



HAL
open science

A survey on automatic dashboard recommendation systems

Praveen Soni, Cyril de Runz, Fatma Bouali, Gilles Venturini

► **To cite this version:**

Praveen Soni, Cyril de Runz, Fatma Bouali, Gilles Venturini. A survey on automatic dashboard recommendation systems. *Visual Informatics*, 2024, 8 (1), pp.67-79. 10.1016/j.visinf.2024.01.002 . hal-04403155

HAL Id: hal-04403155

<https://hal.science/hal-04403155v1>

Submitted on 18 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Graphical Abstract

A survey on Automatic Dashboard Recommendation Systems

Praveen Soni, Cyril de Runz, Fatma Bouali, Gilles Venturini

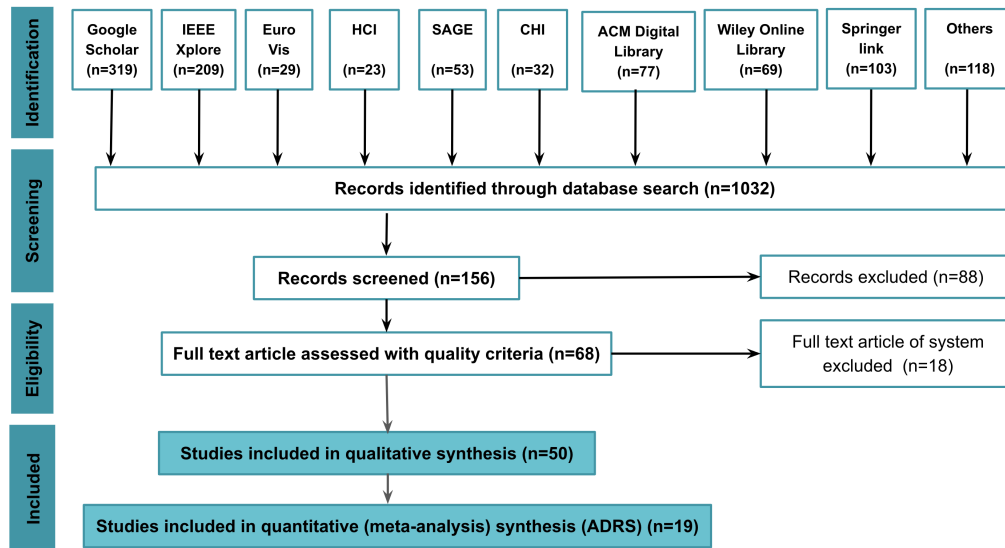


Figure 1: PRISMA flow. Adapted from (Moher et al., 2009)

Response to reviewers:

Dear Editors and dear Reviewers,

We have taken all the required minor changes into account (highlighted in **bold**):

- We use now quotation marks as in this “example” and not as in this ”ex-ample”. We hope this was the suggested change.
- Section 2.4 was shorten and adapted as required
- Section 3.7 was updated to define the values used in table 4
- a few other minor bugs were corrected

Highlights

A survey on Automatic Dashboard Recommendation Systems

Praveen Soni, Cyril de Runz, Fatma Bouali, Gilles Venturini

- Need of automatic dashboard recommendation systems (ADRSs): why, where and who
- Review of surveys that covered systems which support (semi-) automatic recommendation dashboards
- Detailed study and analysis of systems that support automatic recommendation of dashboards, positioning them on 8 dimensions
- Challenges and limitations encountered by existing ADRSs and future scope of ADRSs

A survey on Automatic Dashboard Recommendation Systems

Praveen Soni^a, Cyril de Runz^a, Fatma Bouali^{a,b}, Gilles Venturini^a

^a*LIFAT, University of Tours, 64 avenue Jean Portalis, Tours, 37200, France*

^b*IUT C, University of Lille, 53 Rue de l'Alma, Roubaix, 59100, France*

Abstract

This paper presents a survey on automatic or semi-automatic recommendation systems which help users to create dashboards. It starts by showing the important role that dashboards play in data science, and give an informal definition of dashboards, i.e., a set of visualizations possibly with linkage, a screen layout and user feedback. We are mainly interested in systems that use a fully or partially automatic mechanism to recommend dashboards to users. This automation includes the suggestion of data and visualizations, the optimization of the layout and the use of user feedback. We position our work with respect to existing surveys. Starting from a set of over 1000 papers, we have selected and analyzed 19 papers/systems along several dimensions. The main dimensions were the set of considered visualizations, the suggestion method, the utility/objective functions, the layout, and the user interface. We conclude by highlighting the main achievements in this domain and by proposing perspectives.

Keywords: Automatic Visualization, Dashboard recommendation, Multiple View Interfaces

1. Introduction

In the world of data, visualization plays a vital role to explain abundant and complex data with brief and crucial visual information. One object that is widely used in this context is the dashboard (Pauwels et al., 2009; Sarikaya et al., 2018; Tundo et al., 2020). As we will detail further in this paper, there is not a single definition for this visual representation. We can attempt to simply define a dashboard as a structured set of visualizations, with interactions, that allow users to monitor a process, explore data and get

insights about it. Dashboards are very common in many domains like business intelligence, learning, data analysis, decision making, data storytelling and presentation.

Creating an efficient dashboard is not as easy as it seems. Indeed, in the industry (Mackinlay et al., 2007; Gowthami and Kumar, 2017; QlikTech International, 2013; Sisense Inc, 2004; Tableau Software, 2007; Microsoft, 2015) and the web (Luo et al., 2018; Database Group, 2018), there are many tools to create dashboards. Such tools provide efficient interfaces to edit manually a dashboard: this consists in selecting which data to visualize, then which visualizations to use, how to organize them on the screen, and what dependencies should exist between them like linking or brushing. However, this series of choices is not easy to resolve. In the academic domain, many researchers agree that creating a single visualization is not straightforward. So creating a set of visualizations is indeed even more complex. For several decades, researchers have studied how to automatically generate data visualizations, to fulfill the user’s objectives. Such generative tools are being integrated into the most advanced commercial software that suggest visualizations and dashboards such as Tableau, PowerBI, Qlik Sense or Excel. More recently, academic researchers have also studied how to automatically recommend not a single visualization, but rather a set of visualizations, i.e. a dashboard. Multiple visualizations help users make better decisions (Costa and Aparício, 2019) because the dashboard contains multi-views in contrast to a single visualization. Many viewpoints can be defined in this process of dashboard recommendation: data visualizations, generative algorithms, layout, interfaces, user feedback, and finally the validation of such approaches with user evaluations.

In this paper, we provide a state of the art of automatic dashboard recommendations systems (ADRSs) focusing on approaches that contribute to automatize this process with suggestion or recommendation mechanisms. Section 2 defines the scope of this survey and explains the methodology adopted to select and study papers. Section 3 explains the dimensions we selected to describe the different systems. Section 4 presents a classification of the considered systems. Section 5 discusses some core observations of systems and guidelines for ADRSs, and finally concludes this survey by presenting some perspectives and achievements in this domain.

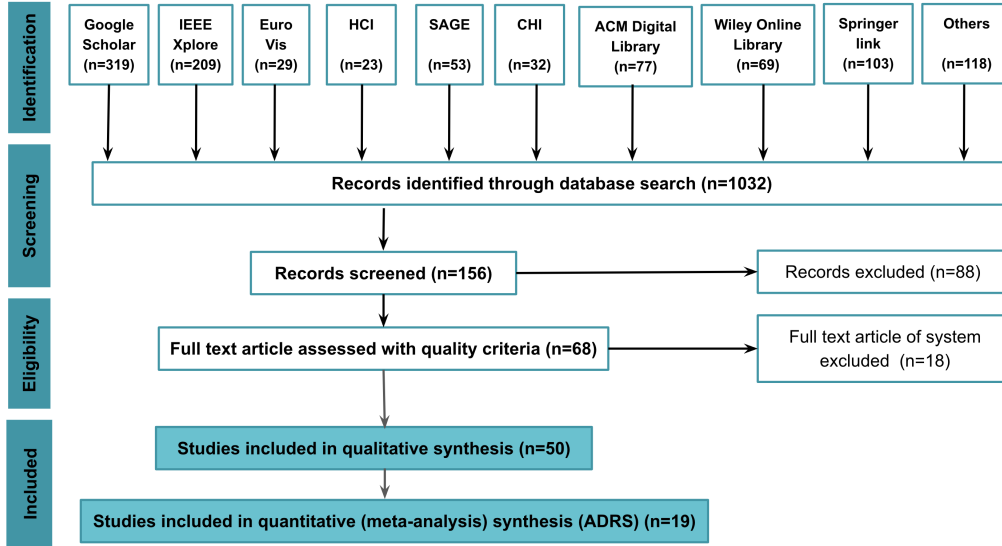


Figure 2: PRISMA flow adapted from (Moher et al., 2009).

2. Scope and methodology

2.1. Problem statement and scope

There is no formal definition of a dashboard (Sarıkaya et al., 2018). Many authors are giving their own definitions, depending on the studied domain: Stephen Few defined a dashboard as “a visual display of the most important information needed to achieve one or more objectives; consolidated and arranged on a single screen so the information can be monitored at a glance” (Few, 2006). Brouns et al. described dashboards as “an easy to read, often single page, real-time user interface, showing a graphical presentation of the current status (snapshot) and historical trends of an organization’s key performance indicators (KPIs) to enable instantaneously and informed decisions to be made at a glance” (Brouns et al., 2015). Others referred to dashboards as “a container of indicators” (Ji et al., 2014) or as “an emerging performance management system, for example, to monitor productivity, analyze cost-effectiveness and improve customer satisfaction” (Park and Jo, 2015).

