Service Recommendation Using Machine Learning Methods Based on Measured Consumer Experiences Within a Service Market

Jens Kirchner
Karlsruhe University of Applied Sciences
Linnaeus University
Email: Jens.Kirchner@hs-karlsruhe.de,
Jens.Kirchner@lnu.se

Andreas Heberle
Karlsruhe University of Applied Sciences
Moltkestr. 30, 76133 Karlsruhe, Germany
Email: Andreas.Heberle@hs-karlsruhe.de

Welf Löwe
Linnaeus University
351 06 Växjö, Sweden
Email: Welf.Lowe@lnu.se

***Abstract*—Among functionally similar services, service consumers are interested in the consumption of the service that performs best towards their optimization preferences. The experienced performance of a service at consumer side is expressed in its non-functional properties. Selecting the best-fit service is an individual aspect as the preferences of consumers vary. Furthermore, service markets such as the Internet are characterized by perpetual change and complexity. The complex collaboration of system environments and networks result in various performance experiences at consumer side. Service optimization based on a collaborative knowledge base of previous experiences of other, similar consumers with similar preferences is a desirable foundation. In this article, we present a service recommendation framework, which aims at the optimization at consumer side focusing on the individual preferences and call contexts. In order to identify relevant non-functional properties for service selection, we conducted a literature study of conference papers of the last decade. The ranked results of this study represent what a broad scientific community determined to be relevant non-functional properties for service selection. We furthermore analyzed, implemented, and validated machine learning methods that can be employed for service recommendation. Within our validation, we could achieve up to 95 % of the overall achievable performance (utility) gain with a machine learning method that is focused on concept drift, which in turn, tackles the change characteristic of the Internet being a service market. Besides the comprehensive and scientific identification of relevant non-functional properties when selecting a service, this article describes how machine learning can be employed for service recommendation based on consumer experiences in general, including an evaluation and overall proof of concept validation within our framework.**

Keywords–Service Selection; Service Recommendation; Machine Learning; Non-functional properties; Performance gain.

I. Introduction

Service-Oriented Computing (SOC), Software as a Service (SaaS), Cloud Computing, and Mobile Computing indicate the development of the Internet into a market of services. In such an anonymous market, service consumers have little to no knowledge about the implementation of a service or the system environment around it. Service functionality can be dynamically and ubiquitously consumed. Besides the actual functionality, service consumers are interested in the performance of a service. The performance of a service is expressed in its non-functional properties (NFPs) such as response time, availability, or monetary costs. In such a service market, the same service functionality may be provided by several competing service providers. Among these functionally similar services, service consumers are interested in the service that fits best to their (NFP) preferences. In particular, consumers are interested in the performance they experience at call side. One of the major characteristics of a service market such as the Internet is perpetual change. Entering and leaving service providers as well as the complexity of service dependencies and environments make service selection and recommendation a challenge. It seems to be impossible to foresee the exact performance of a future service call in a perpetual changing market. Static and single-sided information such as Service Level Agreements (SLAs) is not a good basis for service selection for several reasons (cf. [2]). The first reason is change in general, but also because service providers are interested to embellish the NFPs of their services in order to encourage service consumers to call their services. A further reason is the complex collaboration of various system environments and networks with incidents and coincidences. Since service consumers are interested in the best-fit experience at their side, it is desirable to base service selection on the collaborative knowledge of previous service calls of similar consumers with similar preferences and call contexts. The experienced performance at consumer side is influenced by a consumer's call context, e. g., calling time and/or location. Therefore, the performance has to be predicted based on this context. Furthermore, performance is different for different consumers who value the NFPs of a service differently. For instance, some consumers are more interested in a fast response time and rather neglect higher monetary charges than others who want to have a service for free and rather accept higher response times. Therefore, service value is individual and it has to be determined individually whether a service is actually best-fit in a specific context.

In this article, we present our service recommendation framework, which uses a collaborative knowledge base of consumption experiences of similar consumers in the past to predict the performance of services in a certain consumer-based call context in order to recommend the best-fit service candidate to a consumer, considering his/her preferences [1][2][3]. For the recommendation of services aiming at the optimization of the actual experience at consumers' side, it is important to determine, which NFPs are relevant for service selection. In order to determine these NFPs, we conducted a comprehensive literature survey of scientific conference papers of the last decade. The results of the survey revealed a ranked list of NFPs, which a broad scientific community

described to be relevant for service selection and, hence, for service recommendation. Furthermore, we analyzed two machine learning approaches for their capability to be employed for this service recommendation task. For the more suitable approach, we implemented and evaluated machine learning methods within our framework. In total, this article contributes a comprehensive evaluation of the employment of machine learning within the recommendation of services, including the discussion of the benefits and drawbacks of the general machine learning approaches within the recommendation process, their evaluation, and the implementation and validation of machine learning methods for the most suitable approach. Its results furthermore provide an overall proof of concept for the optimization of service recommendation based on previous experiences considering call contexts and consumer preferences. Moreover, we present the results of a comprehensive survey about relevant NFPs during service selection as the recommendation framework's scientific foundation.

After this introduction, the article is organized as follows: Related work to our research work is outlined in Section II. Section III introduces our service recommendation framework. It describes the relevant aspects of service recommendation in a service market as well as the necessary components. In order to clarify, which NFPs are actually relevant for service selection/recommendation, we present the results of our scientific-community-based study in Section IV. In Section V, we describe how machine learning can be employed for service recommendation based on a collaborative knowledge base. It initially describes the benefits and drawbacks of *classification*- and *regression*-based approaches within this domain. Section VI presents the evaluation of both approaches. The second part of the section then presents the initial implementation within our framework completed with tests about the achievement of the overall possible gain/benefit of the experienced performance at consumer side. Finally, Section VII concludes the article.

II. Related Work

We initially introduced the overall concept of how shared knowledge can benefit service selection/recommendation in general in [2]. In that work, we presented the framework on an abstract level and introduced the recommendation component as a black box. In this article, we describe the architecture of the recommendation component in details and evaluate the employment of machine learning methods within service recommendation aiming at the improvement of service performance experienced at consumer side. Turning measurement data into shared knowledge, it can be used for a consumer-centric optimized service recommendation within an anonymous service market.

The idea for our framework is inspired by our previous contributions on profile-guided composition of desktop applications [4][5]. This framework distributes the infrastructure needed for assessing and optimizing service calls and adopts SOA components for implementing the necessary infrastructure.

There are other considerations about service markets and sub-dependencies that our framework in general focuses and we outline in this article (cf. [6][7]). In contrast to our approach, they consider service brokers and directories as service intermediaries respectively market places for services. In our understanding, intermediaries offer new service functionalities based on the consumption of sub-services while service brokers select the best-fit service among substitutable candidates [2]. Our approach works with NFPs as a foundation for service selection. By the definition of utility functions, consumer preferences can be calculated in order to determine the individual best-fit service instance. The authors of [8] present the results of a test saying that the impact of service attributes to customer satisfaction may change over time for e-services. As for our work, we aim at an automatic recommendation for service consumers in general. With the automation aspect, we primarily address consuming systems or services. Nonetheless, the preferences of service consumers may indeed change over time. In such a case, service consumers have to update their utility functions within our framework.

There are several approaches focusing on a quality-driven selection of services at runtime (late binding), cf. [9][10][11][12][13][14][15]. In most of these approaches, service selection is based on SLAs provided by service providers. These approaches are limited to local integration environments and lack an overall view beyond the borders of a consumer's environment. Some approaches focus on the prediction of NFPs for the detection of SLA violations such as [16]. Still, that work mainly focuses on SLAs to be the foundation for any evaluation. Another approach [17] introduces a framework that uses an SLA broker that negotiates SLAs between service consumers and service providers. During service consumption, SLA breaches are monitored. In such a case, the broker renegotiates or looks for a better service if the renegotiation fails. Although this approach also aims at the optimization of consumers' experiences, negotiation is time-consuming and does not solely focus on the benefit of service consumers' and the convergence of the negotiation remains unclear. Furthermore, optimized service recommendation is supposed to prevent or at least reduce the experience of service failures. However, when it comes to renegotiation, it is already too late and a failed or non-preferred service call is conducted. Furthermore, approaches that require service providers to participate in a single-sided optimization are questionable and they tighten the coupling of service consumers and providers, which conflicts with the decoupling idea of distributed systems. Our approach supports service consumers regardless whether providers actively participate or not and does not require any changes in implementations or architectures; it only provides an extension to the hitherto existing integration environments.

In general, we argue that SLAs are not a sufficient foundation for service selection [2] (also cf. [9][18]). Considering the profit-orientation of service providers, it is tempting for them to embellish their SLAs in order to be more attractive for consumption. Furthermore, the performances of services vary [2] and they are experienced differently due to a consumer's call context. SLAs cannot reflect such aspects since they can only reflect a provider's single-sided view. For a consumer, however, the actual performance experience matters. Also, as SLAs of consuming and providing services (e. g., compound services) depend on the SLAs of sub-providers, deviations of actual NFPs and those specified in SLAs may be propagated and spread even unintendedly and without the control of the providers.

Our framework also aims at self-optimization, which is similar to the goals of autonomic computing (cf. [19][20]).

There are other approaches that focus on self-adaptation and context-orientation affecting the process logic in SOC (cf. [21]). Our approach focuses on the substitution of similar functional services regarding differences in their performance. The approach of [22] aims at the support of the software life-cycle process. By means of aspect-oriented programming, a service provider can observe quality aspects of a software component reflected in the NFPs of a service. The idea of gaining feedback and the measurement methods are similar to ours. However, they focus on the support of service providers and developers, while we focus exclusively on service consumer support. Still, although our framework aims at the optimization at consumer side, it can also be beneficial for service providers for a consumer-oriented optimization of the implementation as well as infrastructure configuration in order to be more attractive for consumption.

Collaborative filtering (CF) approaches for service recommendation also focus on the exploitation of shared knowledge about services for the recommendation of services to similar consumers on an automated basis [18][23][24][25]. Machine learning, in general, can also be used in CF. In contrast to the filtering of external decision results in CF, our approach determines the individual best-fit service based on previously measured performance data, individual preferences, and calculated utility values. With our approach considering call context and utility function, new consumers can already benefit from existing knowledge. CF approaches also do not take into account that consumers can have different optimization goals or preferences. Only some approaches [24][25] consider differences between consumers regarding their context. In [26], the authors tackle the lack of consumer preference considerations. However, they do not take consumer context into account. The authors of [27][28] describe an approach to tackle the mentioned cold-start problem within CF. The prediction of QoS or NFP values using CF is pursued in [29]. The authors assume that consumers who experience similar NFP values for one service also experience similar NFP values for other services. Although we also assume that consumers of a similar call context experience similar NFP values for one service, we claim however that due to the complexity of the Internet and other aspects, call contexts, which affect NFPs, are independent for each service candidate. Differences in call contexts are, hence, not only related to the caller but also the respective callee. For instance, a global player (e. g., Google) may provide a single services that is implemented in a world- or continents-wide load balancing scenario. Consumers all over the world or continents may experience similar NFP values for this service, however, these consumers will then experience other (local, non-balanced) services differently. Furthermore, NFP values may change considering time aspects and other contextual differences even for a single consumer. Similarly, [30] uses the relationships between services, their providers, and service consumers for a bi-directional recommendation basing service recommendation on a satisfaction degree, which is rather subjective than objective. Furthermore, the approach requires provider information that is hidden from service consumers in the SOC domain. Our approach does not require to overcome the concepts of SOC and we base recommendation on objective measurement data. Although CF could be employed for recommending prior recommendation decisions within our recommendation framework, it is disadvantageous, since changes in NFPs over time would not be taken into account. In contrast, within our continuous, objective recommendation knowledge updates, such changes are taken into account.

The authors of [31] use data mining methods for the discovery of services. Trust and reputation are also important aspects for the recommendation of services. Understood in a reliability context, there are approaches focusing on a trust-/reputation-based service recommendation [32][33][34][35][36]. In [33], the authors present a Knowledge-Social-Trust network model to determine the "trustworthiness of a service developer regarding specific user requirements and context" [33]. In [37], a similar concept is presented focusing on the awareness of social influence.

Within our survey, in order to determine relevant NFPs for service selection (Section IV), we discovered only a very few approaches that consider more than one NFP during service selection/recommendation. Outlined within this article, aiming at the various preferences of service consumers, service recommendation has to consider several NFPs. However, multi-NFP consideration within service recommendation is challenging due to the fact that the determination of the best-fit service instance according to service consumers' individual preferences result in a calculation task. Furthermore, NFPs have different scales of measurement and different optimization focuses. Therefore, the complete recommendation process cannot be left to machine learning alone.

III. Recommendation Framework

The first part of this section introduces the optimization aspects on which we focus within our recommendation framework, which is introduced in the second part of this section.

A. Optimization Aspects

In the introduction of this article, we outlined major characteristics of the Internet as an anonymous service market. Services depict to be black boxes. Service consumers have little to no knowledge about the implementation, sub-dependencies, system environment, or usage load. Besides the actual functionality, service consumers only experience the performance of services. *Performance* is experienced in the NFPs of services. NFPs can be measured at consumer side (e. g., response time) or they are stated (e. g., monetary charges of consumption); they can be static or dynamic. NFPs have different scales of measurement. For example, response time is a ratio scale, while the availability for a service at a specific call moment is nominal: a service is either available or not.

The performances of services vary. The complexity of the Internet and the collaboration of various system environments and networks are not evident to consumers, which is part of the design. The limitation of resources, volatile usage and loads of these resources, and incidents in general cause dynamic performance behavior at consumer side. In our analyses of two NFPs [2][3][38], we could determine a time-based behavior of the analyzed Web services. Although service consumers cannot look behind the curtain, they can observe context-based behavior. Within our recommendation framework, we consider a service call's context. In general, a *call context* attribute is an observable/measurable aspect at consumer side at the moment of a service call that may influence the NFPs of the called service. Examples of an open list of call context

attributes are service call date/time, location, and input size. The time stamp (also date aspects such as day of week, day of months, weekday, or weekend, etc.) of a service call can be relevant due to the load of limited resources, which cannot be determined, but experienced. For example, a Web service that provides information about TV programs can have high response times at prime time, when there is a peak load due to a lot of consumers. The location of a service consumer can also affect response times. For example, due to the network topology, service consumers within the same area (country or provider network) may experience better response times than other consumers who call from a different location because of different inter-network bandwidths. Also, whether consumers use a broadband connection or a mobile internet connection can also have an impact on the experienced performance of services. Although we consider solely non-functional aspects, the size of the input data for a service (non-functional aspect) may also influence the NFPs of a service due to functional aspects. For instance, a translation service may need more time to translate a book of several hundred pages than for a single-page document. Although there might be no difference in the transmission time, due to different processing times (processing is non-evident to consumers), service consumers experience different response times.

The third important aspect is *preference*. In general, consumers on any market have different preferences. For instance, for a specific functionality, some consumers are more interested in a fast response time and rather neglect higher monetary charges than others who want to have a service for free and rather experience higher response times. Therefore, whether a service is best-fit for a certain service consumer is an individual aspect. Functionally similar services distinguish themselves among each other in their NFPs, which are experienced at consumer side. Considering a consumer's preferences during service recommendation means considering preferences in the NFPs. As outlined above, NFPs have different scales of measurement. Furthermore, they also have different optimization functions. Recalling the examples of response time and availability above: for the ratio scale of response time, the optimization is focused towards the minimum; while the optimization focus of availability, which is nominal, is to select a service that has the highest (maximum) probability of being available. When the selection of a service instance is based on more than one NFP, NFP data has to be normalized in order to be comparable and calculable. In such a case, not all NFPs are equally important, so their importance has to be weighted and taken into account [1]. Our determination basis for the recommendation of the individually best-fit service is the calculation and comparison of a utility value. A *utility value* expresses the numerical degree of how much a service meets the preference goals of a consumer. The higher a value, the better. For the calculation of this value, we introduce the concept of utility functions. In general, *utility functions* express the mathematical relationship of the expected, normalized NFP values in order to meet the selection preferences of service consumers with a numerical output (interval scale). This definition emphasizes that in general our framework is not limited to a specific structure of utility functions. As an initial implementation, we use a weighted scoring model that expresses the impact of each normalized NFP towards a consumer's preferences. They "can be captured in a vector of real values, each representing the weight of a corresponding quality metric [NFP]" [2]. For instance, lowest response time is more important (weighted: 60 %) than lowest price (weighted: 40 %) would results in a utility function $U(ResponseTime, Price) = 0.6 \times ||ResponseTime|| + 0.4 \times ||Price||$, where $||\cdot||$ normalizes $ResponseTime$ and $Price$, respectively, between 0 and 1 [2]. Within a single-tier recommendation, the experienced NFPs of services are influenced by consumer contexts. Hence, within the same call context (e. g., time, weekday, location, and type/size of input data), consumers with different preferences experience statistically similar NFPs, but the calculated utilities are different due to different utility functions.

B. Framework Components

With the characteristics of the Internet as an anonymous service market, the focus of the design of our framework is to cope with perpetual change. Furthermore, it aims at a fully automated process without any human interaction and it is supposed to work with existing Service-Oriented Architecture (SOA) infrastructures. Due to the general design of SOC with its encapsulation aspects, services are treated as black boxes within our framework.

