24th European Conference on Information Systems, ECIS 2016*.

Kurgan, L., & Musilek, P. (2006). A survey of Knowledge Discovery and Data Mining process models. *The Knowledge Engineering Review*, *21*(01), 1.

Laing, S., & Kühl, N. (2018). Comfort-as-a-service: Designing a user-oriented thermal comfort artifact for office buildings. *International Conference on Information Systems 2018, ICIS 2018*, *Parment 2009*, 1–17.

Lash, M. T., & Zhao, K. (2016). Early predictions of movie success: The who, what, and when of profitability. *Journal of Management Information Systems*, *33*(3), 874–903.

Li, L., Goethals, F., Giangreco, A., & Baesens, B. (2013). USING SOCIAL NETWORK DATA TO PREDICT TECHNOLOGY ACCEPTANCE. *ICIS-RP*, 1–10.

Li, W., Chen, H., & Nunamaker Jr, J. F. (2016). Identifying and profiling key sellers in cyber carding community: AZSecure text mining system. *Journal of Management Information Systems*, *33*(4), 1059–1086.

Li, Z., Hong, Y., & Zhang, Z. (2016). An empirical analysis of on-demand ride sharing and traffic congestion. *Proc. International Conference on Information Systems*.

Lin, Y.-K., Chen, H., Brown, R. A., Li, S.-H., & Yang, H.-J. (2017). HEALTHCARE PREDICTIVE ANALYTICS FOR RISK PROFILING IN CHRONIC CARE: A BAYESIAN MULTITASK LEARNING APPROACH. *MIS Quarterly*, *41*(2).

Lüttenberg, H., Bartelheimer, C., & Beverungen, D. (2018). Designing Predictive Maintenance for Agricultural Machines. *26th European Conference on Information Systems, ECIS 2018*.

Manning, C. D., & Schütze, H. (2000). Foundations of Natural Language Processing. In *Reading*.

Martens, D., & Provost, F. (2014). Explaining data-driven document classifications. *MIS Quarterly*.

Martens, D., Provost, F., Clark, J., & Junqué de Fortuny, E. (2016). Mining Massive Fine-Grained Behavior Data to Improve Predictive Analytics. *MIS Quarterly*, *40*(4), 869–888.

Microsoft. (2020). *Microsoft Team Data Science Process Documentation*. https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 220–229.

Mo, J., Sarkar, S., & Menon, S. (2018). Know when to run: Recommendations in crowdsourcing contests. *MIS Quarterly: Management Information Systems*.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2013). Foundations of Machine Learning. *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699.

Mongan, J., Moy, L., & Kahn Jr, C. E. (2020). Checklist for Artificial Intelligence in Medical Imaging (CLAIM): A guide for authors and reviewers. *ArXiv Preprint ArXiv:2003.05155*.

Neyman, J. (1934). On the two different aspects of the representative method : the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, *97*(4), 558–625.

Oh, C., & Sheng, O. (2011). Investigating Predictive Power of Stock Micro Blog Sentiment in Forecasting Future Stock Price Directional Movement. *Icis*, 1–19. http://www.mendeley.com/catalog/investigating-predictive-power-stock-micro-blog-sentiment-forecasting/%5Cnhttp://misrc.umn.edu/workshops/2011/fall/OliviaSheng_Paper.pdf%5Cnhttp://www.misrc.csom.umn.edu/workshops/2011/fall/OliviaSheng_Paper.pdf

Olbrich, S., Frank, U., Gregor, S., Niederman, F., & Rowe, F. (2017). On the merits and limits of replication and negation for IS research. *AIS Transactions on Replication Research*, *3*(1), 1.

Olorisade, B. K., Brereton, P., & Andras, P. (2017). *Reproducibility in machine Learning-Based studies: An example of text mining*.

Oquendo, M. A., Baca-Garcia, E., Artes-Rodriguez, A., Perez-Cruz, F., Galfalvy, H. C., Blasco-Fontecilla, H., Madigan, D., & Duan, N. (2012). Machine learning and data mining: strategies for hypothesis generation. *Molecular Psychiatry*, *17*(10), 956.