Dashboards can be considered as a special case of multiple view (MpV) visualizations, a set of approaches which make use of several coordinated visualizations to explore data. The distinction between them is not obvious (Wu et al., 2021). One difference is probably the target users and tasks.

While a MpV visualization can include systems with advanced and specific visualizations intended to be used by domain experts in a data mining context, dashboard could consider simpler and commonly used visualizations and a broader audience, i.e., novice users like citizens or business experts, who would like to explore or monitor multidimensional data. However, the recent progresses made in commercial software for instance are such that dashboard creation/management systems are becoming more and more complex, thus making the difference between dashboards and other MpV approaches thinner.

ADRSs are the focus of this survey and can be defined as follows: an ADRS is a system that generates a dashboard with the use of an automatic suggestion mechanism. It is a tool that assists users in the conception and implementation of a dashboard by formulating recommendations. Such recommendations can occur at any of the major steps of dashboard creation: selection of data, selection of visualizations with their data mapping, layout of the visualizations on the screen. Systems that rely only on manual editing, even with user-friendly interfaces, are not considered as ADRSs. A suggestion mechanism has to be used. Indeed, many ADRSs will also include the possibility to edit the generated dashboard. Many of them can thus be considered as semi-interactive or semi-automatic systems, i.e., with the use of a tight coupling between a suggestion mechanism and a user interface. Finally, there exist several systems that suggest single visualizations to users, and not dashboards. Hence we will not include such systems in our survey.

2.2. Methodology

To find a relevant set of papers we performed a systematic literature review following the guidance proposed in (Schwendimann et al., 2016; Toasa et al., 2018; Vázquez-Ingelmo et al., 2019; Abduldaem and Gravell, 2019). Figure 1 illustrates our process of identifying papers from different resources. It finally results in selecting 50 papers from which 19 are relevant to ADRSs. We considered the following sources: Google Scholar, IEEE Xplore, ACM Digital Library, Wiley Online Library, Springer Link, Science Direct, Semantic Scholar, CHI, SAGE, EuroVis, HCI, and others. We were also interested in other online articles/videos to gather information and knowledge. For querying those sources, we used the following main keywords: “dashboard automatic generation”, “visualization recommendation”, “dashboard generation”, “dashboard visualization”, “multiple view visualizations”, “business

intelligence dashboard”, “multiple visualization tools”, “multi-vision dashboard”, “dashboard guideline and challenges”, “need of dashboard”. Our last access to the resources was on 10th May 2022 with latest and updated versions of tools/papers/systems.

Initially, we identified 1032 papers using different keywords from different resources. We screened them on the bases of different dimensions like number of citations, publishers, year of publication, and considered around 156 interesting papers. We discarded more papers based on new dimensions like abstract reading, quick overview of sections, figures, dashboard interface, layout, methodology, evaluation and users feedback. With these dimensions, we got the basic idea of each of these papers which helps us to consider papers for full-text reading. Finally, we came up with 68 papers for full-text reading. We categorized these papers into four categories: survey papers, ADRS, single-view systems, and miscellaneous (needs and challenges in ADRS, critics, pros and cons in ADRS, common algorithms in ADRS, guidelines to develop ADRS, features and components of ADRS). We excluded the single view systems (18) because our survey is specific to MpV visualizations. We finally included 50 papers of three categories, 16 miscellaneous, 15 surveys, and 19 ADRSs.

2.3. Related previous surveys

We reviewed several survey or state of the art papers that deal with dashboards and ADRSs. Table 1 splits them into three parts. In the first part (top of the table), several papers are devoted to dashboards in general, their use in specific domains or in the industry. Such papers contribute to clarify the context, domain and usage of dashboards. In the second part (middle of the table), papers provide surveys with a more specific focus that can be related to ADRSs. In the third part of the table (bottom), papers present recommendation systems, and they can be compared to those of our survey. (Kaur and Owonibi, 2017) reviewed different recommendation strategies but for single visualizations, while we concentrate on complete dashboards. (Law et al., 2020) analyzed 20 papers dealing with automated insights along different dimensions. This paper does not deal with other kind of recommendation systems used for dashboards.

Reference	P	Scope	Dimensions of analysis
(Schwendimann et al., 2016)	55	Learning dashboard	Context, contribution, users, settings, data, eval.
(Toasa et al., 2018)		Real-time information	Visualization techniques and platforms
(Diamond and Mattia, 2017)	4	Software tools	Excel, Power BI, Tableau, IBM Watson Analytics
(Kubernátová et al., 2018)	15	Recommender systems	Data, task, domain Knowledge, user preferences
(Gowthami and Kumar, 2017)	5	BI Tools for enterprise	License, trail, ease of use, training, platform, Integration, price
(Vázquez-Ingelmo et al., 2019)	23	Tailored dashboards	Year, author, type of publication, domain, factors, target, stage, method, test
(Pauwels et al., 2009)		Dashboards as a service	Purpose, user, effectiveness, development, adoption
(Abduldaem and Gravell, 2019)	18	Design of dashboards	Accuracy, completeness, representation, objectivity
(Sarikaya et al., 2018)		About dashboards	Purpose, audience, visual features, data semantics
(Shen et al., 2021)	45	Nat. lang. interfaces	Query interpretation, data transformation, mapping, view, interaction, dialogue, presentation
(Elias and Bezerianos, 2012)		Annotating dashboards	Needs, prototype, design
(Zhu et al., 2020)	22	Recommendations	Data-driven, knowledge-based, hybrid
(Law et al., 2020)	20	Automated data insights	Types, purposes
(Kaur and Owonibi, 2017)		Recommendation	Data, task, knowledge, user preferences
(Qin et al., 2020)	29	Data visualization	Specifications, approaches, recommendation

Table 1: The different survey or state of the art papers considered in our survey, with three categories: on top, general surveys and information about dashboards. On the middle, surveys focused on specific features in dashboards. On the bottom, surveys related to visualization recommendation. P denotes the number of studied papers, when relevant.

(Qin et al., 2020) performed a survey about making data visualization more effective and efficient. (Zhu et al., 2020) performed a survey for an automatic recommendation of infographics and visualizations. First, they broadly classified all the considered systems into two categories: data-driven and knowledge-based. Then they further classified them within these two categories with different dimensions: statistical automatic visualization, automatic storytelling visualization, automatic annotation, etc. Both surveys (Qin et al., 2020; Zhu et al., 2020) included all types of automatic visualization generation systems independent of view types (single view systems and MpV systems).

In our survey, we concentrate on ADRSs, their core recommendation method for dashboards and its integration in a user interface, and provide a more detailed analysis of such systems along 8 specific dimensions.

2.4. Specific inclusion or exclusion of systems

Our survey includes three systems that do not generate dashboards exactly but rather MpV structures. They use recommendation methods of interest: ScagExplorer (Dang and Wilkinson, 2014), DataShot (Wang et al., 2019) and Calliope (Shi et al., 2020).

We have excluded also three systems. Qualdash (Elshehaly et al., 2020) offers design principles and a process to generate dashboards from the analysis of tasks and requirements expressed by users. It uses a Metric Specification Structure, which, once defined, can be automatically turned into a Vega-lite representation of the dashboard. However, this specification structure is determined by human experts, so we decided not to include Qualdash in the detailed analysis, even if it represents a valuable work.

LADV considers the same principle of using initial user specifications but with sketches or images (Ma et al., 2020): the user provides the system with an image that represents the future dashboard, and LADV, using image recognition methods, turns the graphical input into a dashboard. The user must then determine the data to visualizations mapping, or possibly edit the dashboard. Again, the initial specification is given by the user and not by a suggestion algorithm. Hence we have not included LADV in our survey.

Voder (Srinivasan et al., 2018) is a system that helps users discover and highlight data facts in a visualization. Voder has a simple suggestion mechanism for proposing alternative visualizations to represent data facts. However, Voder is more intended to generate one visualization rather than a MpV visualization, so we did not include it in our survey.

Important step/features in ADRS	Studied dimensions
Work origin	System’s name, year and reference
Suggestion of visualizations	Considered visualizations Suggestion method Utility functions
Layout	Layout method
User’s viewpoint	User feedback User interface and evaluation License

Table 2: The 8 studied dimensions for the 19 systems included in our survey.

3. Dimensions used to describe ADRSs

To describe the 19 collected papers/systems, we have defined 8 dimensions that represent key characteristics of ADRS (see Table 2). Those dimensions were derived both from what is observed in existing systems and also from conceptual analysis of the main steps/features that should be present in an ADRS:

- a recommendation method, that will consider a set of visualizations, among which selected visualizations will be proposed by a suggestion method, possibly using utility functions,
- a layout of the suggested visualizations, with several possible methods,
- a user’s viewpoint, with the possibility of providing feedback to the system, the features of the user interface and its evaluation, and the software license of the system which might limit its use.

In the following sections, we give more details about each dimension.

