Statistical Validation of Column Matching in the Database Schema Evolution of the Brazilian Public School Census

Muriki G. Yamanaka, Diogo H. de Almeida, Paulo R. Lisboa de Almeida, Simone Dominico, Leticia M. Peres, Marcos S. Sunye, Eduardo C. de Almeida

[1]Federal University of Paraná (UFPR)
Centro de Computação Científica e Software Livre (C3SL)
Curitiba – PR –Brazil

{dha21,mgy20,paulo,sdominico,lmperes,sunye,eduardo}@inf.ufpr.br

Abstract. *Publicly available datasets are subject to new versions, with each version potentially reflecting changes to the data. These changes may involve adding or removing attributes, changing data types, and modifying values or their semantics. Integrating these datasets into a relational database poses a significant challenge: How to keep track of the evolving database schema while incorporating different versions of the data sources? This paper presents a statistical methodology to validate the integration of 12 years of open access datasets from Brazil's School Census, with a new version of the datasets released annually by the Brazilian Ministry of Education (MEC). We employ various statistical tests to find matching attributes between datasets from a specific year and their potential equivalents in datasets from later years. The results show that using the Kolmogorov–Smirnov test, we can successfully match columns from different dataset versions in about 90% of cases.*

1. Introduction

Integrating open data sources is a complex challenge in developing web information systems. Open data sources may exhibit structural changes over time when made public, including variations in data types, values, semantics, and missing values, requiring constant evolution of the integrated database schema before the ingestion of new data [Garcia-Molina et al. 2009]. The PRISM project reported an average of 217% schema changes over 48 months across 12 large web information systems [Curino et al. 2009, Curino et al. 2013]. For example, the Ensembl Genome project presented over 410 schema versions in 9 years. The Ensembl DB schema contains over 175 individual changes of primary and foreign keys in its schema evolution history.

The evolution of a database schema often leads to mapping errors, compromising the accuracy of stored data and ultimately leading to inconsistencies and inaccuracies in data analysis. Furthermore, differences in data presentation and evolving business needs can significantly hinder the incorporation of new data into existing databases.

In this paper, we introduce a statistical methodology to validate the integration of open-access datasets into the Educational Data Laboratory (*Laboratório de Dados Educacionais*) (LDE) information system[1]. This methodology allows us to track the evolution

---

of the system's database schema across different versions of datasets.The LDE system integrates open-access data from Brazil's School Census to support many studies and public educational policies [Schneider et al. 2023, Alves et al. 2019, Schneider et al. 2020, Silveira et al. 2021]. The LDE database contains 12 years of School Census data and is freely accessible. Each year, MEC publishes the School Census[2], which includes comprehensive data from 179,500 schools, such as the number of students, teachers, and classes at each school. However, the publicly available data files have undergone 416 individual changes in naming conventions, as well as the addition and removal of columns over the years. These changes, driven by evolving government requirements, make it challenging for policymakers and researchers to access a unified and integrated reliable source.

Our methodology employs Goodness-of-fit statistical tests to evaluate the evolution of the LDE database schema enhancing the reliability of column matching. Goodness-of-fit tests are meant to define how well some sample of data fits with another given distribution [D'Agostino 1986]. In our context of data integration, the tests conduct data profiling [Abedjan et al. 2015], analyzing column-matching operations such as detecting additions, removals, and changes. This process helps minimize errors and inconsistencies in the evolution of an integrated database schema.

Overall, our main contributions to this paper are the following:

Validation of the LDE database schema: Our methodology validates the quality of data integrated into the LDE database, thereby supporting the evolution of its schema.

Quality metrics based on statistical tests: Our methodology encompasses metrics from four goodness-of-fit statistical tests to evaluate the matches between the attributes of datasets from different releases: Kolmogorov-Smirnov test [Berger and Zhou 2014], Anderson-Darling test [Anderson and Darling 1952], Welch's t-test [Rayner et al. 2009], and the F-test [Hahs-Vaughn and Lomax 2020].

