



HAL
open science

Operationalization of a Glass Box Through Visualization: Applied to a Data Driven Profiling Approach

Niels Netten, Arjen Suijker, Mortaza S. Bargh, Sunil Choenni

► **To cite this version:**

Niels Netten, Arjen Suijker, Mortaza S. Bargh, Sunil Choenni. Operationalization of a Glass Box Through Visualization: Applied to a Data Driven Profiling Approach. 20th Conference on e-Business, e-Services and e-Society (I3E), Sep 2021, Galway, Ireland. pp.292-304, 10.1007/978-3-030-85447-8_26 . hal-03648126

HAL Id: hal-03648126

<https://inria.hal.science/hal-03648126>

Submitted on 21 Apr 2022

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.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Operationalization of a glass box through visualization: applied to a data driven profiling approach

Niels Netten¹, Arjen Suijker¹, Mortaza S. Bargh¹ and Sunil Choenni¹

¹ Creating 010, Rotterdam University of Applied Sciences, 3011WN, Rotterdam, Netherlands
{c.p.m.netten; a.j.suijker; m.shoae.bargh; r.choenni}@hr.nl

Abstract. The profiles from data-driven profiling applications are a model of the reality. The interpretability of these profiles for end users, e.g. policymakers, is often far from trivial. How and why these models are obtained by the applications are often regarded as a black box. In recent years several profiling applications used by public organizations have led to wrong interpretations of the obtained models and impacted individuals and society adversely. Hence, the research focus has increasingly shifted towards dealing with the trust and interpretability issues of the models. In support of a more careful and proper interpretation of these models, several scholars have advocated a glass box approach that aims at making these models more transparent to end users.

In this paper, we operationalize the glass box approach for a Genetic Algorithm (GA) based profiling application. To enhance the interpretability of the models provided by the application, we aim at facilitating the interaction of domain experts with the models. Hereby domain experts can gain insight to the evolution of the profiles and what happens to the profiles if we change or add a new pieces of information. Adding such an interactive visualization provides more transparency about the derived models, making them more understandable for end users and policymakers. As a result, they can better assess and explain the consequences of those models when they apply to practice.

Keywords: Algorithms, Glass box, Transparency, AI, Profiling

1 Introduction

Nowadays, many organizations in the private and public sector are searching for ways to take advantage of the explosive growth of (big) data. Especially, in the public domain we see a growing urge to apply data-driven analytics. In recent years, we have seen that data-driven analytics is used to examine large datasets for profiling purposes such as fraud detection [1, 2] and predictive policing [3-5]. On the one hand, these new applications of profiling have opened up many new opportunities for organizations to analyze and predict the behavior of people. On the other hand, in the context of government services the outcome profiles of these new applications are sensitive and a matter of public debate, because they can (adversely) impact individuals (see for example [6] and the references there-in [7]). If the profile models are wrongly interpreted and applied to practice, they may lead to decisions that seem mathematically optimal, but are far from

being correct, just and fair. Hence, it is important to take care of a proper interpretation of profile models and apply them appropriately to practice. To this end, it is advocated to adopt a glass box approach [8, 9]. In such a glass box approach, the relevant concepts and their relationships within an advanced application are explained to end users in a meaningfully transparent way. The goal of a glass box approach is to facilitate a proper interpretation of the obtained models so that end users can assess the consequences of the actions taken based on those models.

In this paper, we operationalize the glass box approach for the GA-based profiling tool as reported in [2]. The tool is developed to support the law enforcement staff at Rotterdam municipality in investigating those individuals who unlawfully misuse the municipality's social benefits system. Despite the reliable performance of GA-based profiles, optimizing, interpreting, and applying such profiles to practice remain challenging for domain experts and policymakers [10]. Domain experts usually have a basic understanding of statistics, but often lack deep knowledge about machine learning in general and GAs in particular. This makes it hard for them to scrutinize the resulting profiles. One should realize that deriving and applying group profiles is also subject to further validations based on privacy and ethical laws and constraints. To aid the validation of the resulting profiles, transparency of the used model is essential for experts and policymakers who want to apply the profiles to practice. Although the human-readable profiles developed by [2] add a lot of transparency to the results of the algorithm, they do not explain what goes on inside the GA-based profiling tool. When evaluating the human-readable profiles resulting from the GA-based tool, two major questions were put forward: how does the algorithm come up with these specific profiles? and could users steer the algorithm towards a preferred set of profiles? To address these questions, visualization of the GA algorithm for profiling seems a promising approach.