3.1. Year

This is the year the paper was published. This gives an idea of the recentness of the work. Besides it, we added the paper/system reference.

3.2. Considered visualizations

The visualizations that can be integrated into the dashboard belong to a predefined set. With this dimension, we want to characterize this set. Such visualizations can be:

- standard and typical visualizations, such as scatter plot (SP), table (T), bar chart (BC), pie chart (PC), line chart (LC), map (M), histogram (HT), area chart (AC), gantt chart (GC), radar chart (RC), gauge (G), frequency polygon (FP),
- mark type (MT) like cross, plus, circle, etc. Some systems propose the choice of a mark type to generate visualizations. These mark types are not considered directly as visualization types but ultimately they are the main component of visualization (like a circle in the case of a scatter plot),
- many visualizations (MV), a case in which many visualizations are proposed (including the standard ones and many others).

In this dimension, we also mention the total number of visualizations available in the system. If we did not find this number of visualizations, we indicate it by “not found” (NF).

3.3. *Type of visualization suggestion method*

This is an important dimension as it is a key component of an ADRS. We want to classify here the different methods which are used to propose visualizations to the user. First we make a distinction between ADRSs according to the overall functioning of the suggestion method:

- single and independent (SI), in which the suggestion of the dashboard is done by proposing visualizations one by one, whatever the current content of the dashboard, i.e. the previously selected visualizations,
- single and dependent (SiD), in which the suggestion method also proposes visualizations one by one but taking into account the current state of the dashboard,
- global optimization (GO), in which the method proposes a complete set of co-adapted visualizations, thus performing a global optimization of the suggested visualizations.

We now make distinction between methods according to the core algorithm used to make suggestions. Some methods use simple suggestion algorithms with straightforward heuristics that do not involve learning or specific adaptation to the data distribution:

- direct matching (DM) is one of the simplest methodologies that matches data columns (categorical, numerical, temporal, etc) to visual attributes of a simple visualization (in general BC, PC, SP) with simple and fixed rules. It might use a basic decision model designed by hand from expert knowledge,
- Predefined dashboard templates (T) can be used to propose a set of visualizations with its associated layout. Templates can be determined with expert knowledge or past user behavior.

We distinguish a second set of methods that try to adapt the suggested set of visualizations according to statistical information about data or users, or according to expert knowledge. We call them advanced suggestion methods because they are not strictly fixed algorithms but methods which behavior depends on data characteristics and user objectives. These methods use:

- design from User's Initial Actions (UIAs) is a process that allows users to initially choose particular columns (dimensions or measures) of the given data. Then the system automatically generates and recommends the visualizations relevant to selected data using heuristics often inspired from expert or domain knowledge. This process typically supports user interaction with different system features (drag and drop, selection mechanism) to further manually edit the dashboard,
- visualization query language (VQL) and associated knowledge expressed as constraints. The current situation (data, objectives) is turned into a query and a method such as a constraint solver can output a set of matching visualizations,
- statistical properties of the dataset (SD). Such method evaluates statistical properties (i.e entropy, variance, probability distribution, etc) of the dataset to rank and recommend visualizations,
- behavior-based (BB) approach that considers users' current or past behavior as inputs, and possibly from a large usage history. Such an approach infers knowledge about visualizations, or users' intended task, so as to recommend useful visualizations. They involve simple statistical properties or machine learning (ML) models.

Finally, a third set of complex suggestion methods can be distinguished, and it uses:

- ML models trained on a large set of behavioral observations. Many ML methods are considered in this context: decision tree, learning to rank, neural networks, reinforcement learning, long short-term memory, and deep learning.

3.4. *Utility functions*

During the optimization process of a dashboard, the system might use objective functions to evaluate the possible options before proposing them to the user. These functions are evaluation functions that automatically evaluate a criterion. It can be the expressiveness or effectiveness of the visualization (Mackinlay, 1986), the coverage (are all important data attributes represented in the dashboard), etc. We consider the following cases for utility functions:

- insights (IN), i.e. the potential interest of the discoveries that can be done with this visualization. For instance, to suggest a potentially interesting scatter plot, a utility function can be the correlation between the two axes. In this case, the system will prefer visualizations of correlated attributes,
- “none”, when systems do not involve utility functions,
- “not found” (NF), for systems which do not give a clear idea whether they used utility functions or not.

3.5. *Layout*

A dashboard is a set of visualizations that have to be organized on one or more computer screen(s). This organization raises the problem of finding a correct layout of the visualizations. It means that both the location on the screen as well as the size of each graphic have to be determined. In addition, the layout can be spread on several slides or pages. The possible options for this dimension are:

- manual size (MS) or manual layout (MLa), in which the user has to edit the size and location of the visualizations,
- imposed on a grid (IG) or a specific layout, or with an imposed size (IS) for each visualization,

- optimized Layout (OL) or optimized size (OS), in which an optimization algorithm is used to compute the layout,
- predefined templates (PT), in which the user has to select a template that fits the given dataset,
- “none”, if the system does not support any kind of layout.

3.6. *User feedback*

We distinguish two main possibilities for user feedback:

- direct feedback (DF), in which the ADRS suggests choices and then let the user immediately tell the ADRS if this suggestion is interesting or not,
- behavioral feedback (BF), in which the ADRS can record what choices the user has accepted or rejected, on a longer history, and can use the recorded behavior to improve its suggestions. A typical case of BF consists in collecting and labelling hundreds or thousands of observations so as to form a learning set that can be used to derived statistical properties or to train ML models,
- “none”, if the system does not support any kind of user feedback.

3.7. *User interface and user evaluation*

User interface plays a key role in ADRS. Some ADRS provide features such as:

- drag and drop (DD) of visualizations filters for dimensions selection, icons, etc., in the control panel to design or modify suggested visualizations in a dashboard. These features allow users to edit a dashboard,
- selection mechanisms (SM), such as mark type selection, data dimensions, color combination, global and local filters, at the local (particular visualization) and global (for all the visualization in a dashboard) levels. It makes the dashboard more user-friendly and appealing.

We also summarize in section 4.8 the evaluation performed for each interface, and we use the following notations:

- **“Sole”** when the evaluation includes the studied system only,
- **“Comp.”** when a comparative evaluation with at least one other system is conducted.

3.8. License

ADRSs have different types of license to come into the market: commercial (C), open-source (OS), and freeware (F). If we did not find the license of the system we indicate it as “not found” (NF).

4. A classification of ADRSs

4.1. Overview

In Table 3, we present the 19 systems included in our survey, with the main values for the 8 studied dimensions. Before analyzing each dimension, we present the main design rationale for each system, starting by the academic work.

ScagExplorer (Dang and Wilkinson, 2014) is a system that focuses on scatter plots evaluated by utility functions, and that performs global clustering and optimization to produce a MpV visualization that summarizes such plots. DashBot (Da Col et al., 2021) generates a dashboard considering user immediate feedback and reinforcement learning. (Chen et al., 2020) (denoted by CCP in the following for “Composition and Configuration Patterns”) uses the combination of a behavior-based approach and statistical knowledge for proposing dashboard layout. Exploration Views (Elias and Bezerianos, 2011) proposes initial dashboard templates and a rich user interface to help novice users. DynSpace (El Meseery et al., 2018) is a system designed for novice users so as to ease the user exploration process with multiple workspaces. VizDeck (Key et al., 2012) involves the user intervention as direct feedback to choose the vizlet to add to the dashboard from the recommended hand thumbnails, which in turn helps the system to improve a decision tree used for recommendations. MultiVision (Wu et al., 2021) uses two deep learning models, one siamese network for ranking visualizations with their data mapping, and one LSTM for assessing the quality of single or multiple charts. Then a simple incremental heuristic is used to add chart to the dashboard, maximising the assessment computed by deep neural networks. DeepEye (Luo et al., 2018) performs an exhaustive search of visualizations and uses ML to keep the good ones, and then it defines a partial order on visualizations and selects the top-k to form a dashboard (Database Group, 2018). Draco (Moritz et al., 2018) uses VQL specification for visualization recommendation with the use of logic programming and a constraint solver, but it does not provide a user interface. Likewise, Zenvisage (Siddiqui et al., 2016) also

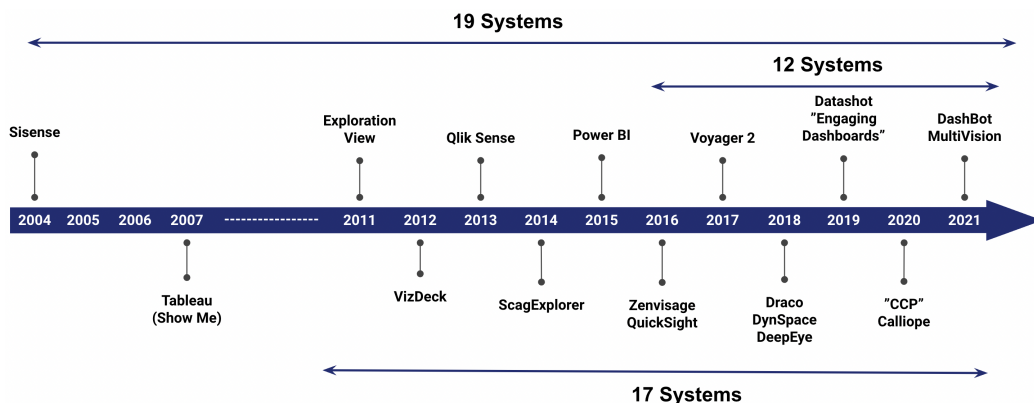


Figure 3: The time line representing the systems studied in our survey.

supports VQL for visualization recommendation but unlike Draco, it provides a user interface with different features. Voyager 2 (Wongsuphasawat et al., 2017) is an advanced version of Voyager. Voyager 2 uses the concept of wildcards for visualizations recommendation. “Engaging dashboards” (Aksu et al., 2019) suggests the visualisations with a predefined simple decision model established with expert knowledge. Datashot (Wang et al., 2019) generates all interesting facts from a data table, each fact being evaluated with utility functions. Then it aggregates facts to form relevant fact sheets, using a simple heuristic and a decision tree to turn facts into visualizations. Calliope (Shi et al., 2020) has a similar behavior to generate data stories, but to scan the huge search space of facts and stories, it uses a more advance optimization algorithm.

