

Analysis of Incremental Learning and Windowing to Handle Combined Dataset Shifts on Binary Classification for Product Failure Prediction

Marco Spieß¹, Peter Reimann^{1,2}, Christian Weber² and Bernhard Mitschang²

¹Graduate School of Excellence advanced Manufacturing Engineering (GSaME), University of Stuttgart, Germany

²Institute for Parallel and Distributed Systems (IPVS), University of Stuttgart, Germany

Keywords: Binary Classification, Combined Dataset Shift, Incremental Learning, Product Failure Prediction, Windowing.

Abstract: *Dataset Shifts (DSS)* are known to cause poor predictive performance in supervised machine learning tasks. We present a challenging binary classification task for a real-world use case of product failure prediction. The target is to predict whether a product, e. g., a truck may fail during the warranty period. However, building a satisfactory classifier is difficult, because the characteristics of underlying training data entail two kinds of DSS. First, the distribution of product configurations may change over time, leading to a *covariate shift*. Second, products gradually fail at different points in time, so that the labels in training data may change, which may be a *concept shift*. Further, both DSS show a trade-off relationship, i. e., addressing one of them may imply negative impacts on the other one. We discuss the results of an experimental study to investigate how different approaches to addressing DSS perform when they are faced with both a covariate and a concept shift. Thereby, we prove that existing approaches, e. g., *incremental learning* and *windowing*, especially suffer from the trade-off between both DSS. Nevertheless, we come up with a solution for a data-driven classifier, that yields better results than a baseline solution that does not address DSS.

1 INTRODUCTION

Dataset Shifts (DSS) (Quiñonero-Candela et al., 2008) relate to the phenomenon that properties and statistical distributions of data change over time (Moreno-Torres et al., 2012). They are common in many real-world application scenarios of non-stationary environments and their study is an active area of research (Ditzler et al., 2015; Bang et al., 2019; Dharani Y. et al., 2019; Losing et al., 2018; Elwell and Polikar, 2011). Literature discusses various types of DSS (Moreno-Torres et al., 2012). Especially two types of DSS occur more frequently in practice than others and are therefore of particular relevance: a *covariate shift* (Dharani Y. et al., 2019) and a *concept shift* (Elwell and Polikar, 2011). A covariate shift is a shift in the distribution of feature values between a training dataset \mathcal{T} and a test dataset \mathcal{T}' , so that single class patterns occur with different frequencies. A concept shift constitutes a change in the decision boundaries of individual class patterns. Usually, DSS cause poor prediction performance in supervised machine learning tasks (Kull and Flach, 2014).

Related work proposes approaches such as *incremental learning* (Losing et al., 2018) and *windowing* (Bifet and Gavaldà, 2007) to address either covariate or concept shifts. Moreno-Torres et al. highlight that no adequate solution exists that addresses combinations of both kinds of DSS, since such combinations seem to be rare in practice (Moreno-Torres et al., 2012). However, data of many real-world use cases, e. g., from product design (Nalbach et al., 2018) or medical diagnoses (Khan and Usman, 2015; Maitín et al., 2020), often suffer from both DSS. Here, such combined DSS make it difficult to build classifiers for improving product design and automating medical diagnoses. So far, no studies investigate how different approaches for dealing with DSS perform when confronted with both covariate shifts and concept shifts.

In this paper, we analyze approaches to deal with combined dataset shifts. To this end, we have developed a real-world use case together with a large global truck manufacturer. The manufacturer finds that an increased number of trucks suddenly fail during the warranty period. A root cause analysis shows that the targeted assembly of more robust product components prevents these failures. Accordingly, the truck manu-

facturer plans to selectively assemble robust parts in those trucks that are likely to fail. This can be framed as a supervised machine learning task for product failure prediction to avoid future failures.

For this task, we collect a training dataset \mathcal{T} , representing trucks that are already in customer use. The feature set of \mathcal{T} corresponds to the truck configurations, while a label indicates whether a truck has failed during the warranty period. The goal is to learn a binary classifier in order to apply it on a test dataset \mathcal{T}' . This test set \mathcal{T}' represents new trucks and their configurations that will soon be produced. The classifier predicts which of these new trucks are also likely to fail with the less robust components. The manufacturer then assembles more robust components into these trucks as a preventive measure. Learning the binary classifier is complex because the use case data characteristics exhibit two types of DSS. **Covariate Shift (CS1):** The training data \mathcal{T} consists of 3,000 features that represent the various components assembled in each truck. However, the truck manufacturer produces a large number of truck variants whose components and configurations change over time. This induces a covariate shift (CS1). **Concept Shift (CS2):** Trucks may fail at different points in time during their use phase. Hence, the labels of trucks that eventually fail gradually shift from "nonfailed" to "failed" over time, i. e., one truck after the other. This leads to a concept shift (CS2).

In this paper, we investigate how to build a classifier with satisfactory prediction performance measured as *True-Positive-Rate (TPR)* under the influence of these two types of dataset shifts in combination. We thereby make the following main contributions:

- We shed light into real-world data characteristics and causes of combined DSS in a use case of product failure prediction. We propose four meta-features that clearly analyze the extents of covariate (CS1) and concept shifts (CS2) in these data.
- We show that existing approaches to *Data Stream Mining* that handle DSS, e. g., incremental learning and windowing, suffer from a trade-off relationship between covariate (CS1) and concept shifts (CS2). With our meta-features, it is possible for the first time to measurably analyze the trade-off behavior of this combined dataset shift.
- We come up with a classifier that yields better results than a baseline that does not explicitly address DSS. This classifier finally increases the True-Positive-Rate and leads to a significant reduction of warranty claim costs by $\sim 13\%$ -points.

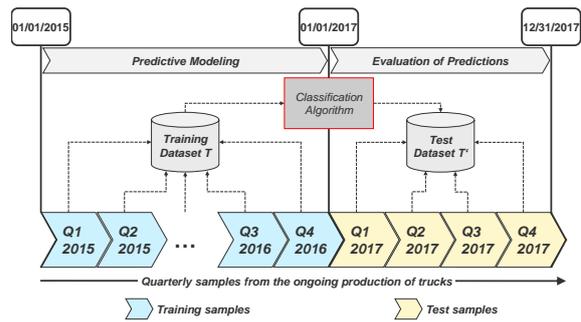


Figure 1: Overview on datasets for training and testing possible classifiers that can predict product failures of trucks.

- As basis for our evaluation, we have generated synthetic datasets. We provide these datasets in a GitHub repository¹ to increase reproducibility.

Our paper is structured as follows: in Section 2, we describe the use case of predicting product failures and the two kinds of DSS. We discuss related work in Section 3. Section 4 then illustrates the design of our experimental study. In Section 5, we discuss the evaluation results and come up with the classifier that outperforms the baseline. We conclude in Section 6.

2 DATASET SHIFTS IN PRODUCT FAILURE PREDICTION TASKS

In this section, we present our use case for product failure prediction (2.1), describe the underlying data characteristics (2.2), and how this leads to a covariate and a concept shift (2.3). Last, we discuss in which other domains both kinds of DSS occur together (2.4).

