Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação
Projeto de Final de Curso

ESCOLA POLITÉCNICA
DE PERNAMBUCO

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 19 de julho de 2018, às 12:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente **OTO COSTA PINHO ALVES**, orientado pelo professor **Fernando Buarque de Lima Neto**, sob título **Feature Engineering for Clustering of Asset Compatibility Analysis Data**, a banca composta pelos professores:

Fernando Buarque de Lima Neto
Caio César Davi

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

☒ Aprovada ☐ Aprovada com Restrições* ☐ Reprovada

e foi-lhe atribuída nota: 9,5 (nove e meio)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 7 dias para entrega da versão final da monografia a contar da data deste documento.

FERNANDO BUARQUE DE LIMA NETO

CAIO CÉSAR DAVI

* Este documento deverá ser encadernado juntamente com a monografia em versão final.

Feature Engineering for Clustering of Asset Compatibility Analysis Data

Oto C. P. Alves, Fernando B. de Lima-Neto
Polytechnic School of Pernambuco - POLI
University of Pernambuco - UPE
Recife, Brazil
Email: {ocpa,fbln}@ecomp.poli.br

***Abstract*—Government auditors responsible for monitoring illegal activities are often faced with a deluge of data. Filtering these data in order to find the relevant information is challenging even for a specialist, so the field of Knowledge Discovery provides an important toolset for support in this endeavor. In this work, we apply Knowledge Discovery techniques to a real world case within local branches of the Brazilian Government, focusing on the Preprocessing and Transformation phases of the Knowledge Discovery in Databases process. Specifically, we conduct a process of Feature Engineering with the aim of improving the performance of Hierarchical Agglomerative Clustering in the context of the Asset Compatibility Analysis problem. We compare three methods, namely, Principal Component Analysis, Independent Component Analysis and Self-Organizing Maps. We show that, within this scenario, Self-Organizing Maps are the best of the three alternatives, providing a significant improvement in the quality of clustering for most levels of the cluster hierarchy.**

Keywords—Feature Engineering, Clustering, Dimensionality Reduction, Self-Organizing Maps, Feature Extraction

I. Introduction

Over the course of the last few decades, the Brazilian Government has devised and implemented a wide array of systems for corruption prevention, detection and mitigation both at municipal, state and federal levels. At all the three levels of government, the role of the *Controladorias* (Comptroller's/Internal Affairs Offices) is of upmost importance. These governmental bodies are in charge of a number of administrative and bureaucratic activities, including, but not limited to, auditing public records and operations in order to seek signs of financial mismanagement, money laundering and irregular activities in general.

From a technical standpoint, the aforementioned monitoring activities can be aided by a broad spectrum of techniques and heuristics, which might in turn feature automation to different degrees. Within the application domain of monitoring illegal activities, both statistical [1] and data-mining-based [2] approaches have established a strong track record of effectiveness over the years.

The foci of this paper are the *Controladoria Geral do Município de Recife* (CGM) and *Secretaria da Controladoria Geral do Estado de Pernambuco* (SCGE), which are the Comptroller-General Offices of the Municipality of Recife and of the State of Pernambuco, respectively. Within these offices, the monitoring activities are still conducted, mostly, in a standard manual fashion. Moreover, they still rely heavily on the possible existence of human-provided hints that can steer the investigation towards areas of interest within the available data, whose volume is already too large for effective manual inspections to be effective.

A. Asset Compatibility Analysis

One of the tasks conducted by CGM/SCGE's auditors is the Asset Compatibility Analysis (ACA) of the public workers within the city's jurisdiction. This process consists in maintaining the workers under constant scrutiny with regards to the compatibility of their income sources and their legally declared assets, even if there's no formal charges being pressed against them. If any incompatibility is spotted, it can be regarded as possible evidence of irregular activities and be used as basis for a formal investigation, which shall then be conducted by the appropriate agencies [3],[4].

Within the context of Brazilian public institutions, ACA has a history of effectiveness. The most noteworthy case is the "*ISS Mafia*", a large-scale corruption scheme uncovered in 2013 by the Comptroller-General Office of the Municipality of São Paulo, the largest city in the country. The initial evidence for this scheme was found through a joint analysis of data from previously separate databases, which indicated asset incompatibilities among several of the city's public workers [5],[6].

ACA is not a trivial process, since there are asset acquisition methods, such as inheritance, which are not immediately evident in the data CGM and SCGE are able to access. As a consequence, ACA is subject to a high likelihood of false-positive errors. Furthermore, the amount of public workers under the jurisdiction of these agencies is much greater than their auditing resources are capable of handling with the currently employed methods. Therefore, it is of upmost importance that automated or semi-automated methods are incorporated in the process of filtering and electing the foci of corruption monitoring and the following investigations.