There are numerous commercial tools for recommending dashboards, and we selected well known tools in our survey: Tableau (Mackinlay et al., 2007), Power BI (Microsoft, 2015), Qlik Sense (QlikTech International, 2013), QuickSight (Amazon Web Services, 2016), and Sisense (Sisense Inc, 2004). They use almost the same approach for visualization recommendation, with a one by one suggestion of visualizations. These tools support imposed and manual layout optimization with a lot of customization features in the user interface. More importantly, these systems are designed for expert users.

4.2. Year

In Figure 3, we analyse the year of publication for each system. After the pioneering work of (Mackinlay et al., 2007), this domain is active

mainly since a decade. It does not represent so many papers compared to other domains. It has been preceded by many works on single visualization recommendation systems, starting in the 90's. During this last decade, an important contribution is done every year, with a slight increase in the recent years.

4.3. Included visualizations

Through our analysis, we find that the minimum number of visualization supported by a system is 1, in the case of ScagExplorer, while the maximum can be hard to evaluate, several tens for commercial software, because of the many possible variants. In this section, we consider the systems ordered by increasing number of proposed visualizations, and we distinguish two kind of systems: those with a small or medium number of proposed visualizations (8 or less), and those with a large number of proposed visualizations. The first group of systems starts with ScagExplorer which supports only one visualization, i.e. scatter plot. This system concentrates on this visualization and is able to perform a precise evaluation of the visualizations with 9 utility functions devoted to scatter plots. Defining such functions for other visualizations could be possible but was beyond the scope of ScagExplorer. Dynspace supports only 3 visualizations, histogram, pie chart and scatter plot, because this system is clearly designed for novice users. DeepEye supports 4 most frequent visualizations, bar chart, line chart, pie chart and scatter plot, because it requires a learning set and expert rules, which take time to establish. It does not specifically mention their target audience, but it can be used by both novices and experts.

Multi-vision supports 5 visualizations, scatter plots, bar chart, line chart, pie chart and area charts, because these are considered the most commonly used visualizations. DashBot supports many visualizations, some of them are: bar charts, pie charts, scatter plots and tables. VizDeck supports 7 visualizations, i.e. scatter plot, histogram, bar chart, pie chart, time-series plot, line chart and maps, and their implementation as vizlets. Zenvisage supports bar charts, scatter plots, heat maps, histograms and line charts, because this system does not have yet a complete user interface. In Elias and Bezerianos (2011), the number of visualizations supported by Exploration Views is not clearly mentioned, but it seems to be around 8. Exploration views is intended to be used by novice users and therefore it integrates common visualizations only.

The second group consists of systems that can propose many visualizations (>8). Voyager 2 supports many visualizations, i.e. histogram, scatter plot, line chart and mark types, by making use and generalizing the vega-lite specifications, as well as Draco. The number of visualizations in such systems increases the effort needed to update the knowledge base. “Engaging Dashboards” supports multiple visualization types: bullet graph, heat map, highlighted table, dot plot, sparkline, line graph, slope graph, stacked bar graph, variance graph, boxplot, histogram, frequency polygon, spatial map. It is designed to propose the most frequent visualizations used in dashboards. CCP (Chen et al., 2020) supports 14 visualizations, 12 types of information visualization as suggested in (Borkin et al., 2013), plus SciVis and Panel. DataShot and Calliope supports many visualizations, and especially those used for storytelling.

Table 3: Overview of the studied ADRSs.

System	Year and Reference	Considered vis.	Suggestion method	Utility functions	Layout	User feedback	User Interface	License
ScagExplorer	2014 (Dang and Wilkinson, 2014)	SP (1)	SD, GO	9, IN	OL, OS	None	SM	NF
DashBot	2021 (Da Col et al., 2021)	T, BC, SP, PC (4)	BB, SD, ML, SiD	3, IN	IS, IG	DF	SM	NF
CCP	2020 (Chen et al., 2020)	T, LC, BC, M, MV (14)	BB, T	None	PT, MS, ML	BF	DD, SM	NF
Exploration Views	2011 (Elias and Bezerianos, 2011)	PC, BC, T, MV (NF)	T, DM	None	PT, MS, MLa	None	DD, SM	C
DynSpace	2018 (El Meseery et al., 2018)	HT, BC, SP (3)	DM	None	IS, IG, MS, MLa	None	DD, SM	NF
VizDeck	2012 (Key et al., 2012)	HT, BC, SP, LC, PC, M (7)	SD, BB, ML	None	IS, IG	DF	SM	C
MultiVision	2021 (Wu et al., 2021)	SP, LC, BC, AC, PC (5)	BB, ML, SiD	4	IS, IG, MS, MLa	None	DD, SM	OS
Tableau (Show Me)	2007 (Mackinlay et al., 2007)	T, BC, SP, GC, MV (14)	UIAs, SI	(2) NF	IS, IG, MS, MLa	None	DD, SM	C
Power BI	2015 (Microsoft, 2015)	LC, T, BC, PC, M, SP, MV (21)	UIAs, SI	NF	IS, IG, MS, MLa	None	DD, SM	C
DeepEye	2018 (Luo et al., 2018)	BC, LC, PC, SP (4)	BB, ML	3, IN	IS, IG	None	SM	F
Draco	2018 (Moritz et al., 2018)	MT (NF)	VQL, ML	None	None	None	None	C
Zenvisage	2016 (Siddiqui et al., 2016)	HM, SP, HT, LC, MV (NF)	VQL	None	IS, IG	None	DD, SM	OS
Qlik Sense	2013 (QlikTech International, 2013)	LC, G, PC, PT, HT, MV (NF)	UIAs	NF	IS, IG, MS, MLa	None	DD, SM	C
Voyager 2	2017 (Wongsuphasawat et al., 2017)	HT, SP, LC, MT, MV (NF)	VQL	2	IS, IG, MS, MLa	None	DD, SM	C
QuickSight	2016 (Amazon Web Services, 2016)	BC, T, SP, HT, MV (NF)	UIAs	NF	IS, IG, MS, MLa	None	DD, SM	C
Sisense	2004 (Sisense Inc., 2004)	SP, PC, HT, LC, MV (NF)	UIAs	NF	IS, IG, MS, MLa	None	DD, SM	C
"Engaging Dashboards"	2019 (Aksu et al., 2019)	SP, HT, HM, FP, MV (13)	DM	None	IS, IG	None	SM	C
Dashboard	2019 (Diamond and Mattia, 2017)	BC, AC, M, SD, PC (14)	BB, SD, ML	2, IN	IS, IG	None	SM	OS
Calliope	2020 (Zeng et al., 2021)	BC, LC, PC, AC, BP (12)	BB, SD, GO	1, IN	IS, IG	None	SM	OS

Commercial tools studied in this paper, namely Tableau, PowerBI, Qlik Sense, Sisense and Quicksight, form a sub-group in this category. They propose many visualizations. Users of such systems are BI experts, data analysts and researchers, who might have a high level in data visualization, so they can work with complex visualization in addition to common ones. Another reason is that such systems charge users for their services, and in return users expect to benefit of more advanced features and services.

In the first group, systems are more focused on limited set of standard visualizations. Their design is more oriented towards novice users, who can more easily understand common visualizations than others. In addition, these systems might not have a fully developed user interface and so the number of proposed visualizations is limited. In the second group, systems include common visualizations but also advanced visualizations that experts can understand and configure. They address different users. The inclusion of complex visualizations also requires an important effort of design and development that other academic works cannot afford.

4.4. Suggestion mechanisms

Here we discuss the automatic or semi-automatic algorithms and methods used for the recommendation of dashboards as mentioned in Table 3, ordered by increasing complexity of suggestion mechanisms (see Section 3.3):

Simple suggestion mechanisms. The systems, in this category, use simple predefined suggestion methods, like direct matching or predefined templates. DynSpace (El Meseery et al., 2018) focused on the management of multiple workspaces to ease the user exploration process. DynSpace uses SD and DM to propose visualizations and mapping, with simple heuristics: “histograms are used for single dimensions, scatter plots for two numerical dimensions, and bar charts are created when one dimension is categorical and the second is numerical”. Once the user has selected a data attribute, DynSpace automatically provides visualizations for this dimension, according to the DM simple strategy.