2.1 Real-world Use Case from Industry: Prediction of Product Failures

A truck manufacturer provided us with data from a real-world use case. The truck manufacturer noticed that powertrain failures happened more frequently since the production year 2015. Usually, components from the powertrain, e. g., injection pumps, turbocharger, gears, etc. fail due to higher loads. To therefore avoid possible real-world hints and to increase readability, this paper considers an unspecified component from the powertrain. In order to tackle this quality problem, the truck manufacturer substituted this component with another more robust one. We call the initial, less robust part \mathcal{A} and its more robust

¹<https://github.com/MarcoS89-dev/Combined-Dataset-Shifts-in-Product-Failure-Prediction>

variant \mathcal{B} . Starting from 01/01/2017, \mathcal{B} is ready for assembly in trucks and prevents particular powertrain damages in the later use phase of trucks. However, due to supply bottlenecks, the availability of part \mathcal{B} is restricted and can only be assembled in a maximum of 50% of all trucks. Figure 1 shows the corresponding production time line for the use case and which data we used as training (\mathcal{T}) and test dataset (\mathcal{T}').

Quality engineers found that individual truck configurations somehow influence the proneness to powertrain damages. The idea is to determine which configurations are more prone to this damage. So, these trucks are then assembled with the more robust part \mathcal{B} . Otherwise, part \mathcal{A} and \mathcal{B} are assembled randomly.

The manufacturer asked data scientists to collect historical training data and to learn a classifier that can predict truck failures. On 01/01/2017, the data scientists built a classifier using the collected training dataset \mathcal{T} of trucks produced in 2015 and 2016 (see Figure 1). They then applied this classifier to the new trucks produced in 2017 to decide whether to assemble part \mathcal{A} or \mathcal{B} in these new trucks. Some years later, in 2022, the labels in the test data \mathcal{T}' have changed and now show stable class patterns and fixed decision boundaries. So, the data scientists finally evaluated their classifier in this year again. The evaluation shows a *True-Positive-Rate (TPR)* of only 55.9%. This is an increment of 5.90%-points compared to the random assembly of \mathcal{A} and \mathcal{B} . However, this 55.9% is a poor predictive performance and not satisfactory at all. The data scientists assume that it is due to DSS.

Research goal. This use case provides the starting point for our study. We investigate different well-known approaches from *Data Stream Mining* that address DSS. The ultimate goal is to determine if it is possible to build a classifier that outperforms the baseline with 55.9% TPR. Please note that the final TPR is only known when enough time has passed. This means in our case, when each truck from the test data has completed its warranty period. Thus, we are looking from today's perspective, i. e., from 2022, into the past. With this retrograde analysis, we examine if it is possible to learn a better classifier with different combinations of sampling techniques and algorithms.

2.2 Data Characteristics

Table 1 shows the attributes of \mathcal{T} and \mathcal{T}' : truck identification number, a class label and the feature space \mathcal{X} . The label of a sample is either "failed" as positive class c^+ or "nonfailed" as negative class c^- . Moreover, \mathcal{T} and \mathcal{T}' include only trucks with part \mathcal{A} , because only these may fail due to the particular failure.

Table 1: Exemplary structure of datasets \mathcal{T} and \mathcal{T}' .

Truck-ID	Class label (c^+ / c^-)	x_1 : Engine ABC	x_2 : Engine XYZ	x_η : Other part
1	failed	1	0	0
2	nonfailed	0	1	0
3	nonfailed	1	0	1
...

Note that the manufacturer only gets data of failed trucks if the customer brings them to a workshop that is under contract with the manufacturer. Within the 36-month warranty period, the customer is restricted to bring his or her trucks to those contracted workshops to claim warranty obligations. However, after the warranty period has expired, the customer is free to choose another workshop. In this case, we cannot capture any more labels for trucks out of warranty.

The feature space \mathcal{X} is one-hot encoded and contains around 3,000 disjunctive features. So, \mathcal{X} consists of η binary variables x_1 to x_η . Each variable represents one particular truck equipment, e. g., a certain type of an engine. The binary values 1 and 0 describe whether a respective equipment is present in a truck or not. Example: truck 1 and truck 3 in Table 1 are equipped with an *ABC engine*, whereas truck 2 has an *XYZ engine*. Each truck consists on average of about 186 binary 1's in \mathcal{X} and about 2,814 binary 0's. The number of 1's per truck ranges from 131 to 262.

Meta-features. Table 2 shows the four relevant meta-features: c^+ ratio, $\emptyset MIS$, $\mathcal{V}S$ and $\mathcal{V}S_{c^+}$.

With our proposed meta-features, we analyze the datasets \mathcal{T} and \mathcal{T}' at two time points: on 01/01/2017 and on 01/01/2022. We use these two different data states in Section 2.3 to illustrate the DSS of our datasets. The training set \mathcal{T} contains 80,948 trucks that have been produced in 2015 (\mathcal{T}_{2015}) or 2016 (\mathcal{T}_{2016}). On 01/01/2017, only 97 of them have already failed and thus get the positive class c^+ as label. As a result, the c^+ ratio, which represents the share of failed trucks to all trucks, is 0.12%. The *Months-In-Service (MIS)* (see Table 2) is the operating time of a truck and correlates with the c^+ ratio because the older a truck, the higher its risk of failure.

The trucks in \mathcal{T} and \mathcal{T}' may differ considerably. We have identified 489 key features that describe a truck in its basic characteristics. These 489 key features split into 11 non-overlapping feature domains, e. g., engine (29 key features), gearbox (30 key features), and axle (41 key features). In each feature domain, we compare the distributions of key features between \mathcal{T} and \mathcal{T}' to develop the meta-feature *Truck Similarity* $\mathcal{V}S$ as a similarity indicator. *Truck Similarity* $\mathcal{V}S$ is the mean of the frequency differences across

Table 2: View on \mathcal{T} and \mathcal{T}' on 01/01/2017 and 01/01/2022 with number of trucks (N), failures (c^+) and the meta-features.

Set	N	c^+	c^+ ratio	\varnothing MIS	$\mathcal{V}S$	$\mathcal{V}S_{c^+}$
Data Status on 01/01/2017						
\mathcal{T}	80,948	97	0.12%	10.28	79.67%	49.95%
\mathcal{T}_{2015}	38,659	82	0.21%	16.63	76.73%	49.06%
\mathcal{T}_{2016}	42,289	15	0.04%	5.29	82.28%	57.40%
Data Status on 01/01/2022						
\mathcal{T}	80,948	7,245	8.95%	71.89	79.67%	65.14%
\mathcal{T}'	42,260	4,585	10.85%	52.11	-	-

 Table 3: Exemplary comparison of the proportions of key features within the engine domain between \mathcal{T} and \mathcal{T}' .