B. Scope of This Work

With the aforementioned necessities in mind, the aim of this work is to aid CGM and SCGE in the conduction of ACA by carrying out of parts of the Knowledge Discovery in Databases (KDD) [7] process with their data. Specifically, we focus on the Preprocessing and

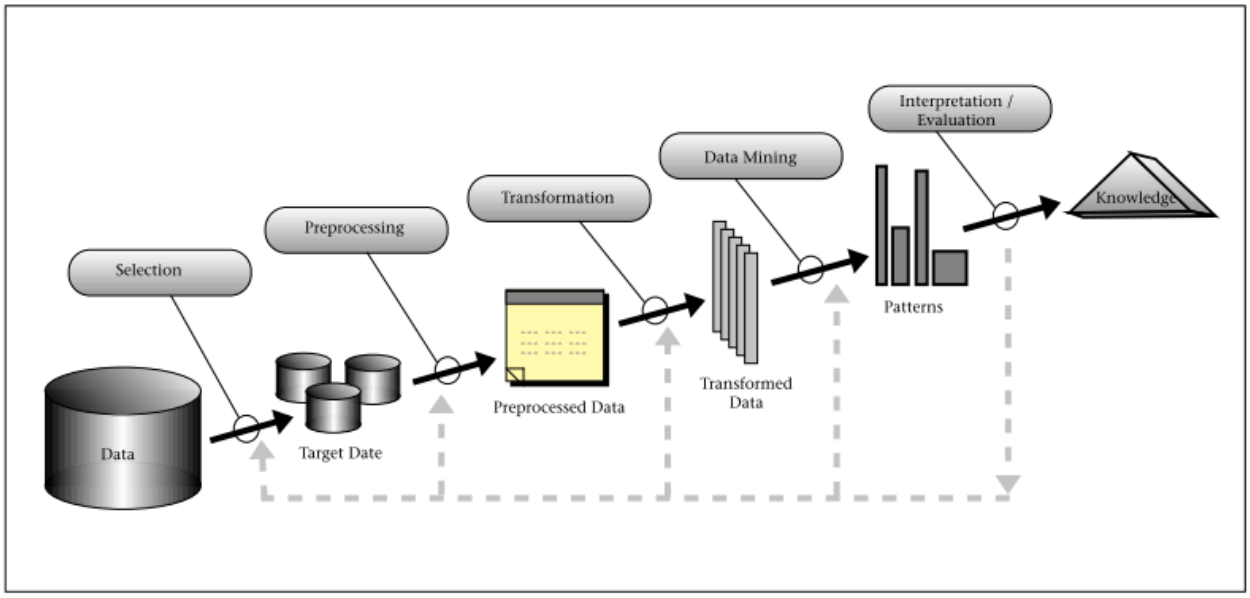

Fig. 1. A diagram describing the KDD Process. Source: [8]

Transformation steps of the KDD process, by conducting Feature Engineering experiments which seek to improve the performance of Machine Learning algorithms applied to CGM/SCGE's datasets.

CMGR's and SCGE's databases contains a broad range of information on all public workers under their jurisdiction, including data on taxes, assets and specific job details regarding each individual. Since the formal investigations that are initiated as a consequence of CGM/SCGE's work are actually conducted by different branches of the government, they have no access to data regarding which of the flagged occurrences were indeed deserving of suspicion. In other words, their datasets have no labels that can aid an algorithm in the task of encoding the traits of demonstrably corrupt individuals.

Due to this limitation, we are restricted to Unsupervised Methods when it comes to incorporating data mining techniques to the current ACA pipeline. Therefore, our Feature Engineering experiment seeks to improve the performance of clustering algorithms applied to CGM/SCGE's databases and, as a consequence, aid auditors in understanding the underlying patterns in the data regarding the subject public workers.

It is worth mentioning that all sensitive or personally-identifying data regarding the individuals in the datasets used in this work were previously obfuscated by CGM via Cryptographic Hash Functions.

II. Knowledge Discovery in Databases

In a seminal paper on the subject, Fayyad et al. have defined the KDD process as "... the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" [7]. KDD is, to some degree, a multidisciplinary endeavor. It involves a mixture of technical data science abilities with domain knowledge within the target application fields. KDD can be divided in five main steps. We shall describe each of them on broad terms, as follows:

A. Phases of the KDD Process

- *Selection:* This step involves selecting the initial dataset(s) which shall be used in the process. Ideally, it also involves the studying and understanding of the application domain.
- *Preprocessing:* During preprocessing, many tasks are conducted in order to clean the data. Missing values, seemingly incorrect readings and outliers are all treated according to the particular requirements of the problem.
- *Transformation:* In this step, the features of the data are re-encoded in a way that the algorithms used in the following phase can work more effectively with them. Several techniques can be used for achieving effective data transformation. Within the context of this work, we shall use three different dimensionality reduction techniques, which are considered part of the Feature Engineering process, which is discussed at length in Section V.
- *Data Mining:* Unlike the previous steps, the focus of Data Mining is no longer the data themselves, but rather the learning of the knowledge they convey. This step might involve different types of problems, including, but not limited to, classification, regression and clustering. The experiment conducted in this work focuses on the latter.
- *Interpretation/Evaluation:* The last step in the KDD process is often subjective and relies heavily on human input from domain specialists. In this step, the knowledge provided by Data Mining has to be validated, interpreted, and contextualized. Data Visualization [9] tools provide significant aid in this process.

III. Unsupervised Learning

Machine Learning Algorithms can be roughly split into three major groups: Supervised Learning, Unsupervised Learning and Reinforcement Learning. Supervised Learning algorithms learn from labeled data and seek to encode relationships between the data's features and the corresponding labels, i.e. a relationship between inputs and outputs. Unsupervised Learning algorithms learn from unlabeled data, and seek to learn and encode the underlying relationships and structures that exist within the data. Reinforcement Learning algorithms aim to learn behavioral

patterns with respect to feedback given by specific external signals [10].