“Engaging Dashboards” (Aksu et al., 2019) uses a similar strategy but slightly more advanced: a simple predefined decision model for automatic visualization recommendations is used. This model for KPIs visualization is established from experts recommendations. It takes the form of a decision tree, in which leaves are visualizations. Then, considering a description of KPIs, it generates dashboards automatically using the decision model.

Exploration Views (Elias and Bezerianos, 2011) also uses DM but in a different context. Initially, Exploration Views proposes predefined dashboard templates to users. Users can choose one of the template from the set of given templates based on their domain knowledge and previous experience. Then users can customize the initial dashboard by selecting data and visualization. DM is used with simple heuristic to provide data mapping. The user interface has interesting features for novice users, like undo or visual feedback.

To summarize, in this category, the target users are novices. The suggestion mechanism has to be autonomous without requiring any complex actions from users. Furthermore, except for “Engaging dashboards” which does not seem to have any user interface, DynSpace and Exploration views have rich user interfaces with many specific features for novice users. These features compensate for the simplicity of the suggestion method and the limited number of proposed visualizations.

Advanced suggestion mechanisms. These systems use more complex statistical models and methods involving statistical or knowledge based approaches. CCP (Chen et al., 2020) uses a BB approach. Authors have analyzed visualizations layouts proposed in many MpV visualization papers. They have created co-occurrence matrices of visualizations that appear often together in the same MpV interface, as well as representative layouts template with their associated frequency. Then, they use this statistical knowledge to help users determine the content and layout of their dashboard.

ScagExplorer is designed to explore data through a huge collections of scatter plots. ScagExplorer is based on two suggestion mechanisms: GO and SD. The system generates all possible scatter plots from the data, and evaluates each of them with nine scagnostics measures which are utility functions. Then ScagExplorer uses a clustering approach to produce groups of scatter plots. Such a group is a list of plots with a leader that is used as a group prototype. To visualize the clusters, a forced-directed layout technique is used on leader scatter plots. About the interaction, users can select a leader to see all similar plots in that cluster or filter scatter plots by their features. During this process, the system uses statistical properties, i.e. density distribution, for filtering scatter plots.

Several systems are using visual query languages to search and propose visualizations. Voyager 2 is an advanced version of the original Voyager (Wongsuphasawat et al., 2015) with a set of two new important extensions, automating visualizations suggestions and avoiding redundant suggestions,

as discussed in (Wongsuphasawat et al., 2017). In Voyager 2, automatic visualization suggestions are supported by the wildcards concept and CompassQL (Wongsuphasawat et al., 2016) query. Wildcards in the dashboard panel allow users to change the properties of a specification given in CompassQL query, thus letting the system suggest visualizations. It uses two standard utility functions, expressiveness and effectiveness, to rank the proposed visualizations. Zenvisage uses VQL for specifying the desired visual patterns to be present in the visualizations. Zenvisage query language let the user formulate a query, and then the system returns visualizations. Using a small number queries, users can explore trends, patterns and insights of the given data.

In this category of systems, one subgroup is formed with commercial software, namely Tableau, Power BI, Qlik Sense, QuickSight and Sisense. In general the user starts by selecting data of interest. Then these systems recommend visualizations for the considered data. They use a one by one strategy to suggest visualizations, and it seems that they do not take into account the current state of the dashboard and the previously selected visualizations. This one by one strategy is highly guided by the user who must take crucial decisions. Visualizations can be deeply customized within a rich interface, both from the visual and interactive points of view. So such systems require in general skilled users. In addition, they do not disclose how their suggestion mechanism works.

Complex suggestion mechanisms. These are systems that use ML-based suggestion mechanisms. The general idea in such systems is to acquire expert knowledge about visualization and dashboards through the learning of a model trained on empirical visualization data. Such data can be for instance the labelling of existing visualizations, or the comparison of several couples of visualizations with the selection of the best one for each couple. This train set can be augmented as users interact with the system.

Draco (Moritz et al., 2018) is based on a VQL used to query a knowledge based model with constraints, hard and soft. Draco compiles a user’s query including the dataset, the partial specification and the task, into a set of facts and combines them with the existing knowledge base in the Answer Set Programming framework. Draco then calls a solver (.clingo) to resolve the constraints and obtain a feasible answer set. Finally, Draco translates the answer set into a Vega-lite specification. To establish the weights associated to soft constraints, Draco uses a support vector machine to learn such weights

from empirical data, with a learning to rank approach that is empirically more efficient than the use of experts weights.

DashBot (Da Col et al., 2021) generates a dashboard on a one-by-one visualization basis using reinforcement learning. DashBot selects the most relevant panel/visualization, i.e. data attributes, using SD (i.e. entropy, variance, and coverage). It takes into account the panels already selected in the dashboard. The user can provide immediate feedback, such as “yes” in which case the panel is added to the dashboard, or “no with explanations”, in which the use can tune the panel, or “no without explanations” in which case a reinforcement learning algorithm is used to generate a new panel. DashBot supports two strategies, ϵ -greedy and soft-max, each of them providing different exploitation and exploration trade-offs for panel generation.

VizDeck (Key et al., 2012) uses the combination of two concepts, statistical function and vizlet model. A vizlet is defined as a 4-tuple which includes two render functions (visualization producing thumbnails and interactive version of the vizlet), a predicate (indicating compatibility with dataset) and a scoring function (one of several signals used for ranking). For recommendation purposes, a decision tree is trained to relate statistical properties, like distinct values, entropy, coefficient of variation, kurtosis, or periodicity, to a particular vizlet. User actions of promoting or discarding vizlet allow the system to improve its learned model by collecting positive and negative examples. VizDeck involves also the use of heuristics based on human perception and these heuristics control vizlets recommendation.

DataShot (Wang et al., 2019) uses a large history of visualizations usage in spread sheets to learn what models are often used and in what conditions. It derives also a decision tree from this model to determine what visualizations to use for a given data fact. It uses an exhaustive search of facts, followed by a clustering/aggregation approach that takes into account the topic covered by facts. Then it generates the corresponding fact sheet with the use of the decision tree to turn facts into visualizations.

Calliope (Shi et al., 2020) also uses as a starting point a large database of story telling videos which have been manually analyzed and annotated. It does not scan the whole search space but it uses rather a tree-based optimization method to select to most relevant facts and their organization into a story (nodes of the tree are facts, a story being a path from root to leaf). It provides a rich user interface to edit stories.

DeepEye proposes the use of two ML models. First, it generates all possible visualizations, and a first decision is made to discard “bad” visualizations

and to keep only the “good” ones. This ML model is trained on a learning set where users have established a ground truth. Then, a partial order is established between the selected visualizations. A heuristic based on expert knowledge, or alternatively a learning to rank approach, is used to select the top-k visualizations that can be proposed to the user. In the evaluation of these two methodologies, authors found the first mechanism based on expert knowledge to perform better than the second one.

MultiVision (Wu et al., 2021) learns deep learning models to guide a recommender algorithm. The deep neural networks are trained with labeled data or data from the logs of the system. These networks can assess the quality of a visualization and its data mapping, or the quality of a MV visualization. The recommender is a simple heuristic that adds visualization one by one so as to maximize the evaluation provided by the deep networks.

4.5. Utility functions

Systems which integrate utility functions use either advanced or complex suggestion mechanisms for visualization generation. This is due to the fact that utility functions are used to automate the evaluation of the generated visualizations, opening the possibility to use optimization or ML algorithms. Hence such functions are involved in either completely automatic systems like ScagExplorer, MultiVision, DeepEye, DataShot, Calliope or systems that require minimal human intervention like Tableau or Voyager 2. DashBot, which is a more interactive system than the previous ones, uses utility functions to improve the suggestions proposed to the user. The literature for commercial systems (QlikTech International, 2013; Amazon Web Services, 2016; Sisense Inc, 2004; Microsoft, 2015) does not reveal anything about the use of utility functions, and the other remaining systems do not seem to use utility functions at all.

4.6. Layout

Finding an efficient layout for a dashboard is a complex problem that is not well documented or studied in the literature. Its impact is also not clear on the user’s performance. Al-Maneea et al. (Al-Maneea and Roberts, 2020) proposed a tool to help users create and re-configure the layouts of different MpV visualisations. In the 19 systems, the layout was determined as follows:

No layout. Draco (Moritz et al., 2018) supports the visualization recommendation method but it has not been explicitly tested within a user interface. Therefore, it does not solve at all the layout problem,

Manual (user editing). In many systems (Tableau, PowerBI, SpagoBI, Qlik Sense), the user has to entirely determine how to organize the visualizations. Users have access to different features: resize, move, delete, edit, hover, etc. MultiVision supports major user editing features: charts editing, lock, move and resize in a responsive grid layout. Dynspace is certainly one the most advanced manual systems, with its specific strategy to manage several workspaces. Other systems (Exploration Views, Voyager 2 and (Chen et al., 2020)) also support manual editing.

Templates and predefined layouts. Exploration Views (Elias and Bezerianos, 2011) proposes directly a compound set of visualizations with a pre-defined template. One could say that expert knowledge is used to propose good templates. Then users can choose a template based on personal knowledge and understanding. In (Chen et al., 2020), the statistical analysis of previous layouts published in research papers is used to suggest which visualization to add and where to place it with respect to the current one.