Oroszi, F., & Ruhland, J. (2010). An Early Warning System for Hospital Acquired Pneumonia. *Ecis*, *2010*. ecis2010.up.ac.za

Pant, G., & Srinivasan, P. (2010). Predicting Web page status. *Information Systems Research*, *21*(2), 345–364.

Pant, G., & Srinivasan, P. (2013). Status locality on the web: Implications for building focused collections. *Information Systems Research*, *24*(3), 802–821.

Pedregosa, F., & Varoquaux, G. (2011). Scikit-learn: Machine learning in Python. In *… of Machine Learning …* (Vol. 12).

Peng, R. D. (2011). Reproducible research in computational science. *Science*, *334*(6060), 1226–1227.

Pineau, J. (2020). *The machine learning reproducibility checklist*. http://www.cs.mcgill.ca/~jpineau/ReproducibilityChecklist-v2.0.pdf

Powers, D. M. (2011). Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, *2*(1), 37–63.

Pröllochs, N., Feuerriegel, S., & Neumann, D. (2015). Generating Domain-Specific Dictionaries using Bayesian Learning. *23rd European Conference on Information Systems, ECIS 2015*, *2015*, 0–14.

Qiao, N. (2019). A systematic review on machine learning in sellar region diseases: quality and reporting items. *Endocrine Connections*, *8*(7), 952–960.

Rahman, M. M., & Davis, D. N. (2013). Addressing the Class Imbalance Problem in Medical Datasets. *International Journal of Machine Learning and Computing*, *3*(2), 224–228.

Ram, S., Wang, Y., Currim, F., & Currim, S. (2015). Using Big Data for Predicting Freshmen Retention. *ICIS*, *Astin 1999*, 1–16.

Rätsch, G. (2004). A brief introduction into machine learning. *21st Chaos Communication Congress*, 1–6. http://www.mva.me/educational/hci/read/ML_reading.pdf

Riekert, M., Leukel, J., & Klein, A. (2016). Online Media Sentiment: Understanding Machine Learning-based Classifiers. *24th European Conference on Information Systems, ECIS 2016*.

Riekert, M., Premm, M., Klein, A., Lyubomir Kirilov, Kenngott, H., Apitz, M., Wagner, M., & Ternes, L. (2017). Predicting the Duration of Surgeries To Improve Process Efficiency in Hospitals. *25th European Conference on Information Systems, ECIS 2017*.

Rupp, M., Gobre, V., Vazquez-mayagoitia, A., Tkatchenko, A., & Lilienfeld, O. A. Von. (2013). Machine Learning of Molecular Electronic Properties in Chemical Compound Space. *New Journal of Physics*, *15*, 1–9.

Samtani, S., Chinn, R., Chen, H., & Nunamaker Jr, J. F. (2017). Exploring emerging hacker assets and key hackers for proactive cyber threat intelligence. *Journal of Management Information Systems*, *34*(4), 1023–1053.

Schooler, J. W. (2014). Metascience could rescue the 'replication crisis.' *Nature*, *515*(7525), 9.

Schwaiger, J., Lang, M., Johannsen, F., & Leist, S. (2017). “WHAT DOES THE CUSTOMER WANT TO TELL US?” AN AUTOMATED CLASSIFICATION APPROACH FOR SOCIAL MEDIA POSTS AT SMALL AND MEDIUM- SIZED ENTERPRISES. *25th European Conference on Information Systems*.

Seebach, C., Pahlke, I., & Beck, R. (2011). Tracking the Digital Footprints of Customers: How Firms can Improve their Sensing Abilities to Achieve Business Agility. *19th European Conference on Information Systems, ECIS 2011*, *2011*, 30–41. http://aisel.aisnet.org/ecis2011/258/

Sharp, J., & Babb, J. (2018). Is Information Systems late to the party? The current state of DevOps research in the Association for Information Systems eLibrary. *AMCIS*.

Shi, D., Guan, J., Zurada, J., & Manikas, A. (2017). A data-mining approach to identification of risk factors in safety management systems. *Journal of Management Information Systems*, *34*(4), 1054–1081.

Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C. T., Gaasenbeek, M., Angelo, M., Reich, M., Pinkus, G. S., Ray, T. S., Koval, M. A., Last, K. W., Norton, A., Lister, T. A., Mesirov, J., Neuberg, D. S., Lander, E. S., Aster, J. C., & Golub, T. R. (2002). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, *8*(1), 68–74.

Shmueli, G., & Koppius, O. R. (2011). Predictive Analytics in Information Systems Research. *Mis Quarterly*.

Spuler, M., Sarasola-Sanz, A., Birbaumer, N., Rosenstiel, W., & Ramos-Murguialday, A. (2015). Comparing metrics to evaluate performance of regression methods for decoding of neural signals. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, *2015-Novem*, 1083–1086.

Stange, M., & Funk, B. (2015). How Much Tracking Is Necessary - The Learning Curve in Bayesian User Journey Analysis. *23rd European Conference on Information Systems, ECIS 2015*, *2015*, 0–15.

Stange, M., & Funk, B. (2016). Predicting Online User Behavior Based on Real-Time Advertising Data. *24th European Conference on Information Systems, ECIS 2016*, 1–14. https://convertcase.net/

Staudt, P., Rausch, B., & Weinhardt, C. (2018). Predicting redispatch in the German electricity market using information systems based on machine learning. *International Conference on Information Systems 2018, ICIS 2018*.

Studer, S., Bui, T. B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., & Mueller, K.-R. (2020). Towards CRISP-ML (Q): A Machine Learning Process Model with Quality Assurance Methodology. *ArXiv Preprint ArXiv:2003.05155*.

Tackett, J. L., Brandes, C. M., King, K. M., & Markon, K. E. (2019). Psychology’s replication crisis and clinical psychological science. *Annual Review of Clinical Psychology*, *15*, 579–604.

Tafti, A., & Gal, D. (2018). Predicting Complainers on Social Media : A Machine Learning Approach. *International Conference on Information Systems*, 1–16.

Timmerman, Y., & Bronselaer, A. (2019). Measuring data quality in information systems research. *Decision Support Systems*, *126*, 113138.

Tripathi, M., & Kaur, I. (2018). Oil Prices Forecasting: A Comparative Analysis. *International Conference on Information Systems (ICIS)*.

Tušar, T., Gantar, K., Koblar, V., Ženko, B., & Filipič, B. (2017). A study of overfitting in optimization of a manufacturing quality control procedure. *Applied Soft Computing*, *59*, 77–87.

Twyman, N. W., Proudfoot, J. G., Schuetzler, R. M., Elkins, A. C., & Derrick, D. C. (2015). Robustness of multiple indicators in automated screening systems for deception detection. *Journal of Management Information Systems*, *32*(4), 215–245.

Urbanke, P., Uhlig, A., & Kranz, J. (2017). A Customized and Interpretable Deep Neural Network for High-Dimensional Business Data - Evidence from an E-Commerce Application. *ICIS 2017 Proceedings*, 1–18. http://aisel.aisnet.org/icis2017/DataScience/Presentations/13

VHB. (2012). *VHB-JOURQUAL3*. VHB. https://vhbonline.org/en/service/jourqual/vhb-jourqual-3/

VHB. (2019). *Complete list of the journals in VHB-JOURQUAL3 in alphabetical order*. VHB. https://vhbonline.org/en/service/jourqual/vhb-jourqual-3/complete-list-of-the-journals/

Voets, M., Møllersen, K., & Bongo, L. A. (2018). Replication study: Development and validation of deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *ArXiv Preprint ArXiv:1803.04337*.

Walden, E., Cogo, G. S., Lucus, D. J., Moradiabadi, E., & Safi, R. (2018). Neural Correlates of Multidimensional Visualizations: An fMRI Comparison of Bubble and Three-Dimensional Surface Graphs Using Evolutionary Theory. *MIS Quarterly*, *42*(4), 1097–1116.

Wang, R. Y., Kon, H. B., & Madnick, S. E. (1993). Data Quality Requirements Analysis And Modeling. *Proceedings of IEEE 9th International Conference on Data Engineering*, *April*, 670–677.