Key Features	Proportions in...		Difference
	\mathcal{T}	\mathcal{T}'	
H290, 10.81	17%	12%	5%
H310, 12.91	-	25%	25%
H460, 15.71	35%	20%	15%
M130, 5.21	7%	7%	0%
M175, 7.81	4%	-	4%
...

the respective key features. For instance, a particular engine may be assembled more often or less in \mathcal{T} than in \mathcal{T}' , leading to a shift of the frequencies of this particular feature value. This shift in feature frequencies and the resulting differences are illustrated in Table 3.

Finally the value of meta-feature $\mathcal{V}S$ (see Table 2) is the result from $100\% - \varnothing$ differences of all eleven unweighted feature domains. In order to measure the truck similarity only between failed trucks from the training (\mathcal{T}) and the test (\mathcal{T}') datasets, we developed $\mathcal{V}S_{c^+}$ as the *Truck Similarity of Class c^+* . The calculation of $\mathcal{V}S_{c^+}$ is the same as for $\mathcal{V}S$, except that we only use the distributions of failed trucks (c^+).

2.3 Dataset Shifts

The characteristics of our use case and of its data establish a non-stationary environment that entails two kinds of *dataset shifts* for product failure predictions. For the mathematical description of the DSS we denote x as the input variables, i. e. features, and y as the target class variable (Moreno-Torres et al., 2012).

Covariate Shift (CS1): $P_{train}(y|x) = P_{test}(y|x)$ and $P_{train}(x) \neq P_{test}(x)$. The functional relationship $f(x)$ between training and test data remains the same over time. However, the distributions within the feature sets x of training and test data are different. Consequently, important class patterns in the test data may be underrepresented in the training data and vice versa. This may usually result in a degradation of the

classification performance, since relevant class patterns in the test dataset cannot be entirely learned by a classification algorithm from the training dataset.

CS1 occurs in our use case because the configurations of trucks especially vary for different areas of application, e. g., construction site or transport trucks. Over individual quarters, different customers order different fleets of trucks, which can vary significantly in terms of their areas of application and thus in their configurations. Therefore, the trucks in the training dataset \mathcal{T} may differ from those in the test dataset \mathcal{T}' . This can be seen by the meta-feature $\mathcal{V}S$ reported in Table 2. The value of $\sim 80\%$ indicates that the distribution of feature values in \mathcal{T} and \mathcal{T}' differ by at least 20%. Even worse, the similarity of failed trucks ($\mathcal{V}S_{c^+}$) is only about 50%. This difference in the distribution of features constitutes a *covariate shift*.

Concept Shift (CS2): $P_{train}(y|x) \neq P_{test}(y|x)$ and $P_{train}(x) = P_{test}(x)$. The distributions of features remain the same between training and test data, but a shift in the mapping function $f(x)$ occurs. This may usually result in a degradation of the classification performance, since the descriptions of the class patterns change over time, although the feature space \mathcal{X} remains the same. This means that the patterns in the training data are described either by different features or value ranges than the actual patterns in the test data.

CS2 occurs in our use case because each sample of \mathcal{T} is initially an element of c^- ("nonfailed") and has a chance to eventually become an element of c^+ ("failed"). On 01/01/2017, the average MIS of the 80,948 trucks in training data \mathcal{T} is about 10 months, and 97 of them failed so far. Until 01/01/2022, when the average MIS is about 72 months, the number of failed trucks increases to 7,245. So, the class label of 7,148 trucks has shifted from c^- to c^+ , so that the c^+ ratio increases from 0.12% to 8.95%. This causes a *concept shift*, as the decision boundaries of class patterns may change with each label shift.

Combination of both Shifts CS1 and CS2: $P_{train}(y|x) \neq P_{test}(y|x)$ and $P_{train}(x) \neq P_{test}(x)$. Moreno et al. currently consider this combined DSS

as impossible to solve. Further, they consider that this DSS is not discussed in literature because it rarely occurs in practice (Moreno-Torres et al., 2012). However, in real-world applications these DSS are more common than literature assumes, and thus we specifically focus on them in this paper. In the next section (2.4), we discuss other domains where both DSS occur, highlighting the importance of this topic.

Trade-off Relationship between CS1 and CS2: To illustrate the interdependence between the covariate and the concept shift in our use case, we compare in Table 2 two subsets of the training dataset \mathcal{T} : older trucks produced in 2015 (\mathcal{T}_{2015}) and newer trucks from 2016 (\mathcal{T}_{2016}). On 01/01/2017, older trucks from 2015 have 82 failures and a c^+ ratio of 0.21%. Newer trucks from 2016 however have only 15 failures and a lower c^+ ratio of 0.04%. Additionally, trucks produced in 2016 have values of $\mathcal{V}S$ and $\mathcal{V}S_{c^+}$ that are with 82.28% and 57.40% about 7%-points higher than those from 2015. This indicates that the older and thus less similar the trucks are, the more significant the negative effect of the covariate shift (CS1) and the lower the negative effect of the concept shift (CS2). Newer trucks show exactly the opposite behavior, i. e., they are rather affected by CS2 than by CS1.

2.4 Dataset Shifts in Other Domains

The characteristics of DSS are common in manufacturing data (Bang et al., 2019). We have found other application areas in the literature in which both DSS occur together. This includes domains such as product design (Nalbach et al., 2018) and medical diagnoses (Khan and Usman, 2015; Maitín et al., 2020).

In the area of product design, Nalbach et al. (Nalbach et al., 2018) have developed a system called *Preventive Quality Assurance (PreQA)*. This system uses machine learning to learn from the products returned by customers in order to improve the design of new products. During the development of new products, the product designer receives a forecast from PreQA that indicates the likelihood if the planned product design may later fail during customer use. To this end, PreQA refers to historical failures and correlates them with the new product design based on similar feature subsets. In analogy to our use case, PreQA also takes covariate shifts (CS1) into account by selecting similar product designs. Moreover, concept shifts (CS2) have a negative impact on the predictive accuracy of the classifiers in this area as well. This is because the training data of PreQA consists of products in customer use. These products likewise gradually fail over time, which may thus change the concepts in the data.

Both types of dataset shifts also occur in the detection of neurological diseases, as exemplified by Alzheimer's (Khan and Usman, 2015) and Parkinson's (Maitín et al., 2020) diseases. For instance, people with certain characteristics, such as old age or diabetes, are more susceptible to Alzheimer's disease. The distribution of these characteristics that predispose Alzheimer's disease change over time among people. For example, more people get diabetes today than a few decades before because they consume more industrial sugar. This leads to a shift in the covariates (CS1). Furthermore, given that the Alzheimer's disease is a gradual disease, it is often detected for a specific person in a late stage of the disease. This means that class labels and thus also their patterns (concepts) shift over time (CS2).

3 RELATED WORK

In this section, we first discuss related work in the area of data-driven prediction of product failures (3.1). Next, we discuss related work on known data stream mining algorithms that address dataset shifts (3.2).

