Optimizing Early Detection of Production Faults by Applying Time Series Analysis on Integrated Information

Thomas Leitner* and Wolfram Wöß†
Institute for Application Oriented Knowledge Processing, Johannes Kepler University Linz
Altenberger Straße 69, Linz, Austria
Email: *thomas.leitner@jku.at, †wolfram.woess@jku.at

***Abstract*—According to the *Industry 4.0* initiative, industry aims for total automation and customizability using sensors for data retrieval, computer systems such as clusters and cloud services for large-scale processing, and actuators to react in the production environment. Additionally, the automotive industry is focusing increasingly on gathering information from the after-sales market using sensors and diagnostic mechanisms. All this information enables more accurate classification of faults when cars malfunction or exhibit undesired behaviour. Since finding systematic faults as quickly as possible is key to maintaining a good reputation and reducing warranty costs, techniques must be established that recognize increasing occurrences of fault types at the earliest possible point in time. Several sources of information exist that store heterogeneous datasets of varying quality and at various stages of approval. Using as much data as possible is fundamental for accurately detecting critical developing faults. In order to appropriately support the combination of these different datasets, information should be treated differently depending on its data quality. To this end, a concept to optimizing early fault detection consisting of four components is proposed, each of them with a particular goal; (i) determination of data quality metrics of different datasets storing warranty data, (ii) analysis of univariate time series to generate forecasts and the application of linear regression, (iii) weighted combination of course parameters that are calculated using different predictions, and (iv) improvement of the system accuracy by integrating prediction errors. This concept can be employed in various application areas where multiple datasets are to be analyzed using data quality metrics and forecasts in order to identify negative courses as early as possible.**

Keywords–data mining; time series analysis; data quality metrics; automotive industry.

I. Introduction

In recent years the capabilities of storing large data volumes that originate in various industries, ranging from the manufacturing industry to social web companies lie beyond the possibilities of analyzing them. From the management perspective, information hidden in raw data from various data sources provides decision support and guidance, and is therefore gaining importance. In order to draw reliable conclusions, new sophisticated ways of processing these large datasets are required.

In cooperation with the industrial partner *BMW Motoren GmbH* (engine manufacturing plant), located in Steyr, Austria, the *Quality - abnormality and cause analysis* (Q-AURA) application has been developed and is currently being improved and optimized. The core functionality of Q-AURA is to shorten the problem-solving time for finding causes of automobile engine faults in the after-sales market. The system consists of several components that support the quality management experts in their daily work. The first step in the Q-AURA analysis process is to find significant faults (i.e., those with negative consequences) to determine which fault types are occurring increasingly in the after-sales market. The result is a set of significant faults, which are analyzed further by calculating histograms and attribute distributions of engines that are affected by the same fault type in order to identify similarities between them. The last step is to analyze bills of materials (BOM) consisting of engine parts, components, and technical modifications from the development department to determine a set of modifications, that is the most likely cause of a particular fault. Q-AURA was evaluated positively and is already being used by quality management experts in their daily work. Although it delivers good results, improvements are being considered to further enhance Q-AURA's functionality. Currently, the application uses only one dataset from one warranty information system to determine critical developing (significant) faults. Since datasets residing in other information sources store warranty and after-sales information at various stages of approval, an extension is needed that integrates them into Q-AURA. This would provide an improved overall view of real-world situations and allow techniques to be used that help to find significant faults earlier, but must be thoroughly validated to achieve robust results [1].

This paper focuses on a concept that uses data quality metrics to determine dataset quality, time series analysis including forecasting methods to reveal trends and predict future values, and weighting mechanisms for optimized combination of multiple datasets. The structure of this paper is as follows: Section II describes the requirements for such a concept and the associated research issues and challenges. Section III gives an overview of related approaches and describes different methods and mechanisms that are addressed and used by the proposed concept. Subsequently, Section IV introduces Q-AURA, details the proposed improvement, and presents its integration into the Q-AURA analysis process. Finally, Section V concludes the paper, providing an outlook on future improvements.

II. Research Issues and Challenges

As mentioned in Section I, the overall goal of the proposed concept is to earlier identify significant faults, which required rethinking the Q-AURA concept. Currently, only historic customer claims (from the last six weeks) are used to determine whether faults are significant. However, improving the approach requires not only data from previous weeks, but also predicting future values. Calculating future values based on past observations is challenging, because it introduces some degree of uncertainty. Therefore, we propose using multiple datasets to improve the prediction process. In detail

the proposed approach consists of four tasks, each of which addresses a particular challenge.

The first task is to *validate each dataset, which stores partially contradictory, complementary, and/or redundant information*. The business process addressed by Q-AURA begins in the early development phase and comprises the development of new, and the improvement of existing, benzine and diesel engine generations. The process ends in the after-sales market, where information about warranty claims and data generated during a car's usage is stored. If a customer experiences a particular fault, the car must be checked at a dealer's workshop. There, information about the car and the fault are retrieved and sent to the car manufacturer. Since BMW sells cars in many countries partly different classifications of faults exist, which may lead to discrepancies. These must be addressed and a solution found to obtain a correct and consistent overall view of the fault data. Additionally, datasets exist that store information at different stages of approval. Data quality metrics must be defined to determine numeric criteria that can be interpreted and used in further processing steps.

The second task deals with the challenge of *detecting critically developing faults as early as possible using time series and regression analysis*. This is important because each week a critical developing fault is identified earlier reduces warranty costs and simultaneously enhances customer satisfaction. In the current Q-AURA implementation regression analysis is performed on data from the latest six weeks, and thresholds are then applied to regression parameters to determine whether the fault is significant. This time period (of six weeks) proved to provide the best trade-off between early detection (using as few recent weeks as possible) and robustness. The only way to improve the concept was therefore to incorporate predictions. Forecasting methods are used to compute the most likely future performance, which is then used as input to regression analysis. The most suitable time series and prediction methods were evaluated and selected.

The third task concerns the development of a *verification and weighting mechanism to determine how different datasets should be treated in the analysis process*. Since multiple datasets are used to obtain more robust results, the best way of combining them must be found. Two types of weighting factors are central to the proposed concept: those based on the overall data quality metric of each dataset and those based on the prediction accuracy of a particular fault's course. The prediction accuracy can be calculated using the prediction of the previous week and the observation of the current week. The proposed concept also defines how the weighting factors are used to combine the data from these different datasets to finally determine whether an analyzed fault is significant or not.

The last task is the *integration of the invented concept into the Q-AURA application and its verification*. Therefore, the new Q-AURA concept is described to demonstrate the benefit of the improved approach.

The resulting approach consists of a set of methods that enables earlier detection of badly developing fault courses. First, data quality metrics of different datasets are calculated, which are then used for weighting. Time series analysis using forecasting mechanisms is applied to predict future values based on historical data from these different information systems. The prediction accuracy is determined on the basis of the following week's observation, which is also used as a weighting factor for the datasets. Finally, the calculated weighting factors and the regression parameters are used to determine the significance of a particular fault.

III. Related Work

This section presents related approaches, information about the methods applied and an overview of the concepts on which the proposed approach is based under three different headings: (i) data quality metrics including their assessment, (ii) analysis of univariate time series, and (iii) determination of forecasting accuracy.

A. Related Approaches

The proposed concept is tailored to the particular needs related to identifying significant faults using time series analysis, forecasts, and regression analysis based on data from multiple information systems. Other approaches exist that focus on similar topics.

Chan et al. [2] presented a case study of predicting future demands in inventory management. They focused on combining different forecasts to improve prediction accuracy compared to only using one forecast. Their approach differs from the presented approach in several aspects: it seems to use only one information source as dataset, applies different time series forecasting methods and calculates weighting factors based on the results of those different methods. A major difference between the introduced concept and the approach used by Chan et al. is that the presented concept is based on different data sources. Thus, different time series are generated leading to different predictions. Further, the combination step of the proposed approach is not carried out at the level of the forecast result, but later after the data has been evaluated. The results are then weighted based not only on accuracy metrics of the forecasts, but also on the data quality of the particular information source.

Research by Widodo and Budi [3] focused on predicting the yearly passenger number for six consecutive years using 11 time series. Their approach uses the mean squared error (MSE) to compare prediction accuracy. In their research work the comparison of forecasts is done using the same dataset. The following points distinguish the proposed approach from theirs: More than one dataset is used in the presented concept. The forecasts are calculated separately for each data source with the same forecasting method and are combined after evaluation. In the proposed concept the different forecasts are combined using two types of weighting factors, (i) weights based on the prediction error, and (ii) weights based on data quality.

In [4], the authors described a method for analyzing the lifetime of products using Weibull distributions. Their application area is focused on electronic components in the automotive industry. The approach employs a day-in-service metric to identify the potential lifetime of the products analyzed. Day-in-service specifies how long a product has already been in use. In the automotive industry the day-in-service metric usually begins with the delivery of the car to the customer. The *bathtub* reliability curve is used, representing lower reliability at the beginning and the end of product life. Their approach has a different objective than the proposed one: They want to know

how long the majority of components will survive before they fail. Hence, they are not interested in what (fault type) occurred and how it developed in recent weeks, but how reliable the products are across all fault types.

Montgomery et al. [5] published a detailed paper about combining forecasts from different methods, and particularly how they can be weighted for the best results in social sciences. They proposed an enhancement to the *ensemble Bayesian model averaging (EBMA)* method that improves accuracy and performance for social science applications. They evaluated their approach in two use cases: prediction of (i) the 2012 US election and (ii) the development of the US unemployment rate. EBMA is mentioned in various research papers and has proved useful in combining different prediction methods. Since the proposed concept in this research work integrates different datasets, data quality metrics must be used, as the quality of those different sources may vary. Also, it has to be outlined that the combination task is performed on the regression analysis parameters of each dataset, which is necessary to identify whether a particular fault is significant or not.