Unsupervised Learning encompasses algorithms that are used for solving several categories of problems, including, but not limited to, cluster analysis, outlier detection and anomaly detection.

In this work, our focus is to improve the performance of clustering algorithms within the domain of ACA. More specifically, we aim to improve the performance of Hierarchical Agglomerative Clustering when applied to a dataset containing information on the public workers under CGM's and SCGE's jurisdiction.

A. Clustering

Clustering can be described as the process of finding groups of similar data samples within a dataset, according to a given metric of similarity. In [11], Jain defines the objective of clustering algorithms as "… to discover the natural grouping(s) of a set of patterns, points, or objects".

As is often the case with Unsupervised Learning techniques, the outputs of clustering algorithms are usually descriptive in nature [10]. Hence, for a large number of applications, there is no obvious error metric. Moreover, for many applications, a human in the loop with domain expertise is still required for interpreting the results of clustering.

B. Hierarchical Clustering

Clustering algorithms whose output is a single set of coexisting clusters, such as K-Means [12], are classified as flat clustering algorithms. In contrast, hierarchical clustering algorithms produce clusters which can be organized in a tree-like structure called a dendrogram, in which each node represents a cluster and the children nodes represent subclusters of the parent node [13].

This type of clustering is especially useful in real-world scenarios where one seeks to find clusters of different granularities, according to the needs presented by the problem. In our specific problem, it is important to identify both the less specific, overarching clusters and the more specific, highly granular subclusters. The former is important because the dataset encompasses a very diverse set of individuals, with different occupation areas and varying levels of income. The latter is important because the auditors are especially interested in identifying small subgroups of individuals that deviate from the norm, since such a deviation could indicate illegal activities.

Hierarchical Clustering is split in two paradigms, namely, Agglomerative Clustering and Divisive Clustering. Agglomerative Clustering is a bottom-up approach where each data point starts as a cluster, and the clusters are progressively merged until the root of the dendrogram is reached. Divisive Clustering performs the procedure in the opposite direction, starting with a single cluster and dividing it gradually until the leaves of the dendrogram are reached. In both approaches, clusters are merged or partitioned with respect to a given similarity measure between their two respective sets of members.

C. Clustering Evaluation Criteria

As mentioned in Section III.A, the task of evaluating the quality of clusters generated by a learning algorithm is not a trivial endeavor. In order to make this analysis less subjective, several quality metrics have been devised over the years.

In general, these metrics can be split into external evaluation criteria and internal evaluation criteria. External evaluation criteria requires the existence of *a priori*, ground-truth information about the data, such as class labels. Their objective is to measure how well-correlated the results of the clustering are with the aforementioned external information. In contrast, internal evaluation criteria work in a purely unsupervised manner, only taking in consideration the features themselves, with no regard to *a priori* information. These criteria usually focus in how homogenous, compact and well-separated from each other the clusters are [14].

In this work, due to the absence of *a priori* information that could be used as a basis for external criteria, we are restricted to the usage of internal criteria.

IV. The Curse of Dimensionality

In a rather counterintuitive twist, many learning algorithms fail to generalize well when applied to data modeled in higher dimensions. As the amount of dimensions in a model increases, concepts like similarity or proximity gradually detach themselves from their intuitive semantics [15]. Much of this effect is due to the way geometry behaves in higher dimensions, twisting the qualitative meaning of traditional distance metrics [16].

Thus, it is difficult for the human brain, which evolved in a three-dimensional setting, to effectively grasp the semantics of Machine Learning models operating in higher dimensions, let alone the outputs of said models [15].

As mentioned in Section III, the output of clustering algorithms is descriptive in nature and, for many practical purposes, subject to human interpretation. Therefore, they are especially affected by this loss of intuitive interpretability. While there is a considerable amount of research on clustering in higher dimensions, most of this field lies beyond the scope of this paper. For a deeper treatment of the subject, we refer the interested reader to [17].

Since the problem at hand involves a large volume of data with high dimensionality, it can benefit significantly from Machine Learning methods. However, it is important to mitigate the undesirable side-effects caused by high dimensionality.

A. Dimensionality Reduction

A common and effective approach to mitigate the effects of the Curse of Dimensionality is a process called Dimensionality Reduction. As the name suggests, it consists in re-structuring the representation of the data in such a way that the information contained within can be encoded in a lower-dimensional space, while also maintaining the loss of information within an acceptable range or even avoiding it altogether.

Dimensionality Reduction does not necessarily imply in a mere removal of dimensions. Techniques such as Principal Component Analysis, Independent Component Analysis and Self-Organizing Maps, which shall be described in Section V, actually replace the pre-existing dimensions with a new set of features derived from the original ones.

V. Feature Engineering

In the context of a knowledge discovery problem, a feature can be any variable, attribute or piece of information present in the dataset's samples that is relevant for the problem at hand. Raw data, be them human-readable or not, are often not expressed in a way that best leverages a learning algorithm's ability to interpret them. By means of removal, creation or alteration of features, one can restructure the data in a way that it is more suitable for usage as input to a learning algorithm. This process is called Feature Engineering, and can be conducted manually, through structured methods or, as is frequently the case, a combination of both [18].