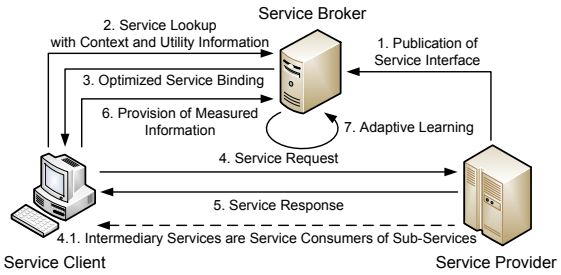


Figure 1. Enhanced SOA Model for Optimized Service Selection. [2]

For our framework, which we introduced in [2], we enhanced the traditional SOA model. The enhanced model, depicted in Figure 1, is extended by the following steps: The service lookup in step 2 is enhanced with call context and utility (preference) information. Based on this information, the service recommendation component of the broker provides in step 3 the service binding of the individually best-fit service based on the knowledge base of previous service calls of similar call contexts and similar preferences. The experienced NFPs during the actual service call in step 4 and step 5 is then submitted to the learning component of the service broker in step 6. Step 7 denotes the learning of the measurement details of the service call in order to update the knowledge base of the recommendation unit. The roles of service consumers and service providers are not always disjunct. In order to provide a certain service functionality, service providers may have further sub-dependencies and consume sub-services, which puts them in the role of a service consumer (step 4.1). Within our framework, we call such value adding services intermediary services.

The architecture of our recommendation framework is illustrated in Figure 2. Since our framework aims at the performance optimization at consumer side, the framework is logically split into a local and a central component. The local component is integrated in SOA environments (integration

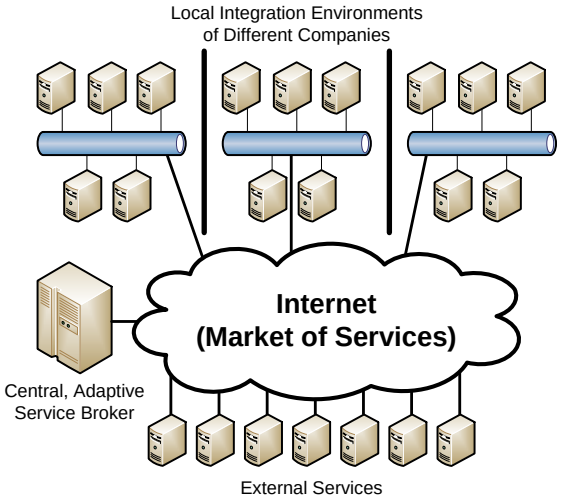


Figure 2. Architecture of Recommendation Framework.

platforms, middlewares, etc.) at consumer side. Its purpose is 1) to manage the call context and selection preferences (in the form of utility functions), 2) to manage dynamic bindings, 3) to measure the performance (NFPs) during service calls, and 4) to submit the observed information to the logically central component. Because of the automation aspect, the framework focuses on objective decision criteria in the form of measured or stated information. Hence, there is no consideration of human end-consumer ratings, which would require human interaction and that could also be rather subjective than objective, although the design could integrate such information in general. As illustrated in Figure 2, the local component is seamlessly integrated in existing infrastructures, our framework does not require any adjustment of existing implementations or systems. Through the extension of existing integration platforms, existing static bindings are replaced through dynamic bindings. Since service calls are dispatched through these integration platforms, calling components only experience optimized service performance due to the recommendation framework, but no changes in configurations or implementations.

The central component is responsible for the centralized functions. Its purpose is 1) to collect the feedback data from the local components of service consumers containing the measurement data including the call context and preference information, 2) to process and integrate the collected information into the recommendation database, 3) to recommend service candidates based on a given call context and utility function, and 4) to notify local components on dynamic binding updates. Within the illustration, the central component depicts to be a bottle neck and a single point of failure. In order to avoid these threats, the logically central component has to use distributed and high availability technologies.

In general, service recommendation aims at the optimization of performance at consumer side, which is a time-critical challenge. For the recommendation of a best-fit service, we follow two approaches. The first approach is dynamic binding. *Dynamic binding* is part of the local component, at client side. Using dynamic binding, the local component registers the desired service functionality, call context, and utility function at the central component and receives an initial best-fit service candidate. In the event of an update, the central component notifies the local component (publish-subscribe pattern). The second concept is dynamic service calls. The *dynamic service call* concept includes the recommendation request as part of each service call. Similar to the dynamic binding, information about service functionality, call context, and utility function has to be provided. In order to tackle the time-critical drawback of service recommendation, we provide a divided architecture of the recommendation unit. The architecture of the central recommendation unit is illustrated in Figure 3. It is separated into a *foreground* and a *background model*. The knowledge preparation for service recommendation contains time-consuming tasks. The decoupling of these time-consuming tasks in the background model from the foreground model, which handles the time-critical recommendation lookups, reduces or even avoids the costs in terms of service time. The tasks in the background model are conducted asynchronously. The output of the background model is the recommendation knowledge, which is used in the foreground model. Since service recommendation within our framework is based on several aspects, which were described in the previous section, the incoming data has to be pre-processed within both models.

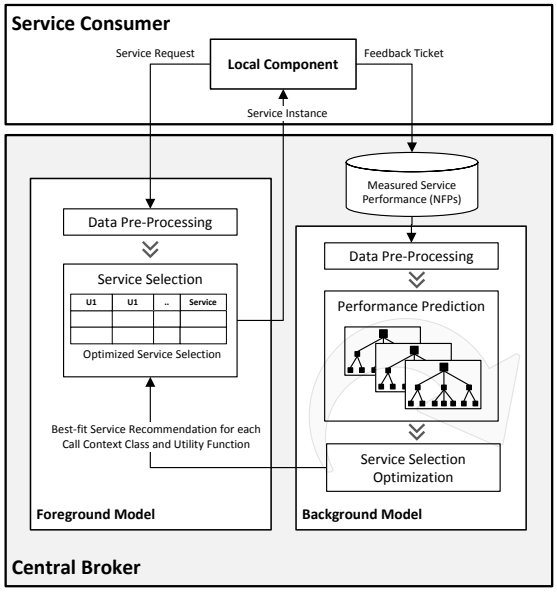


Figure 3. Foreground and Background Model Within our Framework. [1]

Based on the shared knowledge gained from the collected measurement data, the framework prepares recommendation entries with the best-fit service for each combination of call context and preference (utility function), or their cluster. For the prediction of the performance of services or the individual best-fit service, machine learning methods can be employed. Section V describes the general machine learning approaches for their employment. Finally, Section VI presents the evaluation of these machine learning approaches and methods for the employment of service recommendation as well as an initial implementation and validation within our framework. However, before we elaborate on the knowledge preparation for the recommendation task using methods of machine learning, the actual relevant NFPs have to be determined. Section IV presents the results of our literature analysis of scientific

papers in order to identify the relevant NFPs for service selection/recommendation.

IV. Relevant NFPs for Service Recommendation

This section describes the results of our survey about relevant NFPs of services for their consumption and, hence, their selection/recommendation. It is a digest presenting the summary of the aspects that are relevant for the context of this article. The description and the detailed results of the survey can be found in the appendix.

A. Introduction

In order to optimize service consumption at consumer side, it is important to determine the relevant NFPs for service selection/recommendation. Furthermore, it is important to determine whether the relevant NFPs are static or dynamic. While static NFPs can be optimized once, dynamic NFPs are more difficult within service selection since they are likely to change often and require dynamic binding. Within the literature survey, the focus was set on solutions aiming at optimized service selection based on NFPs. The results of the study are based on the analysis of the scientific papers that had been published in the past ten years on conferences in the Service-Oriented Computing (SOC) or related domains. They are founded on service-selection-based conference contributions and, hence, reflect the condensed position of the scientific community in this research area within the recent years. Conference papers are a good foundation because their contributions address state of the art solutions for current problems mostly from researchers but also from industry side. The goal of the survey is to determine the relevant NFPs for service selection and whether the NFPs used within the approaches are theoretically profoundly discussed and validated in practice.

B. Results of the Analysis

Presenting the results of the analysis of our study, the NFPs during service selection as well as during adaptation and consumption from a consumer's perspective are analyzed according to two measures:

Occurrence This measure expresses the amount of conference papers that refer (mention, discussion, or validation) to a specific NFP within all relevant categorized papers.

Count Count represents the amount of overall references of an NFP within all relevant categorized papers. We distinguish count between absolute and relative count. *Absolute count* is the absolute amount of all references. *Relative count* is a normalization of the absolute count within a paper. It is the percentage of references to a specific NFP among all NFP references of a paper.

An NFP with a high *occurrence* can be considered to be widely accepted to be relevant, since it is referred in many papers, while *count* indicates how much text is dedicated to an NFP in absolute terms (*absolute count*) or relatively in the papers (*relative count*). However, these two measures are not sufficient to deduct the quality of the references. In order to satisfy this aspect, we consider the quality of each NFP reference by its occurrence category, cf. Table I. Furthermore, we also differentiate between the papers regarding their semantic quality towards our finding objectives. Therefore, we categorize each paper according to its topic relevance (cf. Table II).

1) NFPs Referred in Conference Papers: Without any consideration of the paper and occurrence categories, we focus on the plain occurrence of NFPs in relevant papers and the plain count of the NFP references in these papers on an absolute and relative basis. In 91 % of the relevant conference papers, *response time* is mentioned (or discussed/validated). Furthermore, the ranked list continues with the following NFPs: *availability* (67 %), *reliability* (45 %), *cost* (40 %), *throughput* (35 %), and *price* (27 %). Considering the textual distribution on an absolute and relative basis, Figure 4 reveals a similar order. However, two things can be noticed: First, *response time* has the highest share among all NFP references. Furthermore, there is a big gap between the relative and absolute count for *trust*. The reason for this is related to the fact that *trust* is extensively discussed in some conference papers. Some researchers argue from their point of view that *trust* had been fiercely neglected compared to other NFPs that are also mentioned frequently in these papers. Comparing absolute and relative count, for some researchers, *trust* is a very important aspect, however for the majority, this NFP is not as relevant as others.

2) Profoundness of the NFP References: For further analysis, we grouped the NFPs in categories. The NFPs within their categories are listed in Table III. So far, we did not take the quality of the NFP references into account. As introduced above, we used paper and occurrence categories in order to reflect the impact quality of the papers and the references themselves into account. Figure 5 lists the grouped NFPs

TABLE I. OCCURENCE CATEGORIES

Category	Description
O-A	NFP used in service selection is *validated in practical or experimental context*
O-B	NFP used in service selection is *theoretically discussed in detail*
O-C	NFP used in service selection is *mentioned but not discussed*

TABLE II. PAPER CATEGORIES

Category	Description
P-A	Conference papers with main focus on *service selection/recommendation*
P-B	Conference papers with main focus on *adaptation of composite services*
P-C	Conference papers with main focus on *service computing* in general
P-D	Conference papers *without main focus* on either of above categories, however, in which NFPs during service selection, adaptation or consumption are mentioned or discussed
P-Z	Conference papers that do not mention NFPs of services at all or that do not fit into the above categories

TABLE III. NFP CATEGORIES

Category	NFPs
Service Time	Delay, Duration, Execution Time, Latency, Performance, Response Time, Timeliness
Service Success	Accessibility, Accuracy, Availability, Dependability, Dependency, Fault Tolerance, Reliability, Successability
Monetary Aspects	Cost, Price
Service Trust/Reputation	Privacy, Reputation, Security, Trust
Service Bandwidth	Bandwidth, Scalability, Throughput
Misc	Energy Consumption, Location, Utilization
Design Aspects	Adaptability, Composability

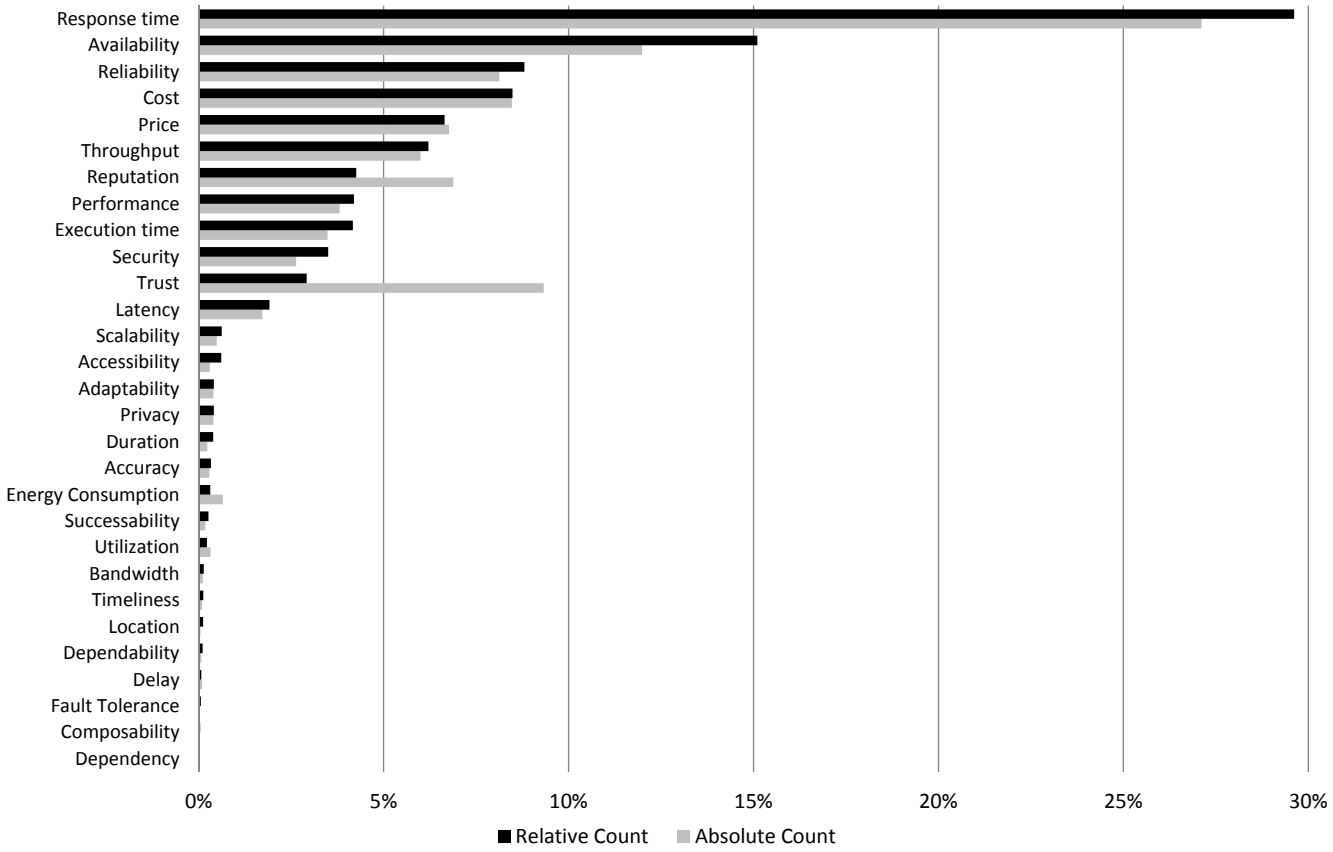


Figure 4. Overall Relative and Absolute Count of NFPs.

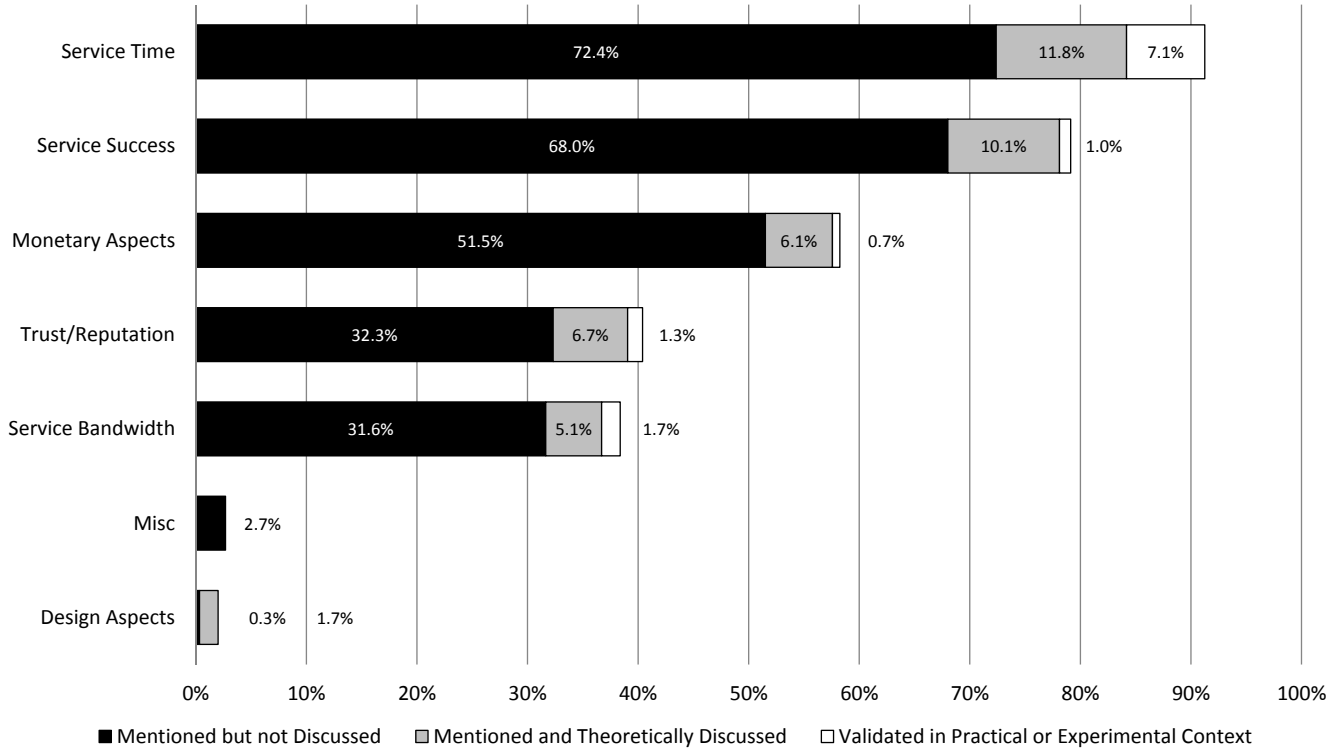


Figure 5. Paper Occurrence of Each NFP Category According to the Occurrence Categories.

regarding their paper occurrence in the form of the percental amount among all relevant papers in which the NFPs of a category occur (mentioned/discussed/validated). If a paper has two or more NFPs that belong to the same NFP category with different occurrence categories, the paper is counted in the highest occurrence category. On average over all NFP categories (disregarding the *Misc* and *Design* categories), 83 % (4 basis points deviation) of all relevant papers only mention NFPs but do not discuss them in detail. On average, 13 % (2 basis points deviation) discuss them in more detail while only 4 % (3 basis points deviation) also validate them in simulations or experiments. This means that a vast majority of relevant conference papers do not elaborate in detail on the NFPs they mention.

