Fairness and Transparency of Machine Learning for Trustworthy Cloud Services

Nuno Antunes[‡], Leandro Balby[†], Flavio Figueiredo[*], Nuno Lourenco[‡], Wagner Meira Jr.[*], Walter Santos[*]
[*] Universidade Federal de Minas Gerais, Belo Horizonte – MG, Brazil
[†] Universidade Federal de Campina Grande, Campina Grande – PB, Brazil
[‡] CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
nmsa@dei.uc.pt, lbmarinho@dsc.ufcg.edu.br, flaviovdf@dcc.ufmg.br, naml@dei.uc.pt, meira@dcc.ufmg.br, walter@dcc.ufmg.br

***Abstract*—Machine learning is nowadays ubiquitous, providing mechanisms for supporting decision making that leverages big data analytics. However, this recent rise in importance of machine learning also raises societal concerns about the dependability and trustworthiness of systems which depend on such automated predictions. Within this context, the new general data protection regulation (GDPR) demands that organizations take the appropriate measures to protect individuals' data, and use it in a privacy-preserving, fair and transparent fashion. In this paper we present how fairness and transparency are supported in the ATMOSPHERE ecosystem for trustworthy clouds. For this, we present the scope of fairness and transparency concerns in the project and then discuss the techniques that are being developed to address each of these concerns. Furthermore, we discuss how fairness and transparency are used with other quality attributes to characterize the trustworthiness of cloud systems.**

I. Introduction

Machine Learning (ML) is nowadays ubiquitous, as most organizations take advantage of it to perform or support decisions within their systems [1], [2]. ML is an area of Artificial Intelligence (AI) in which we use a set of statistical methods and computational algorithms to allow computers to learn from data [3]. ML algorithms can be divided into two main groups: supervised and unsupervised. *Supervised learning* involves the development of computational models for estimating an *output* based on previously known *inputs* and *outputs*. In *unsupervised learning*, the models are built based solely on existing *inputs* but there are no associated *outputs* that may be used for sake of training. We may face fairness and transparency issues for both groups of algorithms.

It is now commonplace to run ML systems in cloud-based infrastructures, motivated by issues such as elasticity, robustness, and ease of operation [4]. In practice, cloud services are fueling big data analytics, allowing organizations to make better and faster decisions using data that previously were hard or impossible to use [5]. This raises many opportunities in today's competitive environment, by offering many services using highly scalable technologies on a pay-as-you-go basis. However, it also *creates new challenges regarding trust*, a paramount concern in critical systems [5].

Regulatory institutions have long focused these properties namely in OECD's fair information practices [6] and in EU Privacy Directive `95/46/EC` [7]. However, such **legislation has never received as much emphasis as now**. The new EU General Data Protection Regulation (`GDPR`) [8] shifts the onus to the organizations, demanding them to demonstrate that they are taking the appropriate measures to protect the legal rights of the individuals and their data, requiring privacy-preserving, fair and transparent systems.

The `ATMOSPHERE` project (atmosphere-eubrazil.eu) aims at developing an *ecosystem to support the design and development of next generation trustworthy cloud services on top of an intercontinental hybrid and federated resource pool*. It considers trustworthiness as depending on many properties such as security, dependability, and privacy assurance among others. Moreover, data become a first class citizen, as trustworthiness also depends greatly on respecting data subject rights.

In this context, **Fairness** and **Transparency** emerge as key properties for the trustworthiness of cloud systems while processing big data. Fairness is concerned with the *assurance of ethical and legal rights*. For instance, a **fair classifier** does not discriminate subjects based on sensitive attributes such as gender or race [9]. Transparency involves multiple sub-dimensions, such as awareness, access, explanation, provenance, auditability and accountability [8]. As ML gains importance, in recent years the Computer Science community (fatml.org, fatconference.org) has invested efforts on fairness and transparency concerns.

In this paper we present how **`ATMOSPHERE` addresses fairness and transparency** towards more trustworthy systems. In practice, it provides an ecosystem for the quick development of large-scale data processing services, endowing them with trustworthy support at the data management and infrastructure layer and with capabilities for trustworthiness monitoring, assessment and adaptation. These services are built using the Lemonade [10] platform. Lemonade is a scalable and efficient visual programming based platform for cloud-based big data analytics. The platform allows users to be compliant with `GDPR`, and its ecosystem gives control over the complete stack of the services, providing the necessary information and means to measure and assess these properties.