While the concept of Feature Engineering considered an informal topic to a certain extent, it is widely regarded as one of the most important and impactful elements of the KDD process [18]. In his 2012 paper *A Few Useful Things to Know about Machine Learning* [15], Domingos regarded it as one of the most important factors in the success of a machine learning project.

A. Feature Extraction

Feature Extraction techniques are structured methods that re-encode information contained in the data in new feature vectors that describe it in a different format, with an acceptable degree of information loss or even no loss at all. In general, this new format is designed to make the data more meaningful to a certain learning algorithm. Even when there is information loss, Feature Extraction methods are designed to retain the relevant information in the dataset, which manifests itself in different forms for different types of data and applications [18].

In many cases, the features obtained by means of Feature Extraction do not translate well to any human-readable semantics, which renders them unsuitable for data visualization or interpretation by a domain specialist. These features might, however, improve the performance of the learning algorithms we feed the dataset to, both in terms of convergence speed and accuracy of models.

B. Feature Selection

Feature Selection consists in selecting a relevant subset of the already existing features in order to achieve dimensionality reduction. By focusing on the most relevant subset of features, a learning algorithm might achieve gains in both accuracy and convergence speed, while also having results that are more human-readable and clear. Feature Selection is different from Feature Extraction in the aspect that it does not involve the creation of any new features, it merely consists in removing the least relevant ones, according to given criteria [19].

C. Principal Component Analysis

Principal Component Analysis (PCA) is a feature extraction technique that applies a linear transformation to the vectors in a dataset in order to express them in a new feature space with an orthogonal basis and reduced redundancy. This redundancy reduction is achieved by a greedy procedure that iteratively extracts dimensions which maximize the variance of the corresponding components of the vectors in the dataset [20].

PCA is initialized by extracting the feature with the highest possible variance within the dataset. After that, it iteratively extracts the feature with the highest variance among set of possible features, i.e. the ones that correspond to dimensions orthogonal to the ones encoded by the previously extracted features.

The output this procedure is the initial dataset encoded in a set of ordered dimensions called Principal Components, sorted by their variances in decreasing order. The level of variance a Principal Component contains can be understood as its level of importance in the representation of the data.

The set of Principal Components obtained by PCA has the same cardinality as the original set of features. A full set of Principal Components encode a perfect representation of the original data. While there is a loss of accuracy when features are removed from this set, for purposes of dimensionality reduction, it is often useful to retain only the first several Principal Components. While this procedure incurs in loss of information, it might improve the performance of learning algorithms upon the dataset [21].

D. Independent Component Analysis

Independent Component Analysis (ICA) is a feature extraction procedure which, similarly to PCA, aims at extracting features that represent the original vectors through a different basis. Unlike PCA, however, the new basis is not made up of orthogonal vectors. Rather, ICA aims to maximize the statistical independence of the components it produces as output [22].

ICA finds ample usage in the processing of signals generated by multiple independent sources, such as audio recordings with ambient noise [23].

E. Self-Organizing Maps

Self-Organizing Maps (SOM) [24] are a family of unsupervised Artificial Neural Network models introduced by Kohonen [25] in 1982 and improved upon by a myriad of theoretical and practical works.

The structure of a SOM consists in a grid of artificial neurons, usually two-dimensional, in which each neuron has N inputs, where N is the dimensionality of the dataset in use. The weights of a given neuron's inputs make up a weight vector, which in practice can be understood as the positioning of this neuron in the N-dimensional space the dataset resides in.

Intuitively, the training of a SOM can be understood as fitting the grid to a higher-dimensional dataset. The N-dimensional surface obtained by this training can be used as a mapping from vectors in the N-dimensional space to vectors in a finite and discrete two-dimensional space, where each possible pair of coordinates correspond to a neuron from the grid. When mapping an N-dimensional vector $\mathbf{v}$ to the new space, we assign to it the coordinates of the neuron whose weight vector $\mathbf{w}$ is the most similar to $\mathbf{v}$, according to some similarity metric.

The aforementioned mapping between spaces provides a representation of the dataset in a lower dimensionality, while also maintaining to some degree the similarity patterns and structures found in the original representation. Therefore, it can be used as a form of dimensionality reduction.