C. Conclusion

As a result of this survey, we determined a list of NFP that are relevant for the selection/recommendation of services. *Response time* is determined to be the most relevant NFP for service selection. It occurs in over 90 % of all relevant categorized papers. Furthermore, with a large gap, its textual distribution with an overall share between 25 % and 30 % also verifies this relevance. The top-3 relevant NFPs are all dynamic. Within the top-6 relevant NFPs, two third of the NFPs are dynamic whereas the rest is rather static. However, *cost* and *price*, which are similar, can also be dynamic depending on the defined price models of service providers.

Considering the quality of the references within the analyzed conference papers, the results reveal that 83 % of the references only mention these NFPs without any further discussion or validation. Only 13 % of the paper occurrences discuss them and even less (4 % with a standard deviation of 3 basis points) validate them in an experimental or practical context.

As the results of a broad research-community-based survey, *response time*, *availability*, *reliability*, *cost*, *price*, and *throughput* are determined to be the top-6 of the relevant NFPs during service selection. Their mostly dynamic characteristic requires dynamic binding and a continuous learning/adaptation of the recommendation knowledge for an optimized service selection. Although response time seems to be the most important NFP among all relevant NFPs, the ranked results list confirms the importance of a multi-NFP service selection, since other NFPs still achieved a considerable relevance.

V. Employment of Machine Learning in Service Recommendation in General

Machine learning can be employed for the recommendation unit within our framework, which we described in Section III. In general, there are two approaches for the preparation of the recommendation knowledge in the background model using machine learning methods: the prediction of a numerical or a nominal value. In machine learning, regression aims at the prediction of numerical values based on attribute values, while classification focuses on the determination of the affiliation to a certain class based on attribute values. Although both approaches fit in general, they have different focuses within the recommendation process. Figure 6 illustrates the different steps that are required for the retrieval of recommendation knowledge. The gray-shaded boxes highlight the steps for which machine learning methods are used.

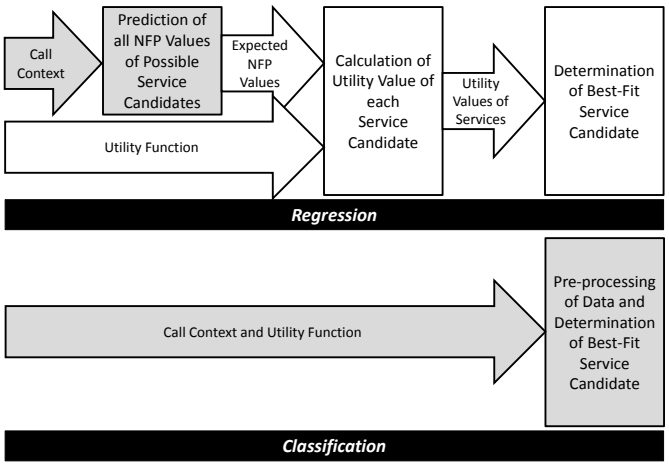


Figure 6. Learning Steps for Service Recommendation Within the Regression-/Classification-based Approach.

A. Regression-based Approach

Within the recommendation focus, regression can be used to predict each NFP value (e. g., the expected response time and the expected degree of availability) based on call context values (e. g., calling time, weekday, and location). The gray-shaded boxes in the regression part of Figure 6 denote the actual determination of the best-fit service is not included in the learning. The drawback is that machine learning has to be conducted for each NFP individually and the actual utility value for the best-fit determination has to be calculated. However, these higher efforts have several benefits at the same time. For each NFP and call context combination, the NFP value has only to be predicted once, while the best-fit service can be calculated for each utility function individually. Furthermore, since the NFP data for all service instances are considered, the ranking of the second, third, etc. best-fit services can be used to achieve a higher overall utility gain. Underdog and quick starter strategies can also be implemented, since the performance data of service calls of the past still remain [3][38].

B. Classification-based Approach

In general, classification focuses on the determination of the affiliation to a certain class based on attribute values. In service recommendation, consumers are ultimately interested in the selection of the best-fit service. For this, classification can be used in order to determine the best-fit service for a specific call context. With this approach, the learning method focuses directly on the best-fit determination. For this, the training set has to be pre-processed: for each combination of call context and utility function, the best-fit service has to be determined based on the measurement data. As a result, classification directly determines the best-fit service (class) within a call context and utility function combination without the consideration of the NFP values. The benefit using classification is to omit the calculation steps after prediction. Disadvantageously, however, the best-fit service has to be learned for each call context and each utility function; whereas having the NFPs predicted for a call context as an intermediate step, the best-fit service can be calculated for other utility functions without new learning. Furthermore, old

service instances are automatically not further considered and, hence, sorted out. Disadvantageously, underdogs can never prove themselves since the approach is only focused on best-fit service recommendation and non-best-fit ones are neglected or not invoked at all. Also, there is no differentiation among non-best-fit services, which is important in a non-accurate prediction in order to still create a high utility gain [3][38].

C. Selection of Machine Learning Frameworks

In [1], we evaluated machine learning methods as well as frameworks that can be employed for our purpose. There are several aspects for the evaluation of machine learning methods such as speed, accuracy, scalability, robustness, and interpretability [39][40]. Table IV lists the requirements that we used for the selection of machine learning methods.

TABLE IV. REQUIREMENTS FOR THE SELECTION OF MACHINE LEARNING METHODS [1]

Speed	describes how efficient the machine learning method performs concerning the training and prediction time. Furthermore, this aspect also concerns the overall machine learning process as a 'critical path' from end-consumer side.
Accuracy	describes how effective the machine learning method performs: Degree of correct classification or coefficient of determination in regression [40].
Scalability	considers the ability of the method to be efficiently applied to a large dataset [40].
Robustness	describes the ability to make correct classifications and predictions, given noisy or missing data value. It also considers whether the method is able to run automatically in a changing environment [40].

In [39], the author published a comprehensive overview of established supervised machine learning techniques. This overview provides useful information for method selection, highlighting the benefits and drawbacks of each method that helped us for further evaluation. For the selection of machine learning frameworks, we focused on a Java integration ability, a high degree of automation, a strong dependence between the library and the implemented method, and a general approach being not limited to specific purposes. Furthermore, we preferred open source frameworks, which are freely available to the public [1].

Because of their extensive collection of classical machine learning methods as well as new algorithms with state of the art concepts for incremental learning, we chose Weka [41] and MOA [42]. Both frameworks provide a high degree of automation and are fully integrated in Java. Furthermore, Weka is also used by other software in this sector. The frameworks are open source and contain different methods and algorithms for pre-processing, classification, regression, clustering, association rules, visualization, and include several state of the art algorithms. Furthermore, MOA also has a focus on online algorithms processing data streams [1].

VI. Evaluation of the General Classification- and Regression-based Approaches

In this section, we analyze the previously outlined two general machine learning approaches, machine learning methods, and strategies for their employment within service recommendation considering optimization aspects, which were identified in Section III-A. At this point, we want to point out that since our research focus is set on the SOC domain, we do not analyze or optimize machine learning algorithms and only use them as black boxes. We analyzed two general machine learning approaches for their capability, benefits, and disadvantages of their employment within service recommendation. Our practical assessment is based on real-world measurement data as well as simulated data in order to conduct a statistically more fine-grained analysis. Based on the evaluation of the two general approaches, the more suitable approach was then implemented and validated in our framework. For this validation in the overall recommendation framework, the three machine learning methods were implemented for their evaluation.

Outlined in Figure 6, we conducted analyses of the two general machine learning approaches using classification- and regression-based machine learning methods for their advantages and drawbacks when using them for service recommendation. This section outlines the evaluation scenario and the results of these studies.

A. Evaluation Scenario

The analyses were described in detail in [3][38]. For more details, interested readers are referred to these references.

1) Objectives of the Analyses: The results in this section are summarized, based on two analyses that focused both on the evaluation of both learning approaches, however, which had different sub-objectives. The first analysis focused on a general comparison between both approaches with their methods. Furthermore, it focused on the simulation of performance data based on the real-world measurement data, in order to conduct a more fine-grained analysis. For this, we had to analyze the service profiles within the measurement data. The second analysis focused on the application of learning strategies. Furthermore, the strengths and weaknesses of each approach within certain presumed and simulated service performance profiles were focused in the analysis.

2) Evaluation Setting: For the conduction of the analyses, we developed a Java-based software using the Weka and MOA frameworks within the evaluation scenario. We implemented the overall recommendation process of each approach illustrated in Figure 6. For the learning task in the processes, we chose the Fast Incremental Model Tree with Drift Detection (FIMT-DD) algorithm for regression. FIMT-DD focuses on time-changing data streams with explicit drift detection [43]. For classification, we chose the implementation of a DecisionStump [44]. We chose both machine learning methods because of the requirements in Table IV as well as good results in initial pre-tests [3][38].

3) Evaluation Criteria: The purpose of both analyses was to evaluate the employment of both machine learning approaches for service recommendation. For their evaluation, we were interested in their overall contribution to the recommendation of best-fit services within the process illustrated in Figure 6; hence, in the benefit of the consumers.

In order to determine the benefit of the recommendation, the accuracy of predicting the best-fit service does not solely reflect the strength of an approach since the recommendation of the second best-fit might be as good as the determination of the actual best-fit if it creates a similarly high optimization benefit. Therefore, within service recommendation, the optimization aims at the experienced performance/utility gain and not at the determination accuracy of the overall best-fit. The evaluation has to take into account how good the optimization is instead

TABLE V. EVALUATION INDICATORS [3][38]

RT Gain Prediction/ Random	This figure indicates the overall percental response time gain when using the machine learning approach including the determination of the best performing service for recommendation in comparison to a random selection of services.
RT Gain Best/Random	This indicator is the overall percental response time gain when continuously choosing the best performing service in comparison to a random selection of services. This is the optimum, i. e., when continuously choosing the best performing service, response time is fully minimized.
Overall RT Gain Achievem.	The ratio between the figures above. It indicates to what degree the response time gain of the prediction achieved the optimum.
RT Ratio Prediction/ Best	It is the ratio between the total response times of prediction in comparison to a continuous selection of the best performing service. Since the optimum is the denominator, this figure is always $>= 100\,\%$.
RT Ratio Random/Best	The ratio between the total response times of a random selection in comparison to a continuous selection of the best performing service. Since the optimum is the denominator, this figure is always $>= 100\,\%$.
Overall Optimization Achievement	It indicates the optimization degree in percent between the response time of the worst service towards the response time of the best service for each prediction case: $\left(1 - \frac{RT_{Prediction} - RT_{Best}}{RT_{Worst} - RT_{Best}}\right) \cdot 100$
Best Choice	It expresses the amount in percent of the prediction of the actual best(-fit) service candidates.

of the best-fit classification accuracy. Therefore, our key figure is *performance gain* respectively *utility gain* when considering preferences. As for this analysis, the focus was set on the evaluation of the machine learning approaches. Since machine learning methods are employed for the determination of NFPs as the input for the calculation of the utility value, which is then used for the determination of the best-fit service, within the regression-based approach respectively for the determination of the actual best-fit service based on the calculation output of the utility function with the input of one or several NFPs, we could simplify the overall process and could use only one NFP for the prediction, in order to reduce unnecessary calculation overhead. In other words, since the calculation of the utility function is not part of machine learning steps within the approaches (cf. Figure 6), it could be simplified for this evaluation. Instead of determining the utility gain, we evaluated the response time gain.

Table V lists the evaluation indicators that we defined for the evaluation of both machine learning approaches. Note that the table also contains the definitions of indicators that we used in the first analysis (RT Gain Prediction/Random, RT Gain Best/Random, Overall RT Achievement, and RT Ration Random/Best) but not in the second analysis and vice versa (Overall Optimization Achievement). The main indicators of the first analysis based the evaluation on a direct comparison between each approach and random selection. However, random selection is also a simple recommendation approach. A comparison based on relative indicator values turned out to be sub-optimal. In order to get comparable indicator values, we used indicators that focus on an absolute scale in the second analysis. Therefore, they are more sufficient for comparisons between different approaches and methods.

4) Preparation and Processing of the Datasets: Analyzing the results of our study in [2], we could discover a pattern-based periodic behavior of the analyzed real-world Web services in natural periods of time such as differences between working days and weekends or time of day. Our hypothesis before the conduction of this analysis was that machine learning approaches contemplating periodic behavior achieve better results. For this, additionally to the basic attributes *date*, *time*, and *response time*, we pre-calculated further attributes and provide enhanced data with focus on natural periods, which can be used by both approaches. These attributes are described in Table VI. While regression is able to focus on a time line, classification is not. However, these additional attributes allow classification to consider such natural time line based aspects for learning and prediction, such as working day or day of week.

Due to the differences between both learning approaches, further pre-processing was necessary. For classification, the dataset had to be prepared in order to train the best-fit service within each call context. In our case, for each date and time (nominal value of hour), the enhanced measurement entry with the minimum response time (desired best-fit service) remained in the dataset. So, the learning method only focused on the fastest (best) services, in which we are ultimately interested. For regression, no best-fit focus was set since the regression approach focused on the actual (NFP/response time) value prediction. However, since each NFP for each service has to be trained, the dataset had to be split for each combination of NFP and service.

After the pre-processing of each dataset (measurement or simulation), the datasets were divided into a training and a validation sub-set after pre-processing. Because of chronological aspects, the datasets could only be split into training and validation sets. N-fold cross validation could not be applied for that reason. While the training dataset was used for the training of the model, the validation dataset was used to validate and evaluate the prediction results of the recommendation processes. In the first analysis [3], the dataset was split in the middle. The first half of the dataset was used for learning, the second dataset was used to validate the prediction results. In contrast to initial analyses, in the second analysis [38], a sliding split point evaluation was conducted. Depending on the split point, the drawback of the initial analysis was that the analysis results varied. This was due to the varying performance profiles during the measurement period. Analyzing the measurement data, some Web services experienced temporal performance changes, which confirmed our focused aspect of the perpetual change characteristic of the Internet as a service market. In

TABLE VI. STATISTICAL ENHANCEMENT OF ATTRIBUTES [3]

DayOfMonth	Extracted day of month from date
Hour	Extracted hour from time
Weekday	Determined nominal day of week from date *(classification only)*
Workingday	Determined whether day is a working day (Monday to Friday) *(classification only)*
RT_Xmin_AVG $X = \{61, 121, 181, 361, 721, 1441, 2881, 7201, 10081\}$	Response time mean of all records (chronologically) within the last 1, 2, 3, 6, 12 hours, and 1, 2, 5, 7 days
RT_X_AVG $X = \{40, 80, 160, 240\}$	Response time mean of the previous x records (chronologically; without consideration of any other attribute)
RT_X_AVG_Hour $X = \{4, 12, 20, 28\}$	Response time mean of the previous x records within the same hour value (considering 1, 3, 5, 7 days of the same nominal hour)
RT_X_AVG_Weekday $X = \{4, 8, 16\}$	Response time mean the previous x records within the same weekday value (considering 1, 2, 4 weeks of the same nominal day of week)

order to achieve statistically generalized results, the split point between training set and validation set was iterated on a day by day basis for each analyzed aspect. Depending on the period (and window sizes), it could result in a statistical mean of up to $\frac{n}{24m} - 1$ iterations (for the measurement input 170 iterations per scenario), for n data entries and m data records per hour (one record for each service) [38].

5) Learning Strategies: Employing machine learning for the recommendation of best-fit services requires learning strategies. Within this employment, it is necessary to analyze the impact on the overall recommendation results when considering the amount of training and prediction data. In other words, how much data is necessary and beneficial for training a model, and for how long is such a trained model reliable for good service recommendation?

In the first analysis [3], there was no focus set on any learning strategy. The results of that analysis are based on a two split (50:50) conduction. The learning model was trained on the first half, while the second half was used for the validation and evaluation of the prediction results.

For the second analysis, the following learning strategies were analyzed. In order to address the research questions related to the optimal learning strategies, a prediction window of various sizes was applied to determine the optimal training/prediction interval ratio for the updates of the foreground model (Figure 3) [38].

Incremental learning This learning strategy continuously updates the learning model. Any strategies on changes and their impact on the model have to be dealt by the learning method such as drift detection.

Sliding window learning This learning strategy applies a fixed window of previous measurements for the training of the learning model.

B. Evaluation Datasets

The machine learning approaches were analyzed and evaluated using measurement as well as simulation data. The latter focused on certain aspects during the evaluation.

1) Measurement Data: For both analyses, the measurement data was gained from four real-world stock quote Web services, which we already partly used in [2]. The services provide similar functionality, so they are functionally substitutable among each other. In the first analysis, the measurement dataset contained 3,223 measurement entries obtained in 34 days. The dataset in the second analysis contained 16,441 measurement entries obtained in a measurement period of 185 days. The dataset of the first analysis is also the initial subset of the second analysis. Therefore, the measurement data of the second analysis is the long term version of the dataset in the first analysis. Each entry contained the *date*, *time*, the consumed *service*, and the measured *response time* of a service call. Within each measurement period, each Web service was called on an hourly basis. If a service was not available or timed out (30,000 ms), its entry was not added to the set. Hence, up to four data entries were obtained per hour for the dataset [3][38].