In this contribution, we concentrate on visualization as a method to show the adaptive internal search process and strategy of the developed GA-based profiling tool to find relevant profiles in the data. The visualization functions are implemented as a graphical user interface that allows users to view the creation and evolution of profiles, interpret the influence of separate profile attributes, and steer the search of the algorithm while it is running.

2 A Strategy for a responsible use of AI models

An increasing trend for users who apply complicated AI models to practice is to focus more on a responsible use of these models and enhance the trust of end-users in such models. A solution direction for a responsible use of the AI models is to provide more model transparency to end-users as we explain in Subsection 2.1. Subsequently we present a short review of the related works in Subsection 2.2.

2.1 Moving from a black to a glass box

Black boxes are systems that hide their internal logic to the user [11-13]. According to the black box approach, data sets and some constraints are fed to the black box and only

the outcomes of the black box are observed. In case that we are not satisfied with the outcomes, some constraints and data may be altered and again fed to the black box. This process may be repeated until the outcomes are satisfactory based on some criteria. In another approach, referred to as the open box, the implementation of the concepts and their relationships are fully documented so that it can be tracked how the outcomes are obtained exactly. In case that the outcomes obtained are not satisfactory, it is possible to find out which concepts and relationships contribute to this dissatisfaction. These concepts and relationships can be adapted accordingly.

We propose to take, however, a view which is in between these two extreme views, referred to as the glass box [8, 9, 14]. In this view, it is not necessary to know precisely all the ins and outs of the implementation of the used concepts and their relationships. Instead, the crucial and relevant concepts and their relationships between them are explained to end users in a meaningfully transparent way.

The goal of a glass box approach is to facilitate a proper interpretation of obtained models in such a way that end users (e.g. policymakers) can assess the consequences of the actions based on those models within their application domain. The transparency of why and how these models are obtained is an important means to realize this goal. Two crucial questions in the context of a glass box are: (1) what concepts and relationships should be made transparent? and (2) what level of transparency is desired? Subsequently the third question would be how to best convey transparency to users. Unfortunately, there are no straightforward answers to these questions as they depend on the application, i.e. the usage scenario, at hand. We distinguish between two extremes: transparency of algorithms and transparency of data.

In the first situation, the focus is on the transparency of the algorithms, while the transparency of the other components can fully be neglected. As an example of a situation in which this is recommended, consider a neural network based application that is trained for recognizing faces of apes (i.e., the possible output classes are apes). One might be disappointed if he/she offers his/her own picture to this application and the application classifies the picture as a gorilla. As the neural network is trained to classify everything as apes, it may be excellent at recognizing the faces of apes but not those of humans. Therefore, it does not matter what pictures you offer to the neural network, it will always recognize each of them as a type of ape. So, to prevent disappointments in classifying faces of people, it is better to be transparent about the algorithm.

In the second situation, the focus is on the transparency of the data and the transparency of the other components are neglected fully. As an example of how this works, suppose that we have an application that predicts who has high chances to become the next Prime Minister of the Netherlands. Like most big data algorithms, assume that the application bases its prediction on the features of the persons that have been elected Prime Minister of the Netherlands in the past. Few people will be surprised if such an algorithm will predict that the next prime minister will be a tall man. In this case, transparency regarding the data is crucial for a proper interpretation and use of the outcome. If most of the previously elected Prime Ministers can be characterized by these features and these features are reflected in the data, then the prediction is understandable. If for some reasons the prediction is undesirable, then the data should be adapted. In this

situation transparency with regards to the algorithms will not help to prevent undesired outcomes.

In the GA-based profiling application that we consider [2], the focus will be to visualize the search process and strategy of the algorithm since the end users of the application, i.e., the policymakers at the municipality, are quite familiar with the data and the quality issues of the data used by the tool.