Analysis of the tests: we present the analysis of the results indicating that our methodology can correctly align the columns of different datasets in about 90% of cases considering the Top 3, and about 85% considering the Top 1, showing high accuracy and effectiveness in the validation of the integrated schema.

This paper is structured as follows: Section 2 outlines the changes in the open-access data files and the potential integration problems in a database schema. It also discusses related work. Section 3 delineates the methodology used in this study. The findings are presented in Section 4. Finally, Section 5 provides a summary of the study and outlines the next steps.

2. Background and Related Work

Although schema evolution literature has long acknowledged the complexity of data source integration, the high computational costs associated with general schema evolution techniques have prevented their practical deployment [Cerqueus et al. 2015a, Scherzinger et al. 2016]. Schema evolution refers to integrating changes to a data source over time, including adding new sources. Examples of source transformations over

---

[2]Brazilian School Census Open data (in Portuguese):   https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos
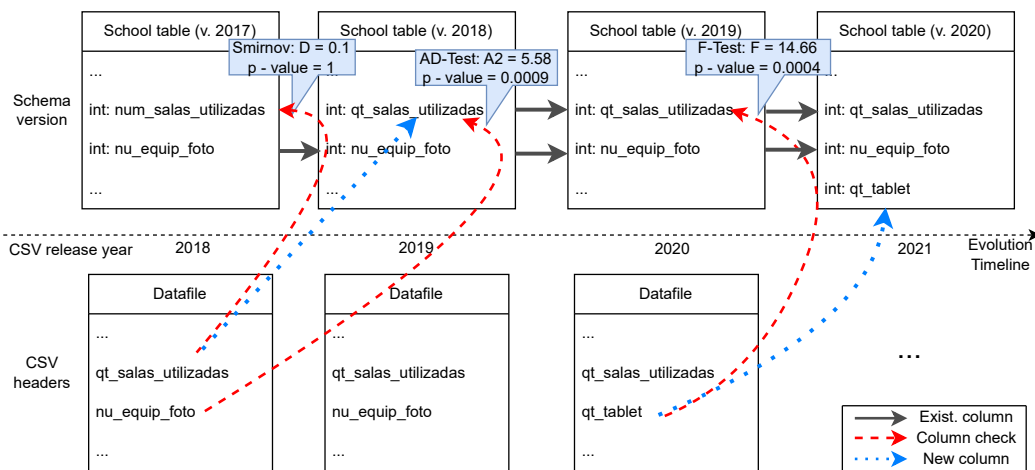
Figure 1. Illustration of schema evolution showing the data file headers from 2018 and 2020, as well as the impact of header changes on the integrated schema. Arrows indicate the mappings. Columns are presented in their original Portuguese names.

time include different column names, changes to the data domain, and their representation. It is also possible for columns or tables to be added as new sources are integrated [Delplanque et al. 2020].

There are many tools to assist in the integration of datasets (a non-exhaustive survey on integration tools is found, here: [Curino et al. 2013]). However, human intervention is often necessary to align open-access datasets containing historical information. This is further complicated by the evolution of the open data file structure over time, including changes in column names, value domains, and additions/removals of columns. This paper focuses on addressing the challenges associated with column name changes and additions/removals.

2.1. The LDE system

The LDE system stores data from the School Census over the past 12 years, compiling a vast amount of educational information. Maintaining this dataset is crucial for monitoring trends over time and gaining valuable insights into the Brazilian educational context. Consequently, the LDE system serves as a key resource for leveraging government open data in academic research. Many projects maintained by MEC and different universities depend on this data, such as the Cost-Student Quality Simulator (SIMCAQ) [3] [Alves et al. 2019]. SIMCAQ evaluates the cost of delivering quality education based on various educational and structural variables, such as class size, teacher salaries, and library resources. Another example of a system that depends on the LDE data is MapFor [Schneider et al. 2023], which tracks teachers' academic backgrounds. These projects demonstrably impact society, highlighting the importance of maintaining high data quality within the LDE database.