3.1 Prediction of Product Failures

Existing reviews on the prediction of product failures show an increased interest for data-driven solutions (Wu, 2013). Khoshkangini et al. (Khoshkangini et al., 2019) state that most of related work relies solely on age-related variables of products to train predictive classifiers. Such variables are, e. g., the *mileage* or *MIS*. Furthermore, they prove that the inclusion of product usage information, i. e., logged on-board data, leads to a performance gain for predictions. The benefits of using product usage information has been shown in other work from Volvo Trucks (Prytz et al., 2015), in which the authors analyzed truck usage data collected over several years.

In our use case however, if we train a classifier with usage information of trucks in the training dataset \mathcal{T} , we could not apply this classifier to the new trucks in the test dataset \mathcal{T}' . This is because the trucks in \mathcal{T}' are not yet produced at the beginning of 2017, so that no usage information is available for them as features. We hence use the individual truck configurations as features, since this is common in both datasets \mathcal{T} and \mathcal{T}' (see Table 1). Additional advantages of using truck configurations as features is that they are ready for analysis from the truck production and remain unchanged during the useful life. Moreover, this static property represents a kind of ground truth in the non-stationary environment of the truck production.

In summary, related work in the domain of product failure prediction conducts their studies with assumptions and data-setups that differ from ours. Moreover, it turns out that the two kinds of DSS are not considered by related work at all, although they are important for the prediction of product failures. The diversity of feature distributions (CS1), i. e., of truck variants and their configurations, leads to different usage patterns (Khoshkangini et al., 2019). This in turn increases the variety of usage data across products produced at different points in time. Furthermore, Wu highlights the difficulties of concept shifts (CS2) by using incomplete warranty data (Wu, 2013).

3.2 Data Stream Mining Algorithms for Addressing Dataset Shifts

Common algorithms addressing dataset shifts implement approaches from *Data Stream Mining* (Homayoun and Ahmadzadeh, 2016), i. e., incremental learning (Losing et al., 2018) and windowing (Bifet and Gavaldà, 2007). In the following, we discuss these algorithms in terms of their ability to address the covariate shift (CS1) and concept shift (CS2) as they are present in our use case given our data characteristics.

Incremental Learning Algorithms: The aim in this approach is to keep the classifiers up to date or to re-train them again with new or changing data (Losing et al., 2018). Related work discusses several adaptations of non-stream classification algorithms so that these batch learning algorithms support incremental learning, e. g., for Decision Trees (Utgoff, 1989), Support-Vector-Machines (Bordes et al., 2005) and *k-Nearest-Neighbors* (*kNN*) (Losing et al., 2018). Besides, tailored algorithms exist to handle DSS, e. g., the *Adaptive Random Forest* (*ARF*) (Gomes et al., 2017). *ARF* is a widely accepted classification algorithm for evolving data streams and extends Breiman’s (Breiman, 2001) original Random Forest algorithm by a drift detection component. If the algorithm detects a drift for one of the decision trees in the ensemble, a new tree is trained in the background until it is ready to replace the tree in the ensemble for which the drift has been detected. An extension exist to handle imbalanced datasets in particular: *Adaptive Random Forests with Resampling* (*ARFRE*) (Boiko Ferreira et al., 2019). Another interesting algorithm to handle DSS are *Fuzzy Hoeffding Decision Trees* (*FHDT*) (Ducange et al., 2021), which is an extension of HDT (Domingos and Hulten, 2002) to address concept shifts in data stream classification.

Windowing Algorithms: To address DSS in data streams, several windowing approaches exist (Bifet and Gavaldà, 2007; Iwashita and Papa, 2019; Lu

et al., 2019). For instance, *ADaptive WINdowing* (*ADWIN*) (Bifet and Gavaldà, 2007) uses a sliding-window approach with a dynamic window size. As long as no concept shift is detected, the window grows accordingly, otherwise it shrinks. A new classifier is trained when a concept shift is detected. This drift detection is performed by comparing different subsets.

Discussion: Incremental learning and windowing are based on the same basic assumption as they prioritize new data more than old data. The reason is that more recent data inherently co-describe the DSS. When predicting product failures, however, the prioritization of certain data subsets is not as clear as literature assumes. This prioritization has to explicitly consider the trade-off between both kinds of DSS: covariate (CS1) and concept shift (CS2) (see Section 2.3).

Regarding the concept shift (CS2), the data must even be prioritized exactly the opposite way round. In fact, data subsets of older trucks are explicitly preferred for training classifiers, given that they have higher MIS values and higher c^+ ratios. Moreover, trucks with a higher MIS are more likely not to fail anymore and thus to be stable in their class label. This also means that the probability for a concept shift decreases. For our use case, incremental learning and windowing need to be adapted, so that older trucks are weighted higher than newer ones. However, these older trucks have lower $\mathcal{V}\mathcal{S}$ values and thus a less similarity to the trucks contained in the test dataset \mathcal{T}' . Hence, preferring older trucks due to the concept shift (CS2) may increase the negative effects of the covariate shift (CS1). Such a non-trivial trade-off relationship is however not explicitly addressed at all by related approaches to addressing dataset shifts.

This trade-off relationship between CS1 and CS2 calls for a more in-depth evaluation of approaches to incremental learning and windowing. This evaluation has to consider how these techniques may address both kinds of dataset shifts at the same time, including their trade-off. To the best of our knowledge, literature however does not comprise any corresponding study. In order to address this issue, we have carried out experiments for our use case of predicting product failures. We detail on this in the following section.

4 EXPERIMENTAL STUDY DESIGN

The aim is to train the best possible classifier \mathcal{M} to classify the 42,260 trucks in the test set \mathcal{T}' , i. e., those produced in 2017. We first present our experimental setup for training classifiers (4.1). Then, we explain

how we evaluate the classification results in scenarios for both incremental learning and windowing (4.2).

4.1 Experimental Setup

The data available for our experiments are high-dimensional, low-sample size (*HDLSS*) datasets. For HDLSS datasets, it is typical that the number of features x is higher than the sample size N (Marron et al., 2007). This applies to the 97 samples of target class c^+ on 01/01/2017. Further, the feature space comprising 3,000 disjunctive features. Therefore, we need to take the typical aspects of HDLSS data into account when choosing the classification algorithm and techniques for sampling and feature selection. With a c^+ ratio of about 0.12% on 01/01/2017, the dataset is highly imbalanced. Therefore, this binary class imbalance must also be taken into account in the setup. *Classification Algorithm:* Our use case implies some constraints that restrict the choice of classification algorithms. The first is the supply constraint which determines how many trucks can be equipped with the more robust part \mathcal{B} . Here, probabilistic classification algorithms are especially suitable. Probabilistic in this context means that each trained classifier outputs a confidence value for each truck in \mathcal{T}' . This indicates how confident the classifier is that a particular truck belongs to class c^+ . The idea is to classify each truck in \mathcal{T}' and to sort the trucks according to their confidence values in descending order. Then, we label all trucks having confidence values above a certain threshold with the positive class c^+ . These trucks are equipped with part \mathcal{B} , and the remaining trucks with lower confidence values get part \mathcal{A} . We thereby set the threshold value as low as possible, so that firstly as many trucks as possible get part \mathcal{B} , but secondly the supply constraint of this part is still satisfied. The Support-Vector-Machine (*SVM*) (Cortes and Vapnik, 1995) is a well-known algorithm to consider as it handles binary features well. Studies (Marron et al., 2007) have shown that SVM suffers from the characteristics of HDLSS data, because SVM encounters difficulties to determine the support vectors with a low number of samples and a high number of features. Thus, Marron et al. developed the Distance-Weighted-Discrimination. However, this approach is not able to handle the negative impacts of combined DSS nor analyze its behaviors.