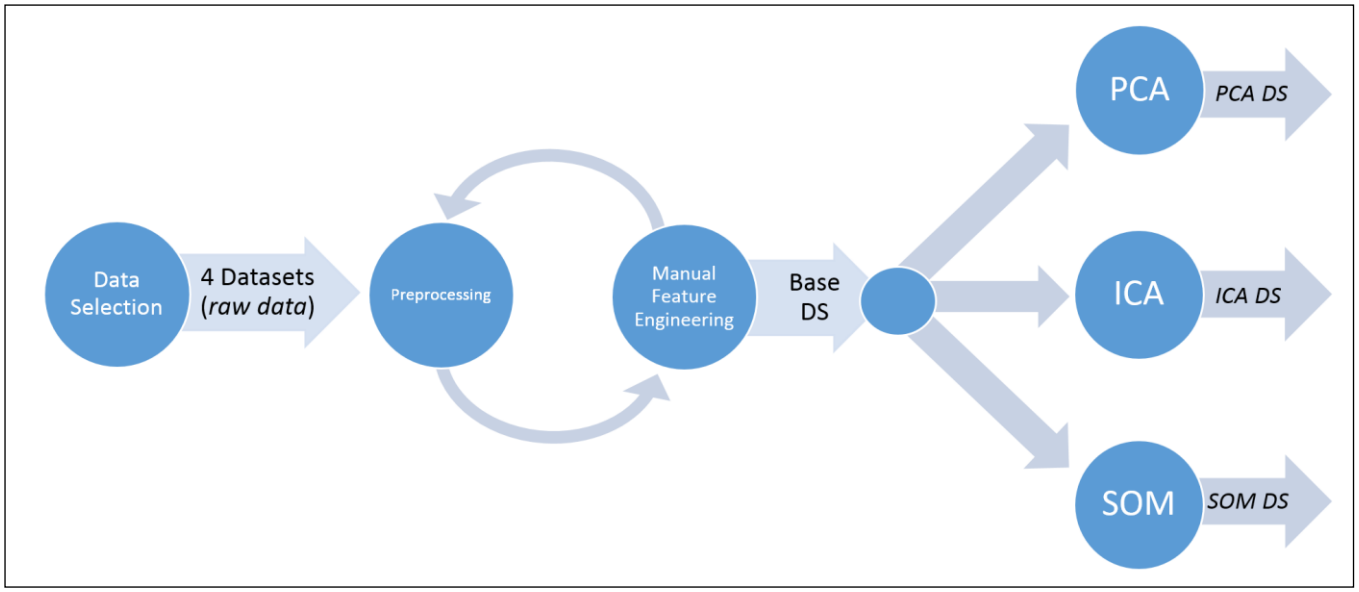

Fig. 2. A flowchart describing the entire Feature Engineering pipeline we employed in sections VI and VII. The outputs of this pipeline are four different representations of the dataset, each one having its own feature space.

VI. Dataset Description and Preprocessing

A. Manual Feature Engineering

Initially, for addressing the ACA problem, we obtained four datasets with samples describing different types of entities, namely, public workers, individual instances of real estate tax debt, individual instances of real estate tax payments, and individual instances of energy bills. In order to create a unified dataset focusing on the public workers themselves, the debts, payments and utility bills were grouped by worker ID in an additive manner.

Furthermore, since some individuals had more than one concurrent job with different time spans, we also coined the **accumulated_wage** feature, which is a sum of the products between the years the individual spent in each specific job and the corresponding yearly wage. It was conceived to encode, in a single feature, a rough metric of the total wealth the individual might have accumulated over the years.

B. Removal of Categorical Features

Multi-valued categorical data such as department name and job titles usually lack a concept of proximity between the possible values. Therefore, they don't work well with clustering algorithms based on distance between data samples. Since this is the case with Hierarchical Clustering, we decided to opt out of using the categorical features in the dataset. While there is a certain level of information loss entailed by this decision, some numerical features are semantically tied to the categorical features removed and can act as rough proxies for them, e.g. an individual's salary being related to their job title.

C. Data Dictionary

After the manual feature engineering operations described in Sections VI.A and VI.B, we obtained a dataset with the features described in Table 1.

TABLE I. Data Dictionary

Feature	*Description*
hash	Number of an identification document, cryptographically hashed by CMGR and SCGE in order to preserve the public workers' privacy
monthly_wage	The current wage the individual receives for all of his public jobs within the city's jurisdiction
tax_paid	Accumulated value of real estate tax paid by an individual over a fixed period of time
retail_value	Accumulated retail value of the individual's real estate properties.
original_debt	Sum of the original values of real estate debts, before interest and fines are applied
total_fines	Total value of accumulated fines on real estate debt
total_interest	Total value of accumulated interest on real estate debt
total_debt	Total value of accumulated real estate debt
kwh_used	Total amount of kWh the individual paid for over a fixed period of time
cip_value	Total amount of money paid for street lighting tax over a fixed period of time
accumulated_wage	The sum of the products between the years the individual spent in each specific job and the corresponding yearly wage

D. Outlier Treatment

Many algorithms are sensitive to outliers, since they might skew the encoding of the patterns in the data. As a consequence, it is a common practice to conduct some form of outlier treatment during the KDD process. However, the objective of this work is ultimately to help CGM and SCGE to find interesting patterns in their financial auditing data. In this application domain, analyzing outliers is desirable, since they have a higher probability of corresponding to individuals involved in illegal activities. For this reason, we refrained from removing outliers from the data.

E. Feature Normalization

Some learning algorithms and feature engineering techniques are susceptible to adverse effects stemming from fluctuating orders of magnitude among the average values of the different features in the space. Therefore, it is important

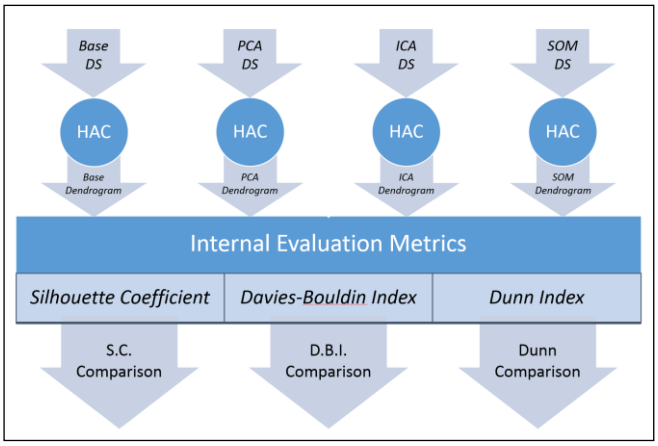

Fig. 3. Flowchart describing the Featuring Engineering evaluation pipeline we employed in order to compare the performance of PCA, ICA and SOM on the dimensionality reduction problem at hand.

to standardize every feature to a single scale before applying any procedure of feature extraction or feature selection. As mentioned in Section VI.D, we maintained the outliers in the dataset. For this reason, we standardized the data by using the Robust Scaler contained in the scikit-learn package [26], which takes in consideration interquartile ranges and is therefore relatively robust to outliers.