To illustrate the changes in column names, consider the following CSV headers released by the MEC open-access files, shown in Figure 1: "$num\_salas\_utilizadas$" and

---

[3]https://www.simcaq.c3sl.ufpr.br/ (in Portuguese)

"*qt_salas_utilizadas*". To illustrate the introduction of new information, consider the column "*qtde_tablet*" added in the 2020 data file. Finally, we illustrate columns when an attribute is no longer included in the data for a particular year. For example, the attribute "*nu_equip_foto*" was removed from the scholar census in 2017.

Properly mapping all these changes in the LDE database is essential to enhancing data quality. In Figure 1, consider a scenario where new information is added to the data files of the scholar census, such as "*qt_salas_utilizadas*". Regardless of whether or not this information is already implied in the existing "*num_salas_utilizadas*" column, there might be a tendency to treat it as a new column. This could lead to the addition of a new column to the LDE database ("*qt_salas_utilizadas*"), resulting in schema evolution. However, mapping to this new column can make it difficult to infer existing information ("*num_salas_utilizadas*") without detailed analysis. When a new column is created (such as "*qt_salas_utilizadas*"), instances from previous years are filled with null values, and subsequent analysis may provide incorrect information, failing to indicate that previous data was present in another column (such as "*num_salas_utilizadas*").

2.2. Related Work

Schema evolution management has been the focus of several works over the years. These works conduct empirical investigations into relational schema evolution [Qiu et al. 2013, Vassiliadis et al. 2015]. In [Klettke et al. 2017], the authors evaluate schema evolution histories over time, examining data from a data lake and schema versioning. Our work analyzes data integration quality and tracks the evolution of the database schema.

Some works have evaluated the schema evolution in the NoSQL database [Meurice and Cleve 2017, Ringlstetter et al. 2016]. In [Cerqueus et al. 2015a], the authors discuss the implementation and customization of verification rules to help developers manage schema evolution and prevent compatibility issues and data loss. [Scherzinger and Sidortschuck 2020, Scherzinger et al. 2016, Cerqueus et al. 2015b] investigate the evolution of NoSQL database schema, focusing on their flexibility, denormalization practices, and changes over development time through empirical analysis of open-source projects. Our work evaluates the quality of schema evolution in a relational database, which implies distinct challenges. The NoSQL schema evolution has greater flexibility and denormalization. However, relational databases enforce constraints and a greater need to maintain integrity.

Prism/Prism++ [Curino et al. 2009, Curino et al. 2013] implements a solution focused on schema evolution in relational databases. Prism uses the data dictionary to track changes in the data schema. It describes an integrated solution to predict and evaluate the impact of schema changes and integrity constraints. The objective is to minimize downtime by automating database migration and documenting schema evolution.

The work of [Delplanque et al. 2020] discusses the challenges of evolving relational database schema. The authors propose a meta-model approach to automate modifications after database changes, providing recommendations to maintain a consistent state. As observed in [Etien and Anquetil 2024], a meta-model for analyzing the impact of changes and ensuring database relational constraints are verified. In contrast, our methodology evaluates the data distribution and other statistical measures without analyzing attribute names. This approach allows us to monitor schema evolution from a

data-centric perspective, providing an understanding of how data changes over time.

3. Goodness-of-fit Schema Evolution Methodology

In this section, we present our statistical methodology used in the integration of open-access datasets into the LDE database. Our methodology employs goodness-of-fit statistical tests to match the columns of the CSV files released each year with the existing columns in the database. First, in Section 3.1, we define the goodness-of-fit tests in the context of schema evolution tests. In Section 3.2, we define our matching algorithm, which uses specific metrics given by the goodness-of-fit tests to determine the correct match of each column for a given year.

3.1. Goodness-of-Fit

Our main hypothesis is that Goodness-of-fit statistical tests ensure reliable *data quality* metrics during column matching. These tests provide information about *data distributions*, *means*, *variances*, and *magnitude* of observed differences. We use the well-known Kolmogorov–Smirnov test, Anderson–Darling test, Welch's t-test, and the F-test to compare the distributions of a column in a given year with possible matches from the next year.