For this purpose, the bootstrap aggregation (bagging) (Breiman, 1996) is a proper method. Due to the statistical properties of bagging, it is thus possible to draw reliable conclusions from the classification results related to the combined DSS behavior. We choose the *Random Forest (RF)* (Breiman, 2001)

algorithm because it is the most prominent bagging technique, requires few hyper-parameters, and still produces stable results. Moreover, authors with similar data already had good experiences with *RF* (Prytz et al., 2015; Hirsch et al., 2019). Gunduz et al. even highlighted RF as a suitable classification algorithm to deal with HDLSS data (Gunduz and Fokoue, 2015).

At each measure point, we determine the optimal hyper-parameters for the RF learning algorithm via grid search and k-fold cross-validation. Then, we train a final model with the tuned hyper-parameters. *Sampling and Parameter Setting:* As listed in Table 2, only 97 failed trucks are available on 01/01/2017. Consequently, 80,851 trucks have not yet failed. From a machine learning perspective, a 50:50 distribution between the two classes is generally considered most favorable for binary classification. To balance both classes, two common ways exist. One is *Random Oversampling (ROS)* (Turki and Wei, 2016), which randomly copies instances from the minority class c^+ . The other way is *Random Undersampling (RUS)* (Hasanin et al., 2019), which randomly removes instances from the majority class c^- .

ROS is not suitable from a domain-specific point of view, because the 97 known instances of class c^+ only have a similarity of about 50% to the failures in \mathcal{T}' . Thus, randomly copying instances of class c^+ does not lead to any improvement. Using RUS, 80,754 trucks of the majority class c^- , i. e., 99.76% of the trucks in \mathcal{T} , must be removed in order to achieve the aspired 50:50 class distribution. This massive removal consequently leads to a loss of information.

To address this issue and to investigate the possible effects of random sampling on a classification result, we parameterize RUS for our study with a self-defined "*Negative Sampling Factor*" (*NSF*). For instance, a NSF of 1 means that for each of the 97 failures (class c^+), one randomly selected counterexample of a nonfailed instance (class c^-) is added to them. The training data thus consists of 97 failures and 97 non-failures. Another example with a NSF of 5 means that five nonfailed trucks are added to each of the 97 failures, i. e., 485 trucks of class c^- . In our evaluation, we carried out a grid search with a value range for NSF between 1-100 with step size 1. In Section 5, we present the best TPR scores with their related NSF. *Feature Selection:* The feature space \mathcal{X} contains 3,000 unique features for truck configurations. Note that \mathcal{X} does not contain a single feature that allows for learning a discriminant function with only this feature, e. g., a particular engine. Nevertheless, in order to mitigate the risk of overfitting, we tested common feature selection techniques such as *Forward Selection*, *Backward Elimination* (John et al.,

1994), *Principal-Component-Analysis (PCA)* (Pearson, 1901) and *Boruta* (Degenhardt et al., 2017). However, none of them improved the quality of \mathcal{T} . In fact, the classification results were even poorer than the baseline. This is due to the concept shift (CS2), which causes techniques to consider many features as unimportant that are actually important. Specifically, this results from the fact that the trucks gradually fail. As a result, decision boundaries identified at different points in time may sometimes deviate more or less from actual patterns. Thus, feature selection does not make sense and is not relevant to us because it biases the classification results until all samples of class c^+ are visible in the data, i. e., when the guarantee period has expired. Therefore, we do not report any results for using feature selection in our experiments.

Hardware and Software Setup: We have carried out all experiments on a computer with an Intel(R) Core(TM) i7-6820HQ CPU with 8 cores @ 2.70GHz, 32 GB RAM and Windows 10 as operating system. The data of the produced trucks in \mathcal{T} and \mathcal{T}' are stored in a relational database using *Microsoft SQL Server 2014* with *Transact-SQL* as programming language for data pre-processing and sampling. For training the classifiers \mathcal{M} , we use R version 3.6.1². The corresponding R packages are `randomForest` version 4.6-14³ and *Classification and REgression Training* (`caret`) version 6.0-88⁴.

4.2 Evaluation Scenarios

According to our findings in Section 3.2, we evaluate two classification scenarios, (1) *Incremental Learning* and (2) *Windowing*. As shown in Figure 1, the use case data is already split into training (\mathcal{T}) and test (\mathcal{T}') datasets. As a performance score, we report the sensitivity, which corresponds to the True-Positive-Rate (TPR) of each classifier \mathcal{M} applied on \mathcal{T}' with data status on 01/01/2022. So, this TPR indicates how correctly a classifier predicts all failures of trucks in \mathcal{T}' that have taken place between 01/01/2017 and 01/01/2022. For each result, we train ten *RF* classifiers and average their TPR to obtain the final TPR.

Scenario 1: Incremental Learning is suitable to our use case in order to assess the impacts of gradual label and therefore concept shifts (CS2). This is due to the fact that incremental learning follows the logic that the more time passes, the more failures are present in the training dataset \mathcal{T} , i. e., more labels have shifted from c^- to c^+ . This in turn may improve the predictive quality. We start our evaluation scenario with the

training of the first classifier \mathcal{M}_1 on 01/01/2017 (t_1) with 97 failures contained in \mathcal{T} . We then apply this classifier to a subset $\mathcal{T}'_1 \subset \mathcal{T}'$ of only those trucks produced in January 2017.

One month later (t_2), we train a new classifier \mathcal{M}_2 with the same set of trucks in training dataset \mathcal{T} . Until then, however, 27 new failures have happened, so that these 27 trucks in \mathcal{T} shift their class label from c^- to c^+ . Hence, \mathcal{T} then comprise in total 124 failed trucks and positive class labels c^+ instead of the previous 97. We use this classifier \mathcal{M}_2 to predict the labels of trucks produced in February 2017 ($\mathcal{T}'_2 \subset \mathcal{T}'$). We correspondingly continue this procedure month by month until the end of 2017, i. e., until 12/2017 (t_{12}). Table 4 shows the evaluation results with various points in time from t_1 to t_{12} . In addition, the c^+ ratio improves with each passing month. Thus, a monthly increase in prediction performance is expected over time. We discuss this hypothesis in Section 5.1.1.