Armstrong [6] published an overview of requirements and the possible ways of combining forecasting methods. Various approaches were analyzed, and it was emphasized that the combination can be achieved using different forecasting methods, different datasets or both. When multiple datasets are analyzed, their heterogeneity may require that more than one forecasting method is used. The approaches investigated address similar use cases, since they seek to improve prediction accuracy using multiple datasets or methods. However, unlike the proposed one, none of these approaches implements a two-step method that uses weighting factors based on data quality evaluation for each dataset and regression analysis (including computed predictions) to determine a significant course.

B. Data Quality

Previously, the data quality of information stored in databases or data warehouses had often been neglected. Redman [7] described the impact of poor data quality at different levels of decision-making and the ensuing problems. Considerable effort has since been put into enhancing data quality and quality assurance, but there remains room for improvement; information derived from data in information systems continues to be of lower quality than expected. Heinrich et al. [8] presented statistics that show various problems due to poor data quality, and mentioned that awareness must be raised.

In many cases decision-makers do not know that the data from a particular information source is of poor quality [7]. Thus, not only must the quality of the stored data be improved as much as possible, but users of this data must be made aware that it is not completely reliable. In modern businesses, many automated procedures and processes exist that transform and aggregate stored data, and compute new values which are then used by other processes to derive and generate new information, decision parameters, and other content. Clearly, if the original data is of poor quality, all the workflows and subsequent processes that use this information generate even poorer results, which may lead to problems, incorrect decisions, or other negative consequences. Hence, it would be advisable that these workflows should not rely solely on the data assuming it is completely correct, but to use quality metrics that determine the level of uncertainty. If multiple information systems exist that store partially redundant information originating from different processes, subsequent processes can use all data from all systems to achieve a better overall view. In order to know how to treat information from these information systems, methods are required that consider and measure the quality of stored data. In the scientific literature, a variety of data quality metrics and dimensions have been defined and specified, each of them tackling a particular aspect of data quality [9][10]. Wang and Strong [11] defined the term data quality dimension as a set of data quality attributes that define a single aspect or construct of data quality. They aimed to categorize data quality metrics in terms of *accuracy of data*, *relevancy of data*, *representation of data*, and *accessibility of data*, while in [12] and [13] the classes were labeled *intrinsic data quality*, *accessibility data quality*, *contextual data quality*, and *representation data quality*. Naumann and Rolker [14] based their distinction on the usage and retrieval process of information, dividing data quality metrics into *subject-criteria scores*, *process-criteria scores*, and *object-criteria scores*. Other publications, among them [15] and [16], investigated dependencies and tradeoffs between data quality metrics. Note that data quality metrics can be determined in a task-dependent and a task-independent manner, depending respectively on whether they are computed with or without the contextual knowledge of their usage [17]. Such context can be included, for instance, by applying business rules or government regulations. Bobrowski et al. [18] distinguished between direct and indirect metrics, where the former are determined directly from the data, and the latter are computed from the former, taking the dependencies between them into account.

In accordance with these classifications, those data quality metrics that are important in the context of the proposed approach are identified and described below. In the application area of the proposed approach data is processed automatically, using a reliable connection. Consequently, data quality metrics concerning the representation or the accessibility of data are not relevant, since they do not describe the data itself. The *intrinsic* and *contextual* categories, however, are important in the addressed context. The *subject criteria* and *process criteria* classes according to Naumann and Rolker [14] are not relevant to the proposed concept, because they seal with how the user perceives the information or how the query processing is treated. In [16], a distinction was made between quality metrics related to a particular user's view and data-related quality metrics. Since the user's view is not important in the presented concept, only metrics that have an impact on the data itself are applied. The remaining data quality metrics that are relevant in the particular application scenario are *Completeness*, *Consistency*, and *Correctness*.

Completeness has been addressed in various research papers, with - in some cases - different interpretations of the definition depending on application area and point of view. Table I lists various contributions with different definitions of completeness. While some concentrate on the presence or absence of entries, others - such as Kahn et al. [19] and Ballou et al. [15] - take a closer look by evaluating whether the amount of information represented by the content is sufficient. Generally, a system is complete if it includes the whole truth. The completeness quality metric is often related to NULL values in databases and information systems. The general understanding is that a NULL value must be treated like a

missing value, but it may also be that it is not known whether it exists or that it does not exist at all, which describes a considerably different perspective on completeness [9]. This means that the conceptual organization of an information system can be seen from two different points of view called *closed world assumption* (CWA) and *open world assumption* (OWA). Under the CWA, all information captured by the information system represents facts of the real world and anything that is not described is assumed to be false. Under the OWA, it cannot be stated whether a fact not stored in the information system is false or whether it does not exist at all. In an OWA-based system that does not store NULL values, identifying the completeness of an information system requires the introduction of a new concept called reference relation. This concept stores all real-world facts with respect to the structure of the particular relation. In comparison to a relation of an information system storing all facts of the real world except one object, the reference relation would contain all information of the relation plus the missing object not captured by the relation. The metric completeness can be defined formally as follows. For a database scheme D, we assume a hypothetical database instance d_0 that perfectly represents all information of the real world that is modeled by D. Furthermore, we assume that one or more instances d_i $(i \geq 1)$ exist, each of them is an approximation of d_0. Next, we consider some views, where v_0 is an ideal extension of d_0 and v_i $(i \geq 1)$ are extensions of the instances d_i. Equation (1) represents this concept, where the absolute values represent the number of tuples [20][21].

$$\frac{|v_i \cap v_0|}{|v_0|} \tag{1}$$

Under the CWA, completeness is defined differently, because NULL values indicate entries that do not exist in the real world. Completeness can therefore be seen from the granularity perspective [10]. The following four types of completeness can be distinguished according to their granularity:

- *Value completeness*: When this type is applied, completeness is determined at the finest-grained level, and the ratio between existing values of particular fields and the total number of fields (including NULL values) is calculated.
- *Tuple completeness*: On a more general level tuple completeness represents the completeness of a particular tuple represented by the tuple's ID. For example, if a relation has four attributes and a particular tuple contains one NULL value, the completeness for this tuple would be 75%.
- *Attribute completeness*: Similar to tuple completeness, this describes the completeness value of a particular attribute. It is calculated as the ratio of existing values and the total number of tuples (containing NULL values).
- *Relation completeness*: This type of completeness is based on the number of NULL values and the total number of values in a whole relation.

It is important to analyze a particular application in detail to determine how NULL values are treated correctly, because they can have different meanings. For example, when the relational model is used there is often a primary key defined for a relation. Since members of the primary key cannot be NULL, missing objects cannot be expressed using NULL entries for these attributes. If a particular attribute is not member of the primary key, it can be NULL (assuming there are no NOT NULL constraints), and therefore it is possible to represent missing objects as NULL values. In the application area of the proposed concept, the scenario is similar, as unknown or non-existent features are represented as NULL values if the particular attribute is not in the set of primary key attributes. If an object exists in the real world but is not represented in the dataset, then no tuple is stored in the database, since primary key attributes cannot be set to NULL.

TABLE I. Completeness definitions in scientific papers.

Reference	Definition
extent to which the value is present for that specific data element	[7]
breadth, depth, and scope of information contained in the data	[11]
presence of all defined content on both data element and dataset levels	[15]
schema completeness is the degree to which entities and attributes are not missing from the schema; column completeness is a function of missing values in a column of a table	[17]
every fact of the real world is represented; it is possible to consider two different aspects of completeness; (i) certain values may not be present at the time, and (ii) certain attributes cannot be stored	[18]
extent to which information is not missing and is of sufficient breadth and depth for the task at hand	[19]
related to the Closed World Assumption (CWA); the information stores the whole truth	[22]
ability of an information system to represent every meaningful state of a real-world system	[23]
degree to which data values are included in a data collection	[24] (via [9])
percentage of real-world information entered in data sources and/or data warehouses	[25] (via [9])
information having all required parts of an entity's description	[26]
ratio between the number of non-NULL values in a source and the size of the universal relation	[27] (via [9])
all values that are supposed to be collected as per a collection theory	[28]

Consistency is a data quality metric whose definition is very similar across different research papers: multiple entries with the same meaning should be represented identically or in a similar way. Interestingly, consistency is often closely related to integrity and integrity constraints. Batini et al. [9] defined consistency as the ratio of values that do not violate specific rules and the overall information set. They stated that these rules can be either integrity constraints (referring to relational theory) or consistency checks in the field of statistics. Integrity constraints can be further subdivided into inter-relational constraints and intra-relational constraints, depending on whether the constraint relates to one or more tables. Pipino et al. [17] also defined consistency as closely connected to integrity constraints (e.g., Codd's Referential Integrity constraint). They proposed that consistency can be calculated as a ratio using the number of violations of a specific consistency check and the total number of consistency checks. Bovee et al. [26] defined consistency as a sub-metric of integrity dealing with different representations of the same information in multiple entries. A summary of the different definitions is listed in Table II.

In the context of the presented approach, consistency is considered as the entries' violation of - or, more specifically, their compliance with - rules that represent consistency checks.

TABLE II. Consistency definitions in scientific papers.

Reference	Definition
refer to the violation of semantic rules over a set of items	[9]
format and definitional uniformity within and across all comparable datasets	[15]
consistency of the same (redundant) data values across tables (e.g., Codd's referential integrity constraint); ratio of violations of a specific consistency type to the total number of consistency checks subtracted from 1	[17]
there is no contradiction in the data stored	[18]
requires that multiple recordings of the value(s) for an entry's attribute(s) be the same or closely similar across time or space	[26]
different data in a database are logically compatible	[28]

TABLE III. Correctness definitions in scientific papers.

Reference	Definition
[accuracy] data are certified error-free, accurate, correct, flawless, reliable, errors can be easily identified, the integrity of the data precisely	[11]
[free-of-error] number of data units in error divided by the total number of data units subtracted from 1	[17]
every set of data stored represents a real-world situation	[18]
[free-of-error] extent to which information is correct and reliable	[19]
[validity] the data sources store nothing but the truth	[22]
[accuracy] refers to information being true or error free with respect to some known, designated, or measured values	[26]
[accuracy] extent to which collected data are free of measurement errors	[28]
[accuracy] data are accurate when the data values stored in the database correspond to real-world values	[29]

It is calculated as the ratio of entries satisfying all consistency checks and the total number of entries. An example of such a consistency check is the proof of duplicates in the dataset.