Let $x : (x_1, x_2, \ldots, x_m)$ and $y : (y_1, y_2, \ldots, y_n)$ be the distributions (collected data between years) being compared of sizes m and n, respectively. Also, let $\overline{x}$ and $\overline{y}$ be the means of x and y, and S_x^2 and S_y^2 be the variances of x and y. Now, we briefly describe each test.

Kolmogorov–Smirnov test: This test verifies if two samples are statistically similar. In our methodology, it determines whether the data in two columns from different years follow the same distribution. This allows us to evaluate data consistency over time by comparing the base year with the following year based on the distribution of the samples. Let F_m and G_n be the empirical Cumulative Distribution Functions (CDFs) for the x and y samples defined as follows:

$$F_m(t) = \frac{\text{number of sample x'} \leq t}{m} \tag{1}$$

$$G_n(t) = \frac{\text{number of sample y'} \leq t}{n} \tag{2}$$

the Kolmogorov-Smirnov test is defined as follows:

$$D = max|F_m(t) - G_n(t)|, min(x, y) \leq t \leq max(x, y) \tag{3}$$

where samples are considered to come from the same distribution if D is small enough [D'Agostino 1986, Berger and Zhou 2014]. Considering the example illustrated by Figure 1, the attributes "$qt\_salas\_utilizadas$" and "$num\_salas\_utilizadas$" present the same distribution and data type. In this particular case, the KS-test shows the $D = 0.1$ and $p - value = 1$.

Anderson–Darling test: Similar to the Kolmogorov-Smirnov test, the Anderson–Darling test considers the differences between the distributions, with the difference

that the Anderson–Darling test gives more weight to the tails of the distributions when compared to the Kolmogorov-Smirnov test. For comparing two distributions, the Anderson–Darling statistic can be computed as follows:

$$A^2 = \frac{1}{N(mn)} \sum_{j=1}^{N-1} \frac{(NX_j - jm)^2 + (NY_j - jn)^2}{j(N-j)} \tag{4}$$

where $N = m + n$, $Z_1 < \cdots < Z_N$ is the pooled ordered sample, and X_j and Y_j are the number of observations in x and y that are not greater than Z_j, respectively [Pettitt 1976]. The Anderson–Darling test applied to "nu_equip_foto" and "$qt\_salas\_utilizadas$" results in a statistic of 5.58 and a p-value of 0.0009, indicating a statistically significant difference in their distributions. This suggests that the columns are unlikely to be compatible.

F-test: By applying the F-test to each column, we compare the data variance across different data file versions (released in different years). If the test statistic F is significant, it implies a substantial difference in data distribution, potentially indicating mismatches or missing data in the following year. The F-test can be computed as follows:

$$F = \frac{S_x^2}{S_y^2} \tag{5}$$

Values closer to 1 indicate similar variances, suggesting that both samples belong to the same distribution [Hahs-Vaughn and Lomax 2020]. In our example, the F-test applied to "$qtde\_tablet$" and "$qt\_salas\_utilizadas$" columns yields a value of 14.66 and a p-value of 0.0004, indicating a statistically significant difference in their distributions.

Welch's t-test: The Welch's t-test is a variation of the Student's t-test, adapted to cases where the samples have unequal variances and sample sizes. The test is similar to the F-test since it compares the variances. But unlike the F-test, this test considers the averaged values and sizes of the distributions. Its value is computed as follows:

$$t = \frac{\overline{x} - \overline{y}}{\sqrt{\frac{S_x^2}{m} - \frac{S_y^2}{n}}} \tag{6}$$

This variant of the t-test relaxes the assumption of equal variances between the two samples [Rayner et al. 2009]. In our example, the attributes "$qt\_salas\_utilizadas$" and "$num\_salas\_utilizadas$" in Figure 1 demonstrate similar distributions, as evidenced by the Welch's t-test statistic of 0.069 and a high p-value of 0.95, indicating no significant difference in their distributions.