2) Simulation Data: In contrast to measurement data, with the ability to adapt the parameters, scenarios can be simulated. The simulation of measurement data enabled to challenge machine learning approaches and to analyze their performance in certain scenarios. Within the measured real-world Web services, the statistical characteristics showed easily distinguishable performance profiles of some services. In order to compare the strengths and weaknesses of the machine learning approaches, more challenging scenarios had to be simulated where the service profiles are harder to distinguish. In both analyses, simulation data was produced in order to analyze certain aspects. The following simulation profiles were generated [3][38]:

In the first analysis [3], the initial goal was to simulate data that closely reflects the characteristics of measured real-world Web services. With such data, it would be possible to adjust certain distinguishable profile characteristics in order to challenge the machine learning methods during the analysis. Based on the analysis of the measurement data and the observations made in [2], two simulation profiles were created [3]:

Normal distribution profiles with similar statistical characteristics of the measured services The visualization of the measurement data revealed that the real-world data had a massive distribution in certain intervals with diverse outliers. We presumed a normal distribution of the measurements with certain extraordinary outliers. We used the Gaussian distribution (cf. [45]) for the simulation of basic interval where most of the values occur and one with the identified outliers using the statistical mean and standard deviation of the services' intervals in order to achieve similarity [3].

Normal distribution profiles with similar statistical characteristics of the measured services and periodicity Additionally to the first profile, we added some periodicity to the profile in order to simulate the in [2] observed differences between certain natural time-/date-based characteristics. For this simulation, we added a periodic component to the normally distributed basis such as a working day pulse, daily periodic waves, and weekly peaks. For the simulation of working day pulse, we used the Fourier series expansion in order to produce a rectangular pulse wave (cf. Equations (1) and (3) for daily periodicity). With these periodic components additionally to the random component, we expected the classification approach to achieve better results because the additional attributes in the pre-processing aims at these natural periods. Classification is in general optimized for such periodic preparations [3].

In order to challenge both approaches, we approximated the statistical mean of all simulated services step-by-step to a defined level (target mean value). Our presumption was that the more the services approximate the statistical mean, the worse the beneficial gain results of the machine learning approaches in comparison to a random selection, since the standard deviation values in the relevant interval are close to each other. The purpose of the approximation was to evaluate, which approach tackles the challenge better and to what approximation degree machine learning approaches can still be beneficial for the recommendation task. For the challenge of the evaluated approaches and methods, the mean values of the service profiles in the first analysis were approximated in the following steps: 50 %, 75 %, 87.5 %, 93.75 %, and 100 %.

$$f(t) = \frac{\tau}{T} + \sum_{n=1}^{1000} \frac{2}{n\pi} \sin\left(\frac{n\pi\tau}{T}\right) \cos\left(\frac{2n\pi}{T}(t+p)\right) \quad (1)$$

$$f(t) = \frac{2}{\pi} \sum_{n=1}^{1000} (-1)^n \sin\left(\frac{(t+p)n2\pi}{T}\right) \quad (2)$$

$$f(t) = \sin\left(\frac{2\pi(t+p)}{T}\right) \quad (3)$$

In contrast to the first analysis, the focus of the second one was set on the explicit analysis of isolated aspects. The analyzed learning approaches with the employed machine learning methods should be challenged for their strengths and weaknesses in the following presumable scenarios. For this, the competing simulated services all had identical profiles with distinguishable differences in each focused aspect. The initially, clearly distinguishable profiles were also approximated in 10 % steps in each conducted iteration until they were fully approximated in order to challenge the learning approaches and methods. In a full approximation, their profiles are identical (up to random noise) [38]:

Normal distribution profiled data As in the initial analysis, we assumed normally distributed response times of Web services around a mean value (with a certain standard deviation and variance). Normally distributed response time data for four services with a similar mean, standard deviation, and variance was created. These response time mean profiles of the services were initially, vertically shifted and were then approximated step by step. Fully approximated, their statistical mean is identical.

Cyclic spikes up/down On the basis of normally distributed response time data around the same mean value, these profiles contained cyclic/periodic spikes that go in one profile up and in the other profile down. Spikes going up simulate services that have suddenly longer response times, while spikes going down simulate sudden response time improvements. For their creation, a saw tooth generator (cf. Equations (1), (2), and [46]) in combination with an iceberg filter that are added to the basic normal distribution line was used. Again, all created services are similar. They distinguish themselves only in their horizontal shift. Fully approximated, their horizontal shift is identical.

Acyclic spikes up/down These profiles have several acyclic spikes and different level shifts in combination with an iceberg filter. Using several cyclic spikes with very long periods in spikes generations in combination with pulse train shifts and the iceberg filter, a complete acyclic/aperiodic behavior could be simulated. Again, all services have the same mean response time and in a fully approximated case and their spikes are overlapping.

C. Evaluation Results

The results presented in this section are extracted from [3][38]. Interested readers are referred to these references for more details on the conduction and results of each analysis. The first analysis [3] used a two-split analysis in the middle of each input dataset for the training and test sub-sets. Additionally, each analysis scenario was re-conducted ten times in order to achieve statistically profound results within the random components. The presented results of this analysis are mean values. In further tests, we discovered that the profiles of the measured services change and that depending on the used split point, the results of the evaluation indicators change. Therefore, we conducted a sliding split point evaluation in the second analysis [38]. While the first analysis used a split point in the middle of each input dataset, the second analysis [38] considered statistically profoundness in terms of the a sliding split point between training sets and validation sets iterated on a day by day basis.

1) Measurement Data: The results of the first analysis using measurement data are presented in Table VII. Within the ten iterations of which the random components (random selection) was re-conducted, the mean values and the values of the best iteration are presented for each approach. Among all indicators introduced in Table V, the most important indicator, which was also used to determine the best iteration, is *Overall RT Gain Achievement*. The second important indicator is *Best Choice Prediction*. Recall, the best choice indicator is a certain kind of accuracy, however, considering the optimization achievement within service recommendation, the actual response gain (or utility/performance gain in general) is more important, since it compares the improvement using machine learning with the overall achievable optimum, which is the RT gain when theoretically choosing always the best-performing service instance. Furthermore, it also takes the optimization degree among the recommended non-best services into account. A positive RT gain is supposed to indicate the percental response time reduction compared to a random selection. Furthermore, the table lists the amount of the selection of services that were not available at the moment of selection. When considering the figures of Table VII, please note that random components are re-conducted in each iteration, while the actual measurement data remains unchanged. The prediction and hence the recommendation results also remain unchanged in each iteration (table columns: RT ration prediction/best; non-AV section in prediction; best choice prediction). Using real-world measurement data, regression achieved better results than classification. If we compare the best choice figures for prediction in the table, we see that classification seems to have a higher accuracy. This might appear odd, when we compare the RT gain and the overall RT achievement figures. Recall, in service recommendation, accuracy is less important than the actual performance gain. The recommendation of the second best service might be in terms of RT gain almost as good as to recommend the actual best. Comparing the RT gain and best choice figure of classification and regression, we see that classification has a higher best choice, while the RT gain is better for regression. This reveals that although regression was weaker in the best choice prediction, it still achieved a higher RT gain, which shows that regression's strength is in a ranked determination, while classification does not consider any ranking. Furthermore, in 1.46 % of the cases (6 times in 17 days), classification recommended a service instance that was not available at that moment. In order to verify our assumption regarding periodic strengths, we had a look at the classification model. The model based its decisions mostly on the sliding window of the previous response time values within the same nominal hour. So, our periodicity assumption was proven.

In the second analysis, the focus was set on optimal

TABLE VII. ANALYSIS RESULTS OF MACHINE LEARNING APPROACHES USING MEASUREMENT DATA [3]

		RT Gain Predict./Rand.	RT Gain Best/Rand.	Overall RT Gain Achvmt	RT Ratio Predict./Best	RT Ratio Rand./Best	Non-AV Prediction	Non-AV Random	Best Choice Prediction	Best Choice Random
Classification	*Mean*	66.57 %	82.33 %	80.86 %	189.17 %	565.97 %	1.46 %	4.61 %	80.09 %	24.51 %
	Best	69.76 %	84.01 %	83.03 %	189.17 %	625.66 %	1.46 %	5.58 %	80.09 %	23.05 %
Regression	*Mean*	75.74 %	82.44 %	91.87 %	138.16 %	569.53 %	0.00 %	4.37 %	66.99 %	25.00 %
	Best	77.78 %	83.92 %	92.68 %	138.16 %	622.05 %	0.00 %	5.58 %	66.99 %	24.27 %

TABLE VIII. DIFFERENT PREDICTION WINDOWS WITHIN INCREMENTAL LEARNING [38]

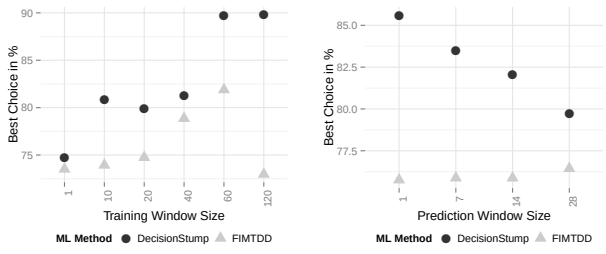
	DecisionStump				*FIMT-DD*			
Win. Size	**Achievement**		**Best Choice**		**Achievement**		**Best Choice**	
Prediction	*mean*	σ	*mean*	σ	*mean*	σ	*mean*	σ
1	97.10 %	6.10 %	82.26 %	22.89 %	97.04 %	3.89 %	73.35 %	21.47 %
28	96.34 %	2.23 %	72.77 %	22.94 %	97.02 %	1.60 %	72.78 %	13.59 %

TABLE IX. SLIDING WINDOW SCENARIO WITH DIFFERENT TRAINING WINDOWS [38]

	DecisionStump				*FIMT-DD*			
Win. Size	**Achievement**		**Best Choice**		**Achievement**		**Best Choice**	
Training	*mean*	σ	*mean*	σ	*mean*	σ	*mean*	σ
1	96.50 %	3.93 %	74.76 %	22.74 %	97.13 %	2.35 %	73.52 %	16.67 %
10	97.29 %	3.21 %	80.85 %	20.83 %	97.23 %	2.36 %	73.94 %	16.89 %
20	97.37 %	3.18 %	79.90 %	22.51 %	97.38 %	2.29 %	74.74 %	16.87 %
40	97.70 %	2.97 %	81.31 %	25.42 %	97.93 %	1.66 %	78.86 %	14.50 %
60	98.32 %	2.53 %	89.72 %	12.70 %	98.16 %	1.63 %	81.89 %	13.20 %
120	97.45 %	4.36 %	89.79 %	4.25 %	97.48 %	2.00 %	72.97 %	13.87 %

TABLE X. SLIDING WINDOW SCENARIO WITH DIFFERENT PREDICTION WINDOWS [38]

	DecisionStump				*FIMT-DD*			
Win. Size	**Achievement**		**Best Choice**		**Achievement**		**Best Choice**	
Prediction	*mean*	σ	*mean*	σ	*mean*	σ	*mean*	σ
1	97.86 %	4.81 %	85.61 %	17.27 %	97.48 %	3.55 %	75.76 %	21.43 %
7	97.42 %	3.52 %	83.48 %	17.33 %	97.54 %	1.95 %	75.87 %	16.27 %
14	97.31 %	2.91 %	82.06 %	18.18 %	97.56 %	1.53 %	75.87 %	12.92 %
28	97.16 %	2.22 %	79.73 %	19.52 %	97.63 %	1.16 %	76.44 %	10.71 %



(a) Training Window Sizes (b) Prediction Window Sizes

Figure 7. Best Choice Means of Different Window Sizes With Measurement Data. [38]

learning strategies and evaluation based on a longer period of measurement data as well as to equalize the results regardless any specific splitting point using the introduced sliding iterations. Tables VIII and IX focus on the evaluation of the training strategies. Table VIII shows the results of the predictions using a prediction window of 1 and 28 days and an incremental learning strategy. Tables IX and X present the results using sliding training windows. Table IX reveals the results analyzing the optimal window size for the training of the model, while Table X focuses on the optimal prediction window size. In contrast to the first analysis, in which we used a relative indicator for making comparisons to another learning approach (random selection), we used an absolute indicator to make the learning approach comparable among each other. The absolute indicator is the *Overall Optimization Achievement*. The best choice indicator remained the same. Comparing the learning strategies in the tables, the overall optimization achievement is more or less similar in all cases. This is due to the fact, that the mean response time value of the worst service is much higher than for the other services. Hence, predicting any service that is not the worst service already achieves quite a high absolute optimization achievement. Comparing on a relative basis, such as in the first analysis, results in easier distinguishable figures in such a scenario. In reality, this indicator is still significant for the general evaluation of an approach. If we directly compare the major two figures between the similar scenarios in the first and second analysis in Table VII and a prediction window of 28 days in Table VIII, the FIMT-DD-based approach could also achieve higher optimization improvements, while the best choice prediction strength of the classification-based approach could not keep its advance in a long term and iteration equalized analysis (second analysis; cf. Table VIII). However, if the prediction window size is one day, the best choice strength is still determinable. In general, a shorter prediction window size is better than a bigger one. While the classification approach gets worse in its predictions, regression-based FIMT-DD remains strong and even gets slightly better on average (illustrated in Figure 7(b)). We assume that the drift detection of this method is responsible for that. As for the training size, for both methods, the prediction results get better the bigger the training window size, up to a size of 60 days (illustrated in Figure 7(a)).

2) Simulation Data: Based on the measurement data, the generated data of the first analysis [3] follows a normal distribution with the same statistical means, standard deviations (both of the main base intervals), and with the same outliers of the services from the measurement data. Starting from the initial vertical level, we approximated step by step the mean values to a target value of 2,000 ms (approximation degrees are represented between 0, no approximation, and 1, full approximation). Within each approximation step, the presented results are mean values of an iteration of ten re-conductions. Tables XI

TABLE XI. ANALYSIS RESULTS OF CLASSIFICATION USING NORM. DISTR. GENERATED DATA WITH PERIODICITY (MEAN VALUES) [3]

Target Mean Value in ms	Approximation Degree	RT Gain in % Prediction/Random	RT Gain in % Best/Random	Overall RT Gain Achievement in %	RT Ratio in % Prediction/Best	RT Ratio in % Random/Best	Best Choice in % Prediction	Best Choice in % Random
2,000	0	69.21	74.89	92.40	122.64	398.33	76.60	25.34
	0.5	40.82	48.57	84.04	115.07	194.47	55.71	26.18
	0.75	29.03	38.13	76.13	114.71	161.64	43.73	23.95
	0.875	22.94	32.31	71.00	113.84	147.74	35.93	25.06
	0.9375	13.75	30.99	44.36	124.98	144.91	34.54	25.06
	1	-5.47	28.36	-19.31	147.23	139.59	32.03	26.18

TABLE XII. ANALYSIS RESULTS OF CLASSIFICATION USING NORMALLY DISTRIBUTED GENERATED DATA (MEAN VALUES) [3]

Target Mean Value in ms	Approximation Degree	RT Gain in % Prediction/Random	RT Gain in % Best/Random	Overall RT Gain Achievement in %	RT Ratio in % Prediction/Best	RT Ratio in % Random/Best	Best Choice in % Prediction	Best Choice in % Random
2,000	0	67.35	76.42	88.12	138.51	424.23	66.01	25.34
	0.5	39.17	51.28	76.38	124.85	205.25	43.73	25.06
	0.75	29.86	43.31	68.95	123.71	176.40	34.54	24.51
	0.875	23.91	37.94	63.03	122.59	161.14	35.93	24.79
	0.9375	13.53	36.98	36.59	137.21	158.69	30.36	23.95
	1	9.33	36.09	25.85	141.87	156.47	28.69	24.79

TABLE XIII. ANALYSIS RESULTS OF REGRESSION USING NORM. DISTR. GENERATED DATA WITH PERIODICITY (MEAN VALUES) [3]

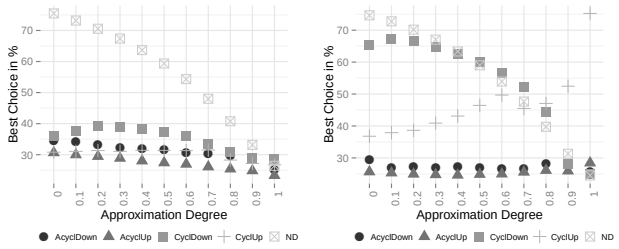
Target Mean Value in ms	Approximation Degree	RT Gain in % Prediction/Random	RT Gain in % Best/Random	Overall RT Gain Achievement in %	RT Ratio in % Prediction/Best	RT Ratio in % Random/Best	Best Choice in % Prediction	Best Choice in % Random
2,000	0	50.83	74.10	68.59	189.85	386.12	38.99	24.51
	0.5	6.98	47.89	14.59	178.49	191.90	27.57	25.06
	0.75	6.28	37.53	16.74	150.03	160.09	27.85	24.51
	0.875	8.75	33.38	26.22	136.96	150.11	28.96	25.34
	0.9375	5.43	30.83	17.61	136.73	144.59	29.52	23.95
	1	3.24	28.12	11.52	134.62	139.13	25.90	25.06

TABLE XIV. ANALYSIS RESULTS OF REGRESSION USING NORMALLY DISTRIBUTED GENERATED DATA (MEAN VALUES) [3]

Target Mean Value in ms	Approximation Degree	RT Gain in % Prediction/Random	RT Gain in % Best/Random	Overall RT Gain Achievement in %	RT Ratio in % Prediction/Best	RT Ratio in % Random/Best	Best Choice in % Prediction	Best Choice in % Random
2,000	0	52.14	76.30	68.34	201.91	421.96	33.14	24.79
	0.5	23.12	51.81	44.63	159.54	207.54	26.46	24.23
	0.75	15.76	44.98	35.04	153.12	181.78	29.52	23.95
	0.875	8.33	39.78	20.94	152.23	166.08	28.13	24.23
	0.9375	0.86	37.70	2.29	159.14	160.53	24.51	25.06
	1	-3.40	35.43	-9.59	160.15	154.88	24.51	25.62

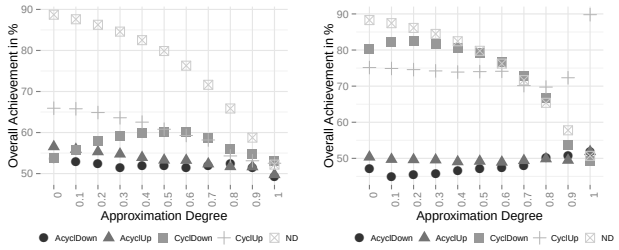
and XII present the data results of the classification-based approach, with and without periodicity on natural time periods. Tables XIII and XIV list them respectively for the regression-based approach. For both machine learning approaches, our assumptions were confirmed. The higher the approximation degree, the less the best choice prediction and the overall RT achievements. Within the approximation, the classification approach achieved now higher RT gain achievements up to a degree value of 0.9375, while within regression the figures got much worse already at a degree of 0.5. For both approaches, since the differences in the response time values between the services also decreases, the gain margin reduces and, therefore, the benefit of recommendation also decreases. Using regression in the periodic/random case, the RT gain is not much better than a random selection already at that degree. In the random-only case, it is much better. Within the generated data in contrast to the measurement data, regression lacks now behind classification. The simulated periodicity was apparently not as much represented in the measurement data than expected. Furthermore, the FIMT-DD does not cope with a sinus-based (natural) periodicity. The classification-based approach could achieve better results due to the in the pre-processing focused natural periods. Nonetheless, this required further analysis. Therefore, in the second analysis [38], a focus was set on each presumed aspect (cf. the introduced simulation profiles in the previous section) individually in order to analyze the strengths and weaknesses within each aspect, but also in order to find out, which aspects can be reflected in the actual measurement data. Like the measurement data in the second analysis, the results are mean values of the sliding point iteration. Similar to the approximation of the simulation data in the first analysis, an approximation was conducted on a 10 % step basis until they were identical (disregarding some random noise). The results of each approximation step reveals how good the learning approaches and scenarios can cope with the challenge that the respective scenario focused on. Figure 8 depicts the best choice results for each machine learning approach. Figure 9 shows the correspondent overall achievement figures. Since the achievement is defined relatively to the best and worst service performances and since these performances are approximated step by step, there is not much difference between both figures with their different accuracy criteria “best choice” and “overall achievement”, resp.; especially, when the approximation approaches 100%. For the cyclic and acyclic profiles, the non-best services perform equally since there is no vertical shift. Hence, the overall achievement depends only on whether finding the best choice or not. Therefore, for these profiles, there is not much difference between the best choice and the overall achievement indicators.