Correctness is a metric that indicates whether the stored information is valid. A summary of different definitions from scientific papers is listed in Table III. Since different terms are often used for the same concept, the original attributes are given in brackets. Pipino et al. [17] provided a very technical definition that explains how the metric is calculated. In [18], the definition was very general, defining correctness of a particular dataset as the presence of a corresponding real-world subject. In this contribution, correctness is also seen as a valid representation of real-world entities. Semantic rules are required to determine whether a particular entry is correct or in the correct range. Since functional requirements can change over time, it is important to modify these rules if necessary [23].

It is very difficult to verify the correctness of data, since tacit information from domain experts is required in most cases. Hence, expert knowledge must be represented as a set of semantic rules, which are applied to the data in information systems to determine whether the content satisfies these conditions. Note that correctness heavily depends on the application area, which means that, even if a particular entry in a dataset complies with all rules of one application area, it might still fail checks of another.

C. Analysis of Univariate Time Series

The proposed approach uses time series analysis to estimate a model that fits the observed data and computes forecasts to determine future values. For this purpose, models must be compared in order to find the most suitable one. Since the application area is based on a single observed variable, we focused on methods that address univariate time series.

Time series analysis is a very popular research field and dates back to 1906, where Schuster recorded sunspot numbers in a monthly schedule, which was one of the first recorded time series. Nowadays, a wide variety of applications exists, ranging from stock analysis and calculations concerning demography to sunspot observations. The basic purpose of a time series is to capture a set of sequential observations over a time period. Methods are needed to compute a model for generating a time series with minimal differences between the observations and the model-generated data points [30]. Time series analysis has two major goals: (i) to express the underlying process that leads to the observations as accurately as possible, and (ii) to obtain a model that predicts future values based on the course of the time series. The smaller the difference between the generated course and the data points the better the model supposedly describes the underlying process. However, this statement is not entirely correct, since a model can also be fitted too closely to the curve (called overfitting), which means that it expresses the observations in too much detail, and also models outliers that might not have a systematic impact. Overfitting results in poorer out-of-sample prediction performance (calculating forecasts) than a model that is fitted less exactly. Time series analysis is closely connected to forecasts, since it focuses on the prediction of future values for a known time series. Weather forecasts are a popular example, where former observations are known and future values are predicted on their basis (considering the laws of physics) [30]. A very basic classification of time series distinguishes between univariate and multivariate types, depending on whether they focus on one or multiple target variables, respectively. Thus, different courses (variables) are analyzed for the same time period, which means that different features are observed at a single point in time (represented as vectors) [31]. The proposed approach focuses on univariate time series and the following time series models were compared to find the most suitable one for the application area.

Box-Cox Transformation, ARMA errors, Trend, and Seasonality (BATS) is a method introduced by De Livera et al. [32]. Since it uses Box-Cox transformations, it does not focus exclusively on linear homoscedastic time series, but also supports nonlinear ones. Furthermore, the method also considers ARMA errors, where ARMA parameters are evaluated and determined in a two-step procedure, as this leads to the best results [33]. Additionally, the trend component is computed using an adaption to the damped trend. The method incorporates mechanisms to deal with seasonal influences, as these often occur in time series. In [32], *Trigonometric BATS* (TBATS) was proposed as an extension to the BATS model, which replaces the seasonal definition of BATS with a trigonometric formulation. A method that was used very often in the past is *Simple Exponential Smoothing* (SES), which applies weights to the individual observations of the time series [34]. As the name indicates, these weights are not equally distributed but decrease exponentially over time giving more

recent observations a higher impact than previous ones. An extension to SES was introduced by Hyndman et al. [35]. They proposed a framework called *Exponential Smoothing State Space Model* that makes it possible automatically determining the best exponential smoothing algorithm and its parameters using state space models. Since their approach delivered good results on the M3-data, it was also investigated and tested in the context of the proposed concept. The quality criteria that is used by this framework is *Akaike's Information Criterion (AIC)* [35]. Another method that was introduced in the application area of demand forecasting is *Croston's Method* (CROSTON) [36][37], which uses multiple single simple exponential smoothing forecasts and treats zero observations separately (in the application area of demand modeling, these are the observations where the demand is zero). *Auto-Regressive Integrated Moving Average* (ARIMA), which belongs to the family of *Auto-Regressive Moving Average* ARMA models, is a popular method for fitting time series and forecasting. ARMA models consist of two components: the auto-regressive component (AR) and the moving average component (MA). The AR-component computes the dependencies between previous values/observations and their impact on the current observation, while the MA-component estimates the smoothing function for the observations in a particular time period. Various modifications to the ARMA model have been proposed, among them ARIMA, which considers also non-stationary processes [38]. Neural networks are used more often for time series analysis. A popular representative is the *feed-forward network with a single hidden layer* (NNETAR). Artificial neural networks are based on inputs and dependent variables; the parameters are transformed, weighted, and combined using one or more hidden or intermediate layers in order to determine the output variable. In [39], the authors presented a comparison of neural networks in different usage scenarios, and - based on recent research - concluded that the risk of over-parameterization is a well-known problem. Hence, they recommended using feed-forward neural networks with a single hidden layer [39].

D. Determination of the Forecasting Accuracy

In the presented concept, assessment of the quality and thus the reliability of a prediction is a key task. In order to determine the reliability of a predicted value, it is important to know how good the particular prediction is. Hence, predictions should be evaluated using new observations as soon as they become available. As this topic is often tightly coupled with time series analysis, many research papers have addressed it. Below, we provide an overview of error terms including their benefits and drawbacks, since these are the terms in which accuracy measures are often considered.

Hyndman and Koehler [40] distinguished between four different types of error measures: (i) scale-dependent measures, (ii) measures based on percentage errors, (iii) measures based on relative errors, and (iv) relative measures (Table IV). In addition to these categories they proposed a scale-independent metric called *Mean Absolute Scaled Error (MASE)*.

The first category of scale-dependent measures includes *Mean Squared Error (MSE)*, *Root Mean Squared Error (RMSE)*, *Mean Absolute Error (MAE)*, and *Median Absolute Error (MdAE)*. The problem with these metrics is that they cannot be compared easily across various time series of different scale. A wide range of applications use these metrics to determine the forecast accuracy of univariate time series [41]. Armstrong and Collopy [42] also addressed the problem arising from scale dependency. The second category is about measures based on percentage errors. Commonly used metrics are *Mean Absolute Percentage Error (MAPE)*, *Median Absolute Percentage Error (MdAPE)*, *Root Mean Square Percentage Error (RMSPE)*, and *Root Median Square Percentage Error (RMdSPE)*. An advantage of these methods is that they are scale-independent and therefore suited to comparing the forecasts of different time series. However, there are also some disadvantages: Firstly, it is not always guaranteed that they are finite or defined. MAPE, for example, encounters problems when a time series is close or equal to zero [39]. Additionally, MAPE and MdAPE come with the drawback that they treat positive errors worse than negative ones, which results in asymmetry. Makridakis [43] described extensions to these metrics in order to find symmetric error metrics, which are called *Symmetric Mean Absolute Percentage Error (sMAPE)* and *Symmetric Median Absolute Percentage Error (sMdAPE)* as an attempt to overcome the asymmetry problem. However, sMAPE and sMdAPE are less symmetrical as their names might imply: It has been shown that the resulting error is greater for overpredictions than for underpredictions by the same amount [39][44]. The third category of forecast accuracy metrics covers measures based on relative errors. Popular metrics of this category are *Mean Relative Absolute Error (MRAE)*, *Median Relative Absolute Error (MdRAE)*, and *Geometric Mean Relative Absolute Error (GMRAE)* [39][40][42]. The advantage of these methods is that the metrics not only compare the times series with the corresponding forecasts, but also compare it with predictions from a different forecasting method that serves as a benchmark method. In many cases, *random walk* is used for this purpose. The fourth category also defines measures on the basis of a comparison between the method applied and a benchmark method. The *Relative Mean Absolute Error (RMAE)* is defined as the ratio between the MAE of the applied method and the MAE of the benchmark method. Similar metrics can be calculated comparing error metrics of the applied model with those of the benchmark method (e.g., *Relative Mean Squared Error (RMSE)*). The im-

TABLE IV. Overview of forecast accuracy metrics.

Category	Metric	Definition
scale-dependent measures	MSE	Mean Squared Error
scale-dependent measures	RMSE	Root Mean Squared Error
scale-dependent measures	MAE	Mean Absolute Error
scale-dependent measures	MdAE	Median Absolute Error
percentage errors	MAPE	Mean Absolute Percentage Error
percentage errors	MdAPE	Median Absolute Percentage Error
percentage errors	sMAPE	Symmetric Mean Percentage Error
percentage errors	sMdAPE	Symmetric Median Percentage Error
percentage errors	RMSPE	Root Mean Square Percentage Error
percentage errors	RMdSPE	Root Median Square Percentage Error
relative errors	MRAE	Mean Relative Absolute Error
relative errors	MdRAE	Median Relative Absolute Error
relative errors	GMRAE	Geometric Mean Relative Absolute Error
relative measures	RMAE	Relative Mean Absolute Error
scale-independent measures	MASE	Mean Absolute Scaled Error

provement provided by the applied method is always expressed in relation to a benchmark method. The drawback of these measures is that they do not indicate an *absolute goodness* of the forecast itself.

IV. Improving Early Detection of Significant Faults in Quality Management

This section covers the Q-AURA analysis process, the invented improvements of it, and their integration into Q-AURA. Q-AURA is a system that supports quality management experts in analyzing faults occurring in the after-sales market. Defect and warranty information is gathered from car dealers who inspect customers' cars and detect faults. The business process relevant for Q-AURA, which ranges from the development of an engine to the after-sales market, is illustrated in Figure 1.