From the context of the project we introduce an initial set of techniques that are being developed not just to support but also to monitor and assess fairness and transparency in the context of ML applications and systems. Finally, we present concrete examples of practical application of these techniques in Lemonade, and how they integrate with other components.

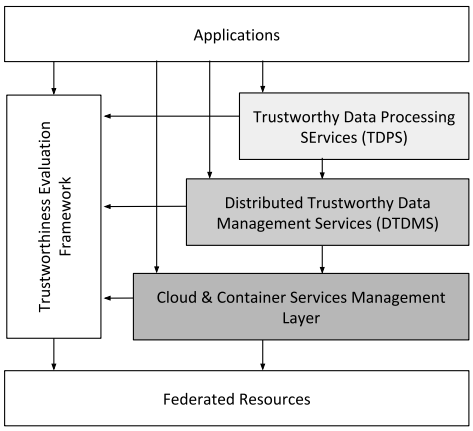

Fig. 1. The model of the ATMOSPHERE project.

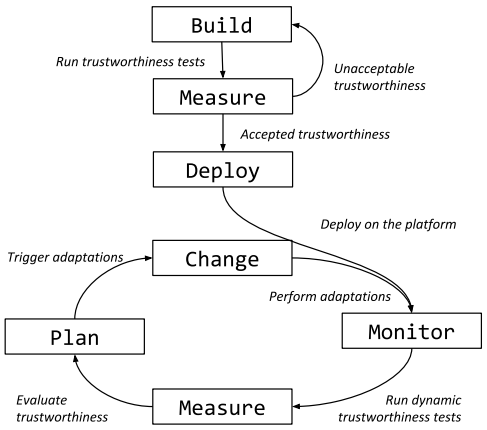

Fig. 2. The lifecycle of ATMOSPHERE applications.

II. Context and Motivation

Trust can be defined as the accepted dependence of a component/user on a set of properties which are provided/implemented by another component, subsystem or system [11]. Then, trustworthiness can be defined as the measure in which a component, subsystem or system, meets such properties [11]. Throughout this section, we describe ATMOSPHERE, as well as fairness and transparency within the context of ML systems.

A. Atmosphere

ATMOSPHERE considers a cloud model comprising three layers and defines the relevant trustworthiness properties for each layer: cloud resources, data management services, and data processing services. Fig. 1 depict these layers and how they interact considering its trustworthiness assessment framework, resources and applications.

As we can observe, there is a trustworthiness assessment framework that receives from all layers of the system the necessary information to assess and monitor the relevant trustworthiness properties. Based on those properties, ATMOSPHERE will provide a continuous, global score of trust for an application, which is a function of the following properties: security, privacy assurance, coherence, isolation, stability, fairness, transparency and dependability.

As depicted in Fig. 2, ATMOSPHERE considers measuring trust both *a priori* (before deployment) and dynamically during execution (at runtime). In practice, applications are deployed only after satisfying the required levels of trustworthiness. After that, the relevant properties are monitored to verify whether the required levels are maintained, as the trust in the application may be affected by changes in the environment and workload, security attacks, and resources' availability.

These scores may be used by the application developer, the application itself or the execution framework for adjusting parameters to increase trust or to react to runtime failures in federated infrastructures, up to the limits on resource allocation that a user may have set - avoiding infinite consumption of resources.

Fairness and transparency are mainly monitored at the layer of the data processing services, which are introduced in the next subsection.

B. Trustworthy Data Processing Services

This layer of the ATMOSPHERE ecosystem will provide a set of tools, libraries and services that will support users to build trustworthy services for processing large data sets. Therefore, this layer will provide primitives, metrics and tools that enable developers to consider privacy, security, fairness and transparency properties in the development of their cloud services.

The key component of this layer is Lemonade (*Live Exploration and Mining Of Non-trivial Amount of Data from Everywhere*) [10]. Lemonade is a visual platform for distributed computing, aimed to enable implementation, experimentation, test and deploying of data processing and machine learning applications. It provides a higher level of abstraction, called operation, to users build processing workflows using a graphical web interface. By using high performance and scalable technologies, such as BSC COMPSs [12] and Apache Spark [13], Lemonade enables the processing of very large amount of data, hiding all back-end complexity from users and allowing them to focus mainly on the construction of solutions, models and services. Details are presented in Section IV-A.

Due to the proximity to the application, this is the layer better supporting the measurement of fairness and transparency in the context of ATMOSPHERE. In practice, as it will be described in Section IV, the tools, libraries and services to be provided will include a set of fairness-aware or transparency-aware versions that are available to the user, and also special components to be added to the data processing workflows. The metrics obtained are then integrated with the trustworthiness assessment framework for the computation of the global scores.