VII. Experiments

After the preprocessing described in Section VI, we compared the performance of three dimensionality reduction methods upon the preprocessed dataset. In order to conduct such comparison, we fed the three resulting datasets to the same clustering algorithm and evaluated the clustering performances over each dataset according to three internal evaluation metrics.

The dimensionality reduction techniques we used were PCA, ICA and SOM. The implementations of PCA and ICA we employed were the ones provided by the scikit-learn package [27], while the SOM implementation we used was the one provided by MiniSom [28]. For training the SOM, we opted for batch-learning, since it tends to converge an order of magnitude faster [24].

We employed Hierarchical Agglomerative Clustering (HAC) as the learning algorithm, since, as briefly mentioned Section III.B, clustering at different granularities while maintaining hierarchical consistency is relevant for the auditing process conducted at CGM and SCGE. Since the results of HAC are deterministic for a given dataset, we did not need to perform multiple runs. The clustering evaluation metrics employed here were the Silhouette Coefficient [29], the Davies-Bouldin Index [30] and the Dunn Index [31].

A. Parametrization

By means of empirical experimentation conducted via grid search, the dimensionality reduction methods were parametrized as shown in tables II, III and IV:

TABLE II. PCA PARAMETER VALUES

Parameter	Value
Number of Components	3

TABLE III. ICA PARAMETER VALUES

Parameter	Value
Number of Components	3

TABLE IV. SOM PARAMETER VALUES

Parameter	Value
Length of Grid	250
Width of Grid	250
Shape of Grid	Square
Epochs	7000
Type of Training	Batch
Neighborhood Function	Gaussian
Sigma	1
Initial Learning Rate	0.5

The HAC algorithm was parametrized as follows:

TABLE V. HAC PARAMETER VALUES

Parameter	Value
Distance Metric	Euclidean
Linkage Criterion	Ward

VIII. Discussion of Results and Future Work

The curves in Figure 4 show the performance of four different feature spaces produced by the Feature Engineering techniques we employed. The Base space is the one obtained right after the preprocessing and manual Feature Engineering steps described in Section VI, while the others are the outputs of PCA, ICA and SOM when applied to the Base Space.

Since Cluster Analysis is an exploratory activity, we do not suggest that auditors merely apply a dimensionality reduction algorithm and blindly use the results. Rather, as shown by Figure 4, an auditor might benefit from taking in consideration the advantages of using different spaces for different levels of clustering granularity.

Next, we present the three methods we used to analyze the clustering results regarding their validity.

A. Silhouette Coefficient

The Silhouette Coefficient of a given clustering scheme is bounded by [1,-1]. Values closer to 1 indicate the clusters are more homogenous and clearly separated from other clusters. Therefore, we seek to maximize it.

Let N be the number of clusters in a HAC run. For $N<10$, the PCA and Base spaces far outclass the other two. The PCA space shows a slight performance gain when compared to the Base space, albeit by a very narrow margin. For $10<N<30$, the ICA space has a significant drop in performance, while the other three become roughly equivalent. For $N>30$, the performance of the SOM space remains stable and vastly superior to that of the remainder spaces.

Therefore, for $N<30$, the Silhouette Coefficient suggests that there is little benefit in applying either dimensionality reduction technique to the data. However, for $N>30$, the usage of Self-Organizing Maps improves the results by a far margin.

B. *Davies-Bouldin Index*

A smaller value of the Davies-Bouldin Index indicates a better quality of clustering, thus, we seek to minimize it. For small values of N, the performance of the Base Space is mostly similar or superior to that of other spaces. However, for N>10, SOM quickly becomes the overall best alternative, albeit not by a significant margin. The PCA space shows very little improvement over the Base one, while the ICA space only does it for very small values of N. Therefore, this metric suggests that, unless we intend to analyze the clusters at the top of the hierarchy, using the SOM space is the best alternative.

C. *Dunn Index*

A higher value of the Dunn Index indicates a better quality of clustering, thus, we seek to maximize it. In this metric, the SOM space greatly outperforms the other three, except for very small values of N. Also, for any value of N, this metric indicates that there is little benefit in applying either PCA or ICA to the data. Therefore, the Dunn Index indicates that the SOM is by far the best alternative.

IX. Conclusion

A. *Contribution*

Our results show that, for analyzing the clusters in the top of the hierarchy, there is little benefit in applying dimensionality reduction to the preprocessed dataset. However, for any level of granularity containing more than 10 clusters, the SOM space has yielded superior values in all three evaluation criteria. For more than 30 clusters, SOM is clearly the best alternative in terms of dimensionality reduction.