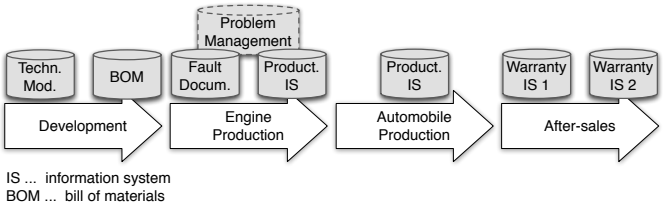


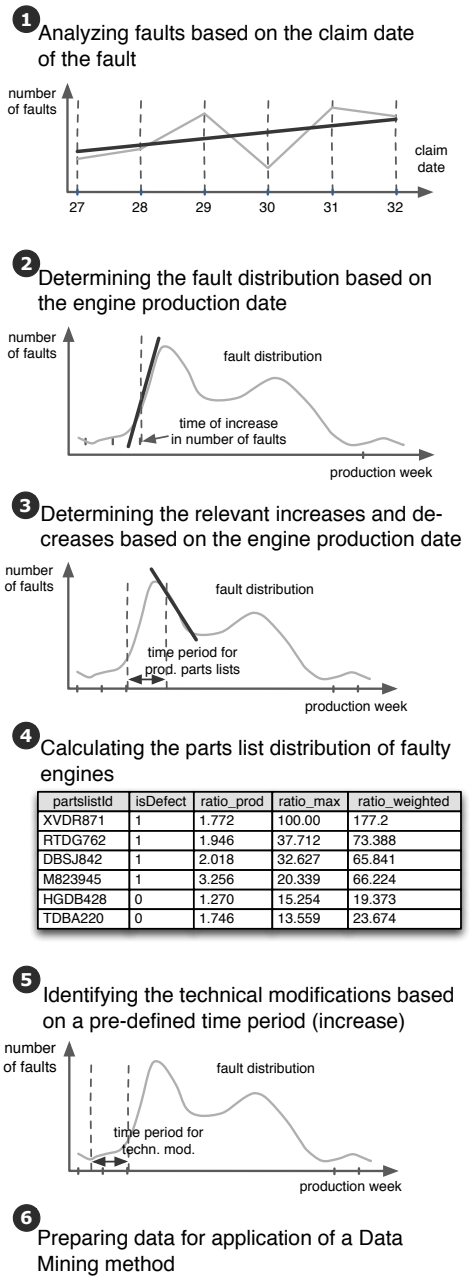
Figure 1. Flow chart of the business process relevant to Q-AURA.

Fault and warranty information is distributed across information systems, which contain partially redundant information. Since partially different data is also stored in the information systems, integration would result in a more complete, holistic view of real-world situations. In combination with Q-AURA's primary aim of identifying significant faults, this extension targets more accurate and robust results if the information is processed and interpreted correctly. Q-AURA's secondary aim of analyzing significant faults further in order to determine technical modifications that might underlie them requires additional information residing in information systems from other process steps. Therefore, these data sources must also be integrated to cover the whole engine lifecycle.

A. Q-AURA Approach

This section describes the Q-AURA approach and its analysis process, which forms the basis of the invented improvement [45]. The underlying analysis process is divided into different steps, which modify the information such that (i) data mining methods can be applied and (ii) the most appropriate representation of the data can be found. These six process steps are illustrated in Figure 2.

The first step is the identification of significant faults that occur in the after-sales market, which are then further analyzed (cf. Figure 2-1). The term *significant* is used for faults with negative consequences that have developed in recent weeks. The information base that is used for this step covers cars that were manufactured in the last three years. To detect faults that have occurred recently and indicate current problems, the last six weeks are considered. These boundaries were set carefully in order to take those cars into account that influence the ongoing development process. Since various engine types exist and since fault types have a different distribution depending on the car brand (e.g., BMW and MINI), the appropriate level of granularity for the analysis had to be found: finally, the



partslistId	isDefect	ratio_prod	ratio_max	ratio_weighted
XVDR871	1	1.772	100.00	177.2
RTDG762	1	1.946	37.712	73.388
DBSJ842	1	2.018	32.627	65.841
M823945	1	3.256	20.339	66.224
HGDB428	0	1.270	15.254	19.373
TDBA220	0	1.746	13.559	23.674

Figure 2. Q-AURA process in detail.

result was to classify faults according to fuel type, car brand, and engine type. Thus, faults that occur in BMW automobiles are not in the same analysis set as faults in engines that have the same engine type and fuel type, but are built into MINI automobiles. Regression analysis is used to determine significant faults [46]. Three different approaches to regression analysis based on convex functions, smoothing functions, and a straight line were tested to find the best method. The evaluation was done using fault courses from most of the analysis sets for diesel engines over several weeks. Experts from the diesel quality management department, who helped in finding the method that best identifies significant fault courses were contacted weekly. The evaluation revealed that the straight-line approach outperformed the others. Different

metrics of the regression line can be calculated to determine its characteristics. Q-AURA previously used *gradient*, *mean value*, and *coefficient of determination*. The *coefficient of determination* (indicating how well the regression line fits the actual course) and the mean value were replaced with a new metric called *gradient_ppm* (Equation (2)). This value is calculated as the ratio between the gradient and the number of faults (regardless of the fault type) of the engine type ($n_{enginetype}$).

$$k_{ppm} = \frac{k * 1.000.000}{n_{enginetype}} \quad (2)$$

Those faults that exceed specific thresholds are analyzed in more detail. These thresholds were investigated and evaluated carefully together with quality management experts. Faults that are not classified as significant are not analyzed further.

For each significant fault, the production week histogram is calculated in the second step (cf. Figure 2-2). The histogram is based on cars that were produced in the preceding three years with claims from the last two years. It shows the number of produced engines with the particular fault in relation to the total number of produced engines of the same class (according to fuel type, car brand, and engine type). This is done to take production fluctuations into account, because an increase in the number of engines produced will most likely affect the number of faults, but does not necessarily indicate a systematic failure during engine development. The course is then normalized by the highest value in order to identify more clearly the highest fault peaks in time. A 5-point smoothing function is applied to eliminate outliers. The resulting course forms the basis for identifying the critical time periods, which are bound by an initial significant increase and a decrease. An increase of the course, which is defined as the ratio between faulty engines and the total number of engines produced, indicates that one or more negative effects have occurred that influence product quality (e.g., a new technical modification that changed the engines). The identification of significant increases is illustrated in Figure 2-2.

Afterwards (cf. Figure 2-3), the decreases of the course are determined. Both steps (finding increases and decreases) are performed using sliding windows and calculating the slope. Subsequently, interesting time periods can be identified, each of which is bound by a significant increase and the subsequent decrease. Such a time period represents the time when most of the engines affected by the particular fault were produced.

In the next step (cf. Figure 2-4), the faulty engines identified for a time period are investigated in more detail. In order to determine more exactly which subset of them is affected most by a particular fault, the distribution of the engine material number is analyzed. The engine material number represents a particular bill of materials (BOM) and, therefore, defines an engine in much detail. The bill of materials specifies all components and parts that are necessary for assembling an engine. A BOM entry contains information such as part number, number required, and unit. More interesting for Q-AURA is that a BOM also stores the technical modification identifier. A technical modification describes the reason why a particular part is in the BOM and which former part it substitutes (if it is a substitution). Possible reasons could be a new supplier or that the former part lead to a quality issue. The BOM distribution is put in relation to the engines produced with the same engine material number to select those material numbers that have a bad ratio. The ratio is then normalized to identify the BOMs that must be analyzed further, since they are affected most by the analyzed fault.

Step 5 in Figure 2 illustrates how the technical modifications are selected. Not every technical modification that occurred throughout the whole time period analyzed is relevant, since a technical modification that was implemented months after the significant increase, cannot be the cause of the fault. Therefore, the time period from which technical modifications are selected can be limited, which is important because the number of technical modifications made over time is vast, which prevents application of intelligent methods and makes drawing meaningful conclusions difficult. In order to avoid being too strict and selecting insufficient modifications (and possibly missing the causative modification), a three-month period starting two months before a significant increase is used. This period was defined and evaluated together with quality management experts.

In the last step, the number of technical modifications is limited to those most likely to have provoked the fault (cf. Figure 2-6). Using the modifications determined in step 5 and the engine classification according to their engine material numbers, two alternatives were implemented that determine the relevant set of technical modifications. The first is a descriptive approach that identifies modifications that are covered by most of the significant engine material numbers, while the second uses association rules. More detailed information about these two methods can be found in [45].

This analysis concept, which forms the core of Q-AURA, is already in daily use by quality management experts at different engine production plants. The evaluation of the tool showed that it provides a significant benefit. The problem solving time for engines produced in the plant in Steyr was recorded in two consecutive years (before Q-AURA was applied and after its introduction). It showed that the reduction was approximately 2% [45].

B. Optimized Early Detection

This section describes the new improved concept in detail and shows the advantages over the current Q-AURA implementation. Clearly, early detection of faults that occur during development or production is crucial, since in most cases they result in negative effects for the company. As described in Section IV-A, Q-AURA is an application that identifies current problems (represented as engine faults) and automatically analyzes them in detail to gather more information about possible causes. This means that early detection is also an important task for Q-AURA. Since finding the causes of a particular fault is very time-consuming, improvements by a single day or even a week are highly beneficial. Thus, an approach was invented, which optimizes (i.e., accelerates) Q-AURA's fault detection method. The improved concept consists of four components, each fulfilling a different task: (i) assess information systems based on data quality, (ii) analyze univariate fault time series and compute forecasts, (iii) determine whether a particular fault is significant using predictions based on multiple information systems, and, (iv) evaluate the prediction accuracy to determine the quality of the forecasts.