2.2 Related work

Some of the earliest influential work on the subject of visualizing GA's was done by [15]. He discusses several standard techniques, which include visualizations of the increase of fitness over several generations. This is still the most common basic way to visualize the progress of GAs. Another common visualization technique focuses on showing the individuals in each generation, while displaying their respective fitness, genetic make-up or other individual information [16-18]. Usually, the same tools display information that allows users to explore the family tree of successful individuals. Again, this type of information can be useful to illustrate the abstract technical workings of the algorithm, but the results are hard to interpret without any knowledge about GAs.

Most of the data that provides real insight exists in many dimensions, since most practical implementations of GAs are used with big datasets consisting of many variables. This presents a challenge for visualization of GAs as we usually only have two dimensions available on a screen. There are several methods to display more dimensions, such as adding color, 3D-projection and time-based visualizations. However, none of these methods scale very well when the number of possible dimensions starts to exceed five [15]. Another method employed by [15] is to map only the dissimilarities existing in higher dimensions unto lower dimensions. This vastly reduces visual complexity, but also neglects a lot of useful information. The main information that can be gained from these visualizations is whether the GA is converging or diverging. A similar method is used by [19], who transform variables into arbitrary polymorphs. A visualization is then based on these polymorphs, displaying mostly (dis)similarity of variables over generations. The resulting visualization is very abstract and hard to interpret, because the link between the variables and the algorithm is removed.

Although these approaches have merit, none of them appears very suitable for visualizing the GAs used for profiling due to the following shortcomings: (a) the knowledge of GAs is required for comprehension, (b) the results cannot be shown on the fly, (c) they cannot be used to guide the GA when users want to get insight into a certain impact, and (d) they abstract away too much of the underlying attributes, which makes the profiles unintelligible.

3 Profiling: using a GA-based algorithm

Municipalities and government agencies in the Netherlands have embraced data-driven analytics in recent years to exploit their available data in order to be more effective and efficient in their task execution. The rapid deployment of those data-driven analytics

applications in practice, however, brought forward some challenges for these organizations. In [2] the first step towards addressing the transparency challenge was taken by choosing the algorithm that suits the preconditions of the problem at hand.

3.1 Challenges of data-driven profiling in practice

The municipality of Rotterdam and the Dutch IRS are two public organizations that embraced the new opportunities of data-driven profiling. One of the challenges they both encountered was how to deal with profiles and make use of the opportunities provided by these profiles responsibly. A profile is a data driven conclusion that applies to a group of people with a confidence value that may or may not be a realistic probability value. The municipality of Rotterdam made use of the Risk Indication System (abbreviated as SyRi in Dutch) to support the municipality in reducing the risk of social assistance fraud within a household. The data (on houses, labor, education, detention, benefits, debts, taxes and receipts allowances) pertained to the residents of the districts Bloemhof and Hillesluis were combined to make a risk analysis of who may be involved in committing social benefits fraud. At the Dutch IRS, people were tracked down that possibly cheated on their child support benefits, on the grounds of dual nationality and other forms of ethnic profiling.

In both cases it is unclear how both organizations dealt with the interpretation of the results of the profiling systems. In these situations, it is crucial to adequately interpret the concept of opportunity and profile. After all, even if a system delivers a result that someone has a 98% chance to commit fraud or that he/she fully complies with a fraud profile, it cannot simply be concluded that the fraud is actually committed or is going to take place. Another challenge in such applications is that in advance it is known that the analyses will sometimes be wrong. The question then is how do you deal with the so-called “*false positives*” and “*false negatives*”. Because both public organizations mentioned above did not provide an adequate answer to the interpretation challenges of the results, the application of the results to practice has led to the disadvantage of citizens, stigmatization of groups and social commotion. As a result, the profiling applications at the municipality of Rotterdam as well as at the Dutch IRS were shut down.

To enable these organizations to better deploy their data driven profiling in practice, we shall provide some answers for them via creating more transparency for their profile models. In [2] the first step towards this was taken with the GA-based profiling tool. This tool shows that it is feasible to derive meaningful and sensible profiles from data analytics.