C. Data Analytics and Machine Learning

Tools such as Deep Learning [14] and Generative Adversarial Networks [15] (GANs) have gained momentum as major

techniques in the ML field. In spite of the unquestionable success that these techniques have achieved, they are not exempt from criticisms. For instance, although recent attempts to understand the models learned from data using Deep Learning or GANs do exist [16], [17], for the majority of end-users and services these models are still viewed as black-boxes. A black-box is an algorithm where its implementation is opaque, i.e., it is difficult to understand the inner workings of of the method. People have accepted the black-box nature of Deep Learning models over the years due to the their good performance on many tasks.

Nevertheless, when our goal is to develop ML methods where interpretability is a necessity, we might move towards other type of ML algorithms such as Decision Trees [18] or a Logistic Regression. The resulting models of both algorithms allow some degree of understanding of its inner workings. For instance, trees can be specified in terms of if-then-else rules, whereas Logistic Regression learns interpretable parameters from data.

An interpretable ML model is a model that is able to explain its reasoning in comprehensible terms to a human [19]. This is very important, because it can be used to measure the fairness of a system. In this case, fairness means that the model is not biased, nor discriminating a certain group or groups. It is also possible to link interpretability to transparency, since, in principle, the more interpretable the model is, the more transparent it is. Over the next section we discuss ATMOSPHERE's approach to tackle Transparency and Fairness requirements.

III. Fairness and Transparency Assessment and Assurance in ATMOSPHERE

Before delving into the details of the approaches employed for the assessment and assurance of fairness and transparency, it is necessary to analyze the scope of our concerns and actions. As the problem is rather broad, this analysis must be performed according to multiple dimensions, as follows.

The first dimension is regarding **who** we are concerned or trying to help. Three main agents should be considered:

- *Compliant Organization* – organizations that plan to provide fair and transparent services, but may lack the resources or knowledge. These organizations need cost-effective solutions.
- *Non-Compliant Organizations* – organizations are not concerned with providing fair and transparent services. Sooner or later (e.g. due to legal reasons), it will be necessary to assess the level to which they are disrespecting fairness and transparency principles.
- *Malicious Users* – attackers may exploit weaknesses in the models to cause fairness and transparency issues. It is necessary to provide organizations with tools that help them understand how resilient their systems are to attacks.

The second dimension is in which **phase of the application development lifecycle** it is possible to act. An overview of this dimension is presented in Fig. 3. As it is possible to observe, two main phases are considered, mapping to the lifecycle presented in Section II-A: *development* (which happens before the deployment of the application), and *production* (which represents the execution of the application after deployment).

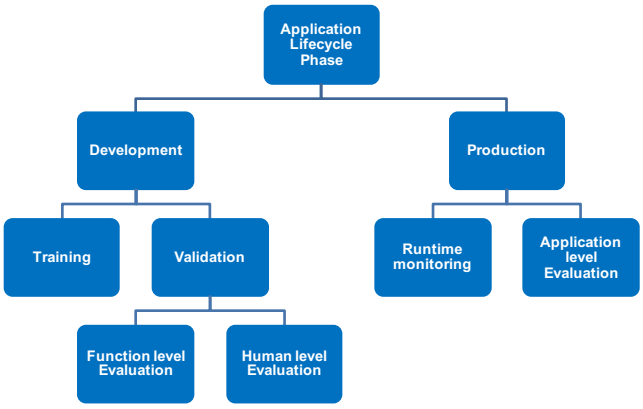

Fig. 3. Fairness and transparency in the applications development lifecycle.

During model *development* two main phases are usually considered: training and validation. While training, it is possible to use techniques for recommending models that better suit the characteristics of the dataset and domain, avoid the use of protected attributes. When validating the model, two main activities may be performed: function level evaluation, which do not depend on humans, and human level evaluation, which, instead, relies on human input.

During *production*, the activities are divided into runtime monitoring, which allows to passively obtain information that may be used for assessment, and at application level evaluation, which depends on human input, namely domain expertise.

The final dimension is the **type of strategy** that will be employed. Fig. 4 provides an overview of the systematization of the potential strategies to address these challenges.

As we can observe, there are two main types of techniques: the ones that use *interpretable models* and the ones that are *agnostic to the model*.