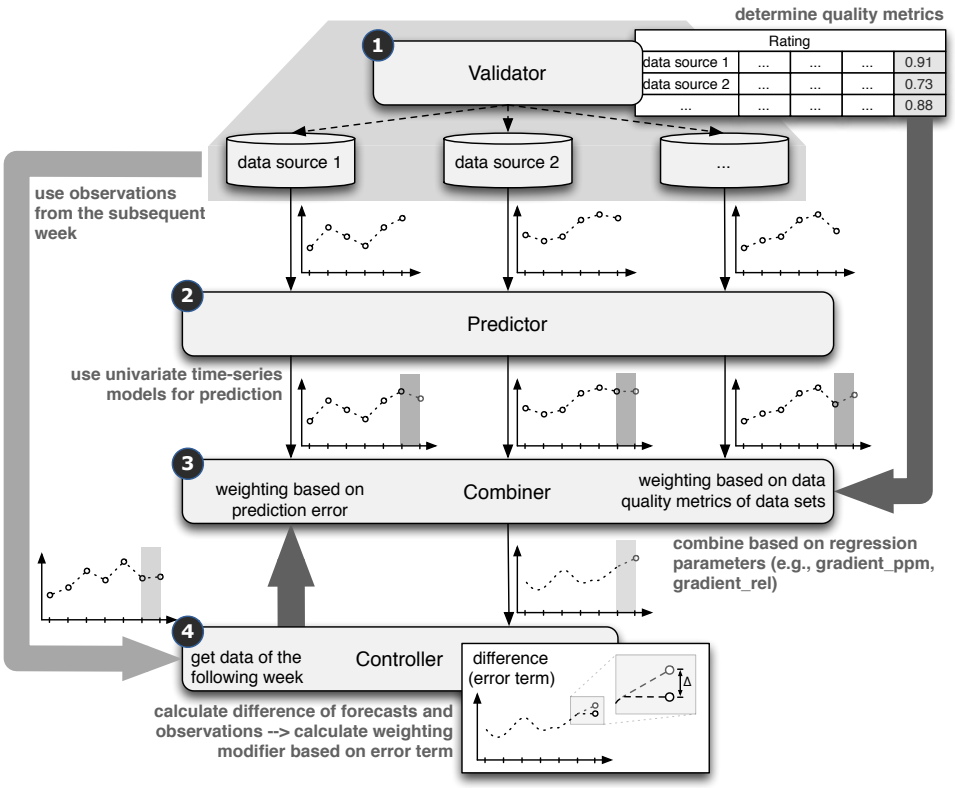


Figure 3. Concept for optimizing early detection.

The overall concept is illustrated in Figure 3. The *Validator* (cf. Figure 3-1) is responsible for determining a specific information system's data quality. Different data quality metrics are used (completeness, correctness, and consistency) to calculate the component's result, which constitutes an overall data quality metric for the particular information system. The *Predictor* (cf. Figure 3-2) analyzes the fault time series for each fault in each data source. This means that a model must be generated that describes the process underlying the time series as well as possible in order to be able to calculate a forecast (out-of-sample prediction). A single value is forecasted, which is then used to determine the significance of the particular fault. Regression analysis is applied considering a six-week period (containing the forecast value as the most recent one). In the subsequent step, the *Combiner* integrates weighting factors and the regression parameters of each data source's regression line to calculate an overall significance metric that indicates whether a particular fault is significant. Finally, the *Controller* determines the accuracy of each forecast. This is achieved by comparing new entries in the information systems from the following week. The prediction error is calculated for each data source using the new value and the predicted value of the previous week. This prediction error is then used to compute a weighting factor that is required by the combiner component.

1) Validator: The validator is responsible for determining the data quality of a particular information system. Various quality metrics from the scientific literature were compared to identify quality metrics that are relevant to the proposed approach. As described in Section III-B, the completeness, correctness, and consistency quality metrics are applied to compute the overall data quality metric.

Completeness is a data quality metric that has different interpretations in research because it can be seen from different perspectives. In the proposed concept multiple datasets exist that store partially redundant warranty and fault information. In industry, data that is used for intensive analytical processing is usually stored in an aggregated form in *data warehouses* (DWHs). Data warehouses are often designed to store historical information, while operational information systems capture only a short time period (to increase performance and throughput) [47]. In many cases, data marts are developed, which do not satisfy the third normal form of relational algebra, since they are organized to improve the performance of analytical queries and transformations. Figure 4 illustrates the DWH concept. Each intermediate step between the original information source and the data warehouse is a source of potential errors that may occur while transforming and cleansing data.

At the bottom-most level, various operational systems store the data as it is being generated. The data models support a particular business case, ensure that relevant information about real-world objects is inserted correctly, and verify the completeness at a particular level (primary key constraints, foreign key constraints, and not-NULL constraints are basic options to ensure this). At the next level, data warehouses are set up to provide an analytical basis for different business aspects. ETL processes extract, transform, and load data in preparation for DWH use cases. During the ETL process, some information may be filtered or left out due to unrequested transformation errors. Thus, completeness of the target DWH is reduced. Since different DWHs that store redundant information exist

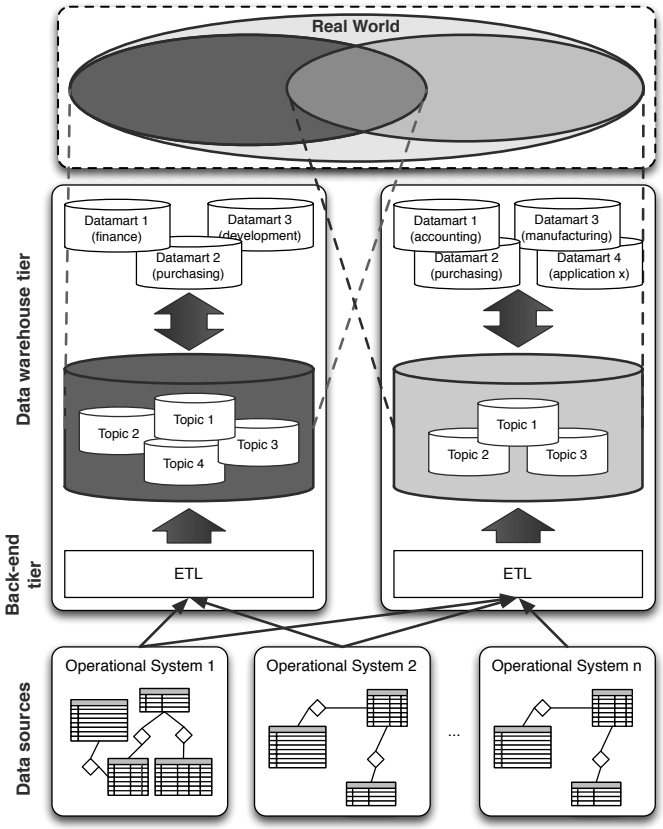


Figure 4. Completeness in data warehouses.

in the addressed application scenario, each of them may have a different view of the real world. As illustrated in Figure 4, it may be necessary to combine these views to obtain the best possible representation of the real world. This concept assumes that data in the information systems does not represent false information, since this would lead to a false representation of the real world. In the addressed application area, the processes are well supported, and in the past the most likely problem was data missed rather than false data. The resulting information base can be seen as a *reference dataset* (similar to the reference relation concept explained in Section III-B). The reference dataset is defined as shown in Equation (3).

$$d_r = \bigcup_{i=1}^{n} d_i \tag{3}$$

d_i are the instances stored in a particular data source (DWH) and d_r represents the total number of records in the reference dataset. In this case, DWHs are considered under the OWA, since it is not exactly known whether information is missing. If different DWHs store data from the same application area, a combination of these entries would lead to a better overall view (reference dataset). In order to calculate the completeness data quality metric for a single information system, the amount of information must be checked against the reference dataset. Equation (4) illustrates how the completeness metric (Q_{comp,d_i}) for a particular data source d_i can be obtained.

$$Q_{comp,d_i} = \frac{|d_i \cap d_r|}{|d_r|} \tag{4}$$

The second data quality metric used in the proposed concept is consistency, which is closely connected with integrity constraints. A perfectly designed data model would apply integrity constraints such as unique, primary key, and referential integrity to prevent inclusion of false data. Some information systems do not implement constraints and, therefore, inconsistencies may occur. An important consistency constraint is referential integrity, which guarantees the existence of a value in the corresponding database table. The consistency metric is calculated as illustrated in Equation (5).

$$Q_{cons,d_i} = \frac{|d_{i[conspos]}|}{|d_{i[all]}|} \tag{5}$$

In the mentioned equation the numerator $|d_{i[conspos]}|$ is defined as the absolute value of the entries that passed the consistency checks, and the denominator is the total number of entries of the dataset. Like the other quality metrics this calculation is applied to each information source.

The third and final data quality metric used to evaluate the data sources is correctness, which is based on semantic checks in the proposed approach. Semantic checks depend on the application scenario and the context in which the data is used. For example, if an attribute is defined as a value between 1 and 5 (e.g., indicating a grade given in Austrian schools) and a field contains the value 6, it is obvious that this information is false. Further, consider an attribute that has a strictly defined structure: the value has six signs, the first one being a letter between A and D, the next three signs between 1 and 6, and the last two signs alphanumeric. As another example, consider an application dealing with dates, where a particular attribute contains only past dates; if an entry contained a future date, it would have to be false. A check of two date attributes would have to verify that they are in sequence, meaning that one must precede the other. These examples show that considerable contextual knowledge is necessary to determine whether a particular entry is correct. More formally, the following check types can be identified:

- Range check: proves whether a particular value is in the correct range, e.g., only past dates are allowed or an integer range between 1 and 5.
- Structure check: evaluates whether entries of a particular attribute satisfy a given format, e.g., total value length is six or it must be a numeric value.

These checks need not necessarily to be static for all instances. It is very important that attributes can also depend on each other. An example is information about pupils, their residence, and their grades. If the residence of a pupil is in Austria, then the grades must be in the range between 1 and 5 (from the set of natural numbers). However, for residents of Switzerland, the range is 1 to 6 (with steps of 0.5).

Note that contextual or semantic changes (in the business process) imply that the checks for correctness must also be adapted to avoid a false correctness metric that would decrease the overall data quality metric of the data source and lead to false results of the proposed concept. Equation (6) shows the

calculation of the correctness quality metric (Q_{corr,d_i}) for a particular data source d_i.

$$Q_{corr,d_i} = \frac{|d_{i[corrpos]}|}{|d_{i[all]}|} \quad (6)$$

In the presented equation, the numerator $|d_{i[corrpos]}|$ represents the absolute value of the entries that proved correct, and $|d_{i[all]}|$ is the total number of entries in the data source.