Before we focus on the differences between both learning approaches, we compare the differences between the different scenarios. Both approaches cope well with the normal distribution scenario. This is the only scenario approximating a vertical shift (response time mean), and both methods and their approaches get worse when the response time means are approximated. All other scenarios approximate a horizontal shift. That means that their normal distribution component is and remains similar. They only distinguish themselves in their performance spikes (response time up for worse performance; response time down for improvements). In the acyclic spike scenarios, both approaches are not able to cope with these



(a) DecisionStump (b) FIMT-DD

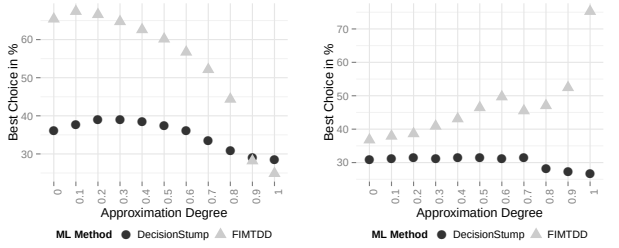
Figure 8. Best Choice Mean Using Sliding Windows Within the Scenarios. [38]



(a) DecisionStump (b) FIMT-DD

Figure 9. Overall Achievement Mean Using Sliding Windows Within the Scenarios. [38]

spikes. No matter whether the spikes go up or down, both approaches remain on a best choice rate of around 25 %, which is not much better than random selection in the first analysis. However, the DecisionStump approach achieves slightly better results. Comparing the remaining cyclic scenarios, the FIMT-DD can show its strengths (see CyclDown and CyclUp in Figures 8(b) and 10). Compared to the classification-based approach illustrated in Figure 8(a), the FIMT-DD achieves much higher best choice (and overall achievement) figures. The profiles of each service are taken into account, while this information is lost using the classification approach, which is illustrated in the charts in Figures 8 and 9. One of our question, whether it makes any difference if the spikes go up (a service gets suddenly worse) or the spikes go down (a service gets suddenly better), could be answered. According to



(a) Cyclic Spikes Down (b) Cyclic Spikes Up

Figure 10. Best Choice Comparison Between Both Learning Methods in the Cyclic Down/Up Scenario.

the results, illustrated in the figures, it does make a difference whether the spikes go up or down. FIMT-DD is in both cases significantly better. However, services getting suddenly worse among similar services (spikes down) can be learned better than the other way around. It seems to be easier to learn an outstanding service whereas it seems to be more difficult to recognize a service getting worse within the optimization focus of similar well performing services. Having a closer look at the cyclic up illustration in Figures 8(b) and 9(b), the indicator values get better with a higher approximation degree. This seems to be odd. One explanation could be that the regression-based approach focuses on the prediction of the performance behavior of each service as a pre-step for the actual best service determination, while the classification-based approach only focuses on the direct learning of the best fit service. However, the spikes up scenario simulates the opposite. Furthermore, considering the generation of the cyclic down and up scenario, their profiles are inverted on a higher level. The differences between the results in the figures also appear to be inverted. Still, the results for this scenario require further analysis, since a total approximation of this profile and its normal distribution part should develop similarly to the fully approximated normal distribution scenario [38].

D. Validation of the More Suitable Approach With Machine Learning Methods in the Framework

As a result of the evaluation of the classification-based vs. regression-based approach, both approaches can be employed in general. However, the regression-based approach revealed several benefits. Therefore, for the evaluation within our framework, we focused the regression-based approach, using machine learning methods for the prediction of the performance of services. The benefit of this approach is that the estimated NFPs within a certain moment (call context) can then be used for the calculation of the utility value for different preferences. Furthermore, with this approach, a ranking between the service candidates can be conducted since it is more important to gain a higher utility than to achieve a high best-fit accuracy. With the selected regression-based approach in general, machine learning methods had to be evaluated for their NFP prediction within the overall knowledge retrieval and recommendation process. As a result, the evaluation of the recommendation process with the employment of the selected learning approach together with at least one appropriate machine learning method is also a proof of concept of the overall framework.

1) Validation Scenario: With the learning focus set on the prediction of NFP values, we implemented three machine learning methods in our framework for further evaluation. In contrast to the first evaluation, we also changed the training and evaluation phase to the recommendation scenario in reality. Like in reality, we used a continuous learning strategy.

After initial, various pre-tests, we selected besides FIMT-DD [43] also Naïve Bayes and Hoeffding Tree [47] for the implementation and validation within our framework [1]. The Naïve Bayes classifier is a simple probabilistic classifier based on Bayesian statistics (Bayes' theorem) with strong independence assumptions [48]. The Hoeffding tree or Very Fast Decision Tree (VFDT) is an incremental, anytime decision tree induction algorithm that is capable of learning from massive data streams, assuming that the distribution generating examples do not change over time. It exploits the fact that

TABLE XV. EVALUATION RESULTS OF THE MACHINE LEARNING METHODS NAÏVE BAYES, HOEFFDING TREE, AND FIMT-DD WITHIN THE OPTIMIZED SERVICE SELECTION/RECOMMENDATION [1]

	Naïve Bayes	Hoeffding Tree	FIMT-DD
TOP1 Accuracy (in %)	58.634	59.837	69.287
TOP2 Accuracy (in %)	90.163	90.421	93.471
Mean Absolute Error (Utility)	1.656	1.660	1.049
Recommend. Table Updates	659	647	1.189

a small sample can often be enough to choose an optimal splitting attribute. This idea is supported mathematically by the Hoeffding bound, which quantifies the number of observations needed to estimate some statistics within a prescribed precision [47][49]. The FIMT-DD, which focuses on time-changing data streams with explicit drift detection [43], was again used because of its focus on drift detection. With respect to the requirements in Table IV, these methods were chosen due to their outlined characteristics (simplicity vs. incremental, anytime decision with capability of massive data streams vs. drift detection).

For the evaluation of the learning methods, the actual best-fit service instance has to be known at each call context (location, weekday, time, etc.) with each utility function. This is a challenge when it comes to a real-world validation. In reality, service calls over the Internet cannot be repeated under the exactly identical conditions as the various kinds of networks and infrastructures build complex systems with variable behavior. At a certain, unique moment, the load of a service instance's system environment and the network load or any incident are combinations of coincidences, which cannot be repeated. However, such aspects have an impact on the experienced NFPs at consumer side. For a strict validation, all service calls that are supposed to be compared had to be made at the same, identical call context, which is practically infeasible, especially when there are several competitive service instances [1].

Such a situation can only be derived within a simulation scenario, where the characteristics of NFP behavior are known for evaluation. In order to achieve such a scenario, where the validation process retrieves exactly the best-fit service instance for validation at each moment considering call context and utility function, we developed a simulator that creates NFP measurements for services based on pre-defined behavior profiles within a period. The implementation of this simulator follows a periodic behavior influenced by statistical random-based deviation. Similar to the previous simulation, the periodic behavior of the simulated Web services is based on our initial measurements and considered day/night time, weeks, month, work day, and weekend aspects. The random-based deviation is supposed to simulate unexpected incidences such as network traffic jams, high/low usage of a service's limited infrastructure [1].

For this evaluation, we focus on the machine learning within the overall recommendation process. Recall, within the regression-based approach, which we preferred for the outlined reasons, the machine learning methods are used for the prediction of the NFP values as the input for the calculation of the utility value. The amount of NFP inputs have no impact on the machine learning steps. Therefore, based on the results in Section IV, we selected the top-2 of the relevant NFPs: response time and availability. A further advantageous aspect for their use is that their measurement scales are different and both can be measured, hence, they are variable. Furthermore, both of their measurement scales are all also used by the actual top-6 relevant NFPs: ratio scale for response time, cost, price, and throughput; nominal scale for availability and reliability. So, by their usage for the evaluation and validation, all possible measurement scales for the top-6 relevant NFPs are included in the analysis. In the appendix, Figure 18 illustrates the characteristics of the simulated response times within the whole period. Note that the line is only a visual orientation to depict the concept drift of each service instance. For the recommendation, the actual best-fit service instance at each time is important and not the averaged value of each service instance. Figure 19 (appendix) depicts the statistical value of availability with a focus on weekday and daytime periods.

2) Results: The evaluation results are based on the process described in Section III and Figure 6 (upper process), in which learning is only used for the prediction of the NFPs of a service. The learning of the expected NFPs is based on incremental, continuous learning with each evaluated learning method. The calculation of the individual utility values and the best-fit service determination are done in intervals and updated in the foreground model. Listed in Table XV, the results of the FIMT-DD achieved around 70 % of correct predictions on average. Note that the calculated utility ranges from 0–100. TOP 1 accuracy is the prediction accuracy of the actual best-fit service, while TOP 2 accuracy is fulfilled when predicting the best-fit or second-best-fit service.

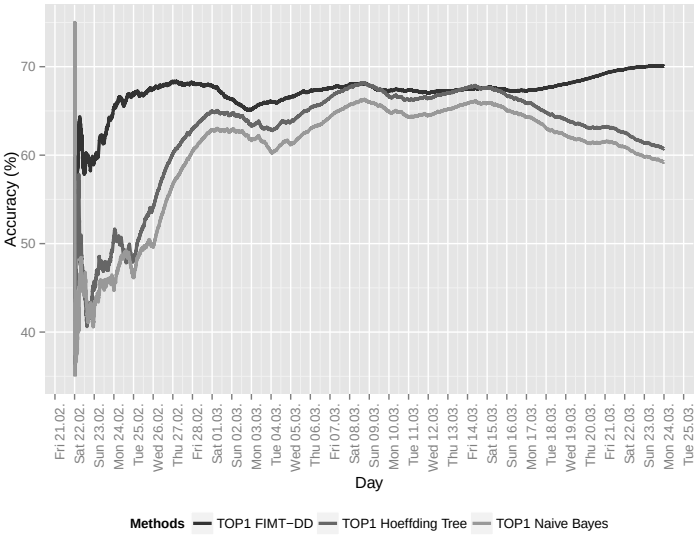


Figure 11. Service Recommendation Accuracy of the FIMT-DD, Hoeffding Tree, and Naïve Bayes Algorithm in the Course of Time. [1]

TABLE XVI. UTILITY GAIN WITH SERVICE RECOMMENDATION USING THE FIMT-DD ALGORITHM IN COMPARISON [1]

After selecting ...	Average experienced utility value	FIMT-DD comparison in percent
the FIMT-DD recommended instance	86.79	100.0 %
the perpetual best instance at each time	91.86	94.5 %
the perpetual worst instance at each time	29.22	297.0 %
the statistically best instance statically	81.96	105.9 %
an instance randomly	64.08	135.4 %

Comparing all methods, there is not much difference between Naïve Bayes and Hoeffding Tree. The FIMT-DD shows very good results. It has the highest update rate of the foreground table, which is an indication that it reacts quicker and more fine-grained on change than the other methods [1]. The cold start problem applies to service recommendation. However, good recommendation results are also supposed to be achieved with a small set of records at the beginning. Comparing the graphs in Figure 11, we can see that for the TOP 1 indicator in the overall recommendation process, the FIMT-DD quickly achieves a high accuracy in the recommendation of the best-fit service. The drift detection of the FIMT-DD seems to work at the end of the period, where some service instances change their performance behavior (see Figure 18) [1].

Figure 12 reveals more insight in the accuracy measure. The figure shows the degree of accuracy of the utility prediction. Once again, the best-fit service is the one with the highest utility value regarding a service consumer's individual preferences, which are expressed in a utility function. When comparing the prediction quality of machine learning methods within our framework, the accuracy of the NFP prediction is relevant. Since the utility value is calculated on that basis, a method is better, the closer the utility value based on the prediction is to the one based on the actual NFPs. Comparing the method's graphs, we see that for Naïve Bayes and Hoeffding Tree the predicted utility values at each time are both quite similar and do not reflect the curves of the actual values. In contrast, the graph of FIMT-DD depicts that the prediction is very close to the actual values. The intercepts of the curves show, that FIMT-DD does cope with change rapidly. In all cases, intercepts – which denote a change in the best-fit ranking – are also reflected in the prediction quite accurately [1].

For the evaluation of service recommendation in general, the actual utility gain is an important measure. Since the selection of service instances are based on several NFPs, the utility value as a basis for the individual preferences is an appropriate measure to benchmark service recommendation. In Table XVI, the average experienced utility value after the service recommendation based on the FIMT-DD algorithm is compared with other scenarios. The table reveals good results. As written above, within this evaluation scenario, the overall best and worst services can be determined at each time. Once again, such comparisons are only possible within such a scenario; this is not possible in reality. Comparing the figures, we see that the FIMT-DD-based recommendation is able to achieve 94.5 % of the maximum achievable utility value. It is 35.4 % better than a random selection approach and even 5.9 % better than the statistically best service instance when statically using it. Note, that choosing the statistically best service instance is also a kind of learning [1].

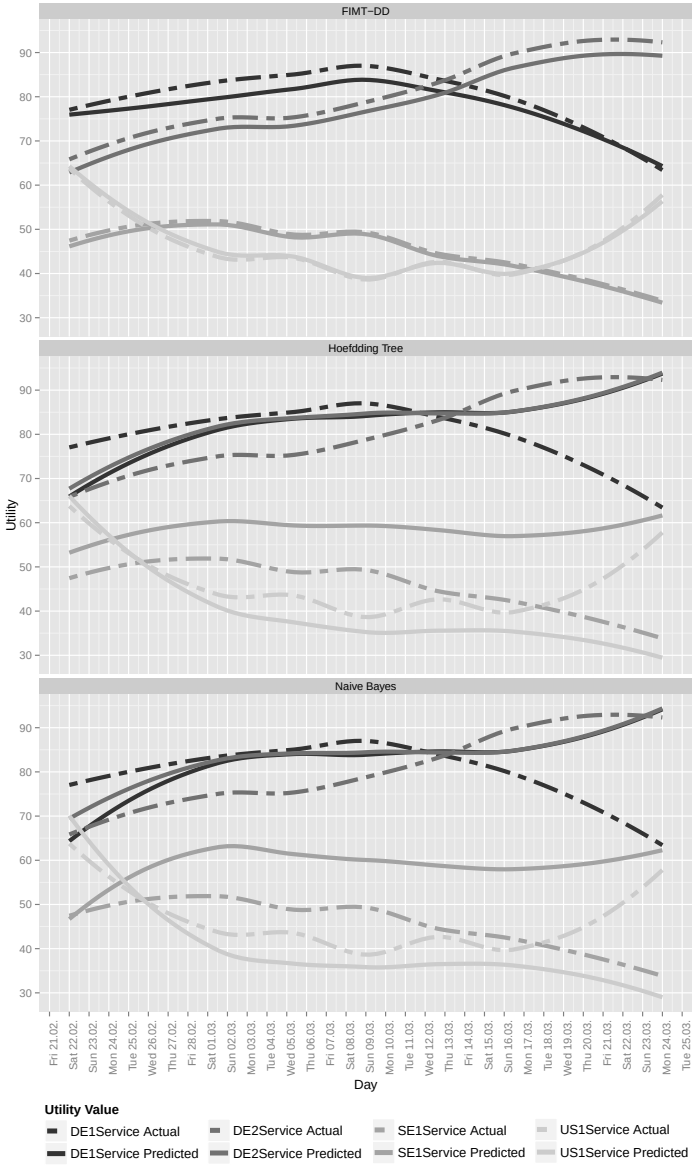


Figure 12. Detailed Overview About the Predicted and Actual Utility Values. [1]

VII. CONCLUSION

With the introduction of our recommendation framework, we aim at the optimization of consumer experienced performance of Web services at consumer side. Thereby, our framework considers the preferences and call contexts. It uses the shared knowledge of previous service calls of other, similar consumers in order to optimize the benefit of (other) service

consumers. Furthermore, the framework does not require any changes in existing implementations or systems and does not interfere with the encapsulation concept of the distributed world.