It should be noted that, for any value of N within the problem at hand, the SOM space has maintained very stable values in both the Silhouette Coefficient and Dunn Index, indicating that it maintains a consistent level of quality for any level of granularity. For simplicity, working only with the SOM space is a viable alternative.

It is also noteworthy that the usage of SOMs has two drawbacks. First, the method is computationally more expensive than PCA and ICA, especially if we use a large grid or high amount of training epochs. There are, however, ways to mitigate that, such as using batch-learning [24] or running the algorithm on GPU hardware [32].

B. *Future Work*

We have demonstrated that Self-Organizing Maps tend to be both an efficient and consistently stable method of dimensionality reduction for the clustering problem at hand. The SOM implementation we used was the standard version of the algorithm. A direct extension of this work could be made by applying other variants of the SOM algorithm, such as Growing Self-Organizing Maps [33], which uses dynamic grids that create new nodes on-demand during the training process.

Another direct extension of this work could be made by using clustering algorithms other than HAC for the evaluation of the feature engineering process. It could also benefit from the usage of biclustering algorithms [34], which conduct clustering while also selecting a custom set of features for each individual cluster, thus taking in consideration the specific semantics and significance of each feature for different types of individuals.

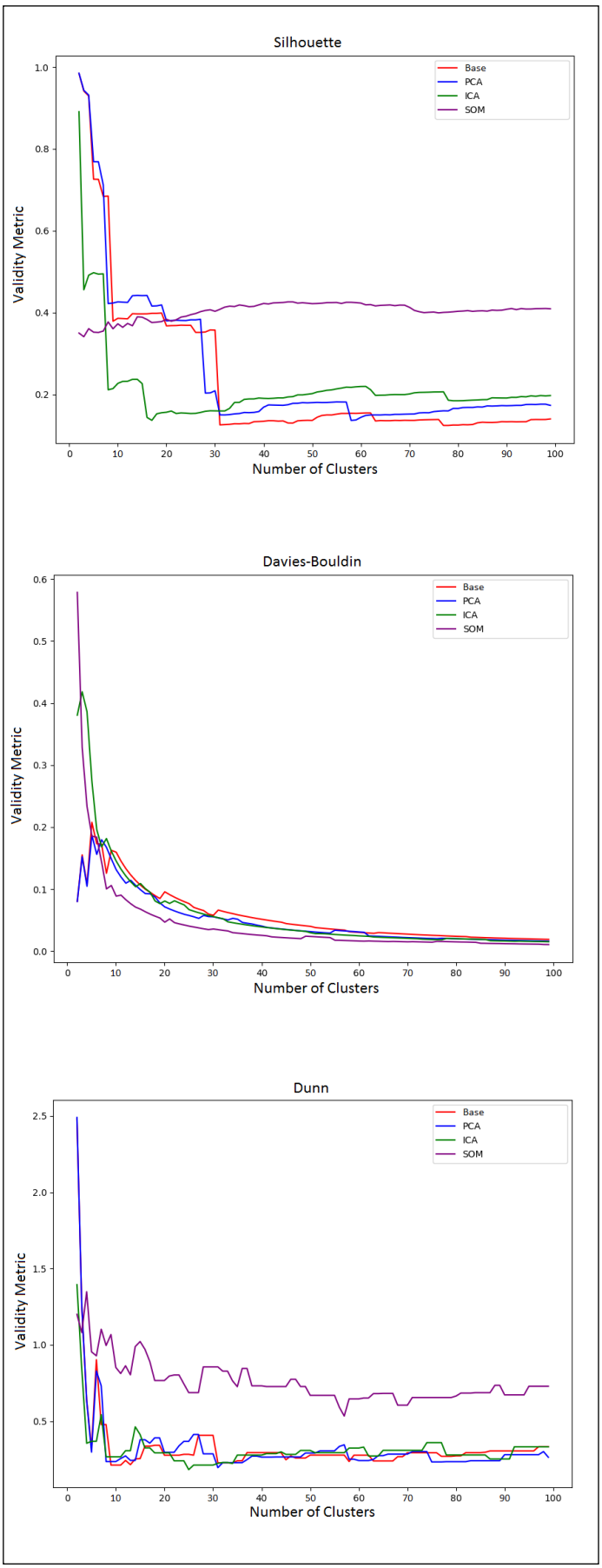

Fig. 4. Evaluation of the dendrogram produced by HAC over the four different feature spaces, according to three different clustering evaluation metrics

Moreover, the breadth of available Feature Engineering and Clustering Evaluation techniques can be significantly enlarged by the acquisition of ground-truth information for a reasonably large amount of samples. Such information could be provided by, for instance, labels that indicate if an individual has been found guilty of any irregular activities by formal investigations. It should be noted, however, that it is not advisable to use class labels as the only source of clustering evaluation, and that finding patterns not related to existing labels might be desirable [35].

Provided there is access to enough ground-truth information, future work might also benefit from the usage of semi-supervised learning algorithms [36] in the knowledge discovery process.

Lastly, one might also seek to conduct similar Feature Engineering experiments on auditing datasets from other branches of the government. Since additional data yields additional information, it would also be in the best interest of the auditors to run the same experiment with future versions of CGM/SCGE's dataset.

REFERENCES

[1] R. J. Bolton and D. J. Hand, "Statistical Fraud Detection: A Review," *Stat. Sci.*, vol. 17, no. 3, pp. 235–249, 2002.