Scenario 2: Windowing is suitable to investigate the trade-off relationship between CS1 and CS2 in more detail. We expect negative effects of CS1 when using older trucks in \mathcal{T} for classifier training. This is because these older trucks are less similar in their feature values to those in \mathcal{T}' than newer trucks. Conversely, we expect negative impacts from CS2 when using newer trucks as training data because these newer trucks have a lower c^+ ratio.

In order to investigate this empirically, we divide the entire training dataset \mathcal{T} into 8 subsets \mathcal{T}_i , each with a fixed window size of one quarter, i. e., three months (see Table 5). We also examined a window size of 1 month each. However, we chose to present the results with quarters as window size instead of single months, as both provide the same insights. In addition, the representation is clearer with quarters as window sizes. The study starts with samples of trucks that have been produced in Q1/2015 (\mathcal{T}_1). After that, the window goes one quarter forward and takes the trucks produced in Q2/2015 (\mathcal{T}_2), followed by trucks produced in Q3/2015 (\mathcal{T}_3), and so on. For each quarter, we train a classifier \mathcal{M}_i with each subset \mathcal{T}_i of the training data. We then apply this classifier \mathcal{M}_i on the whole test dataset \mathcal{T}' to obtain the TPR. This way, we examine whether the older or the newer subsets \mathcal{T}_i of the training data are more likely to result in better TPRs. This helps to assess the negative impacts of CS1 (older data) and of CS2 (newer data), as well as the trade-off between them.

Baseline for the evaluation is the TPR of 55.9% (see Section 2.1). This baseline corresponds to a solution that does not explicitly address DSS. We may use this baseline to examine whether approaches to

²<https://www.r-project.org/>

³<https://rdocumentation.org/packages/randomForest/>

⁴<https://rdocumentation.org/packages/caret/>

Table 4: Incremental Learning: Overview of TPR scores and relevant meta-features of \mathcal{T} at monthly training times t_i .

Time of training \mathcal{M}_i	c^+	c^+ ratio	$\varnothing MIS$	$\mathcal{V}S_{c^+}$	TPR	NSF
t_1 : 01/2017	97	0.12%	10.90	48.75%	61.21%	5
t_2 : 02/2017	124	0.15%	11.64	49.59%	60.92%	9
t_3 : 03/2017	152	0.19%	12.48	49.15%	62.00%	22
t_4 : 04/2017	188	0.23%	13.48	43.38%	58.80%	19
t_5 : 05/2017	222	0.27%	14.47	50.24%	58.74%	1
t_6 : 06/2017	272	0.34%	15.47	46.69%	59.26%	12
t_7 : 07/2017	320	0.40%	16.47	48.89%	59.46%	24
t_8 : 08/2017	379	0.47%	17.47	46.44%	59.50%	18
t_9 : 09/2017	440	0.54%	18.47	48.04%	58.55%	5
t_{10} : 10/2017	488	0.60%	19.47	39.59%	57.92%	17
t_{11} : 11/2017	554	0.68%	20.47	45.41%	57.90%	3
t_{12} : 12/2017	640	0.79%	21.47	40.36%	57.88%	2

incremental learning or windowing really address our DSS and thus lead to improvements in the TPR or not.

5 EVALUATION

We report the results of our experimental study according to the setup described in Section 4. First, we discuss the classification results of the two approaches of incremental learning and windowing in the respective evaluation scenarios (5.1). Then, we evaluate data-driven classifiers from a domain-specific point of view, i. e., the concrete added value for truck manufacturers by saving monetary warranty costs (5.2).

5.1 Results for Evaluation Scenarios

Basis for the two approaches of incremental learning and windowing in the evaluation scenarios is the use case introduced in Section 2.1. Here, data scientists train a model early in 2017. So, they use the training data \mathcal{T} with a data status in this year, i. e., the statuses t_1 to t_{12} for the incremental learning approach (see Table 4). For the evaluation of windowing (Table 5), we use the fixed data status on 01/01/2017. For the discussions of the evaluation results, we use the meta-features c^+ ratio, $\varnothing MIS$ and $\mathcal{V}S_{c^+}$ to characterize the impacts of the two dataset shifts CS1 and CS2. Note that we use $\mathcal{V}S_{c^+}$ instead of $\mathcal{V}S$, because our paramount interest are trucks that will fail in future. Furthermore, we have listed the best NSF value for each measurement in both Table 4 and Table 5.

5.1.1 Results for Incremental Learning

Here, we train different classifiers monthly with new labels updated each month (t_{1-12}). We then ap-

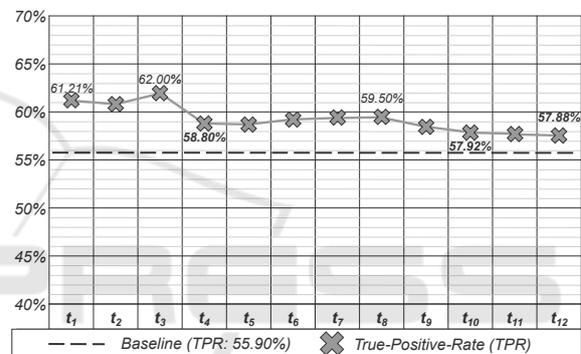


Figure 2: Incremental Learning: the y-axis shows the TPR scores and the x-axis the different points in time t_1 to t_{12} .

ply each classifier to the next batch of manufactured trucks (\mathcal{T}'_i). With the monthly changing data statuses in 2017, i. e., t_1 to t_{12} , the expectation is to start with a less performant model at t_1 . Nevertheless, the more time passes, the better the predictive performance is supposed to become, since the c^+ ratio increases monthly. This is because the formation of current to final class patterns converges over time.

Figure 2 shows the TPR scores for each point in time t_i . At time point t_1 , the TPR is 61.21% and can even improve to 62% by t_3 . So, this at first glance corresponds to the expected behavior that the TPR increases over time. However, this expectation is contradicted with following time points, where the TPR tends to get worse. With 57.88%, the final incremental TPR at the end of the time line (t_{12}) is even lower than the one at t_1 . This incremental TPR is significantly influenced by two previous time points: t_4 and t_{10} . The two drops in performance at t_4 and t_{10} can be explained by the meta-feature of truck similarity $\mathcal{V}S_{c^+}$ (see Table 4). On average, the value of $\mathcal{V}S_{c^+}$ is 46.38%, measured from t_1 to t_{12} . Especially at

Table 5: Windowing: Overview of TPR scores and relevant meta-features for time-consecutive training subsets (windows) \mathcal{T}_i . The data status for training is always 01/01/2017.