The easiest and straightforward way to achieve interpretability in machine learning is to use only a subset of algorithms that are well known to be interpretable. Typical model interpretable types are Logistic Regression and Decision Trees. The main drawback of this approach is that such models have limited expressiveness and thus might be unsuited to model

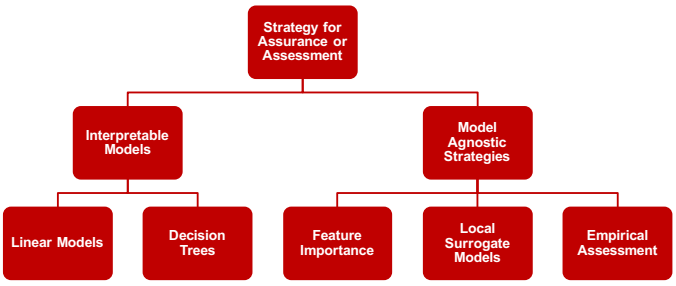

Fig. 4. Classes of the techniques to use for fairness and transparency.

complex data.

Model agnostic interpretability models, on the other hand, refer to separating the explanations from the machine learning model [20]. Thus, the machine learning developer does not need to sacrifice performance for interpretability and is free to use any machine learning model she likes. Two representatives of this kind of approach are: (i) feature importance, that measures in a model agnostic way the importance of features (which are used for explanations) [21], and (ii) local surrogate models, that fit local, interpretable models that can explain single predictions of any machine learning model [22].

In empirical assessment we include techniques based on tests that may use specially crafted inputs adapted for sake of fairness validation, in a similar fashion to what is employed nowadays for reliability and dependable systems. Although less complete, these techniques usually are more flexible and scalable, and therefore applicable to more situations.

A. Transparency Assessment and Assurance

One of the first attempts to formulate evaluation strategies for assessing interpretability in Machine Learning is proposed by Doshi-Velez and Kim [19]. They propose a three-level taxonomy of evaluation approaches, briefly summarized next.

- Application level evaluation: The explanations are integrated into the outputs of the final model such that end users (in this case domain experts) can evaluate it. This requires a clear idea on how to assess the quality of explanations, for example, comparing to the explanations given by domain experts to the same decision.
- Human level evaluation: This corresponds to a simplified version of the application level evaluation. The difference is that here domain experts are not required, thus making experiments cheaper and more feasible.
- Function level evaluation: Here humans are not required. This is more efficient when the models used are already well understood by humans. If it is known, for example, that users understand well decision trees, a measure of explanation quality might be the depth of the tree, i.e., shallow trees would get a better explainability rating.

Transparency can also be measured based on the minimum description length (MDL) of the model. For this, it is necessary to build an interpretable model from the outputs of the original one and calculate its MDL as a transparency metric.

Finally, there are aspects of transparency related to the purpose for which data are used, as *GDPR* also contemplates the right of the data subject to "be informed of the existence of the processing operation and its purposes" [8]. For this, it is necessary to track at runtime the application for which each data element is used. This can to be done at multiple granularities, on a trade-off between information and performance.

B. Fairness Assessment and Assurance

Complementing Transparency, Fairness evaluation is also a challenge. That is, Fairness issues may rise directly due to biases in the data [23]. Also, Fairness issues arise when loss functions are ill-defined and feedback-loops exist in the system [23]. We now detail such issues within the scope of the ATMOSPHERE project. We also present proposals on how to evaluate them. In order to understand fairness, consider for instance a classifier trained for the task of identifying health risks based on subject features.

- Selective Sampling/Labelling: The classifier above may be trained with data that was either sampled or labelled selectively. For instance, crowd-workers may label data based on pre-conceived relations between gender and health risks. Also, the dataset may be biased towards certain social-demographic variables. On the deployment phase, identifying under-represented or over-represented groups of subjects may mitigate such issue. Moreover, dropping labels where no significant agreement between labelers exist can also help. At the scope of the ATMOSPHERE project, we shall evaluate the representation of subjects/labels, helping end-users identify such settings and act accordingly.
- Unfair Features: Columns of a table, or features, may also present Fairness issues. Some of such features can even be illegal to train ML algorithms on [23]. Dropping such columns and/or anonymizing others are a natural procedure to deal with such a problem. By leveraging representation metrics (e.g., features which are skewed), as well current regulations, the ATMOSPHERE model will help end-users decide whether columns will be explored or not.
- Loss Functions: The loss function of ML algorithms may also be subject to biases and fairness issues. In fact, loss functions prone to less biases are gaining the attention of the scientific community (see fatml.org). Recently, novel techniques exist which can deal with representation issues directly in the training phase [24], [25]. Such techniques can be provided to the end-user within the Lemonade framework.
- Fault Injection: studies have shown that models can be easily affected by adding small perturbations in the inputs [26]. In line with these findings, and inspired in what has been done in computing systems for dependability assessment, we will propose techniques to systematically generate fault models focused on fairness, for the specific application domain in question taking advantage of the input datasets.
- Feedback loops: Finally, feedback loops occur when user actions guided by ML algorithms decrease trustworthiness. In our example, medical actions guided by an algorithm (e.g., a correlation between race and a certain risk), can increase biases when such data is used to retrain new models. Even though feedback loops rise due to user actions, it is possible to employ recent techniques as to detect their presence [27].