[2] F. Bonchi, F. Giannotti, G. Mainetto, and D. Pedreschi, "A Classification-based Methodology for Planning Audit Strategies in Fraud Detection," in *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 175–184.

[3] Controladoria-Geral da União, *Manual de Processo Administrativo Disciplinar*. [Online]. Available: https://www.cgu.gov.br/Publicacoes/atividade-disciplinar/arquivos/manual-pad.pdf. [Accessed: 24-Apr-2018].

[4] B.F. Cabral and D.D.D. Cangussu, "Os caminhos da sindicância patrimonial para apurar indícios de enriquecimento ilícito de agentes públicos," *jus.com.br*, Sep-2011. [Online]. Available: https://jus.com.br/artigos/19954/os-caminhos-da-sindicancia-patrimonial-para-apurar-indicios-de-enriquecimento-ilicito-de-agentes-publicos/1. [Accessed: 24-Apr-2018].

[5] F. Pereira, "Em dois anos, Controladoria Geral do Município garante recuperação de R$ 270 milhões," *Prefeitura de São Paulo*, 05-May-2015. [Online]. Available: http://www.capital.sp.gov.br/noticia/em-dois-anos-controladoria-geral-do-municipio. [Accessed: 24-Apr-2018].

[6] Controladoria Geral do Município de São Paulo, *Controladoria em Casos: experiências inovadoras para o combate à corrupção e a promoção da integridade na cidade de São Paulo*. [Online]. Available: http://www.prefeitura.sp.gov.br/cidade/secretarias/upload/controladoria_geral/arquivos/CC_Final2.pdf. [Accessed: 31-May-2018].

[7] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The KDD process for extracting useful knowledge from volumes of data," *Commun. ACM*, vol. 39, no. 11, pp. 27–34, 1996.

[8] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Mag.*, pp. 37–54, 1996.

[9] U. M. Fayyad, A. Wierse, and G. G. Grinstein, *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, 2002.

[10] K. P. Murphy, Machine Learning: A Probabilistic Perspective. The MIT Press, 2012.

[11] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[12] J. Macqueen, "Some methods for classification and analysis of multivariate observations," *Proc. Fifth Berkeley Symp. Math. Stat. Probab.*, vol. 1, no. 233, pp. 281–297, 1967.

[13] F. Murtagh, "A Survey of Recent Advances in Hierarchical Clustering Algorithms," *Comput. J.*, vol. 26, no. 4, pp. 354–359, 1983.

[14] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988.

[15] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, p. 78, 2012.

[16] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the Surprising Behavior of Distance Metrics in High Dimensional Space," pp. 420–434, 2001.

[17] H.-P. Kriegel, P. Kröger, and A. Zimek, "Clustering high-dimensional data," *ACM Trans. Knowl. Discov. Data*, vol. 3, no. 1, pp. 1–58, 2009.

[18] J. Brownlee, " Discover Feature Engineering, How to Engineer Features and How to Get Good at It," *Machine Learning Mastery*, 26-Sep-2014. [Online]. Available: https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/. [Accessed: 24-Apr-2018].

[19] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1157–1182, 2003.

[20] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 4, pp. 433–459, 2010.

[21] J. Shlens, "A Tutorial on Principal Component Analysis," 2014.

[22] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, no. 4–5, pp. 411–430, 2000.

[23] J. Shlens, "A Tutorial on Independent Component Analysis," 2014.

[24] T. Kohonen, "Essentials of the self-organizing map," *Neural Networks*, vol. 37, pp. 52–65, 2013.

[25] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982.

[26] scikit-learn developers, "Compare the effect of different scalers on data with outliers," *scikit-learn*. [Online]. Available: http://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html. [Accessed: 25-Jun-2018].

[27] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[28] G. Vettigli, "Minisom: minimalistic and numpy based implementation of the self organizing maps," http://github.com/JustGlowing/minisom, 2013. [Online]. Available: http://github.com/JustGlowing/minisom. [Accessed: 20-Jun-2018].

[29] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. C, pp. 53–65, 1987.

[30] D. L. Davies and D. W. Bouldin, "A Cluster Separation Measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[31] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, 1973.

[32] S. McConnell, R. Sturgeon, G. Henry, A. Mayne, and R. Hurley, "Scalability of self-organizing maps on a GPU cluster using OpenCL and CUDA," *J. Phys. Conf. Ser.*, vol. 341, no. 1, 2012.

[33] D. Alahakoon, S. K. Halgamuge, and B. Srinivasan, "Dynamic Self-Organizing Maps with Controlled Growth for Knowledge Discovery," *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 601–14, 2000.

[34] S. C. Madeira and A. L. Oliveira, "Biclustering algorithms for biological data analysis: a survey.," *Trans. Comput. Biol. Bioinforma.*, vol. 1, no. 1, pp. 24–45, 2004.

[35] I. Färber et al., "On Using Class-Labels in Evaluation of Clusterings," Proc. 1st Int. Work. Discov. Summ. Using Mult. Clust. (MultiClust 2010) conjunction with 16th ACM SIGKDD Conf. Knowl. Discov. Data Min. (KDD 2010), Washington, DC, USA, p. 9, 2010.

[36] X. Zhu, "Semi-Supervised Learning Literature Survey," 2005.