Production window \mathcal{T}_i	c^+	c^+ ratio	$\varnothing MIS$	$\mathcal{V}S_{c^+}$	TPR	NSF
\mathcal{T} : 2015/16	97	0.12%	10.90	49.95%	55.90%	5
\mathcal{T}_1 : Q1-2015	9	0.11%	21.35	50.65%	44.83%	17
\mathcal{T}_2 : Q2-2015	29	0.30%	18.32	52.43%	51.19%	11
\mathcal{T}_3 : Q3-2015	27	0.26%	15.53	53.89%	55.71%	8
\mathcal{T}_4 : Q4-2015	17	0.16%	12.40	54.35%	61.51%	4
\mathcal{T}_5 : Q1-2016	8	0.08%	9.13	60.31%	69.34%	3
\mathcal{T}_6 : Q2-2016	7	0.06%	6.20	65.11%	56.27%	1
\mathcal{T}_7 : Q3-2016	0	0.00%	3.17	-	-	-
\mathcal{T}_8 : Q4-2016	0	0.00%	0.90	-	-	-

the time points t_4 , t_{10} and t_{12} , the $\mathcal{V}S_{c^+}$ values are however much lower than this average with 43.28%, 39.59% and 40.36%, respectively. Thus, despite increasing c^+ ratio, the failure behavior represented by t_4 , t_{10} and t_{12} is increasingly different from the failure behavior of current production batches (\mathcal{T}'_i). We hence conclude that the covariate shift (CS1) has a significant impact on the poor TPR scores.

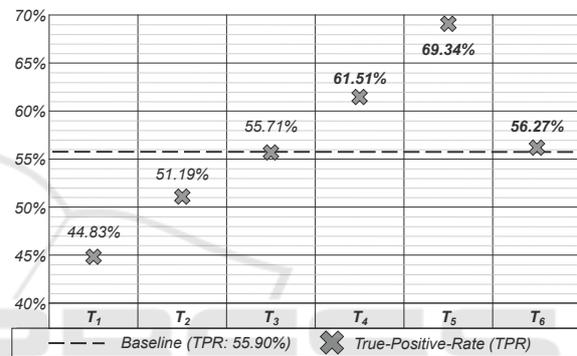
The final TPR score with incremental learning is the 57.88% at time point t_{12} . Although this is much lower than expected, it still constitutes an increase of 1.98%-points compared to the baseline. Note that this small improvement may be achieved despite the above-mentioned negative effects of the covariate shift (CS1). So, incremental learning seems to address the other dataset shift, i. e., the concept shift (CS2), at least to a moderate degree. This is also evident, as the c^+ ratio increases monthly (see Table 4).

Altogether, the behavior of the incremental TPR and our discussion above prove the non-trivial trade-off relationship between CS1 and CS2 and that it constitutes a challenge for approaches to incremental learning. In the following, we investigate this trade-off in more detail via the approach of windowing.

5.1.2 Results for Windowing

Here, we split the dataset \mathcal{T} into eight training subsets (\mathcal{T}_{1-8}). Since the subsets \mathcal{T}_{7-8} do not contain failed trucks, we can only train classifiers with the subsets \mathcal{T}_{1-6} , respectively the windows (\mathcal{T}_i). We then apply each trained classifier on all trucks in \mathcal{T}' and measure the resulting TPR scores for each window \mathcal{T}_i .

Figure 3 shows the TPR scores for each window \mathcal{T}_i . For windows \mathcal{T}_1 to \mathcal{T}_5 , it is noticeable that the TPR scores get better with each window and thus with decreasing truck age. This is due to the fact that although the c^+ ratio is higher in the first windows, the lack of similarity leads to a weaker predictive perfor-


 Figure 3: Windowing: the y-axis shows the TPR scores and the x-axis the different windows \mathcal{T}_1 to \mathcal{T}_6 .

mance (TPR). Thus, older trucks in \mathcal{T}_1 are less similar to the failed trucks in \mathcal{T}' than newer trucks from \mathcal{T}_5 .

Going further with the subsequent window \mathcal{T}_6 , the TPR value decreases by about 13%-points although the $\mathcal{V}S_{c^+}$ value is higher than at \mathcal{T}_5 . This drop in performance can be explained by the decrease in c^+ ratio due to the low average MIS of the trucks in \mathcal{T}_6 . Among all subsets, the covariate shift (CS1) has the lowest negative impact on the TPR at \mathcal{T}_6 ($\mathcal{V}S_{c^+}$: 65.11%), because these trucks are most similar to the current production. This is contrasted by the concept shift (CS2), which has the highest negative impact on the TPR at \mathcal{T}_6 (c^+ ratio: 0.06%), among all subsets.

Moving from subset \mathcal{T}_5 to the left, the negative effect of CS2 decreases by the fact that the trucks have already been in use for a longer time. So, they had a higher chance to fail and the c^+ ratio is usually higher for subsets with older trucks. However, it is evident from the $\mathcal{V}S_{c^+}$ values in Table 5 that older failed trucks in early subsets \mathcal{T}_i are less similar to the failed trucks in the test set \mathcal{T}' . So, an higher average age of trucks leads to an increase in the negative effects of CS1. Summarized, both dataset shifts have

negative impacts on the True-Positive-Rate. Thus, we cannot state which dataset shift is more dominant.

Only the training of classifiers on subsets \mathcal{T}_4 and \mathcal{T}_5 leads to an increase of the TPR compared to the baseline. Training the classifier on \mathcal{T}_4 leads to an increase of the TPR by 5.61%-points, whereas training on \mathcal{T}_5 leads to an increase of 13.44%-points. The window \mathcal{T}_5 represents the most balanced dataset, as both c^+ ratio and truck similarity $\mathcal{V}S_{c^+}$ have high values (see Table 5). Trucks in \mathcal{T}_5 are old enough (\emptyset MIS: 9.13) to have an adequate number of failures (c^+ ratio: 0.08%), and those failed trucks have a adequate similarity ($\mathcal{V}S_{c^+}$: 65.11%) to the failed trucks in \mathcal{T}' .

5.2 Domain-specific Evaluation

Now we discuss the concrete added value that data-driven classifiers imply in terms of saving monetary warranty costs. In this context, we compare the potentials of cost savings yielded by the employed approaches to incremental learning and windowing with the baseline. This baseline is a classifier trained on the entire training dataset \mathcal{T} without explicitly addressing the two dataset shifts and results in a TPR of 55.9%.

In order to highlight the potential cost savings in practical use, we assume that each repair of the particular damage leads to warranty costs of about \$2,500. Note that this is a conservative assumption because the scope of components that may fail is very wide. Corresponding to the 4,585 failed trucks in \mathcal{T}' , this results in a total of \sim \$11.4 million in warranty costs.

The final TPR achieved by the approach to incremental learning is 57.88%. So, it can reduce the failure rate by 1.98%-points compared to the TPR of the baseline. With 4,585 failed trucks contained in \mathcal{T}' (see Table 2), this leads to about 91 less truck failures. It hence results in potential cost savings of about \$0.2 million. The cost savings yielded by windowing are however much higher. Here, the best TPR score of 69.34% is achievable with the training subset \mathcal{T}_5 , resulting in a reduction of the failure rate by 13.44%-points. So, we may prevent with windowing 616 more failures, resulting in potential cost savings of approximately \$1.5 million. This is a significant reduction of avoidable warranty claims and consequential costs.