3.2. The Schema Matching Algorithm

Algorithm 1 is designed to identify changes in data columns across different years. It compares columns from later years with the "base schema" (lines 2-5). The "base schema" represents the operational schema following a successful evolution and data integration. When the p-value from a comparison exceeds a specified threshold (line 7), the algorithm

flags the column from the later year as a potential match and selects it as the best candidate for mapping to the corresponding base year column (lines 8-10).

Most importantly, the algorithm enables the classification of data columns into three categories to guide integration decisions: identical columns, new columns, and columns without data. Identical columns exhibit consistent data across different years. New columns are introduced in the following year with no corresponding column in the base year. Columns without data are present in the base schema but absent in the subsequent year.

The comparison of column matches falls under the broader domain of data profiling, which involves analyzing columns [Abedjan et al. 2015, Pena et al. 2021]. In data profiling, the number of potential column comparisons can grow exponentially with the number of attributes in a relation. While our algorithm inherits this complexity, it focuses on the specific task of comparing two columns, resulting in a worst-case scenario of quadratic complexity when dealing with identical schemas.

Algorithm 1: COLUMNMATCH($curr, new, gd, threshold$).

Input: $curr$: the current database columns; new: the new columns that arrived that must be matched; gd: goodness-of-fit method to be used; p_thresh: minimum p_value to accept the column as a possible match.

Result: A map matching the columns in new with the columns in $curr$.

1   $matches = empty\ set$
2   **for** c_column in $curr$ **do**
3     $chosen\_col = NULL$
4     $metric = NULL$
5   **for** n_column in new **do**
6       $n\_metric, p\_value = gd(c\_column, n\_column)$
7   **if** $p\_value \geq p\_thresh$ **then**
8   **if** *(chosen_col is $NULL$ or n_metric is better than metric)* **then**
9           $metric = n\_metric$
10           $chosen\_col = n\_column$

    // the tuple (c_column, n_column) is a match between the current and the arrived column
11     $matches = matches \cup (c\_column, n\_column)$
12   remove $chosen\_col$ from the set new

  // Mark the remaining as no match (new columns)
13   **for** n_column in new **do**
14     $matches = matches \cup (NULL, n\_column)$
15   **return** matches

---

4. Experimental Results

In this section, we delve into the experiments we conducted using statistical techniques on data retrieved from the LDE database. We describe the experimental protocol we followed and report the results obtained from the evaluation process.[4]

---

[4]In this link, we provide access to data and the complete source code of the LDE system: https://dadoseducacionais.c3sl.ufpr.br/ (in Portuguese).

4.1. Experimental Protocol

We used the R implementation for the Goodness-of-fit methods used in this work described in Section 3.1. Some goodness-of-fit approaches may be sensible to outliers (e.g., the variance-based approaches, such as the F-Test), necessitating their removal during data preprocessing. We employed the Interquartile Range (IQR) method for outlier detection by finding the first (Q1) and third (Q3) quartiles, representing 25% and 75%, respectively. The IQR is the difference between Q1 and Q3. We identify and remove outliers as values falling below Q1 subtracted by 1.5 times the IQR, or above Q3 added by 1.5 times the IQR.

Instead of feeding the columns data directly to the Algorithm 1, which could lead to problems when estimating the $p - value$ (indicating the confidence of a given column to be a correct match), we first transform the data of the columns in a *10-bins* histogram. We defined $p\_thresh = 0.9$ (i.e., $\alpha = 0.1$) in Algorithm 1, which is a common practice when accepting/rejecting the NULL hypothesis (i.e., the columns come from the same distribution).

4.2. Results – Matches Considering the Previous Year

In Table 1, we show the results, considering the accuracy of the Top 1. We define a successful match as an exact match between a column and its corresponding column from the previous year. The accuracy is defined as $acc = \frac{hits}{\#columns}$. In Table 1, column *Year* defines the reference year, which we need to match the columns with the previous year. The column *Changes* shows $[x]+$ as the number of new, $[y]-$ as the number of removed, and $[z]c$ as the number of changed columns when compared with the previous year (considering the ground-truth).

For example, consider 2019 in Table 1. This year, when analyzing the official data made available from the MEC and comparing it with the previous year (2018), no columns changed their names; 17 new columns appeared, presenting data that was not collected in the previous year, and eight columns disappeared, presenting data that was no longer collected.