3.2 Mining with a Genetic Algorithm

In [2, 20] a data-driven profiling approach was adopted that uses a GA to search for meaningful and human understandable (group) profiles, i.e., in the use case of [2] finding groups of clients who make unlawful use of the municipality’s social service. In that context a profile, which is specified by a set of data attributes (and their values), represents a group of people who unlawfully use social services. An example of such a profile might be “*young men who live in ZIP code 1234AB*”. To search for the profiles,

we model the databases as a search space and tailor a GA to walk through the space. To facilitate the interpretation of the profiles, a profile will be presented in IF-THEN-ELSE-RULES. An example rule might be: *IF sex=male AND age IN [18,24] AND zipcode=1234AB THEN unlawful use of social service.*

Via a case study, paper [2] showed that the resulting profiles can be comprehensible for law enforcement officers to apply them to their daily practices, for policymakers to analyze their efficiency and legality, and for others to scrutinize their fairness. Nevertheless, it also concluded that deriving and using the profiles should be done with great care according to, among others, privacy and non-discrimination laws and guidelines. Hence, in addition to providing a comprehensible outcome, one should assure and validate the resulting profiles in being fair and accurate enough before applying them (i.e., one should manually investigate those persons who are identified by the validated profiles, before imposing any sanctions on them).

Essentially, this approach was successful at making the output of the algorithm more comprehensible, but it also highlighted the need for transparency of the algorithm's process and subsequent control over it. We exploit data visualization as a tool to show the internal search process of the algorithm. Additionally, this visualization functions as a graphical interface that allows users to modify the search direction of the algorithm while it is searching through the data.

The profiling algorithm that was used in [2] is a GA. GA's are considered more easy to understand for data scientists, since their internal search process can be tracked and analyzed. However, their internal representation is often too abstract for domain experts to understand. In terms of performance, this doesn't have to be a problem. The evolution of the profiles during the search process, however, is still very hard to grasp in this way. Therefore, we look at the discipline of data visualization to make this search process more transparent. In the following section this will be explained in more detail.

4 A Glass-box for a GA profiling algorithm

Based on desktop research and interviewing the domain experts at Rotterdam municipality, we gathered the requirements about the GA-visualization that would allow them to secure the interpretation of the creation and evolution of profiles during the search process (see Section 4.1). We noticed an interest in the aspects that are considered important in determining a high prediction risk by the GA and an eagerness in knowing what will happen to the profile search if new pieces of information are added. To address this transparency, it is worthwhile to develop a robust, interpretable, and interactive GA visualization. Visualization methods that we can employ to satisfy these requirements are given in Section 4.2. The implementation of GA-Viz in Section 4.3 and a hypothetical use case is given in Section 4.4.

4.1 Requirements of GA-Viz

Work on the role of visualizations in data-driven profiling exists, but most of it focuses on the exploration of the dataset using general visualization algorithms [21]. Instead, we intend to follow and steer an algorithm in its path through the data in order to find

a solution, i.e., a profile for a problem at hand. Based on several interviews with domain experts and through literature review, we elicited a number of requirements for the visualization of the search process and strategy of the GA. Together, these requirements should result in a visualization module on top of the GA so that end users can obtain more insight into the algorithm's search process and can more actively steer the search. The resulting design requirements are:

1. **Coverage analysis:** Users should be able to determine which parts of the data have been considered by the algorithm and which have been ignored [22]. This control helps to prevent a number of issues, like the GA developing a bias, getting stuck in a local optimum, or ignoring the areas in the dataset that end users know to be of interest. We intend to show the coverage of the algorithm in two ways: by showing an ordered list of the variables the algorithm is exploring the most, and by clustering similar profiles together so that less-explored variables can be seen as lying further out.
2. **Convergence/divergence:** Users should be able to determine whether an algorithm is diverging or converging on solutions [22]. This helps them to decide on the optimal number of cycles to run the GA. To make this possible, the visualization should be on the fly. This also allows them to see that the clusters of similar profiles either grow or shrink, and that the list of different explored variables either grows or shrinks.
3. **Interaction with the algorithm:** A popular, well-performing paradigm for systems designed for domain experts is the human-in-the-loop approach [23,24]. This means that the decision process is designed in such a way that the algorithm can work semi-independently, but that it benefits greatly from human input in its decision process. Our visualization accomplishes this by keeping users updated on every decision by the algorithm and allowing them to interact with the algorithm by steering it in certain directions based on their domain knowledge.
4. **Real-time operation:** the visualization of the algorithm needs to happen in real-time to make real interaction possible. Simply restarting the algorithm with different parameters won't have the same effect, since non-deterministic approaches such as GA will lose the unique progress that they make.
5. **Confirmatory and exploratory analysis:** The visualization should allow for both confirmatory and exploratory analysis, meaning that end users can not only let the program run its course and learn from its process, but can also easily change its hyperparameters and run it again to check certain hypotheses [25]. Because the search process of the GA can be followed, both confirmatory and exploratory analysis not only are possible, but also are easy since end users don't need the algorithm to finish completely before their analysis. Exploration can also be made easier by making the algorithm interactive, thereby allowing end user to steer the direction of their exploration.

4.2 Conceptual design of GA-Viz

How the requirements have been translated into the architecture of our GA-based profiling tool on an abstract level is shown in Figure 1. On the left side in Figure 1 is the

overall workflow of the GA. It starts with the random initialization of the profile population with completely random solutions. Then the GA selects pairs of profiles (parents) to evaluate their fitness (i.e. the score it tries to optimize) with regard to the defined search problem. Next the two parents are subjected to the manipulation operators of crossover and mutation. The two resulting offspring profiles are also evaluated on their fitness and the best one is passed on to the next generation together with the parent profile with the highest score. For more GA details see [2].

The visual design of the interface revolves around a force-based lay-out, where each profile is represented as a circle. The distance between the circles is based on the overlap between their target groups: all profiles represent a group of people, and the distance between these profiles is decided by the number of people they have in common. This creates visual clusters that give users information about the (dis)similarity of the profiles. The size of the circles is determined by the size of the target group, since this is an important indicator for the usefulness of a profile. In Section 4.4 these visual design choices are illustrated.

The coverage can be analyzed by seeing which profiles are being added and removed. The convergence and divergence can be checked by seeing increasing or decreasing similarity between the profiles that continually added, changed and removed. The interaction with the algorithm is made possible in real time, because the profile overview is updated with every new generation, which allows the user to react to this by changing the likelihood of the propagation of specific properties, leading to visible changes in the next iteration. The confirmatory and exploratory analysis are made possible through the same process.

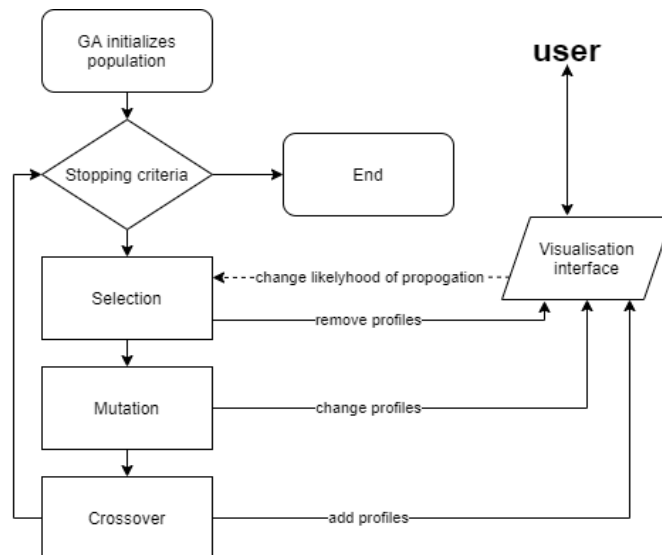


Fig. 1. Interaction between user and GA-model

4.3 Implementation of GA-Viz

The GA runs on a server using the Genie package for Julia language. The visualization runs in a browser. This makes the system flexible and modular. The GA and its visualization are connected through websockets, which makes the communication between them near-instantaneous. The GA starts to run, and whenever genetic individuals are created, mutated or removed, it sends an update to the client browser in the JSON format. The client browser runs JavaScript, using d3 for its clustering algorithm. Normally, d3 works with dynamic svg-images. This leads to performance issues when too many objects are displayed at the same time. Therefore, we used PixieJS to render our visualization instead, since it uses WebGL to achieve superior performance. In this way, we were able to implement all the requirements mentioned in Section 4.1, apart from the interaction with the algorithm. Although the infrastructure and the design are there, implementing the actual interaction is left for future editions of the tool. This means that the dotted line in Figure 1 is still in the implementation phase.