Imposed layout. Several systems (Da Col et al., 2021; Key et al., 2012; Luo et al., 2018; Siddiqui et al., 2016; Aksu et al., 2019; Wang et al., 2019; Shi et al., 2020) do not let the user choose the layout, or propose a limited set of alternatives (Wang et al., 2019). Instead, the system uses a fixed strategy. In DashBot, the layout is a grid, to which the generated visualizations are added one by one. Likewise, in the case of VizDeck, the user selects vizlets one by one and the dashboard layout automatically changes with every add-on of vizlet. DeepEye generates a multi-page dashboard with a list of visualizations based on the selected suggestion mechanism. Voyager 2 suggests focus and related views which are automatically positioned on the dashboard without users' intervention. Like DeepEye, Voyager 2 also suggests the list of visualizations in a multi-page dashboard.

Optimized layout. In this survey, only one system (Dang and Wilkinson, 2014) proposes an optimized visualization layout without users' intervention. ScagExplorer automatically determines the layout of the representative scatter plots with a data projection method that takes into account the similarity between scatter plots to position them in 2D and the cardinality of the cluster to set the size of the thumbnail.

4.7. *User feedback*

Dashboard recommendation is a process that follows a proper sequence of steps during which the user feedback can be taken into account to improve the suggestion of visualizations and other options. A special case of feedback is the immediate feedback, but as far as we know only DashBot really implements this strategy. Hence, this system is able to take into account user’s advices given to the system. A second type of feedback is the behavioral feedback. Here a large usage history is considered to learn or improve an ML model. A slight distinction can be made between VizDeck or Multivision and the other systems. In VizDeck, the systems learns from the interaction between the user and the suggestion strategy. Recommendations that are kept (respectively not kept) by the user are considered as positive (respectively negative) examples that can be added to a training set. The system strategy can be regularly updated and could be adapted to the current user. Other systems consider large usage history of external systems (e.g. from Excel), in order to learn knowledge about visualizations design and efficiency. Here the behavioral feedback is obtained from other systems than the current recommendation strategy. Hence adaptation is not specific to the current users.

4.8. *User interfaces and evaluation*

In this section, we have ranked the studied systems from the simplest to the most complex user interface. We start with Draco because it does not provide any interface, but it outputs a Vega-lite code. Then some systems do not provide any editing interface, but instead a direct visualization of the suggested dashboard, as in “Engaging Dashboards”. Zenvisage and Voyager 2 provide minimal user interfaces.

Then a second group of systems provides advanced users interfaces, often with a focus on specific features. Dynspace concentrates on multiple workspaces and other features for novice users, as does Exploration Views. Other systems provide rich user interfaces with detailed view of the proposed dashboards (ScagExplorer, MultiVision, DeepEye, DataShot, Calliope), advanced interactions devoted to user feedback (DashBot, VizDeck) or interactive options selection (CCP).

Finally, commercial systems (Mackinlay et al., 2007; Gowthami and Kumar, 2017; QlikTech International, 2013) support the most advanced user interfaces with many customization features compared to other scientific tools. These commercial tools propose a highly versatile dashboard interface. In

System	Type	U	Mode	Findings
CCP	Sole	20	Online	Completion time, Ease of use
Expl.View	Sole	15	Offline	Ease of use, chart creation and customization
DynSpace	Sole	8	Offline	Selecting dimensions preferred
VizDeck	Comp.	32	Offline	Completion time
MultiVision	Sole	12		Useful, convenient, easy to learn
Zenvisage	Comp.	12	Offline	Completion time, accuracy, usability and satisfaction
Voyager 2	Comp.	16	Offline	Supports focused analysis, related views + wildcards helpful
Eng. Dash.	Sole	3	Online	Dashboard improvement
Tableau	Sole	10	Online	Show Me + Add to Sheet useful
DataShot	Sole	10	Offline	Effective data fact extraction, visual design, interactions
Calliope	Comp.	16	Offline	Logicity, memorability, comprehension, engagement, dissemination

Table 4: A summary of the user evaluations performed in ADRSs. U denotes the number of users involved in the evaluation.

return, the user has to be familiar with complex technical terminology and interface set-up.

About the evaluation of interfaces, commercial systems do not provide evaluation results, except for Tableau. For Draco, DataShot and ScagExplorer, no user was involved. The evaluation was a demonstration of functionalities on use cases. For DeepEye, a machine based evaluation was performed. For the remaining 11 systems, a user evaluation was performed, as summarized in Table 4. 7 studies test the functionalities of the systems, while only 4 perform a comparative study involving up to 3 different systems.

4.9. Market accessibility

Some systems (Dang and Wilkinson, 2014; Da Col et al., 2021; Chen et al., 2020; El Meseery et al., 2018) are not available practically to be tested and used. ADRSs have different types of licenses to come into the market:

- Commercial: systems (Mackinlay et al., 2007; Gowthami and Kumar, 2017; Elias and Bezerianos, 2011; QlikTech International, 2013; Aksu

et al., 2019; Amazon Web Services, 2016; Sisense Inc, 2004) are commercial, they do not allow users to modify their source code. And users have to pay to use their services,

- Open Source: only two systems (Siddiqui et al., 2016; Wu et al., 2021) are open source. This kind of systems allows users to make modifications to their source code for personal customization. MultiVision and Zenvisage share their source code on GitHub. MultiVision is under development,
- Freeware: Only one system (Luo et al., 2018) is a freeware that provides its service free of cost and without time constraints. Commercial systems provide a free trial for a few days with time constraints.

4.10. *Transverse categorization of systems based on methodologies and features*

In complement to the previous analysis of descriptive dimensions, we propose in this section different and complementary viewpoints on the analyzed systems.

Commercial systems. These systems form a group with similar features. They all propose many visualizations and advanced interfaces. They are in general designed for experts users and are more difficult to use by novice users. They generate visualizations in a dashboard with a one by one strategy that relies on user choices to select the proposed options. They let the user do a lot of customization in a user interface with many complex features. They do not seem to use immediate feedback and in general they do not disclose their suggestion method.

Proposing dashboards templates and layout. Several systems are proposing predefined templates, either according to expert knowledge (Elias and Bezeiranos, 2011) or usage history like CCP. This is helpful for users. However, users need some expertise to select the right template that fits the data, and manually edit the dashboard to finalize it. These two limitations are the main difficulties for novices to use this kind of systems. In other systems, the layout either relies on user manual editing or is imposed, usually with a grid or tiled format.

ML and Deep Learning algorithms. Several systems (Da Col et al., 2021; Wu et al., 2021; Luo et al., 2018) use ML methods for visualization recommendation. These systems reduce human efforts and suggest precise visualizations related to data. They avoid to formulate complex knowledge bases as in Draco for instance. The authors of Draco also admit this fact as they use a support vector machine to set up weights that are too difficult to be humanly determined. They also show that human settings are experimentally less efficient. But these systems involve complex algorithms that do not explicitly acquire knowledge. They can not explain why a suggestion is interesting, or why a visualization is better than another one. They work as black boxes, with an abstraction of the whole process, and users are only aware of input and output. Another drawback is that these systems require high computation costs in the learning phase as well as data acquisition, labelling and preprocessing.

In our survey we have only one system (Wu et al., 2021) which uses deep learning (Wu et al., 2021), with two different architectures (Siamese network and LSTM). As for other ML approaches, Deep Learning has the same drawback and advantages. It is not clear whether the actual quantity of available labelled data is in favor to Deep Learning compared to standard ML. Yet it is difficult to compare Deep Learning and ML algorithms because this would require the same data and evaluation procedure.

Statistical algorithms. Many systems focus on the statistical properties of the dataset for improving the quality of suggestions. They look for insights, relevant information or common behavior in the data (Dang and Wilkinson, 2014; Da Col et al., 2021; El Meseery et al., 2018; Key et al., 2012). VizDeck (Key et al., 2012) uses 5 statistical properties to generate top-k visualization. DashBot (Da Col et al., 2021) uses aggregation function (min, max, count, sum, avg) to compute other statistical properties like entropy and variance in the data preprocessing phase (offline process) of dashboard generation. ScagExplorer (Dang and Wilkinson, 2014) uses statistical properties for filtering the scatter plots. Similar to ML algorithms these systems also involve high computation of datasets to find different statistical properties. The algorithms behind calculating SD are not as complex as that of ML methods.

Query based systems. Several systems (Da Col et al., 2021; Luo et al., 2018; Moritz et al., 2018; Siddiqui et al., 2016) use knowledge based approaches

with query languages. Those systems, in contrast to ML based approaches, have an explicit representation of knowledge. Formulating such knowledge can only be done by experts, it requires a lot of work, and such knowledge bases might be difficult to maintain. But once established and validated, it can be used to propose visualizations and to explain the results. Such system do not require high computation to acquire knowledge, however, the inference they perform might require a long time.

Completely automatic systems. Complete automation of dashboard creation has two main advantages: for novice users, it is easier to obtain a complete dashboard than to edit it, a task that requires knowledge of visualizations and understanding of the user interface. Another advantage of such an approach is that it can cope with the huge search space involved in dashboard creation. However, such systems are not so numerous, even if their presence is increasing over the recent years: ScagExplorer (Dang and Wilkinson, 2014), “Engaging dashboards”, DeepEye (Luo et al., 2018), DataShot (Wang et al., 2019) or Calliope (Shi et al., 2020) support complete automation for dashboard generation. These systems only ask users to upload a dataset as an input and generate dashboards as the output. These systems generate, recommend, and optimize the dashboard on the basis of dimensions and attributes of the dataset. It is interesting to see that these systems however do not support yet user interfaces with features like drag and drop, resize, move, add, and delete the visualization. During the whole process of dashboard recommendation, these systems do not allow any kind of user intervention.