As one can observe in Table 1, the Kolmogorov–Smirnov (K-S) test presented the best results when considering the averaged results, correctly fitting the columns in over 80% of the cases. The results in Table 1 also show that even in the event of many changes occurring in a given year, such as in the year 2019, where 19 new columns appeared and 2 columns disappeared, all tested approaches were able to correctly detect most of the correct fits, columns that should be created (new columns), and columns that should be disregarded (discontinued ones).

During the tests, we observed an interesting phenomenon: even when the column is not perfectly fitted with the previous year, the correct fit still presented a high probability (according to each test) of belonging to its proper fit. In light of this, we make a Top 3 analysis in Table 2. The Top 3 consider a hit if the predicted fit appears in the three most probable fits for a given approach. The Top 3 can present a more realistic scenario than the Top 1 since it can show the most probable fits to a specialist, who will choose the correct one according to his/her domain knowledge.

The Top 3 results in Table 2 show that both the Kolmogorov–Smirnov and Anderson–Darling tests correctly fit the columns in almost 90% of the cases considering the

505

Year	Changes	K–S Test	A-D Test	Welch's Test	F-Test
2007	-	-	-	-	-
2008	*no change*	0.667	0.333	0.5	0.333
2009	*no change*	1.0	0.333	0.333	0.0
2010	*no change*	0.833	0.667	0.333	0.333
2011	*no change*	0.333	0.667	0.667	0.167
2012	*no change*	1.0	0.667	0.667	0.333
2013	[0]c [7]+ [0]−	0.846	0.692	0.692	0.769
2014	*no change*	0.923	0.769	0.462	0.231
2015	[14]c [0]+ [0]−	1.0	0.846	0.538	0.308
2016	*no change*	1.0	0.846	0.385	0.462
2017	*no change*	0.615	0.538	0.462	0.231
2018	[7]c [0]+ [0]−	1.0	0.769	0.692	0.846
2019	[0]c [17]+ [8]−	0.824	0.882	0.706	0.706
2020	*no change*	1.0	1.0	0.636	0.591
2021	*no change*	0.818	0.727	0.364	0.273
Average (stdev)		0.847 (0.189)	0.701 (0.185)	0.531 (0.139)	0.393 (0.228)

Table 1. Accuracy Considering the Top 1 results.

Top 3 results. This result shows that these goodness-of-fit tests used in combination with our proposed Algorithm (Algorithm 1) can significantly decrease the manual work of the domain's specialist, who will find the correct match in the first proposals of the algorithm instead of needing to find the proper fit considering all possible columns available for a given year.

4.3. Results – Matches Considering the Accumulated Years

In this Section, we follow the same protocol as in Section 4.2, with the difference that when trying to match the nth reference year, all the data from the first year to the $nth − 1$ year are used to create the distribution to be compared. For instance, when trying to match the reference year of 2010, the data distribution from 2010 is compared with the distributions of the years 2007, 2008, and 2009 combined. To make this possible, we first normalize the histograms by dividing each bin by the number of tuples used to create the histogram.

The concept of this test is that by grouping more data to compare, we could get closer to the real underlying distribution of the data. The Top 1 results of this test are shown in Table 3.

As one can observe in Table 3, besides the good results of the Kolmogorov–Smirnov and Anderson–Darling tests, there was a decrease in the accuracy for all the tested approaches when compared with the results when the columns were matched considering only the previous year (Section 4.2)[5]. We hypothesize that external factors, such as data collection policy changes over time, made it more difficult to make a correct match when including the data from previous years in the distribution for comparison.

---

[5]We identified a similar behavior in the Top 3 results, which are not included in this work due to length restrictions.