IV. Practical Application

As explained in Section II, it is expected that end-users will access the ATMOSPHERE model through an ML framework. Currently, we are extending Lemonade for supporting trustworthiness properties. In this section, we detail the Lemonade [10] platform and discuss the approach to address Fairness and Transparency.

A. Lemonade

Lemonade is a visual programming ML framework. A service, in Lemonade platform, has two distinct moments regarding its execution: design time and production time. In design time, a business specialist, data scientist or any other professional working with data, uses the tool to explore data or perform experiments. Many adjustments may be performed, for example, model generation tuning and settings related to privacy. Any time, the professional may modify, execute the workflow and receive feedback (success/failure, samples of data, visualizations) from Lemonade. In general, this is a cycle that repeats until the professional is confident about the result, for example, when it produces the correct outcomes or a metric is satisfied. When the design process resumes, the resulting workflow may be deployed as a service. In this case, the resulting workflows of Lemonade are executed in production time. The professional can deploy a Lemonade workflow by pressing a button in the interface and configuring the deployment parameters of the service. Lemonade deploys the workflow as a microservice that provides a REST API. Finally, applications may be built by the composition of different Lemonade microservices.

Under the hood, Lemonade includes a source code generator. It generates code compatible with Python language, targeting the chosen execution platform. For example, a user may choose Apache Spark as execution platform and, in this case, Lemonade generates a valid PySpark (Python code that includes Spark API calls). In design time, the generated source code is submitted to a job scheduler (Apache Mesos, Apache Yarn) or a container orchestration service (Kubernetes). The job scheduler (or container service) allocates resources (processing nodes, memory, CPUs, GPGPUs) according to QoS parameters specified in Lemonade and then executes the code. In production time, the process is similar, but, in this case, to guarantee the continuous execution of the Lemonade service, a high availability service, such as Mesosphere Marathon, is used. It restarts Lemonade service in case of failures and notify administrators in case of fatal failures.

B. Fairness and Transparency in Lemonade

We foresee at least 3 different strategies to implement trustworthiness assurance in Lemonade: (1) by injecting validations/control code during the source code generation; (2) by wrapping existing operations as new operations and including the validation/control code in the new implementation (composition of operations); (3) by extending the workflow primitives in Lemonade to include a operation IF/Decision that evaluates a conditional when the code is executed.

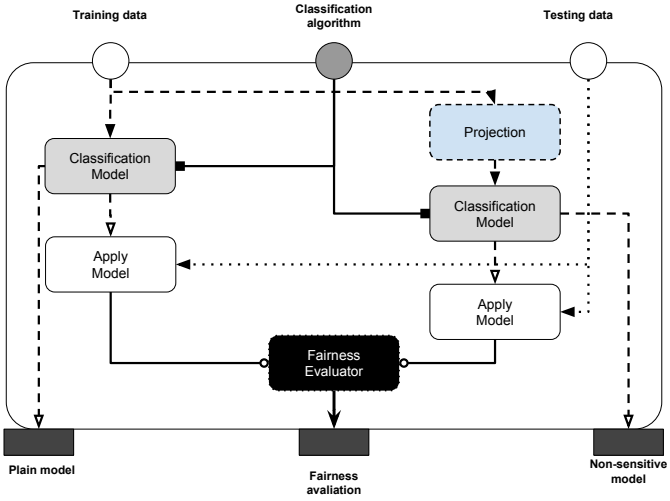

Fig. 5. Representation of a Fairness aware classifier in Lemonade

All of the three approaches above will operate either as a decorator or as a composition pattern to guarantee trustworthiness. That is, Lemonade currently provides end-users with several ML algorithms to be exploited.