User technical skills. In the evaluation section of the studied papers (Elias and Bezerianos, 2011; El Meseery et al., 2018; Luo et al., 2018), are defined different levels (beginner, advanced, expert) of expertise for different users (novice and expert) to understand BI tools. Technical skills (in data analysis, data modeling, visualizations pros and cons, terminology) possessed by users help them to decide their preference for a given system. (Elias and Bezerianos, 2011) state that it is difficult for novices to understand the technical terminology used in the system interface to define different elements and components of the dashboard. On the other hand, experts are familiar with these technical terms. Similarly, (Saket et al., 2018) states that after mathematical experiment and calculation, authors find that a scatter plot is the best option to find anomalies in data because of its good performance (in terms of time, precision and accuracy). But users equally preferred bar

charts instead of scatter plots for the same task because users are more familiar with bar charts and they can easily interpret bar charts in contrast to scatter plots. Systems (Elias and Bezerianos, 2011; El Meseery et al., 2018) are more preferred by novice users because their terminology is easier to understand in comparison to commercial systems (Mackinlay et al., 2007; Gowthami and Kumar, 2017; QlikTech International, 2013) and open-source systems (Wu et al., 2021) whose major users are experts and researchers.

5. Conclusion and perspectives

5.1. *About this survey*

In this paper, we have focused on automatic dashboards recommendation systems. This survey contributes to researchers and BI experts, acting as a resource and guideline for the design and development of ADRSs. It also helps readers to categorize the level of considered visualizations and complexity of suggestions methods used in ADRSs.

We found only 19 ADRSs starting from a pool of 1036 papers. It shows the scarcity of such systems in the field of data visualization. We identify 8 dimensions to analyze these systems. The results of this analysis show that there are many challenges that could be addressed in the future of ADRS.

5.2. *Analysis of system launched year*

Regarding previous surveys (Shen et al., 2021; Zhu et al., 2020) which showed the evolution of automatic visualizations recommendation systems (from 2001 to 2021 and from 2012 to 2020 respectively) and with our analysis presented in Table 3, we can conclude that in last decade (2011-2021) there is an increase in the number of ADRS. In addition to this growth, there is also a transformation in technology used to develop such ADRSs, from simple and manual to advance and automatic suggestion mechanisms respectively, with the use of ML. It attracts researchers, data scientists and business intelligence experts to develop new and advanced ADRSs. Based on this fact, we think that the work in this domain will continue to increase over the years.

5.3. *No dominant methodology and lack of experimental comparison*

Various methodologies are used in ADRS. Indeed ML, Deep Learning and Statistics are the most used and recent trends for developing ADRSs. However, in our opinion, knowledge based approaches are still competitive to other ML models, and this will continue as long as collected data about

visualization usage will not be informative and voluminous enough, and as long as explanations of ML models will not be clear for users.

In addition, it is hard to determine which approach is the best. Yet the only possibility to compare these systems is to study their evaluation methodology and obtained results, but due to different users, data and hardware, this could be biased. Every system uses different evaluation methodologies and dimensions to evaluate its performance and user experience. The most valuable results of the evaluation are found through users' feedback. But it would be more beneficial to evaluate different ADRSs with the same datasets, tasks, participants and hardware. In other words, to compare different systems performances, there is a need to perform experiments with all systems in similar conditions, including quantitative results and users' feedback. Zeng et al. (Zeng et al., 2021) have proposed an evaluation focused framework to contextualize and compare visualization recommendation algorithms. They have demonstrated the proposed framework with 5 visualization recommendation systems only. Alternatively, it would be interesting to develop a testing platform that could use the same grounds for comparing all systems.

So there is not an approach that dominates the others, and studied systems are in a way complementary. This shows that we are still at an early stage on the research about ADRS.

5.4. *Novice users*

Recommendation systems are useful for all users, but they play a more important role for novice users who need more guidance and help than for others. System users (novices/experts) and the terminology of the system directly affect each other. If users of the system are experts, in such a case technical words used in the dashboard interface are fine, like in the case of commercial systems (Mackinlay et al., 2007; Gowthami and Kumar, 2017; QlikTech International, 2013). But on the other hand, if users are novices in this case it can be very difficult for them to deal with the dashboard interface. (Elias and Bezerianos, 2011) highlighted this issue during the evaluation through user feedback after the experiment. In addition, systems that propose a complete dashboard without user intervention can be interesting for novices who will not have to take difficult decisions. On the other hand, expert users are more likely to make good suggestions to recommendation systems through user feedback or manual editing.

5.5. Dashboard interface

Dashboards can be extended and improved with specific visual and interactive features. Such possibilities are a bonus for users because they help them understand the data story quickly and improve user interactions with the system.

Linking and brushing are common techniques to many systems, but additional and more specific features can be found, like dashboard annotation, animation of visualizations, color combination, drag and drop, local and global filters, undo/redo and visualization caption. (L’Yi et al., 2019) explains and compares how the effects of annotation and animation help users understand visualization quickly. (Yuan et al., 2021; Few, 2003) use a selection of the right color pallet for different view types (widgets), which also help users to quickly understand insights of the dashboard. We observe that commercial dashboard interface supports many such features (pulse feature in Sisense, narrate feature in Qlik Sense, story feature in Tableau, etc) compared to other systems of Table 3. Hence the dashboards designed with commercial tools are very appealing.

5.6. User feedback

The integration of user feedback is still rare but it is a major challenge for future ADRSs. Only 4 systems (Da Col et al., 2021; Chen et al., 2020; Key et al., 2012; Aksu et al., 2019) support user feedback. More specifically, the most popular commercial systems (i.e. Tableau, Power BI, Sisense, Qlik Sense, QuickSight) do not propose user feedback feature. In addition, comparative evaluation of user feedback methods is difficult, because of the different testing conditions of each ADRS.

User feedback, either immediate or from logs recording visualizations usage, represents an important feature to help systems generate better suggestions. Immediate user feedback involves the users actively during the process of visualization recommendation and it can help the system to dynamically understand the user’s aim and objectives. It can improve the quality of suggested visualizations. User feedback based on large usage history concern any kind of users and domains. It would be interesting to collect richer data, and more information about users and their context. The collection of such enriched data will open the way to more efficient ML models, that will not be general to all users but more specific to some users. It could be interesting to know about user context and level to make suggestions that are better personalized.

From this point of view, reinforcement learning could play an important role because it is a learning paradigm that can take into account immediate feedback. Also another paradigm such as semi-supervised learning or active learning could be useful to personalize models to a given user.

5.7. *Lack of open source and freeware ADRS*

There are more commercial ADRSs in comparison to open source and freeware systems. Indeed ADRSs attract researchers but also major investors and enterprises. But there is still a scarcity of open source and freeware systems. Due to this scarcity, common users deprive of analyzing household data using ADRSs, as these common users do not want to use enterprise versions (commercial systems) and invest money to perform their domestic tasks (electricity bills, daily sales analysis for street vendors, monthly expenses, etc). Even in the last decade, we have observed that there is less focus on open source and freeware ADRSs development in contrast to commercial tools.

Acknowledgment

This research work has been supported and funded by the ANR with the BI4PEOPLE project.

References

- Abduldaem, A., Gravell, A., 2019. Principles for the design and development of dashboards: literature review. Proceedings of INTCESS , 1307–1316.
- Aksu, Ü., del Río-Ortega, A., Resinas, M., Reijers, H.A., 2019. An approach for the automated generation of engaging dashboards, in: OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”, Springer. pp. 363–384.
- Al-Maneea, H.M.A., Roberts, J.C., 2020. A tool to help layout multiple view visualisations guided by view analysis, in: Eurographics & Eurovis 2020.
- Amazon Web Services, 2016. Quicksight.
<https://aws.amazon.com/quicksight/>.

- Borkin, M.A., Vo, A.A., Bylinskii, Z., Isola, P., Sunkavalli, S., Oliva, A., Pfister, H., 2013. What makes a visualization memorable? *IEEE transactions on visualization and computer graphics* 19, 2306–2315.
- Brouns, F., Zorrilla, M., Álvarez, E., Solana-González, P., Cobo, A., Rocha, R., Viaña, M., Hoyos, C., Silva, M., Lazo, C., et al., 2015. D2. 5 learning analytics requirements and metrics report. Technical Report. Technical Report.
- Chen, X., Zeng, W., Lin, Y., Ai-Maneea, H.M., Roberts, J., Chang, R., 2020. Composition and configuration patterns in multiple-view visualizations. *IEEE Transactions on Visualization and Computer Graphics* 27, 1514–1524.
- Costa, C.J., Aparício, M., 2019. Supporting the decision on dashboard design charts, in: *Proceedings of 254th The IIER International Conference 2019*, pp. 10–15.
- Da Col, S., Ciucanu, R., Soare, M., Bouarour, N., Amer-Yahia, S., 2021. Dashbot: An ml-guided dashboard generation system, in: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 4696–4700.
- Dang, T.N., Wilkinson, L., 2014. Scagexplorer: Exploring scatterplots by their scagnostics, in: *2014 IEEE Pacific visualization symposium*, IEEE. pp. 73–80.
- Database Group, 2018. Deepeye. <https://deepeye.tech/>.
- Diamond, M., Mattia, A., 2017. Data visualization: An exploratory study into the software tools used by businesses. *Journal of Instructional Pedagogies* 18.
- El Meseery, M., Wu, Y., Stuerzlinger, W., 2018. Multiple workspaces in visual analytics, in: *2018 International Symposium on Big Data Visual and Immersive Analytics (BDVA)*, IEEE. pp. 1–12.
- Elias, M., Bezerianos, A., 2011. Exploration views: understanding dashboard creation and customization for visualization novices, in: *IFIP conference on human-computer interaction*, Springer. pp. 274–291.