Wang, T., Kannan, K. N., Ulmer, J. R., Wang, T., Kannan, K. N., & Ulmer, J. R. (2013). The Association Between the Disclosure and the Realization of Information Security Risk Factors The Association Between the Disclosure and the Realization of Information Security Risk Factors. *Information Systems Research*, *May 2016*.

Weinhardt, C., van der Aalst, W. M. P., & Hinz, O. (2019). Introducing Registered Reports to the Information Systems Community. *Business and Information Systems Engineering*.

Winkler-Schwartz, A., Bissonnette, V., Mirchi, N., Ponnudurai, N., Yilmaz, R., Ledwos, N., Siyar, S., Azarnoush, H., Karlik, B., & Del Maestro, R. F. (2019). Artificial intelligence in medical education: best practices using machine learning to assess surgical expertise in virtual reality simulation. *Journal of Surgical Education*, *76*(6), 1681–1690.

Witten, I. H., Frank, E., & Hall, M. a. (2011). Data Mining: Practical Machine Learning Tools and Techniques. In *Annals of Physics* (Vol. 54, Issue 2).

Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, P., Parkhe, M., & others. (2018). Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.*, *41*(4), 39–45.

Zhou, J. (2017). Data Mining for Individual Consumer Credit Default Prediction under E-commence Context : A Comparative Study. *Icis*, 1–18.

About the Authors

Niklas Kühl is leading the Applied AI in Services Lab at the Karlsruhe Service Research Institute (KSRI) at the Karlsruhe Institute of Technology (KIT). He is also working as a Data Scientist for IBM Global Business Services in diverse industry projects. Niklas' goal is to continuously facilitate the exchange between academia and industry by publishing highly relevant work for both worlds. He has been researching applications in machine learning for the past 5 years in different domains. Niklas did his PhD in designing and implementing a machine learning based tool capable to automatically identify customer needs in social media data, e.g. for e-mobility needs expressed via Twitter. Currently, he and his team of nine researchers are working on different AI solutions within industrial services, sales forecasting, production lines and many other examples.

Robin Hirt is the Co-Founder and Co-CEO of prenode GmbH, a German-based company that provides solutions for decentralized machine learning and has been acknowledged by Gartner as one of the few companies that enable privacy-preserving cross-organizational machine learning with its unique technologies. He is also an associated researcher at the Applied AI in Services Lab at the Karlsruhe Service Research Institute at the Karlsruhe Institute of Technology (KIT). Robin did his PhD in designing and developing algorithms and systems for realizing privacy-preserving horizontal and vertical federated machine learning and collaborated with the MIT-IBM AI Lab as a visiting researcher.

Lucas Baier is a Research Associate in the Applied AI in Services Lab at the Karlsruhe Service Research Institute (KSRI) which is located at the Karlsruhe Institute of Technology (KIT). He holds a bachelor and a master's degree in Industrial Engineering from KIT. His research is concerned with solving the challenges of deployed machine learning services in real-world settings with a special focus on concept drift. He has developed various concept drift handling algorithms for domains such as mobility or e-commerce and his ideas have been implemented in various research projects funded by both industry and government.

Björn Schmitz is a Data Science Manager and Senior Data Scientist in the Cognitive & Analytics practice of IBM Services. In his role, Björn supports clients in diverse industries and markets in designing, developing and operating solutions that utilize Machine Learning and Artificial Intelligence. He acquired a PhD in Information Systems at the Karlsruhe Institute of Technology (KIT) focusing on estimating cost uncertainties in industrial full-service contracts. Since then, Björn has collaborated closely with the scientific community in various research projects and is currently working as a lecturer for computer vision at KIT.

Gerhard Satzger is Director of the Karlsruhe Service Research Institute, an "industry-on-campus" initiative focused on innovation in IT-based services, and professor at the Karlsruhe Institute of Technology (KIT) in Germany. His research is concerned with conceiving and developing digital services and corresponding business models - with a particular focus on human-centered design for services as well as on creating value from data via AI. Drawing both on academic qualifications and experience in IS as well as on a multi-year industry track record in various national and international roles at IBM, he strives to drive effective innovation by collaboration between industry and academia. His work is published in IS and service journals and finds its application in projects with a variety of industry partners.