To tailor and test the visualization, we utilized a similar use case and dataset as those of [2] to visualize the corresponding GA. However, in order to showcase the genericness of our approach and due to the classified nature of that dataset, we decided to create and test another use case based on public U.S. census data [26].

4.4 A Hypothetical use case

To illustrate our visualization, we present a hypothetical use case. It not only shows how the system may work in practice, but also illustrates how the visualization satisfies the requirements in Section 4.1. We will refer to these requirements as (req. 1) through (req. 5). Figure 2 shows a screenshot of the user screen during the search process. In the use case, a car producer is doing market research for a new series of high-end cars. They are looking for buyers in a higher income bracket. They set this bracket at an income of 50K \$ per year. Using available census data, they want to create interesting profiles of potential customers so that they can adapt their marketing accordingly.

The visualization shows the progress of the GA through the data, while it ‘searches’ for the best (i.e., the highest scoring) profiles. The GA is optimized to work with profiles made of demographic data and to find those profiles that fit a large number of high-income persons. For example, a profile looks like the one shown in Table 1. This profile could be represented in our visualization as shown in Figure 2. Since it has a high fitness there, it is represented by a relatively large circle. The GA uses these profiles as genetic individuals, tries out new combinations, and strives to generate new profiles with an increasingly good fit. A fitness function is used to optimize the search process for a (predefined) ratio of the number of people with an income above or below the 50K \$ threshold that a profile represents.

Table 1. Profile example (consisting of 5 attributes)

Attribute	Value
Wage per hour	10.17 – 18.82
Age	2 : 16
Reason for unemployment	new entrant
Education level	associates degree / academic program
Race	black

The end user of the profiling system is a marketing expert we will call John. Starting the GA, it immediately shows ten blue circles of varying sizes. Each circle is also labeled with a sequential number to ease identification. John hovers over one of the circles with his mouse, and he is presented with information about the profile that that circle represents, like its attributes, its fitness and the size of its target class (i.e., the number of people it targets). John is curious about some of the smaller circles, so he scrolls his mouse wheel and drags the canvas to pan and zoom in on his areas of interest.

**Fig. 2.** A screenshot of our application illustrating the use case.

The profile updates from the GA are animated, drawing attention to the changes (req. 4). Whenever a new profile is created, it ‘shoots’ onto the screen. When a profile mutates, it shakes. When a profile is removed, it fades out. John sees that many of the uninteresting profiles are being replaced by new ones, but also that some of the profiles are changing and becoming more interesting and better performing. This shows that the algorithm is slowly converging (req. 2). John sees that a cluster of profiles is starting to form, all of which include the attribute sex =”female”. He is interested in this and he marks them by clicking them. This clicking makes the color of the profile cluster red so that their progress can easily be tracked (req. 5). After a while, however, he notices that many of these profiles are being replaced by different profiles that represent highly-educated white males because they tend to have higher average incomes. Since this is

already an over-saturated market, he adjusts a slider, so that sex =”male” is negatively weighted in next generations (req. 3).

Gradually different profiles start to emerge and, besides the main clusters, some small profiles (i.e., targeting few persons) at the edges start to appear and disappear. These small profiles seem interesting but are not selected for reproduction causing some areas of the data to be underexplored (req. 2). John clicks one of them and adjusts a slider that increases their reproduction likelihood (req. 3). Another important aspect of the visualization is that it keeps an updated list of attributes. This list shows all of the attributes that are currently involved in at least one of the profiles. It updates whenever the visualization changes. Included in this list is the total number of the profiles using each parameter, as well as the average fitness of all of these profiles.