- Elias, M., Bezerianos, A., 2012. Annotating bi visualization dashboards: Needs & challenges, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1641–1650.
- Elshehaly, M., Randell, R., Brehmer, M., McVey, L., Alvarado, N., Gale, C.P., Ruddle, R.A., 2020. Qualdash: adaptable generation of visualisation dashboards for healthcare quality improvement. *IEEE Transactions on Visualization and Computer Graphics* 27, 689–699.
- Few, S., 2003. Information dashboard design. S. Few .
- Few, S., 2006. Information dashboard design: The effective visual communication of data. volume 2. O'reilly Sebastopol, CA.
- Gowthami, K., Kumar, M.P., 2017. Study on business intelligence tools for enterprise dashboard development. *International Research Journal of Engineering and Technology (IRJET)* 4, 2987–2992.
- Ji, M., Michel, C., Lavoué, E., George, S., 2014. Ddart, a dynamic dashboard for collection, analysis and visualization of activity and reporting traces, in: European Conference on Technology Enhanced Learning, Springer. pp. 440–445.
- Kaur, P., Owonibi, M., 2017. A review on visualization recommendation strategies, in: International Conference on Information Visualization Theory and Applications, SCITEPRESS. pp. 266–273.
- Key, A., Howe, B., Perry, D., Aragon, C., 2012. Vizdeck: self-organizing dashboards for visual analytics, in: Proceedings of the 2012 acm sigmod international conference on management of data, pp. 681–684.
- Kubernátová, P., Friedjungová, M., Duijn, M.v., 2018. Constructing a data visualization recommender system, in: International Conference on Data Management Technologies and Applications, Springer. pp. 1–25.
- Law, P.M., Endert, A., Stasko, J., 2020. Characterizing automated data insights, in: 2020 IEEE Visualization Conference (VIS), IEEE. pp. 171–175.
- Luo, Y., Qin, X., Tang, N., Li, G., 2018. Deepeye: Towards automatic data visualization, in: 2018 IEEE 34th international conference on data engineering (ICDE), IEEE. pp. 101–112.

- L'Yi, S., Chang, Y., Shin, D., Seo, J., 2019. Toward understanding representation methods in visualization recommendations through scatterplot construction tasks, in: Computer Graphics Forum, Wiley Online Library. pp. 201–211.
- Ma, R., Mei, H., Guan, H., Huang, W., Zhang, F., Xin, C., Dai, W., Wen, X., Chen, W., 2020. Ladv: Deep learning assisted authoring of dashboard visualizations from images and sketches. *IEEE Transactions on Visualization and Computer Graphics* 27, 3717–3732.
- Mackinlay, J., 1986. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.* 5, 110–141. doi:10.1145/22949.22950.
- Mackinlay, J., Hanrahan, P., Stolte, C., 2007. Show me: Automatic presentation for visual analysis. *IEEE transactions on visualization and computer graphics* 13, 1137–1144.
- Microsoft, 2015. Powerbi. <https://powerbi.microsoft.com/>.
- Moher, D., Liberati, A., Tetzlaff, J., Altman, D.G., Group*, P., 2009. Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. *Annals of internal medicine* 151, 264–269.
- Moritz, D., Wang, C., Nelson, G.L., Lin, H., Smith, A.M., Howe, B., Heer, J., 2018. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE transactions on visualization and computer graphics* 25, 438–448.
- Park, Y., Jo, I.H., 2015. Development of the learning analytics dashboard to support students' learning performance. *Journal of Universal Computer Science* 21, 110.
- Pauwels, K., Ambler, T., Clark, B.H., LaPointe, P., Reibstein, D., Skiera, B., Wierenga, B., Wiesel, T., 2009. Dashboards as a service: why, what, how, and what research is needed? *Journal of service research* 12, 175–189.
- Qin, X., Luo, Y., Tang, N., Li, G., 2020. Making data visualization more efficient and effective: a survey. *The VLDB Journal* 29, 93–117.

- QlikTech International, 2013. Qlik Sense. <https://www.qlik.com/us/products/qlik-sense>.
- Saket, B., Endert, A., Demiralp, Ç., 2018. Task-based effectiveness of basic visualizations. *IEEE transactions on visualization and computer graphics* 25, 2505–2512.
- Sarikaya, A., Correll, M., Bartram, L., Tory, M., Fisher, D., 2018. What do we talk about when we talk about dashboards? *IEEE transactions on visualization and computer graphics* 25, 682–692.
- Schwendimann, B.A., Rodriguez-Triana, M.J., Vozniuk, A., Prieto, L.P., Boroujeni, M.S., Holzer, A., Gillet, D., Dillenbourg, P., 2016. Perceiving learning at a glance: A systematic literature review of learning dashboard research. *IEEE Transactions on Learning Technologies* 10, 30–41.
- Shen, L., Shen, E., Luo, Y., Yang, X., Hu, X., Zhang, X., Tai, Z., Wang, J., 2021. Towards natural language interfaces for data visualization: A survey. *arXiv preprint arXiv:2109.03506* .
- Shi, D., Xu, X., Sun, F., Shi, Y., Cao, N., 2020. Calliope: Automatic visual data story generation from a spreadsheet. *IEEE Transactions on Visualization and Computer Graphics* 27, 453–463.
- Siddiqui, T., Kim, A., Lee, J., Karahalios, K., Parameswaran, A., 2016. Effortless data exploration with zenvisage: an expressive and interactive visual analytics system. *arXiv preprint arXiv:1604.03583* .
- Sisense Inc, 2004. Sisense. <https://www.sisense.com/get/free-trial/>.
- Srinivasan, A., Drucker, S.M., Endert, A., Stasko, J., 2018. Augmenting visualizations with interactive data facts to facilitate interpretation and communication. *IEEE transactions on visualization and computer graphics* 25, 672–681.
- Tableau Software, 2007. Tableau (showme). <https://www.tableau.com/products/trial>.
- Toasa, R., Maximiano, M., Reis, C., Guevara, D., 2018. Data visualization techniques for real-time information—a custom and dynamic dashboard for analyzing surveys’ results, in: *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)*, IEEE. pp. 1–7.

- Tundo, A., Castelnovo, C., Mobilio, M., Riganelli, O., Mariani, L., 2020. Declarative dashboard generation, in: 2020 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), IEEE. pp. 215–218.
- Vázquez-Ingelmo, A., García-Peñalvo, F.J., Therón, R., 2019. Tailored information dashboards: A systematic mapping of the literature, in: Proceedings of the XX international conference on human computer interaction, pp. 1–8.
- Wang, Y., Sun, Z., Zhang, H., Cui, W., Xu, K., Ma, X., Zhang, D., 2019. Datashot: Automatic generation of fact sheets from tabular data. *IEEE transactions on visualization and computer graphics* 26, 895–905.
- Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J., 2015. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE transactions on visualization and computer graphics* 22, 649–658.
- Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J., Howe, B., Heer, J., 2016. Towards a general-purpose query language for visualization recommendation, in: Proceedings of the Workshop on Human-In-the-Loop Data Analytics, pp. 1–6.
- Wongsuphasawat, K., Qu, Z., Moritz, D., Chang, R., Ouk, F., Anand, A., Mackinlay, J., Howe, B., Heer, J., 2017. Voyager 2: Augmenting visual analysis with partial view specifications, in: Proceedings of the 2017 chi conference on human factors in computing systems, pp. 2648–2659.
- Wu, A., Wang, Y., Zhou, M., He, X., Zhang, H., Qu, H., Zhang, D., 2021. Multivision: Designing analytical dashboards with deep learning based recommendation. *IEEE Transactions on Visualization and Computer Graphics* 28, 162–172.
- Yuan, L., Zhou, Z., Zhao, J., Guo, Y., Du, F., Qu, H., 2021. Infocolorizer: Interactive recommendation of color palettes for infographics. *IEEE Transactions on Visualization and Computer Graphics* .
- Zeng, Z., Moh, P., Du, F., Hoffswell, J., Lee, T.Y., Malik, S., Koh, E., Battle, L., 2021. An evaluation-focused framework for visualization recom-

mendation algorithms. *IEEE Transactions on Visualization and Computer Graphics* 28, 346–356.

Zhu, S., Sun, G., Jiang, Q., Zha, M., Liang, R., 2020. A survey on automatic infographics and visualization recommendations. *Visual Informatics* 4, 24–40.