Finally, the overall data quality metric of the data source can be calculated as the multiplication of the three quality metrics discussed (Equation (7)). The resulting quality metric of data source d_i is represented by Q_{d_i}, and the completeness, consistency, and correctness component quality metrics are denoted by Q_{comp,d_i}, Q_{cons,d_i}, and Q_{corr,d_i}, respectively. This metric is then used as a weighting factor W_{qual,d_i} for the combiner component.

$$W_{qual,d_i} = Q_{d_i} = Q_{comp,d_i} * Q_{cons,d_i} * Q_{corr,d_i} \quad (7)$$

An example output of the validator component is shown in Figure 5.

data source	Q_{comp}	Q_{cons}	Q_{corr}	Q_d
data source 1	0.85	0.91	1.00	0.77
data source 2	0.94	0.97	0.98	0.89
data source 3	0.98	0.89	0.92	0.80
...	...	...	...	...

Figure 5. Example results of the validator component.

2) Predictor: The predictor's tasks of forecasting future values based on a particular dataset's values and of performing regression analysis are implemented as two steps: (i) determining the value of the following week for the various fault types based on their number from the previous weeks and (ii) regression analysis using the previous five weeks and the calculated forecast.

In order to generate future values, the contextual requirements must be known to investigate and determine which time series method best suits the use case. In the proposed concept, the prediction of how many faults will occur in the following week is performed based on the number of faults in the after-sales market from an appropriate time period. The specific fault analysis set is bound by the particular fault type, fuel type, car brand, engine type, and the period to be used in the prediction task. In this scenario, the period was set to one year, a relevant period in the investigation process of the quality management experts. Different time series analysis methods which are capable of performing the prediction task are listed in Section III-C. An evaluation identified the most appropriate approach, which depends on the underlying process and the given time series. To this end, two types of quality checks were applied: one is based on the *Diebold-Mariano Test* [48], which compares the prediction quality of two methods, and the other calculates *Goodness-of-Fit* measures (e.g., MAPE, sMAPE, MAE). The test scenario was established as follows:

- Different time series defined as a sets of fault type, fuel type, car brand, and engine type were evaluated.
- Every possible pairwise combination of time series methods was used in the Diebold-Mariano test to obtain a matrix that shows how they perform in relation to each other. The *h* value was set to one, which specifies that only a one-point forecast was evaluated, since this is also the aim in the application scenario. The alternative hypothesis method was set to *greater*, which means testing whether method two is more accurate than method one. The loss function power was set to two, a commonly chosen value.
- To determine how good the different predictions perform using the *Goodness-of-Fit* metrics, in-sample predictions were computed, where the most recent week (observation) was left out for the comparison task. The different quality metrics were then calculated using the left-out observation and the forecast value.
- The results were ranked to see which prediction method outperforms the others in the particular use case.

The results revealed that it cannot be clearly determined which prediction method is the best, since this heavily depends on the course of the time series. The *Goodness-of-Fit* metrics could not establish a clear winner: the best methods were ARIMA, TBATS, and Croston's method. The Diebold-Mariano tests identified ARIMA and TBATS as superior methods; hence, the two are favored by the proposed concept. ARIMA is used for the prediction task, since it is also provided by a tool already in use by the business partner.

The second task of the predictor component is to perform regression analysis of data from a six-weeks period. As in the current implementation of Q-AURA, linear regression using a straight line was chosen, since this yields the best results and has been applied and evaluated for two years. The period used for regression analysis includes the most recent five weeks observed and the value predicted for the next week. The characteristic values *gradient*, *mean value*, and *coefficient of determination* are computed. The gradient and the mean value are calculated using the equation for a straight line (Equation (8)).

$$y = k * x + d \quad (8)$$

The parameters x and y represent a two-dimensional coordinate in the diagram, where x corresponds with being the time value and y is the observed (or predicted) value of the focused measure. The characteristic value k is the gradient and represents the average increase between two subsequent points in the diagram. d is the offset and describes the initial or start value y at $x = 0$. Another characteristic value of the regression line is the mean value $\bar{y}$, which is computed by averaging the data points over the time period. In the use case of the proposed concept this period consists of five observations and the predicted value. A previous version of this approach also calculated the *coefficient of determination* [46]. This value describes the steadiness of the regression line. In the proposed concept the regression line depends only on one variable, therefore the coefficient is equal to the square of *Pearson's Correlation Coefficient* r_{xy}^2 (Equation (9)) [49].

$$R^2 = r_{xy}^2 = \frac{s_{xy}^2}{s_x^2 s_y^2} \quad (9)$$

Based on the gradient, two new values are calculated, which provide a more detailed view of the course over six weeks. The first is an extension of the gradient, since it determines the relative value based on the mean value of the six weeks (of the analysis set). The mean value is interpolated based on the observations from the previous five weeks, because the most recent week is predicted, and thus has no underlying number of observed faults (Equation (10)).

$$n_{6weeks} = n_{5weeks} * \frac{6}{5} \tag{10}$$

This value is then used to determine the relative gradient of the six weeks (Equation (11)).

$$k_{rel} = \frac{k}{n_{6weeks}} \tag{11}$$

The second value is called *gradient parts-per-million* (k_{ppm}), which is also based on the gradient (k) of the regression line. The idea behind this metric is the identification of faults with a high value when compared to the particular engine type. Since the regression line is based on the analysis set consisting of fault type, fuel type, car brand, and engine type, it limits data to a fine-grained but appropriate set of faults. While k_{rel} determines the average gradient based on this analysis set, k_{ppm} takes the whole number of faults for the particular engine type $n_{enginetype}$ into account as given in Equation (12). Since the result is a very low value, it is expressed as ppm (multiplied by 1,000,000).

$$k_{ppm} = \frac{k * 1,000,000}{n_{enginetype} * \frac{6}{5}} \tag{12}$$

These computations are performed for each analysis set (fault type, fuel type, car brand, and engine type) from each dataset. An example of an output from the resulting data structure is shown in Figure 6.

data source	k	$\bar{y}$	R^2	k_{rel}	k_{ppm}
data source 1	1.92	12.34	0.32	0.69	123.23
data source 2	2.45	32.91	0.19	0.23	453.21
data source 3	0.64	24.54	0.98	0.76	91.23
...	...	...	...	...	...

Figure 6. Example results of the predictor component.

3) Combiner: The third component of the proposed concept is the combiner, which decides whether the analyzed fault is significant. As explained above, the predictor uses regression analysis and calculates the corresponding characteristic metrics, k_{rel} and k_{ppm}. The combiner uses these two parameters in addition to weighting factors from the validator and the controller component. The overall weighting factor for a particular fault is computed as the product of the data quality metric ($W_{qual,i}$) and the weighting factor based on the prediction accuracy ($W_{cont,i,ft}$) (Equation (13)).

$$W_{i,ft} = W_{qual,i} * W_{cont,i,ft} \tag{13}$$

In the proposed approach, two concepts have been developed with different granularities to determine the overall result that decides whether the fault is significant.

- *Parameter-driven approach*: In the first step the differences between defined thresholds and the characteristic parameters (k_{rel} and k_{ppm}) are calculated, which are then multiplied by the corresponding weighting factors and divided by the sum of the weighting factors over the different information sources (Equation (14) and Equation (15)). A fault is significant if both resulting values are greater than 0, and insignificant otherwise (Equation (16)).

$$R_{rel,ft} = \frac{\sum_{j=1}^{n} W_{i,ft} * (k_{rel,thr} - k_{rel,i,ft})}{\sum_{i=1}^{n} W_{i,ft}} \tag{14}$$

$$R_{ppm,ft} = \frac{\sum_{j=1}^{n} W_{i,ft} * (k_{ppm,thr} - k_{ppm,i,ft})}{\sum_{i=1}^{n} W_{i,ft}} \tag{15}$$

$$S_{ft} = \begin{cases} R_{rel,ft} > 0 \cap R_{ppm,ft} > 0, & 1 \\ R_{rel,ft} \leq 0 \cup R_{ppm,ft} \leq 0, & 0 \end{cases} \tag{16}$$

Figure 7 shows an example database table resulting from the parameter-driven approach. The result is defined on the analysis set that consists of fault type, fuel type, car brand, and engine type.

fault	fuel	brand	e_type	R_{rel}	R_{ppm}	S_{ft}
f1	d	b1	e1	0.23	0.42	1
f2	b	b1	e2	-0.12	0.18	0
f1	d	b2	e3	-0.50	-0.34	0
...	...	...	...	...	...	...

Figure 7. Example results of the combiner component based on the parameter-driven approach.

- *Result-driven approach*: This concept is based on the significance result of each information source. First, the fault must be classified as significant or not depending on the characteristic metrics. The result indicates whether based on the dataset the fault would be classified as significant (1 = significant, 0 = insignificant) (Equation (17)). Each result is multiplied with the weighting factor of the corresponding fault/data source-combination and the results of the data sources are aggregated. The last step is to divide the value by the sum of the weights of the data sources. The fault is significant if the result is greater than 0.5, and insignificant otherwise (Equation (18)).

$$S_{i,ft} = \begin{cases} k_{rel,i,ft} > k_{rel,th} \cap k_{ppm,i,ft} > k_{ppm,th}, & 1 \\ k_{rel,i,ft} \leq k_{rel,th} \cup k_{ppm,i,ft} \leq k_{ppm,th}, & 0 \end{cases} \tag{17}$$

$$S_{ft} = \begin{cases} \frac{\sum_{j=1}^{n} W_{i,ft} * S_{i,ft}}{\sum_{i=1}^{n} W_{i,ft}} > 0.5, & 1 \\ \frac{\sum_{j=1}^{n} W_{i,ft} * S_{i,ft}}{\sum_{i=1}^{n} W_{i,ft}} \leq 0.5, & 0 \end{cases} \quad (18)$$

Figure 8 shows an example output table of the combiner component based on the result-driven approach. As illustrated, the result is defined for the particular analysis set, which consists of fault type, fuel type, car brand, and engine type.

fault	fuel	brand	e_type	W_{sum}	S_{sum_w}	S_{ft}
f1	d	b1	e1	2.53	2.02	1
f2	b	b1	e2	1.45	0.58	0
f1	d	b2	e3	2.67	0.51	0
...	...	...	...	...	...	...