By default, this list is sorted by the average fitness value of the profiles that incorporate that attribute. This gives an indication about that specific attribute, which is often an overall good predictor. The list can also be sorted by the number of the profiles that incorporate that attribute. This gives an indication of how much the currently running GA is focusing on that attribute (req. 1). This give users an overview of the current state of the algorithm, and it also allows users to check if the algorithm is missing opportunities. For example, John notices that the inclusion of attribute ‘education’ produces high fitness, but when he sorts the list by number of profiles, he sees that the algorithm is not really focusing on that attribute. He adjusts a slider to include the likelihood of those profiles to reproduce (req. 3). When a user clicks on the attributes in this list, the profiles that incorporate that attribute are highlighted and can be tracked throughout the GA search process. John uses this functionality, and notices that education performs especially well in conjunction with sex =”female”, forming small target groups with a high fitness. He decides to use this finding as the foundation of his new marketing campaign, using the generated profiles to further specify some niches.

5 Conclusion and Future Work

In this paper we operationalize a glass box concept for a GA-based profiling tool. To this end, we exploited the visualization of the search process and strategy of GA to find profiles. The visualization provides ends user insights and a better understanding of why and how certain profiles are selected by the GA. These insights help them to interpret the selected profiles by the GA. Furthermore, end users are given the possibility to steer the GA into a desired direction.

Experts appeared to be interested in how the application comes up with the profiles as well as playing with the application to see what the effect is when one starts turning the knobs. So far, we described the first part of a glass box extension of a GA-based profiling application. A next step is to implement a more comprehensive interaction functionality for our GA-based profiling application (i.e., the GA-Viz tool). Once this is implemented, end user tests can start to determine the efficacy of our approach.

In the current design, we have made all the information available in one coherent visualization module. This visualization provides an overview of all profiles that are being explored by the algorithm. Although it provides a lot of information, this could

be improved upon by offering users multiple ‘views’. For example, a view could be offered that gives more information about a specific profile, or a view that shows more information about the general progress and history of the current algorithm. These could be shown together on a dashboard to increase informativity and clarity, which has been shown with great success by previous work like [27].

Not only can a visualization make this specific profiling tool much more transparent, but also, we believe, many profiling algorithms would benefit from being more transparent. While developing the visualization, we aimed to make it generic enough to be used not only on GAs as used by [2,20], but on all profiling algorithms based on supervised machine learning. Examples include Support Vector Machines, Multiple Regression and Random Forests. It could even be used to visualize Neural Network-based algorithms, although additional work would be needed to extract the necessary data.

References

1. Jans, M., Lybaert, N. & Vanhoof, K. (2008). Data Mining for Fraud Detection: Toward an Improvement on Internal Control Systems?
2. Netten, N., Bargh, M.S., Choenni, S. (2018) Exploiting Data Analytics for Social Services: On Searching for Profiles of Unlawful Use of Social Benefits. In: Proceedings of the 11th International Conference on Theory and Practice of Electronic Governance. pp. 550–559. ACM, New York, NY, USA.
3. PredPol. (2018). PredPol 2018. <http://www.predpol.com/>. Accessed 10-3-2021
4. Lonkhuyzen, L. (2017). To predict crime, it is possible, NRC, May. www.nrc.nl/nieuws/2017/05/16/misdaad-voorspellen-het-kan-echt-9100898a1558837
5. Willems, D. and Doeleman, R. (2014). Predictive policing: wens of werkelijkheid. *Tijdschrift voor Politie*, 76, 4.
6. Hildebrandt, M. (2006). Profiling: From data to knowledge. *Datenschutz und Datensicherheit-DuD* 30.9: 548-552.
7. Leese, M. (2014). The new profiling: Algorithms, black boxes, and the failure of anti-discriminatory safeguards in the European Union. *Security Dialogue*, 45(5), 494-511.
8. Choenni, S., Netten, N., Bargh, M., Choenni, R. Challenges of Big Data from a philosophical perspective. In Proceedings of the 2018 International Conference on Multidisciplinary Research (MyRes), Pamplemousses, Mauritius, 22–23 June 2018, pp. 71–81
9. Choenni, S., Netten, N., Shoaebargh, M., & Choenni, R. (2018). On the Usability of Big (Social) Data. 2018 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), pp: 1167-1174.
10. Choenni S., Netten N., Bargh M.S., van den Braak S. (2021) Exploiting Big Data for Smart Government: Facing the Challenges. In: Augusto J.C. (eds) Handbook of Smart Cities. Springer, Cham. https://doi.org/10.1007/978-3-030-15145-4_82-1
11. Guidotti, R, Monreale, A, Ruggieri, S., Turini, F, Giannotti, F. and Pedreschi, D. (2018) A survey of methods for explaining black box models. *ACM Computing Surveys*, 51, no. 5.
12. du Boulay, B., O’Shea, T., & Monk, J. (1981). The black box inside the glass box: presenting computing concepts to novices. *International Journal of man-machine studies*, 14(3), 237-249.