The experimental analyses show that the best classifier to deal with combined DSS is trained with the Random Forest algorithm, NSF sampling, quarter-wise windowing, and grid search hyperparameter optimization. In practice, it is however difficult and complex to find the best training subset, i. e., \mathcal{T}_5 . This is due to the fact that a final TPR score can only be measured after waiting a period of several years or at least the warranty period of the future trucks. There-

fore, further research is needed to devise approaches that may early identify those windows yielding a high predictive performance of the classifier without waiting that long period of time. In contrast, incremental learning can directly be applied and save \$0.2 million.

6 CONCLUSION AND FUTURE WORK

In this paper, we discuss our experimental study for a real-world use case of a data-driven prediction of product failures. We provide insights into the data characteristics in such real-world scenarios and into two kinds of dataset shifts (DSS) that result from these data characteristics: a covariate shift (CS1) and a concept shift (CS2). In contrast to the assumptions made in literature, these two kinds of dataset shifts usually occur together in real-world use cases. Furthermore, both DSS show a trade-off relationship, i. e., choosing data to avoid one, risks making the other one worse. With our experimental study, we prove that existing approaches to addressing DSS, e. g., incremental learning and windowing especially struggle with this trade-off relationship between CS1 and CS2. Nevertheless, our evaluation shows that both approaches may still be used to train classifiers that yield better results than the baseline of a classifier that does not address DSS at all. For instance, the use of incremental learning leads to a True-Positive-Rate (TPR) of 57.88%. This still outperforms the baseline by 1.98%-points. Although these TPR scores are low from a data science perspective, this has the potential to realize monetary cost savings for manufacturers.

In future, we are going to develop novel approaches to pre-process the data in order to adequately address both covariate shift (CS1) and concept shift (CS2). For instance, a domain-specific sampling strategy may incorporate the meta-features $\mathcal{V}S_{c^+}$ and MIS to select only trucks for the training dataset that are less affected by the two dataset shifts.

ACKNOWLEDGEMENTS

The authors thank the German Research Foundation (DFG) and the Ministry of Science, Research and Arts of the State of Baden-Wuerttemberg for financial support of this work within the Graduate School of Excellence advanced Manufacturing Engineering.

REFERENCES

- Bang, S. H., Ak, R., Narayanan, A., Lee, Y. T., and Cho, H. (2019). A survey on knowledge transfer for manufacturing data analytics. *Computers in Industry*, 104:116–130.
- Bifet, A. and Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SDM*.
- Boiko Ferreira, L. E., Murilo Gomes, H., Bifet, A., and Oliveira, L. S. (2019). Adaptive random forests with resampling for imbalanced data streams. In *2019 IJCNN*, pages 1–6.
- Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6:1579–1619.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- Degenhardt, F., Seifert, S., and Szymczak, S. (2017). Evaluation of variable selection methods for random forests and omics data sets. *Briefings in bioinformatics*, 20.
- Dharani Y., G., Nair, N. G., Satpathy, P., and Christopher, J. (2019). Covariate shift: A review and analysis on classifiers. In *2019 GCAT*, pages 1–6.
- Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. (2015). Learning in nonstationary environments: A survey. *IEEE CIM*, 10(4):12–25.
- Domingos, P. and Hulten, G. (2002). Mining high-speed data streams. *Proceeding of the Sixth ACM SIGKDD*.
- Ducange, P., Marcelloni, F., and Pecori, R. (2021). Fuzzy hoeffding decision tree for data stream classification. *Int. Journal of Comput. Intell. Systems*, 14(1):946.
- Elwell, R. and Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531.
- Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Embreck, F., Pfharinger, B., Holmes, G., and Abdessalem, T. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9-10):1469–1495.
- Gunduz, N. and Fokoue, E. (2015). Robust classification of high dimension low sample size data. *arXiv: 1501.00592*.
- Hasanin, T., Khoshgoftaar, T. M., Leevy, J., and Seliya, N. (2019). Investigating random undersampling and feature selection on bioinformatics big data. In *2019 IEEE BigDataService*, pages 346–356.
- Hirsch, V., Reimann, P., and Mitschang, B. (2019). Data-driven fault diagnosis in end-of-line testing of complex products. In *2019 IEEE DSAA*, pages 492–503.
- Homayoun, S. and Ahmadzadeh, M. (2016). A review on data stream classification approaches. *Journal of Advanced Computer Science & Technology*, 5(1):8.
- Iwashita, A. S. and Papa, J. P. (2019). An overview on concept drift learning. *IEEE Access*, 7:1532–1547.
- John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In Cohen, W. W. and Hirsh, H., editors, *Machine Learning Proceedings 1994*, pages 121–129. Morgan Kaufmann, San Francisco (CA).
- Khan, A. and Usman, M. (2015). Early diagnosis of alzheimer’s disease using machine learning techniques: A review paper. In *2015 IC3K*, volume 01, pages 380–387.
- Khoshkangini, R., Pashami, S., and Nowaczyk, S. (2019). Warranty claim rate prediction using logged vehicle data. In Moura Oliveira, P., Novais, P., and Reis, L. P., editors, *Progress in AI*, pages 663–674.
- Kull, M. and Flach, P. (2014). Patterns of dataset shift. *First International Workshop on Learning over Multiple Contexts (LMCE) at ECML-PKDD*.
- Losing, V., Hammer, B., and Wersing, H. (2018). Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing*, 275:1261–1274.
- Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363.
- Maitín, A. M., García-Tejedor, A. J., and Muñoz, J. P. R. (2020). Machine learning approaches for detecting parkinson’s disease from eeg analysis: A systematic review. *Applied Sciences*, 10(23).
- Marron, J., Todd, M., and Ahn, J. (2007). Distance-weighted discrimination. *Journal of the American Statistical Association*, 102:1267–1271.
- Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., and Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530.
- Nalbach, O., Linn, C., Derouet, M., and Werth, D. (2018). Predictive quality: Towards a new understanding of quality assurance using machine learning tools. In *Business Information Systems*, pages 30–42. Springer International Publishing.
- Pearson, K. (1901). LIII. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.
- Prytz, R., Nowaczyk, S., Rögnvaldsson, T., and Byttner, S. (2015). Predicting the need for vehicle compressor repairs using maintenance records and logged vehicle data. *Engineering Applications of Artificial Intelligence*, 41:139–150.
- Quiñonero-Candela, J., Sugiyama, M., Schwaighofer, A., and Lawrence, N. D., editors (2008). *Dataset Shift in Machine Learning*. The MIT Press.
- Turki, T. and Wei, Z. (2016). A greedy-based oversampling approach to improve the prediction of mortality in mers patients. In *2016 Annual IEEE Systems Conference (SysCon)*, pages 1–5.
- Utgoff, P. E. (1989). Incremental induction of decision trees. *Machine Learning*, 4(2):161–186.
- Wu, S. (2013). A review on coarse warranty data and analysis. *Reliability Eng. & System Safety*, 114:1–11.