Figure 8. Example results of the combiner component based on the result-driven approach.

4) Controller: The controller component calculates the prediction accuracy of the fault time series' forecasts for each information source. This accuracy metric is used to obtain a weighting factor as required by the combiner component. In the proposed approach, the prediction method is a one-step out-of-sample forecast that computes a value for the following week, the new value can be observed and compared with the prediction from the previous week. In Section III-D, different prediction accuracy metrics were discussed. According to the classification proposed there, *relative errors* and *relative measures* do not meet the requirements, because they represent relative values between the accuracy of the method applied and a benchmark method. The drawback is that if the benchmark method leads to poor results, the calculated metric would possibly indicate a good accuracy. This is even more serious in the proposed approach, since the goal is to weight predictions based on different data sources. For example, if the benchmark method of a data source achieves poor results and the prediction is relatively good in comparison, the data source will be weighted more favorably than a data source where the benchmark method performs very well and the method used for the prediction is not as good in comparison. Metrics that belong to the class *scale-dependent measures* are also excluded, since they are scale dependent. For example, when a particular fault occurs more often in one data source than in another, this difference would influence the outcome, because it is not possible to compare them. Since the *MASE* metric needs more than one prediction for computation, it cannot be used in the proposed approach. Consequently, the remaining errors in the *percentage errors* category are *MAPE*, *MdAPE*, *sMAPE*, *sMdAPE*, *RMSPE*, and *RMdSPE*. Since the error metric in the proposed concept is calculated for a single forecast, there is no difference in the results between the versions using the mean and those using the median. Therefore, for this approach three different relevant metrics can be distinguished *APE*, *sAPE*, and *RSPE*.

The calculation of the *Mean Absolute Percentage Error (MAPE)* is given in Equation (19) [39].

$$e_{MAPE} = \frac{1}{n} * \sum_{i=1}^{n} \frac{|X_i - F_i|}{X_i} * 100 \quad (19)$$

The *symmetric Mean Absolute Error (sMAPE)* is defined as shown in Equation (20) [50].

$$e_{sMAPE} = \frac{1}{n} * \sum_{i=1}^{n} \frac{|X_i - F_i|}{(X_i + F_i)/2} * 100 \quad (20)$$

This equation shows that the sMAPE can take values between 0 and +200 (or - without the multiplier at the end - values between 0 and 2). A drawback of the error metric is that it is not symmetrical: Let us assume that observation X_i is the same for two information sources and has the value 50. The first data source predicts a value of 45 and the second data source predicts 55. Thus, both predictions have the same difference of 5, but one is too high and one too low. The sMAPE for the first data source is then 10.5% and for the second 9.5%. Despite this asymmetry in the results, sMAPE is used in scientific papers to determine the quality of forecasts (e.g., in the M3-Competition [50][51]).

The computation of the *Root Mean Square Percentage Error (RMSPE)* is given in Equation (21) [40].

$$e_{RMSPE} = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} (\frac{|X_i - F_i|}{X_i})^2} * 100 \quad (21)$$

When dealing with a single future prediction the equation can be reduced to (M)APE, as the square root and the power of two can be eliminated.

Since sMAPE constitutes a good measure that can be transformed to the range between 0 to 1 (by removing the multiplier in the denominator), it is a good weighting factor for the proposed approach. Prediction accuracy can thus be calculated as shown in Equation (22).

$$P_{sMAPE} = 1 - \frac{|X_i - Fi|}{X_i + F_i} \quad (22)$$

Alternatively, MAPE could be used for this purpose. However, it is not ideal as a weighting factor, because it cannot be accurately transformed to the range between 0 and 1. A way of using MAPE to determine the prediction accuracy is shown in Equation (23).

$$P_{MAPE} = \begin{cases} \frac{|X_i - F_i|}{X_i} \leq 1, & \frac{|X_i - F_i|}{X_i} \\ \frac{|X_i - F_i|}{X_i} > 1, & 0 \end{cases} \quad (23)$$

Note that not only the prediction accuracy of the current week should be considered in the calculation of the weighting factor. The following example explains why: Let us assume that a specific information source achieved good prediction accuracy in recent weeks and performs poorly in the current week. If the accuracy based on a single week were used, the quality indicator of the data source would decrease drastically. Conversely, if an information source with very low prediction accuracy in previous weeks performs well in the current week, then the weighting should not be based only on this single (good) result. Therefore, the calculation in the proposed

concept of the weighting factor takes also previous prediction accuracies into account as shown in Equation (24).

$$W_{cont,d_i,fault} = \frac{P_{t-1} + P_t}{2} \quad (24)$$

Figure 9 illustrates the structure and example instances of the controller output table. An entry is defined by the dataset (represented by the data source column) and the analysis set (the attributes fault, fuel, car brand, and engine type). The remaining columns define the results of the controller component, and W_{cont} stores the final weighting factors used by the combiner component.

data source	fault	fuel	brand	e_type	P_{t-1}	P_t	W_{cont}
data source 1	f1	d	b1	e1	0.87	0.91	0.89
data source 1	f2	b	b1	e2	0.99	0.58	0.79
data source 2	f2	b	b1	e2	0.69	0.78	0.74
data source 3	f1	d	b2	e3	0.71	0.83	0.77
...	...	...	...	...	...	...	...

Figure 9. Example results of the controller component.

C. Q-AURA Integration

This section focuses on the integration of the presented concept into the Q-AURA application. Q-AURA comprises six steps: The first identifies which faults are significant and should be analyzed further. The presented concept optimizes this task by enabling earlier detection. The interface between this and the subsequent step is defined on a metric that indicates whether an analysis set is significant. Since the proposed concept uses the same representation of results, the original step can be substituted with the new approach. The improved approach including the optimization is illustrated in Figure 10.

Using an interface between the first and the second step eases this substitution. The second step needs only information about which faults are significant depending on fuel type, car brand, and engine type.

V. Conclusion and Future Work

The presented concept, which is currently evaluated, improves Q-AURA by an earlier identification of faults with negative trends. Q-AURA has been developed in cooperation with the industrial partner *BMW Motoren GmbH* (engine manufacturing plant) and is already in daily use by quality management experts at different engine manufacturing plants. It is an application that identifies significant faults, which are then examined in more detail. Bills of materials containing information about parts, components, and technical modifications are analyzed to determine modifications that are most likely the cause of a particular fault.

In this work, a concept has been proposed that addresses the challenge of earlier detection of critical faults. At the heart of the presented approach is the integration of different datasets that provide different views of warranty data. Data quality metrics are used to determine how accurate and correct the information from the different datasets is. Next, the fault course of each dataset is analyzed to predict the most likely value for the following week. Regression analysis is applied to a six-week period (using the predicted value and the last

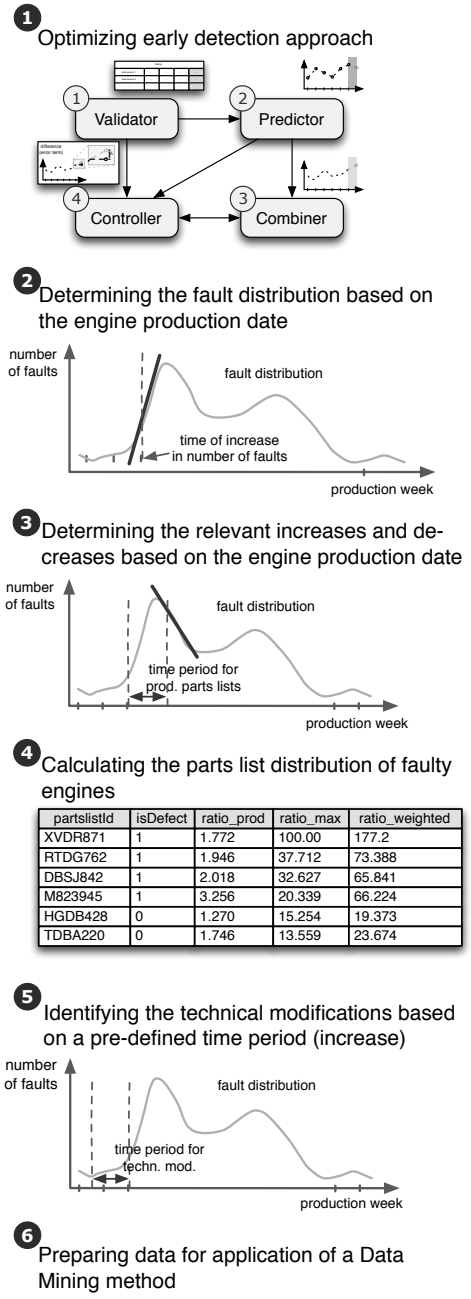


Figure 10. Integration of the proposed approach into Q-AURA.

five observations), which yields the characteristic values of the resulting regression line. The prediction accuracy is determined using predictions from the previous week and observations from the current week. In order to decide whether a fault is significant, weighting factors based on the calculated data quality metric and the prediction accuracy are used in addition to the results of the regression analysis. The approach is applied on warranty information in the automotive industry, but the concept could also be used in other application areas where time series and forecasts from different datasets must be combined to determine whether a particular course is significant. The definition of *significance* must be evaluated and determined in each application area. Depending on the use

case, other data quality metrics are potentially interesting for integration in the overall data quality metric. The three metrics used in this concept were chosen with care to be data-centric and to not take data representation or feature availability into account.

A further possible improvement would be the integration of an additional weighting factor depending on expert input. In some cases, domain experts have additional information about the datasets and would prefer an additional weighting factor that represents their view. This means that three factors would be used to determine the weighting: (i) the overall data quality metric of the dataset, (ii) the prediction accuracy of the dataset's time series analysis, and (iii) the preference metric based on expert input.