Besides the introduction of the framework, we conducted an analysis of relevant NFPs when selecting a service among functionally equivalent candidates. The results of this survey were based on the scientific conference contributions of the last ten years. Therefore, the results reflect the condensed opinion of the scientific community in this research area about what are the relevant NFPs for service selection and, hence, for the recommendation of services. When retrieving this information, we furthermore analyzed whether the NFP candidates had been just mentioned, theoretically discussed, or even validated in a practical scenario.

Within our framework, we comprehensively evaluated the employment of machine learning methods. Initially, we analyzed the two machine learning approaches classification and regression, which aim at different aspects within the overall recommendation process. In our real-world measurement data and simulated data evaluation scenarios, we found out that both approaches can be employed for service recommendation. However, both approaches have benefits and drawbacks.

For the implementation within our framework, we employed machine learning approaches and methods for the prediction of the NFPs of services within a certain call context. The actual determination of the best-fit service is then calculated, based on the predicted NFPs. This approach has several benefits such as a ranked determination of best-fit services and the easy determination of best-fit services for new preferences for existing contexts. The first aspect aims at the fact that service consumers are interested in the increase of their individual utility. Therefore, it is not necessary to recommend always the best-fit service if the second-best-fit is determined to achieve an almost as high utility value. Since consumers' preferences can vary, the second aspect is important in order to reduce the cold-start problem.

Based on the outcome of the NFP analysis, we validated the employment of machine learning methods within our framework. Employing the FIMT-DD algorithm, we could achieve up to 95 % of the overall achievable utility gain using our framework. Due to architectural- and method-based incremental learning and knowledge extraction, the strengths of this algorithm regarding drift detection could prove its capability towards perpetual change within the Internet as an anonymous service market. On the basis of our analysis of relevant NFPs within service selection, response or service time is the most important optimization aspect. However, service recommendation is time-consuming. Therefore, we optimized our framework regarding time aspects. With an architectural optimized model for speed, we reduced or even avoid the cost of service recommendation.

Summarizing all outlined aspects, our framework together with machine learning methods can be used for the optimization of service selection focusing on the benefit of service consumers addressing consumer-centric differences such as call contexts and preferences implemented in existing and prospective future real-world scenarios.

As future work, we currently analyze the application of our approach to a multi-tier scenario, comprising process structures of composite services on each level as well as compensations and transactions. In addition to our utility-value-driven achievement validation, a direct comparison of our recommendation approach with other recommendation techniques, such as CF, is desirable for future work.

Appendix I. Paper Survey for the Determination of Relevant NFPs

In addition to the brief digest in Section IV, this section provides the design and full results of the paper survey in order to determine a ranked list of relevant NFPs during service selection.

A. Design of the Survey

Our survey is mainly based on the "Preliminary Guidelines for Empirical Research in Software Engineering" [50][51]. With a few amendments from our side, due to context-specific conditions, the usage of these guidelines is supposed to ensure the quality of the survey and its analytical results. According to these guidelines, the clear and precise description of the objectives, the design including subjects and objects as well as the analytical process is described in this section.

1) Objectives: The main objective of this survey is to determine the relevant NFPs for service consumption in general and for service selection/recommendation in particular. We put a strong focus on whether or not approaches to optimized service selection based on NFPs are applicable in practice. We identified two issues helping to answer this main question.

First, what are the NFPs used in the approaches? This is interesting because of two reasons. NFPs could be statically known, like e. g., the security level of a service, or change dynamically based on input, e. g., the response time of a service. If the (majority of the) relevant NFPs is statically known, static service binding approaches are sufficient. Otherwise, dynamic service binding approaches are necessary. The latter are more complex since NFP prediction, service selection, and service binding are then runtime issues.

Second, is the approach discussed theoretically and validated in practice. This is interesting because ad-hoc approaches that are not discussed in theory might be not mature for a general application in practice and neither are theoretical approaches that have not been tested experimentally and practically.

2) Identification of the Population: For the conducted survey, we chose conference publications of up to ten years in the SOC context. Table XVII lists the conferences that we used with all their publications for the population of our study according to their length of existence and their size in terms of overall publications in descending order. The total amount of the processed publications was 4,407 conference papers. We believe that conferences contributions are a good basis, since a wide range of scientific approaches of a broad scientific community can be analyzed. Additionally to the scientific community, conference contributions from industry side enrich the results of the analysis with real-world applicability perspectives.

The selection of these conferences was based on our related work knowledge in this field for the past years and does not claim to include all relevant conferences. We consider papers of these conferences to be good candidates for finding answers

TABLE XVII. SELECTED CONFERENCES FOR THE POPULATION OF THE STUDY

Conference		Period
IEEE International Conference on Web Services	ICWS	2003 – 2012
International Conference on Service Oriented Computing	ICSOC	2003 – 2012
IEEE European Conference on Web Services	ECOWS	2003 – 2012
IEEE International Conference on Services Computing	SCC	2004 – 2012
ACM Conference on the Quality of Software Architectures	QoSA	2005 – 2012
IEEE Asia-Pacific Services Computing Conference	APSCC	2006 – 2012
IEEE World Congress on Services	SERVICES	2007 – 2012
European Conference on Service-Oriented and Cloud Computing	ESOCC	2012

to our main question and both sub-questions discussed before. Although we might have missed some relevant papers in this field, we believe that we obtained the condensed opinion of a broad scientific community.

3) Process by Which the Subjects and Objects are Selected: Our survey is based on SOC-related conference papers. For each paper, we assessed its general relevance with respect to service selection based on NFPs and the profoundness of which an approach of service selection was discussed and validated.

For the evaluation of the general relevance, each paper was classified in one of the five categories, which are listed in Table II. Category P-A to P-D are *relevant*-marked categories with graduated significance; category P-Z comprises non-relevant-marked papers.

Besides, the relevance of a paper, the profoundness of an approach to service selection is important. Is an NFP just mentioned, is it furthermore discussed in detail, or even validated in a practical or experimental context? We took account of how thoroughly an NFP is discussed and therefore how good the quality of the reference is. The referred NFPs within a paper were each classified according to the categories listed in Table I.

Since a completely manual analysis of all conference papers would have consumed too much time, we employed a more efficient two-stage approach. The first stage was an automated pre-classification of all papers into *relevant* and *non-relevant*. The second stage was a manual classification of the paper and the occurrence categories of all papers marked *relevant*.

a) Automatic Pre-classification: For the pre-classification of the papers, we used keyword extraction methods of computational linguistics [52]. The idea was to find a top-k hit list with keywords that are highly represented only in *relevant* conference papers. Such a hit list could then be used for the automatic pre-classification of all conference papers within our survey.

When processing natural languages, there are some aspects that needed to be taken into account such as letter cases, lemmatization, acronyms, but also typographic challenges such as ligatures, and the extractions of stop words and references. Therefore, a certain pre-processing was necessary to bring the content of each conference paper into a normalized form for further processing. In order to achieve a top-k hit list of relevant keywords that are salient in the majority of *relevant* conference papers, not only single keywords but also compound keywords needed to be considered. Compound (key)words can be represented in several consecutive words. The consideration of consecutive words as a compound unit is called n-grams where n is the amount of consecutive words. For the automatic pre-classification, we analyzed n-grams up to a level of three (uni-, bi-, and trigrams).

We constructed a *gold set* of 202 randomly selected and *manually* pre-classified papers. Papers of categories P-A, P-B, P-C, and P-D were considered *relevant*. Other papers were considered *non-relevant*. Note that P-D papers do not have the main focus on the targeted topic and only discuss it in parts. Therefore, they are of less importance during the extraction of relevant keywords, since the distinguishable relevant keywords are not as highly represented as in papers that have a main focus on the targeted topics. 40 papers were manually marked *relevant* and 162 *non-relevant*. This gold set was then used for learning and validation of the pre-classification approach.

Within the gold set, each conference is almost equally represented with 4.5 % on average of its overall publications. At the same time, since the conferences have diverse amounts of publications, their share in the population and in the gold set is accordingly. For the relevant-marked papers of the gold set, keyword extraction provided us with a list of relevant keywords (uni-, bi-, and trigrams). The keyword extraction was based on the Lucene [53] and the Stanford CoreNLP [54] frameworks. The keyword extraction is described in details with pre- and post-processing in Algorithm 1.

Algorithm 1 Keyword Extraction Algorithm

```
Remove URLs
Remove references
Substitute acronyms by their full phrases
Substitute upper cases by lower cases
Extract 1-grams, 2-grams, and 3-grams and their frequencies using Lucene's ShingleFilter; no n-grams starting or ending with stop words; Lemma Form of each Token using the Stanford CoreNLP framework.
Pick the 27 most frequent 1-grams, 2-grams, and 3-grams.
for all n ∈ {1, 2} do
  for all m ∈ {n + 1, ..., 3} do
    for all kw ∈ n-gram keyword list do
      if ∃kw′ ∈ m-gram keyword list ∧kw ⊂ kw′ then
        frequency(kw) := frequency(kw) − frequency(kw′)
      end if
    end for
  end for
end for
Merge 1-grams, 2-grams, and 3-grams lists
Sort merged list descendingly in updated frequencies
```

We applied the same algorithm to the non-relevant-marked papers. Due to the small amount of relevant-marked papers in the gold set, we had to semantically revise and amend the

initial list in order to distinguish less distinctive keywords from highly relevant and distinctive ones. This left us with a top-11 hit list of relevant keywords. From a semantic aspect, the keywords are not equally important. Therefore, we manually added significance weights for each of the keywords in the list. The final keyword list with significance weights is listed in Table XVIII.

TABLE XVIII. FINAL LIST OF DISTINGUISHABLE KEYWORDS WITH THEIR CORRESPONDENT SIGNIFICANCE WEIGHT

Keyword	Weight
non-functional	0.8
response time	0.7
quality of service	0.68
service level agreements	0.68
composition	0.65
service selection	0.65
service consumer	0.4
service provider	0.4
monitor	0.3
request	0.3
resource	0.3

The automatic pre-classification of textual documents of a natural language raised a challenge (cf. [55]). Whether a document or parts of it are relevant or not depends on the determination of the actual meaning of the text. So, the relevance of a text cannot only be determined by a text's vocabulary but also its semantics in coherence with the grammatical structure. Hence, the demarcation between a *relevant* and a *non-relevant* paper is even for manual classification not an easy task due to the fact that relevance considered among all conference paper is a rather graduate characteristic.

Nonetheless, a manual conduction of all conference papers was infeasible. We tried classical classification approaches for pre-classification, but their achieved precision and recall indicators were not satisfying. The following approaches were tested and rejected: C4.5 for learning decision trees using the concept of information entropy [56], Naïve Bayesian classifier, a simple probabilistic classifier based on Bayesian statistics (Bayes' theorem) with strong independence assumptions, cf. [48], and the Multi-Nomial Bayes extension with a multinomial feature model where feature vectors represent the frequencies of features. The main problem with the classification methods was the challenge that all conference papers in this context are based on the same specific jargon.

Since the design of the second part of the study includes manual processing, we could neglect the semantic and grammatical structure of a paper. The main purpose of the automatic pre-classification was to reduce the amount of non-relevant conference papers, without sorting out relevant papers. Even for manual processing, keywords still remained a good foundation to determine relevant papers. However, we focused on their significance weight as well as their percental occurrence within a paper for which we computed a document score.

The document $score$ of each paper p was computed as follows. For each paper, we computed a *keyword list*(p) applying the keyword list extraction Algorithm 1. Each keyword frequency was normalized by computing its share of all keywords in percent. The $score$ of each paper p was then computed as the sum over the percentile occurrence (*pfreq*) of all its keywords

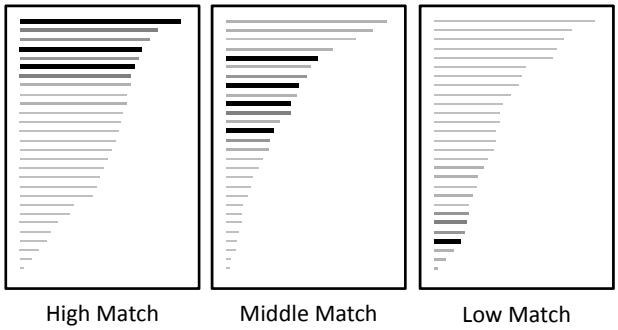


Figure 13. Descendingly Ordered Keyword (Uni-, Bi-, Trigrams) Occurrence Representations of Three Conference Papers With Different Keyword Matches.

multiplied with the manually defined significance (*sig*) of that keyword.

$$\text{score}(p) = \sum_{kw \in \text{keyword list}(p)} \text{pfreq}(p, kw) \times \text{sig}(kw)$$

Note that the significance of keywords that do not occur in the list of relevant keywords is zero.

Figure 13 illustrates the ordered top-k keyword lists of three conference papers. The bulks represent keywords, their length illustrates the percentile occurrence of the respective keyword, and their saturation the keyword's significance. Obviously, the paper on the left has a higher score than the one in the middle and the one on the right. I. e., the paper on the left matches the profile of a relevant paper with higher probability than the paper on the right.

In order to deterministically classify papers as relevant or not, we learned a threshold document score. For this, we used the gold set again. The threshold value was set to the mean of the average document scores of the 40 relevant papers and the 162 non-relevant papers.

b) Accuracy of the Automatic Pre-classification: For the evaluation of the quality of the pre-classification algorithm, we used *precision*, *recall*, and the F_β *measure* (cf. [57]) from the pattern recognition and information retrieval field with the following definitions:

$$\text{precision} = \frac{|\{\text{rel.-classified papers}\} \cap \{\text{rel. papers}\}|}{|\{\text{relevant-classified papers}\}|} \quad (4)$$

$$\text{recall} = \frac{|\{\text{rel.-classified papers}\} \cap \{\text{rel. papers}\}|}{|\{\text{relevant papers}\}|} \quad (5)$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \quad (6)$$

Precision scores close to 1 indicate that found papers are also relevant. *Recall* close to 1 indicates that all relevant papers are also found. The F_β *measure* is a geometric mean of *precision* and *recall* weighted with β. For our purpose, recall is more important, because the classification algorithm is supposed to sort out the non-relevant conference papers for the manual steps of the survey. It is hence less important

to have non-relevant papers remaining in the set for further processing, than sorting out too many of relevant conference papers. Therefore, recall was weighed much higher in the F_2 measure ($\beta = 2$).

With the above described gold set (4.584 % of population), we used two validation methods: *Leave One Out Cross-Validation* and *Two Fold Cross-Validation*. For each paper in the gold set, the former method learned the threshold on remaining papers and classifies the paper, while the latter repeatedly split the gold set randomly into two halves, learned the threshold on one half, and classified the papers in the other halves. Results are listed in Table XIX. With respect to the rather small amount of highly relevant conference papers in the gold set, we consider the figures of the *Leave One Out Cross-Validation* to be more accurate and still sufficient for further processing with manual classification.

c) Manual Classification of Relevant Conference Papers: From the previous automatic pre-classification of the complete population, the *relevant*-marked conference papers were the foundation for the manual processing. The main focus in this step is the semantic analysis of the papers in total and in parts. Especially, the determination of NFPs and the quality of their references (cf. Table I). But also the identification of the main focus of each conference paper (cf. Table II). For the manual analysis, we defined and conducted a strict algorithm in order to avoid bias and improve traceability of the survey.

The procedure described in Algorithm 2 was conducted as follows: For each conference paper pre-classified as *relevant* is scanned for NFPs. If an NFP is found, the quality of the occurrence is determined according to the categories listed in Table I. These categories O-A, O-B, and O-C are descendingly sorted according to their relevance. For each referred NFP, the highest category of all references for this specific NFP within the paper is chosen. Recall that occurrence categories apply individually to each referred NFP in a paper. This ensures that the reference quality of each NFP within a paper can be determined. Additionally to the occurrence category, the absolute amount of occurrences of each NFP is registered. The relative occurrence, which is the basis for later comparison and evaluation, can be calculated from the absolute occurrence. Furthermore, each manually processed conference paper is also classified according to the paper categories listed in Table II. If a paper does not contain any NFP according to the listed occurrence categories, the paper is in category P-Z. Each conference paper that mentions or discusses any NFP is at least in category P-D. However, if the abstract and the introduction of a paper focus on service selection it belongs to category P-A. If the focus is on the adaptation of composite services, it belongs to category P-B. And if the focus is on the consumption of services in general, its category is P-C. All

TABLE XIX. ACCURACY OF THE AUTOMATIC PRE-CLASSIFICATION

	Leave One Out Cross-Validation	Two Fold Cross-Validation
Precision	0.49	0.52
Recall	0.71	0.88
F_2 **measure**[1]	0.65	0.77

[1] $\beta = 2$

Algorithm 2 Manual Classification Algorithm

```
1: P = set of all conference papers automatically pre-classified relevant
2: A = ∅ // set of category P-A papers
3: B = ∅ // set of category P-B papers
4: C = ∅ // set of category P-C papers
5: Z = ∅ // set of category P-Z papers
6: N = ∅ // set of quadruples (p, i, x, c) of a paper (p), an NFP (i), its occurrence count (x), and the papers highest occurrence category (c)
7: for all p ∈ P do
8:   for all found NFP i in p do
9:     if i is validated in practical or experimental context then
10:      d ='a'
11:    else
12:      if i is mentioned and theoretically discussed then
13:        d ='b'
14:      else
15:        d ='c'
16:      end if
17:    end if
18:    if (p, i, ∗, ∗) ∈ N then
19:      N = N\(p, i, x, c)
20:      x = x + 1
21:      c = max(c, d) // recall that 'a' > 'b' > 'c'
22:    else
23:      x = 1
24:      c = d
25:    end if
26:    N = N ∪ (p, i, x, c)
27:  end for
28:  if abstract and introduction of p focus on service selection then
29:    A = A ∪ p
30:  else
31:    if abstract and introduction of p focus on adaptation of composite services then
32:      B = B ∪ p
33:    else
34:      if abstract and introduction of p focus on service consumption in general then
35:        C = C ∪ p
36:      else
37:        if p mentions NFPs in a service selection, adaptation, or consumption context, but without main focus then
38:          D = D ∪ p
39:        else
40:          Z = Z ∪ p
41:        end if
42:      end if
43:    end if
44:  end if
45: end for
```

remaining papers stay in category P-D.