Year	Changes	K–S Test	A-D Test	Welch's Test	F-Test
2007	-	-	-	-	-
2008	*no change*	1.0	0.833	0.5	0.667
2009	*no change*	1.0	1.0	0.5	0.0
2010	*no change*	0.833	1.0	0.667	0.333
2011	*no change*	1.0	1.03	0.833	0.33
2012	*no change*	1.0	1.0	0.667	0.5
2013	[0]c [7]+ [0]−	0.846	0.692	0.462	0.538
2014	*no change*	0.846	0.846	0.846	0.462
2015	[14]c [0]+ [0]−	0.923	0.923	0.846	0.538
2016	*no change*	0.923	0.923	0.769	0.615
2017	*no change*	0.769	0.846	0.692	0.385
2018	[7]c [0]+ [0]−	1.0	1.0	0.846	0.923
2019	[0]c [17]+ [8]−	0.824	0.765	0.706	0.647
2020	*no change*	0.818	0.818	0.591	0.727
2021	*no change*	0.773	0.773	0.727	0.591
Average (stdev)		0.897 (0.087)	0.887 (0.101)	0.689 (0.13)	0.519 (0.21)

Table 2. Accuracy Considering the Top 3 results.

Year	Changes	K–S Test	A-D Test	Welch's Test	F-Test
2007	-	-	-	-	-
2008	*no change*	1.0	0.333	1.0	0.167
2009	*no change*	0.667	0.667	0.333	0.0
2010	*no change*	0.667	1.0	0.167	0.0
2011	*no change*	0.667	1.0	0.167	0.0
2012	*no change*	1.0	1.0	0.167	0.0
2013	[0]c [7]+ [0]−	1.0	1.0	0.615	0.462
2014	*no change*	0.923	0.692	0.231	0.077
2015	[14]c [1]+ [0]−	0.923	0.923	0.0	0.154
2016	*no change*	0.692	0.692	0.154	0.154
2017	*no change*	1.0	0.846	0.769	0.231
2018	[7]c [0]+ [0]−	1.0	0.846	0.846	0.769
2019	[0]c [17]+ [8]−	0.348	0.348	0.304	0.609
2020	*no change*	0.267	0.267	0.1	0.533
2021	*no change*	0.267	0.2	0.067	0.6
Average (stdev)		0.744 (0.271)	0.701 (0.287)	0.351 (0.309)	0.268 (0.26)

Table 3. Accuracy Considering the Top 1 accumulated results.

507

The Kolmogorov-Smirnov test corroborates this hypothesis since the results got worse for the latest years when compared with Section 4.2, since in the latest years, there were more old data aggregated in the columns to make the comparison.

5. Conclusion and Future Work

In this paper, we presented a methodology for identifying matches between attributes of the census datasets from a given year and their possible matches in a dataset released in a subsequent year. We hypothesized that using statistical tests to evaluate the evolution of the LDE database schema enhances the reliability of column matching. Indeed, our methodology finds the matching attributes and shows what the changes are, such as adding new data or data columns not present in the year evaluated. Results showed that our approach, combined with the Kolmogorov–Smirnov test, significantly reduced the manual effort required by specialists. In our methodology, specialists can identify the correct matches rather than having to evaluate all possible columns available for a given year.

In future work, we plan to evaluate attributes that have binary and categorical data. These data require the application of different statistical tests than those used for numerical data. Besides, we also plan to automate the analysis of schema quality as necessary by evaluating column matching. At this point, we intend to study the use of machine learning to facilitate the integration of databases and manage schema evolution.

References

Abedjan, Z., Golab, L., and Naumann, F. (2015). Profiling relational data: a survey. *VLDB J.*, 24(4):557–581.

Alves, T., Silveira, A. A. D., Schneider, G., and Fabro, M. D. D. (2019). Financiamento da escola pública de educação básica: a proposta do simulador de custo-aluno qualidade. *Educação E Sociedade (in Portuguese)*, 40.

Anderson, T. W. and Darling, D. A. (1952). Asymptotic Theory of Certain "Goodness of Fit" Criteria Based on Stochastic Processes. *The Annals of Mathematical Statistics*, 23(2):193 – 212.

Berger, V. W. and Zhou, Y. (2014). Kolmogorov–smirnov test: Overview. *Wiley statsref: Statistics reference online*.