Another possible enhancement is to investigate whether applying different time series and forecasting methods for each dataset and subsequent combination of the forecasts yields more robust predictions and thus better results. Various research papers ([3][6][52][53]) have addressed such combination approaches, which are already used in the field of machine learning [54].

REFERENCES

[1] T. Leitner, C. Feilmayr, and W. Wöß, "Early Detection of Critical Faults Using Time-Series Analysis on Heterogeneous Information Systems in the Automotive Industry," in Third International Conference on Data Analytics, F. Laux, P. M. Pardalos, and C. Alain, Eds. International Academy, Research, and Industry Association (IARIA), 2014, pp. 70–75.

[2] C. K. Chan, B. G. Kingsman, and H. Wong, "The value of combining forecasts in inventory management - a case study in banking." European Journal of Operational Research, vol. 117, no. 2, 1999, pp. 199–210.

[3] A. Widodo and I. Budi, "Combination of time series forecasts using neural network," in International Conference on Electrical Engineering and Informatics (ICEEI), 2011, pp. 1–6.

[4] A. Kleyner and P. Sandborn, "A warranty forecasting model based on piecewise statistical distributions and stochastic simulation," Reliability Engineering and System Safety, vol. 88, no. 3, 2005, pp. 207–214.

[5] J. M. Montgomery, F. M. Hollenbach, and M. D. Ward, "Calibrating Ensemble Forecasting Models with Sparse Data in the Social Sciences (accepted and in press)," International Journal of Forecasting, -.

[6] J. S. Armstrong, "Combining Forecasts," in Principles of forecasting. Springer US, 2001, pp. 417–439.

[7] T. C. Redman, "The Impact of Poor Data Quality on the Typical Enterprise," Communications of the ACM, vol. 41, no. 2, 1998, pp. 79–82.

[8] B. Heinrich, M. Kaiser, and M. Klier, "How to measure Data Quality? A Metric Based Approach," in Proceedings of the 28th International Conference on Information Systems (ICIS). Montreal, Canada: Association for Information Systems, 2007.

[9] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for Data Quality Assessment and Improvement," ACM Computing Surveys, vol. 41, no. 3, 2009, pp. 1–52.

[10] C. Batini and M. Scannapieco, Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications). Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[11] R. Y. Wang and D. M. Strong, "Beyond Accuracy: What Data Quality Means to Data Consumers," Journal of Management Information Systems, vol. 12, no. 4, 1996, pp. 5–33.

[12] D. M. Strong, Y. W. Lee, and R. Y. Wang, "Data Quality in Context," Communications of the ACM, vol. 40, no. 5, 1997, pp. 103–110.

[13] Y. W. Lee, D. M. Strong, B. K. Kahn, and R. Y. Wang, "Aimq: A methodology for information quality assessment," Information and Management, vol. 40, no. 2, 2002, pp. 133–146.

[14] F. Naumann and C. Rolker, "Assessment methods for Information Quality Criteria," in Fifth Conference on Information Quality (IQ 2000), Cambridge, MA, USA, 2000.

[15] D. P. Ballou and H. L. Pazer, "Modeling Completeness Versus Consistency Tradeoffs in Information Decision Contexts," IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 1, 2003, pp. 240–243.

[16] M. Ge and M. Helfert, "A review of information quality research – develop a research agenda," in International Conference on Information Quality, 2007, pp. 76–91.

[17] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," Communications of the ACM, vol. 45, no. 4, 2002, pp. 211–218.

[18] M. Bobrowski, M. Marré, and D. Yankelevich, "A Homogeneous Framework to Measure Data Quality," in Information Quality, Y. W. Lee and G. K. Tayi, Eds. MIT, 1999, pp. 115–124.

[19] B. K. Kahn, D. M. Strong, and R. Y. Wang, "Information Quality Benchmarks: Product and Service Performance," Communications of the ACM, vol. 45, no. 4, 2002, pp. 184–192.

[20] A. Motro and I. Rakov, "Estimating the Quality of Data in Relational Databases," in Proceedings of the Conference on Information Quality. MIT, 1996, pp. 94–106.

[21] A. Motro and I. Rakov, "Estimating the quality of databases," in Proceedings of the Third International Conference on Flexible Query Answering Systems (FQAS), T. Andreasen, H. Christiansen, and H. L. Larsen, Eds., vol. 1495. Springer Verlag, 1998, pp. 298–307.

[22] A. Motro, "Integrity = Validity + Completeness," ACM Transactions on Database Systems, vol. 14, no. 4, 1989, pp. 480–502.

[23] Y. Wand and R. Y. Wang, "Anchoring Data Quality Dimensions in Ontological Foundations," Communications of the ACM, vol. 39, no. 11, 1996, pp. 86–95.

[24] T. C. Redman, Data Quality for the Information Age, 1st ed. Norwood, MA, USA: Artech House, Inc., 1996.

[25] M. Jarke, M. Lenzerini, Y. Vassiliou, and P. Vassiliadis, Fundamentals of Data Warehouses. Springer Verlag, 1995.

[26] M. Bovee, R. P. Srivastava, and B. Mak, "A conceptual framework and belief-function approach to assessing overall information quality." International Journal of Intelligent Systems, vol. 18, no. 1, 2003, pp. 51–74.

[27] F. Naumann, Quality-driven Query Answering for Integrated Information Systems. Berlin, Heidelberg: Springer-Verlag, 2002.

[28] L. Liu and L. Chi, "Evolutional data quality: A theory-specific view." in International Conference on Information Quality, C. Fisher and B. N. Davidson, Eds. MIT, 2002, pp. 292–304.

[29] D. P. Ballou and H. L. Pazer, "Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems," Management Science, vol. 31, no. 2, 1985, pp. 150–162.

[30] R. H. Shumway and D. S. Stoffer, Time Series Analysis and Its Applications: With R Examples, 3rd ed. Springer Texts in Statistics, 2011.

[31] P. S. P. Cowpertwait and A. V. Metcalfe, Introductory Time Series with R, 1st ed. Springer Publishing Company, Incorporated, 2009.

[32] A. M. De Livera, R. J. Hyndman, and R. D. Snyder, "Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing," Journal of the American Statistical Association (JASA), vol. 106, no. 496, 2011, pp. 1513–1527.

[33] A. M. D. Livera, "Automatic forecasting with a modified exponential smoothing state space framework," Monash University, Department of Econometrics and Business Statistics, Monash Econometrics and Business Statistics Working Papers 10/10, 2010.

[34] A. B. Koehler, R. J. Hyndman, R. D. Snyder, and K. Ord, "Prediction intervals for exponential smoothing using two new classes of state space models," Journal of Forecasting, vol. 24, no. 1, 2005, pp. 17–37.

[35] R. J. Hyndman, A. B. Koehler, R. D. Snyder, and S. Grose, "A state space framework for automatic forecasting using exponential smoothing methods," International Journal of Forecasting, vol. 18, no. 3, 2002, pp. 439–454.

[36] J. D. Croston, "Forecasting and stock control for intermittent demands," Operational Research Quarterly, vol. 23, no. 3, 1972, pp. 289–303.

[37] L. Shenstone and R. J. Hyndman, "Stochastic models underlying

Croston's method for intermittent demand forecasting," Journal of Forecasting, 2005.

[38] H. Thome, "Univariate Box/Jenkins-Modelle in der Zeitreihenanalyse (Univariate Box/Jenkins-models in time series analysis)," Historical Social Research, vol. 19, no. 3, 1994, pp. 5–77.

[39] J. G. D. Gooijer and R. J. Hyndman, "25 years of time series forecasting," International Journal of Forecasting, 2006.

[40] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," International Journal of Forecasting, vol. 22, no. 4, 2006, pp. 679–688.

[41] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler, "The accuracy of extrapolation (time series) methods: Results of a forecasting competition," Journal of Forecasting, vol. 1, no. 2, 1982, pp. 111–153.

[42] J. S. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons," International Journal of Forecasting, vol. 8, no. 1, 1992, pp. 69–80.

[43] S. Makridakis, "Accuracy measures: theoretical and practical concerns," International Journal of Forecasting, vol. 9, no. 4, 1993, pp. 527–529.

[44] P. Goodwin and R. Lawton, "On the asymmetry of the symmetric MAPE," International Journal of Forecasting, vol. 15, no. 4, 1999, pp. 405–408.

[45] T. Leitner, C. Feilmayr, and W. Wöß, "Optimizing Reaction and Processing Times in Automotive Industry's Quality Management - A Data Mining Approach," in International Conference Data Warehousing and Knowledge Discovery (DaWaK), ser. Lecture Notes in Computer Science, L. Bellatreche and M. K. Mohania, Eds., vol. 8646. Springer-Verlag, 2014, pp. 266–273.

[46] G. U. Yule, "On the Theory of Correlation," Journal of the Royal Statistical Society, vol. 60, no. 4, 1897, pp. 812–854.

[47] A. Vaisman and E. Zimányi, Data Warehouse Systems: Design and Implementation. Springer-Verlag, 2014.

[48] F. X. Diebold and R. S. Mariano, "Comparing Predictive Accuracy," Journal of Business & Economic Statistics, vol. 13, no. 3, 1995, pp. 253–263.

[49] M. Mittlböck and M. Schemper, "Explained Variation for Logistic Regression," Statistics in medicine, vol. 15, no. 19, 1996, pp. 1987–1997.

[50] S. Makridakis and M. Hibon, "The M3-Competition: results, conclusions and implications," International Journal of Forecasting, vol. 16, no. 4, 2000, pp. 451–476.

[51] M. Hibon and T. Evgeniou, "To combine or not to combine: selecting among forecasts and their combinations," International Journal of Forecasting, vol. 21, no. 1, 2005, pp. 15–24.

[52] R. T. Clemen, "Combining forecasts: A review and annotated bibliography," International Journal of Forecasting, vol. 5, no. 4, 1989, pp. 559–583.

[53] J. M. Bates and C. W. J. Granger, "The Combination of Forecasts," in Operational Research Society, vol. 20, no. 4, 1969, pp. 451–468.

[54] I. H. Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.