4) Threats to Validity: Our survey is based on the scientific research contributions in the field of SOC of up to ten years. Therefore, the outcome does not rely on the work of a few scientists, but on the comprehensive outcome of the whole SOC research community. The results are based on quantifiable and qualifiable measures minimizing any bias. In order to achieve this, we designed a pre-classification that was fully automated and a (semi-)formal and specific manual classification procedure with a narrow interpretation scope.

However, recall is less than one. This reveals that there are probably relevant papers that are not pre-classified as such in the automated stage of our survey. Also, the population of papers is based on selected conferences that we considered to be relevant. It is possible that we missed some important papers this way. Still, since we focused on the determination of relevant NFPs based on the work of a broad scientific community, it is less important if we missed a few relevant papers.

B. Detailed Results of the Analysis

With the outcome of the automatic pre-classification, the population set for the manual analysis was reduced to 993 conference papers (3,414 conference were pre-classified to be non-relevant); with a presumed precision of 0.49 and a recall of 0.71. Within the manual classification, described in Algorithm 2, 297 conference papers were classified among the *relevant* categories; P-A: 104, P-B: 34, P-C: 46, and P-D: 113. Presenting the results of the analysis of our study, the NFPs during service selection as well as during adaptation and consumption from a consumer's perspective are analyzed according to occurrence and count, which were both introduced in Section IV.

Recall, an NFP with a high *occurrence* can be considered to be widely accepted to be relevant, since it is referred in many papers, while *count* indicates how much text is dedicated to an NFP in absolute terms (*absolute count*) or relatively in the papers (*relative count*). As described in Section IV, these two measures are not sufficient to deduct the quality of the references. In order to satisfy this aspect, we consider the quality of each NFP reference by its occurrence category, cf. Table I. Furthermore, we also differentiate between the papers regarding their semantic quality towards our finding objectives. Therefore, we categorize each paper according to its topic relevance (cf. Table II).

1) NFPs Referred in Conference Papers: First, we determined a ranked list of relevant NFPs in descending order due to the amount of papers in which they occur among all relevant categorized conference papers. Figure 14 lists the relevant NFPs according to their paper occurrence without any consideration of the quality of the references (occurrence categories) or focus of the paper (paper categories; except non-relevant papers, which are in P-Z). During the manual, semantical analysis, we discovered that in virtually all papers, time-relevant NFPs were used as synonyms for response time. Therefore, in Figure 14, *response time* encompasses several expressions that are used in a synonymous meaning. Without aggregation, the synonym NFPs would occur with the following amount within all relevant conference papers: *response time*: 75.1 %; *performance*: 22.2 %; *execution time*: 16.5 %; *latency*: 12.1 %; *duration*: 3.4 %; *timeliness*: 1.4 %; *delay*: 1.0 %. Note, since these NFP expressions are also used synonymously within a single paper, these percentages cannot simply be added up. Figure 14 represents the correct percentage of the amount of papers that mention or discuss response time synonyms.

With the knowledge of a ranked NFP list according to the amount of papers in which an NFP is referred, we now have a closer look at the textual distribution within all references. This indicates how much text is dedicated to what NFP in comparison to all other NFP references. As outlined above, we distinguish between absolute and relative count. We argue that an absolute count is more relevant if one is interested in the absolute amount of research documentation about an NFP, while the relative count is more relevant if all papers should have the same impact on the results. Although the list in Figure 4, which lists the relevant NFPs ranked regarding their occurrence counts, is similar to the paper occurrence list in Figure 14 in its ranking, there is a big gap between the counts of response time and the other NFPs. Response time is therefore dedicated more text than the other NFPs. At least, it is more often mentioned. The full list descendingly ordered according to the relative count is depicted in Figure 4. For most NFPs, there is not much difference in the ranking between absolute and relative count. However, for *reputation* and *trust*, there is a big gap between relative count and absolute count. The reason for this is related to the fact that trust and reputation is extensively discussed in some conference papers. With the main focus on these two related NFPs, some researchers argue from their point of view that these two NFPs had been fiercely neglected compared to other NFPs that are also mentioned frequently in these papers. When comparing absolute and relative count, we notice that for some researchers, trust and reputation is a very important aspect. However, for the majority, these NFPs are not as relevant as others.

2) Profoundness of the NFP References: With an overview about the NFPs being widely discussed and regarding their dedicated text, we now focus on the quality of the references. As described above, we consider the paper and occurrence categories in which an NFP is mentioned or discussed.

In Figures 4 and 14, many NFPs are still closely related to each other. In order to get a better overview, for the remaining figures, we grouped all NFPs into several categories, which all represent a certain aspect. Table III lists each NFP aspect category with its containing NFPs.

In Figure 15, the occurrences of each NFP category are related to each paper category (cf. Table II). We can see that a high share of the papers (35.5 % on average among the top-5 categories; 2.6 deviation) does not have its main focus on composition, adaptation, or selection of services and an (almost) equally high share of papers (37.3 % on top-5 average; 2.5 deviation) focuses on service selection. Note, the listed figures in the bars are percentage points.

Within the results of the counts related to each paper category, there is a big gap between relative count and absolute count among conference papers that have the main focus on service selection for *trust*/*reputation*. It affirms again that some researchers put a strong focus on *trust* and *reputation* as important NFPs for service selection.

In Figure 5, the occurrences of each NFP category are related to each occurrence category (cf. Table I). When a

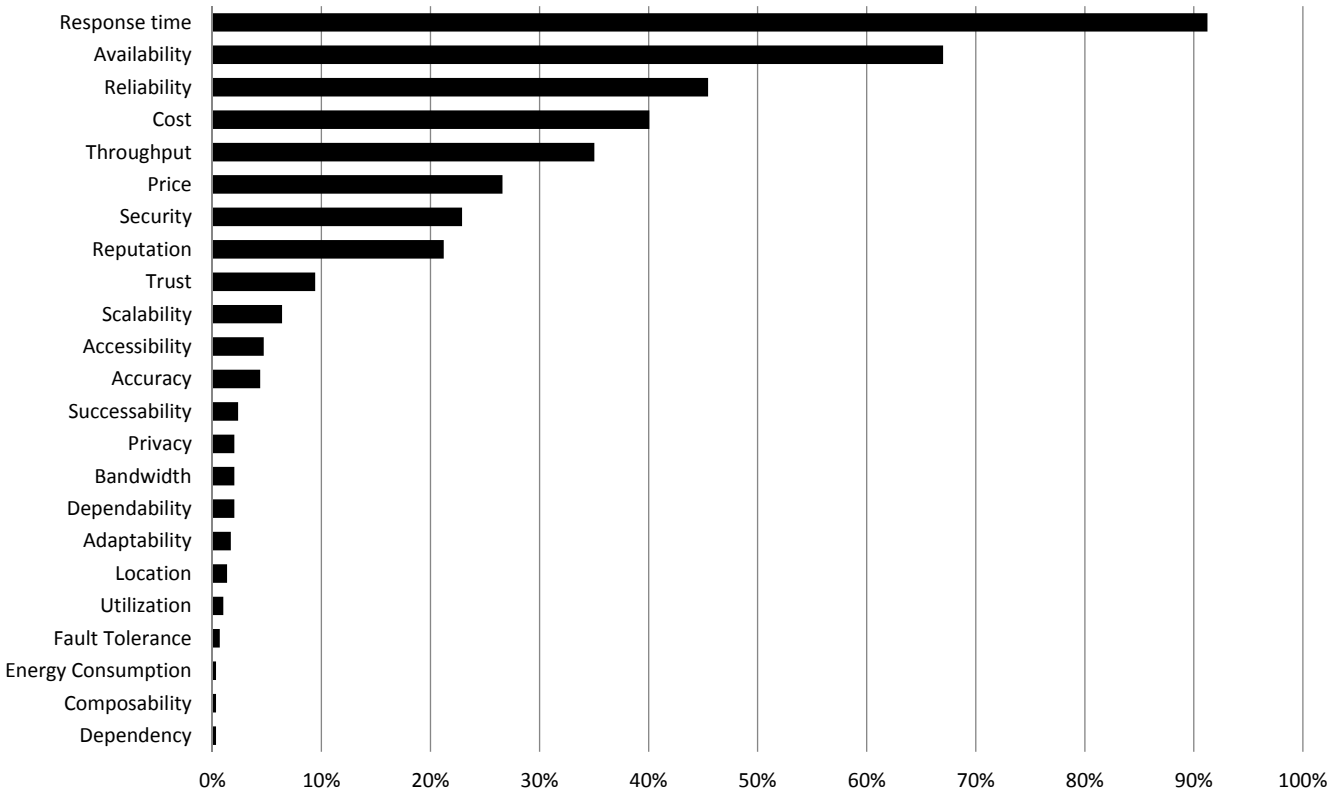


Figure 14. Paper Occurrence of Each NFP Within all Relevant Categories.

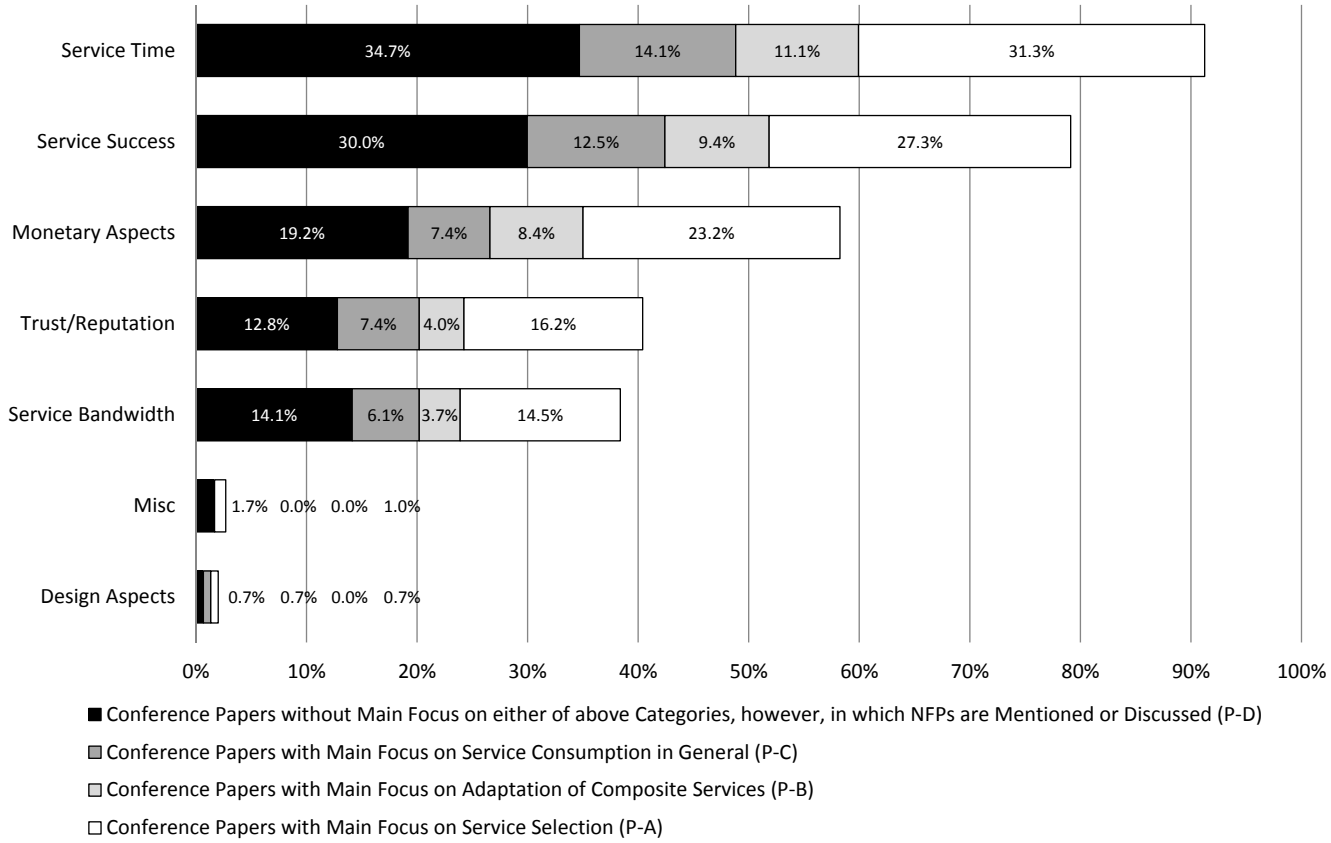


Figure 15. Paper Occurrence of Each NFP Category According to the Paper Categories.

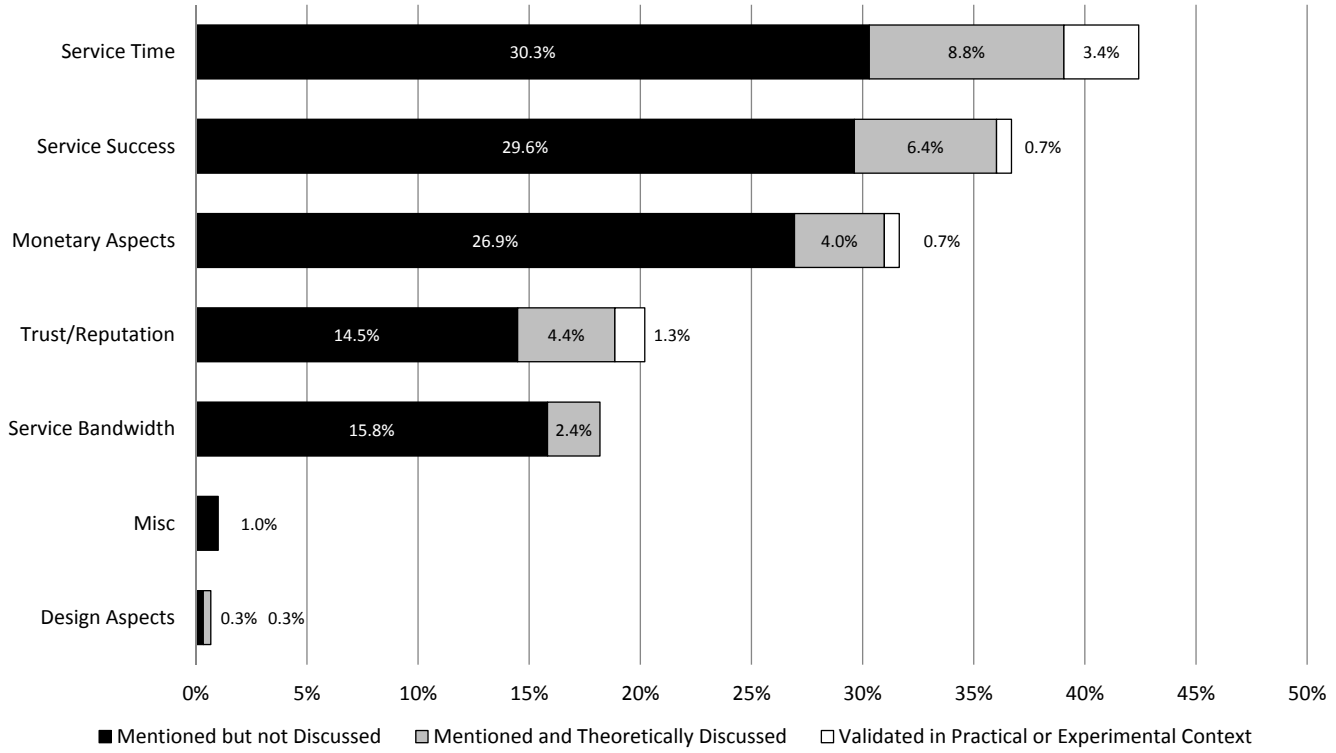


Figure 16. Paper Occurrence of Each NFP Category Within Highly Relevant Categories (P-A and P-B) According to Their Occurrence Category.