13. Sokol, K., & Flach, P. A. (2018, January). Glass-Box: Explaining AI Decisions With Counterfactual Statements Through Conversation With a Voice-enabled Virtual Assistant. In *IJCAI* (pp. 5868-5870).
14. Rai, A. (2020). Explainable AI: From black box to glass box. *Journal of the Academy of Marketing Science*, 48(1), 137-141.
15. Pohlheim, H. (1999) Visualization of Evolutionary Algorithms - Set of Standard Techniques and Multidimensional Visualization. In: *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Vol 1*. pp. 533–540. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA
16. Wu, A.S., De Jong, K.A., Burke, D.S., Grefenstette, J.J., Loggia Ramsey, C.(1999) Visual analysis of evolutionary algorithms. In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99* (Cat. No. 99TH8406). pp. 1419–1425 Vol. 2. ieeexplore.ieee.org
17. Cruz, A., Machado, P., Assunção, F., Leitão, A.: ELICIT: Evolutionary Computation Visualization. In: *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*. pp. 949–956. ACM, New York, NY, USA (2015)
18. Hart, E., Ross, P.(2001) GAVEL-a new tool for genetic algorithm visualization. *IEEE Trans. Evol. Comput.* 5, 335–348.
19. Daneshpajouh, H., Zakaria, N. (2017) A Clustering-based Visual Analysis Tool for Genetic Algorithm. In: *VISIGRAPP (3: IVAPP)*. pp. 233–240. researchgate.net
20. Choenni, S. (2000) Design and implementation of a genetic-based algorithm for data mining. In: *VLDB*. pp. 33–42.
21. Ross, G, Morrison, A. and Chalmers, M. (2004) Coordinating views for data visualisation and algorithmic profiling," *Proceedings. Second International Conference on Coordinated and Multiple Views in Exploratory Visualization*, London, UK, 2004, pp. 3-14.
22. Shine, W.B., Eick, C.F. (1997) Visualizing the evolution of genetic algorithm search processes. In: *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*. pp. 367–372. ieeexplore.ieee.org
23. Llorà, X., Sastry, K., Alías, F., Goldberg, D.E., Welge, M.(2006) Analyzing Active Interactive Genetic Algorithms Using Visual Analytics. *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. pp.1417–1418. ACM, New York, NY, USA
24. Hayashida, N., Takagi, H. (2000) Visualized IEC: interactive evolutionary computation with multidimensional data visualization. In *26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 21st Century Technologies*. pp. 2738–2743 vol.4.
25. Keim, D.A., Mansmann, F., Schneidewind, J., Thomas, J., Ziegler, H (200) *Visual Analytics: Scope and Challenges*. In: Simoff, S.J., Böhlen, M.H., and Mazeika, A. (eds.) *Visual Data Mining: Theory, Techniques and Tools for Visual Analytics*. pp.76–90. Springer Berlin
26. U.S. Census Bureau (2000). *Census-Income(KDD) Data Set*. Retrieved from <https://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29>
27. Google Research (2021) *People + AI Research (PAIR)*. url:<https://research.google.com/big-picture/attacking-discrimination-in-ml/> Last accessed 10-3-2021