Cerqueus, T., de Almeida, E. C., and Scherzinger, S. (2015a). Safely managing data variety in big data software development. In *1st IEEE/ACM BIGDSE*, pages 4–10.

Cerqueus, T., Scherzinger, S., and de Almeida, E. C. (2015b). Controvol: Let yesterday's data catch up with today's application code. In *WWW Companion*, pages 15–16.

Curino, C., Moon, H. J., Deutsch, A., and Zaniolo, C. (2013). Automating the database schema evolution process. *VLDB J.*, 22(1):73–98.

Curino, C., Moon, H. J., and Zaniolo, C. (2009). Automating database schema evolution in information system upgrades. In *2nd ACM HotSWUp 2009*.

D'Agostino, R. (1986). *Goodness-of-Fit-Techniques*. Statistics: A Series of Textbooks and Monographs. Taylor & Francis.

Delplanque, J., Etien, A., Anquetil, N., and Ducasse, S. (2020). Recommendations for evolving relational databases. In *CAiSE 2020*, pages 498–514.

Etien, A. and Anquetil, N. (2024). Automatic recommendations for evolving relational databases schema. *arXiv preprint arXiv:2404.08525*.

Garcia-Molina, H., Ullman, J. D., and Widom, J. (2009). *Database systems - the complete book (2. ed.)*. Pearson Education.

Hahs-Vaughn, D. and Lomax, R. (2020). *Statistical Concepts: A Second Course*. Routledge.

Klettke, M., Awolin, H., Störl, U., Müller, D., and Scherzinger, S. (2017). Uncovering the evolution history of data lakes. In *IEEE BigData 2017*, pages 2462–2471.

Meurice, L. and Cleve, A. (2017). Supporting schema evolution in schema-less nosql data stores. In *24th IEEE SANER*, pages 457–461.

Pena, E. H. M., de Almeida, E. C., and Naumann, F. (2021). Fast detection of denial constraint violations. *Proc. VLDB Endow.*, 15(4):859–871.

Pettitt, A. N. (1976). A two-sample anderson-darling rank statistic. *Biometrika*, 63(1):161–168.

Qiu, D., Li, B., and Su, Z. (2013). An empirical analysis of the co-evolution of schema and code in database applications. In *ACM SIGSOFT*, page 125–135.

Rayner, J., Thas, O., and Best, D. (2009). *Smooth Tests of Goodness of Fit: Using R*. Wiley series in probability and statistics Smooth tests of goodness of fit using R. Wiley.

Ringlstetter, A., Scherzinger, S., and Bissyandé, T. F. (2016). Data model evolution using object-nosql mappers: folklore or state-of-the-art? In *2nd IEEE/ACM BIGDSE*, page 33–36.

Scherzinger, S., de Almeida, E. C., Cerqueus, T., de Almeida, L. B., and Holanda, P. (2016). Finding and fixing type mismatches in the evolution of object-nosql mappings. In *EDBT/ICDT Workshops*, volume 1558.

Scherzinger, S. and Sidortschuck, S. (2020). An empirical study on the design and evolution of nosql database schemas. In *Conceptual Modeling*, pages 441–455. Springer Inter. Publishing.

Schneider, G., Gallotti Frantz, M., and Alves, T. (2023). Infraestrutura das escolas públicas no brasil: desigualdades e desafios para o financiamento da educação básica. *Revista Educação Básica em Foco (in Portuguese)*, 17(2).

Schneider, G., Silveira, A. A., and Alves, T. (2020). Mapeamento da formação de docentes no paraná: um olhar para o indicador de adequação. *Jornal de Políticas Educacionais (in Portuguese)*, 1(3).

Silveira, A. D., Schneider, G., and Alves, T. (2021). *Simulador de Custo-Aluno Qualidade (SimCAQ): Trajetória e Potencialidades*. Inep/MEC (in Portuguese).

Vassiliadis, P., Zarras, A. V., and Skoulis, I. (2015). How is life for a table in an evolving relational schema? birth, death and everything in between. In *Conceptual Modeling*, pages 453–466.