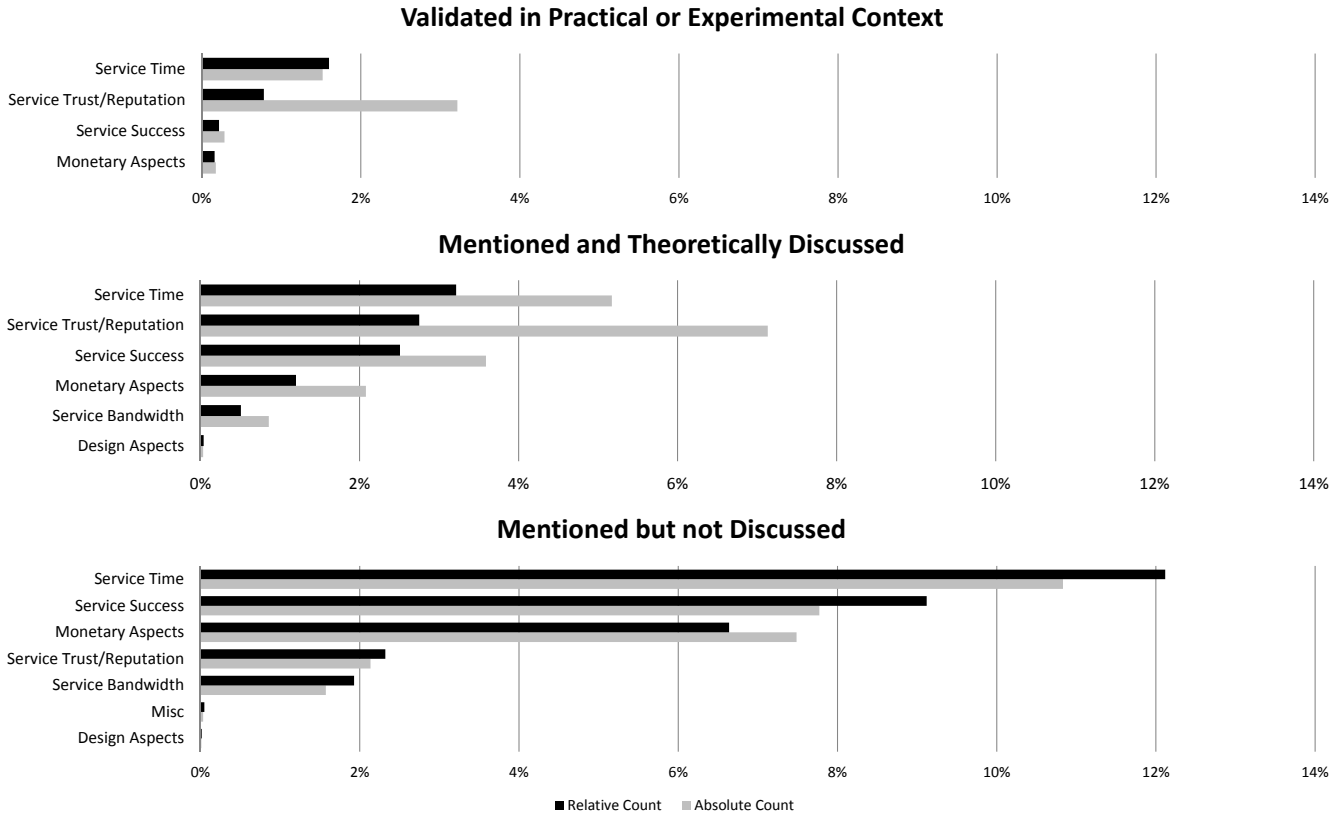


Figure 17. Overall Relative and Absolute Counts of NFPs Among Highly Relevant Categories (P-A and P-B) According to Each Occurrence Category.

paper has two or more NFPs that belong to the same NFP category with different occurrence categories, the paper is counted in the highest occurrence category. On average over all NFP categories (disregarding the *Misc* and *Design* categories), 83 % of all relevant papers only mention NFPs, but do not discuss them in detail. On average, 13 % discuss them in more detail while only 4 % also validate them in simulations or experiments. This means that a vast majority of relevant conference papers do not elaborate in detail on the NFPs they mention.

Finally, we set the focus on highly relevant conference papers (P-A and P-B). These conference papers have a main focus on service selection/reputation or service adaptation, which is very closely related to service selection. Figure 16 shows again the overall paper occurrence per occurrence category. The overall average shares among all NFP categories (without *Misc* and *Design aspects*) are 79 % for occurrence category O-C, 17 % for O-B and 4 % for O-A. There is only a minor difference to the mean figures of Figure 5. Still, the majority of conference papers only mention NFPs without further discussion or validation. Considering the overall counts within the highly relevant papers listed in Figure 17, the vast amount of references do not discuss an NFP in detail. Again, apart from service time, there is a big lack of validation.

Appendix II. Performance Profiles of Simulated Services for the Validation of Machine Learning Methods in the Framework

For the validation of machine learning methods within our framework, we simulated the performance behavior of four services. In order to challenge the methods, their profile change over time and are, in contrast to measurement data, not easy to distinguish:

To get a situation where the validation process retrieves exactly the best-fit service instance for validation at each moment considering call context and utility function, we developed a simulator that creates service instance measurements for a certain time period based on predefined behavior profiles. The implementation of this framework follows a periodic behavior influenced by statistical random-based deviation. The periodic behavior of the simulated Web services follows our initial measurements in [2] and considers: day/night time, weeks, months, work days and weekends. The random-based deviation is supposed to simulate unexpected incidences such as network traffic jams, high/low usage of a service's limited infrastructure. The random-based influence over a period was also evidenced in our real-world service tests [2]. For a multi-NFP service selection, two NFPs were simulated, which are response time and availability [1].

Figures 18 and 19 depict an overview about the simulated NFPs. The simulated validation dataset comprises a period of 30 days and has a total set of 460,800 records (40 records/hour × 24 hours/day × 30 day × 16 unique clients). The records contain information about day, time, response time in millisecond and availability (Boolean). Within the simulation, between each record there is a time interval of 90 seconds. Figure 18 shows in a condensed form the response time of all services instances within the whole period. Note that the line is only the trend. Within the recommendation process, the actual best-fit service instance at each time is important and not the averaged value of each service instance. The line is therefore only a visual orientation for us to determine the concept drift of each service instance within the period (e. g., DE2Service). Figure 19 shows the statistical value of availability with a focus on weekday and daytime periods [1].

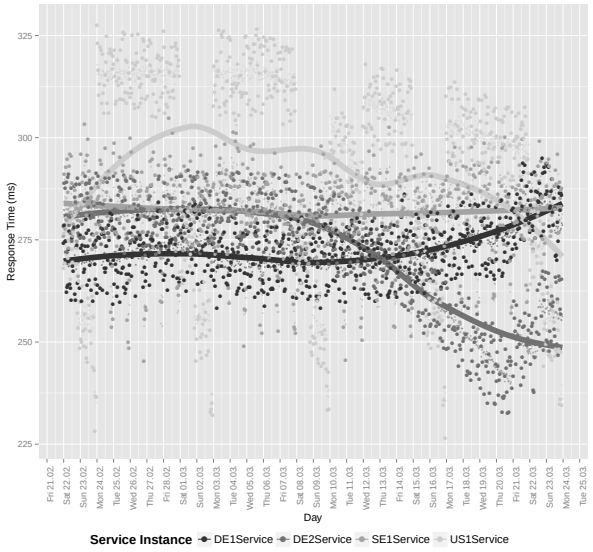


Figure 18. Overview About the Simulated Response Time of Four Service Instances and Their Trend Over the Whole Period. [1]

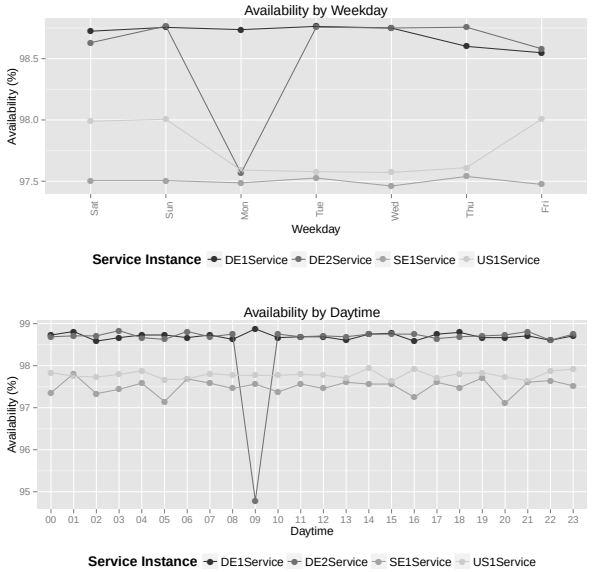


Figure 19. Overall Periodic Behavior Regarding the Availability of the Simulated Service Instances With Weekday and Daytime Aspects. [1]

References

[1] J. Kirchner, P. Karg, A. Heberle, and W. Löwe, "Appropriate machine learning methods for service recommendation based on measured consumer experiences within a service market," in The Seventh International Conferences on Advanced Service Computing (SERVICE COMPUTATION), 2015, pp. 41–48.

[2] J. Andersson, A. Heberle, J. Kirchner, and W. Löwe, "Service Level Achievements – Distributed knowledge for optimal service selection," in Ninth IEEE European Conference on Web Services (ECOWS), 2011, pp. 125–132.

[3] J. Kirchner, A. Heberle, and W. Löwe, "Classification vs. Regression – Machine learning approaches for service recommendation based on measured consumer experiences," in IEEE 11th World Congress on Services (SERVICES), 2015, pp. 278–285.

[4] J. Andersson, M. Ericsson, C. Kessler, and W. Löwe, "Profile-guided composition," in 7th International Symposium on Software Composition (SC), 2008, pp. 157–164.

[5] C. Kessler and W. Löwe, "Optimized composition of performance-aware parallel components," Concurrency and Computation: Practice and Experience, vol. 24, no. 5, 2012, pp. 481–498. [Online]. Available: http://dx.doi.org/10.1002/cpe.1844

[6] W. van den Heuvel and M. Smits, "Transformation of the software components and Web services market," in eMergence, Proceedings of the 20th Bled eConference, 2007, pp. 246–258.

[7] C. Legner, "Is there a market for Web services? – An analysis of Web services directories," in Proceedings of the 1st International Workshop on Web APIs and Services Mashups, 2007, pp. 29–42.

[8] L. Nilsson-Witell and A. Fundin, "Dynamics of service attributes: a test of Kano's theory of attractive quality," International Journal of Service Industry Management, vol. 16, no. 2, 2005, pp. 152–168.

[9] B. L. Duc et al., "Non-functional data collection for adaptive business processes and decision making," in MWSOC '09: Proceedings of the 4th International Workshop on Middleware for Service Oriented Computing. New York, NY, USA: ACM, 2009, pp. 7–12.

[10] L. Zeng et al., "Monitoring the QoS for Web Services," in Service-Oriented Computing – ICSOC 2007, 2008, pp. 132–144.

[11] L. Zeng et al., "QoS-aware middleware for Web services composition," IEEE Trans. Softw. Eng., vol. 30, no. 5, 2004, pp. 311–327.

[12] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Z. Sheng, "Quality driven Web services composition," in Proceedings of the 12th International Conference on World Wide Web. ACM, 2003, pp. 411–421.

[13] L. Li, J. Wei, and T. Huang, "High performance approach for multi-QoS constrained Web service selection," in Service-Oriented Computing – ICSOC 2007, 2008, pp. 283–294.

[14] S. Reiff-Marganiec, H. Yu, and M. Tilly, "Service selection based on non-functional properties," in Service-Oriented Computing - ICSOC 2007 Workshops, 2009, pp. 128–138.

[15] D. Mukherjee, P. Jalote, and M. G. Nanda, "Determining QoS of WS-BPEL compositions," in Service-Oriented Computing – ICSOC 2008, 2008, pp. 378–393.

[16] P. Leitner, B. Wetzstein, F. Rosenberg, A. Michlmayr, S. Dustdar, and F. Leymann, "Runtime prediction of service level agreement violations for composite services," in Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops, 2010, pp. 176–186.

[17] D. Robinson and G. Kotonya, "A runtime quality architecture for service-oriented systems," in Service-Oriented Computing – ICSOC 2008, 2008, pp. 468–482.

[18] R. Yang, Q. Chen, L. Qi, and W. Dou, "A QoS evaluation method for personalized service requests," in Web Information Systems and Mining, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6988, pp. 393–402.

[19] H. A. Müller, L. O'Brien, M. Klein, and B. Wood, "Autonomic computing," Software Engineering Institute, Carnegie Mellon University, Tech. Rep., 2006, ftp://ftp.sei.cmu.edu/public/documents/06.reports/pdf/06tn006.pdf; Retrieved: 25 January 2010.

[20] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing – degrees, models, and applications," ACM Comput. Surv., vol. 40, no. 3, 2008, pp. 1–28.

[21] E. Di Nitto, C. Ghezzi, A. Metzger, M. Papazoglou, and K. Pohl, "A journey to highly dynamic, self-adaptive service-based applications," Automated Software Engineering, vol. 15, no. 3-4, 2008, pp. 313–341.

[22] H. van der Schuur, S. Jansen, and S. Brinkkemper, "Becoming responsive to service usage and performance changes by applying service feedback metrics to software maintenance," in Automated Software Engineering – Workshops, 2008. ASE Workshops 2008. 23rd IEEE/ACM International Conference on, 2008, pp. 53–62.

[23] Z. Zheng, H. Ma, M. Lyu, and I. King, "QoS-aware Web service recommendation by collaborative filtering," Services Computing, IEEE Transactions on, vol. 4, no. 2, 2011, pp. 140–152.

[24] M. Tang, Y. Jiang, J. Liu, and X. Liu, "Location-aware collaborative filtering for QoS-based service recommendation," in Web Services (ICWS), IEEE 19th International Conference on, 2012, pp. 202–209.

[25] L. Kuang, Y. Xia, and Y. Mao, "Personalized services recommendation based on context-aware QoS prediction," in Web Services (ICWS), IEEE 19th International Conference on, 2012, pp. 400–406.

[26] G. Kang, J. Liu, M. Tang, X. Liu, B. Cao, and Y. Xu, "AWSR: Active Web service recommendation based on usage history," in Web Services (ICWS), IEEE 19th International Conference on, 2012, pp. 186–193.

[27] Q. Yu, "Decision tree learning from incomplete QoS to bootstrap service recommendation," in Web Services (ICWS), IEEE 19th International Conference on, 2012, pp. 194–201.

[28] T. Ahmed and A. Srivastava, "A data-centric and machine based approach towards fixing the cold start problem in web service recommendation," in Electrical, Electronics and Computer Science (SCEECS), IEEE Students' Conference on, 2014, pp. 1–6.

[29] Z. Zheng, H. Ma, M. R. Lyu, and I. King, "QoS-aware web service recommendation by collaborative filtering," IEEE Transactions on Services Computing, vol. 4, no. 2, 2011, pp. 140–152.

[30] J. Cao, Z. Wu, Y. Wang, and Y. Zhuang, "Hybrid collaborative filtering algorithm for bidirectional web service recommendation," Knowledge and Information Systems, vol. 36, no. 3, 2013, pp. 607–627.

[31] R. Nayak and C. Tong, "Applications of data mining in web services," in Web Information Systems – WISE. Springer Berlin Heidelberg, 2004, vol. 3306, pp. 199–205.

[32] J. Yao, W. Tan, S. Nepal, S. Chen, J. Zhang, D. De Roure, and C. Goble, "Reputationnet: A reputation engine to enhance servicemap by recommending trusted services," in Services Computing (SCC), IEEE Ninth International Conference on, 2012, pp. 454–461.

[33] J. Zhang, P. Votava, T. Lee, S. Adhikarla, I. Kulkumjon, M. Schlau, D. Natesan, and R. Nemani, "A technique of analyzing trust relationships to facilitate scientific service discovery and recommendation," in Services Computing (SCC), IEEE International Conference on, 2013, pp. 57–64.

[34] K. Huang, J. Yao, Y. Fan, W. Tan, S. Nepal, Y. Ni, and S. Chen, "Mirror, mirror, on the web, which is the most reputable service of them all?" in Service-Oriented Computing, 2013, vol. 8274, pp. 343–357.

[35] L. Li, Y. Wang, and E.-P. Lim, "Trust-oriented composite service selection and discovery," in Service-Oriented Computing, 2009, vol. 5900, pp. 50–67.

[36] Q. He, J. Yan, H. Jin, and Y. Yang, "Servicetrust: Supporting reputation-oriented service selection," in Service-Oriented Computing, 2009, vol. 5900, pp. 269–284.

[37] W. Chen, I. Paik, T. Tanaka, and B. Kumara, "Awareness of social influence for service recommendation," in Services Computing (SCC), IEEE International Conference on, 2013, pp. 767–768.

[38] J. Kirchner, A. Heberle, and W. Löwe, "Evaluation of the employment of machine learning approaches and strategies for service recommendation," in Fourth European Conference on Service-Oriented and Cloud Computing (ESOCC), 2015, pp. 95–109.

[39] S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," Informatica, no. 31, 2007, pp. 249–268.

[40] J. Han and M. Kamber, Data Mining: Concepts and Techniques, 2nd ed. Elsevier, Morgan Kaufmann, 2006.

[41] Machine Learning Group at the University of Waikato, "Weka – Data mining with open source machine learning software in Java," http://www.cs.waikato.ac.nz/ml/weka/.

[42] University of Waikato, "MOA Massive Online Analysis," http://moa.cms.waikato.ac.nz/.

[43] E. Ikonomovska, J. Gama, and S. Džeroski, "Learning model trees from evolving data streams," Data Mining and Knowledge Discovery, vol. 23, no. 1, 2011, pp. 128–168.

[44] Weka, "Weka Javadoc – DecisionStump," date of retrieval: 25 Oct 2014; http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/DecisionStump.html.

[45] M. Simons, "Java implementation of excels statistical functions norminv," http://info.michael-simons.eu/2013/02/21/java-implementation-of-excels-statistical-functions-norminv/; Last Retrieval: 20 June 2015.

[46] MathWorld Team, Wolfram Research Inc., "Fourier series – triangle wave," http://mathworld.wolfram.com/FourierSeriesTriangleWave.html; Last Retrieval: 20 June 2015.

[47] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2001, pp. 97–106.

[48] E. J. Keogh and M. J. Pazzani, "Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches," 1999.

[49] Weka, "Weka Javadoc – Hoeffding Tree," date of retrieval: 25 Oct 2014; http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/HoeffdingTree.html.

[50] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El-Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," National Research Counsil Canada Publications Archive, 2001. [Online]. Available: http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?action=rtdoc&an=8914084

[51] B. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El-Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," Software Engineering, IEEE Transactions on, vol. 28, no. 8, Aug 2002, pp. 721–734.

[52] R. Litschko, "Automated analysis of scientific papers using machine learning and computational linguisitics," Bachelor's Thesis, Karlsruhe University of Applied Sciences, Germany, October 2013.

[53] "Apache Lucene," http://lucene.apache.org.

[54] The Stanford Natural Language Processing Group, "Stanford CoreNLP," http://nlp.stanford.edu/software/corenlp.shtml.

[55] M. Bates and R. M. Weischedel, Eds., Challenges in Natural Language Processing. Cambridge University Press, 2006.

[56] J. R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

[57] C. J. van Rijsbergen, Information Retrieval. Butterworth-Heinemann, 1979.