One example of Fairness assessment using Lemonade is shown in Figure 5. Based on the figure, we foresee a Fairness evaluation component which compares the output of two models. One model (on the right), is trained using the original dataset. The other model is trained on a projection of the data which deals with Fairness as described in the previous section. For instance, this projection may be a sub-sample of the data after features dropped or biases are removed (e.g., by re-sampling). Both models are applied and evaluated resulting in three outputs: (1) a plain model; (2) a non-sensitive (or fair) model; and, (3) a fairness evaluation which compares the efficacy of the two models.

In Figure 6 we present an example focused on Transparency.

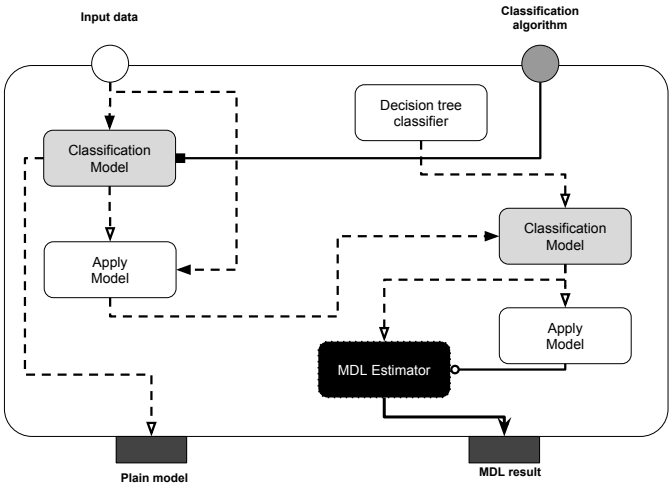

Fig. 6. Representation of a transparency aware/model interpretable classifier in Lemonade

Again, two models are trained on the same input data. However, in this setting, models use the same input data. Then, the Minimum Description Length (MDL) of both models may be computed and compared. MDL captures a trade-off between model accuracy and simplicity. As in Occam's Razor, simpler explanations should be favored in contrast to complex ones. Simpler models will *possibly* be more interpretable and useful for end-users.

These examples show how we are exploring Lemonade as the framework users will explore in ATMOSPHERE. In the next section, we shall conclude the paper presenting some of our proposals for the future of ATMOSPHERE's model.

V. Conclusions and Future Work

Fairness and transparency are key properties in face of the increasing adoption of machine learning algorithms to processing the large amounts of data available.

In this paper we presented the research plans for the way ATMOSPHERE will address fairness and transparency in its ecosystem for trustworthy clouds. Lemonade has a key role in this process, as it will allow developing, testing and deploying of data processing and machine learning applications. By endowing Lemonade with fairness and transparency capabilities, we will be allowing the development of better applications.

These properties will be handled together with other properties that may have conflicting objectives, such as confidentiality. Although this represents a challenging proposition, it will allow that fairness and transparency are considered in real applications for relevant scenarios.

Acknowledgment

This work has been partially supported by the project **ATMOSPHERE** (atmosphere-eubrazil.eu), funded by the Brazilian Ministry of Science, Technology and Innovation (Project 51119 - MCTI/RNP 4th Coordinated Call) and by the European Commission under the Cooperation Programme, Horizon 2020 grant agreement no `777154`. It is also partially supported by the project **METRICS**, funded by the Portuguese Foundation for Science and Technology (FCT) – agreement no `POCI-01-0145-FEDER-032504`.

References

[1] A. Bundy, "Computational thinking is pervasive," *Journal of Scientific and Practical Computing*, vol. 1, no. 2, pp. 67–69, 2007.

[2] M. Rich, "Machine Learning, Automated Suspicion Algorithms, and the Fourth Amendment," *University of Pennsylvania Law Review*, vol. 164, no. 4, p. 871, 2015.

[3] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[4] "User Stories - OpenStack Open Source Cloud Computing Software." [Online]. Available: https://www.openstack.org/user-stories/

[5] E. O. o. t. President, *Big Data: Seizing Opportunities, Preserving Values*. CreateSpace Independent Publishing Platform, 2014.

[6] OECD, "Guidelines on the Protection of Privacy and Transborder Flows of Personal Data." [Online]. Available: http://www.oecd.org/sti/ieconomy/oecdguidelinesontheprotectionofprivacyandtransborderflowsofpersonaldata.htm

[7] European Parliament, "Directive 95/46/EC," Nov. 1995. [Online]. Available: http://data.europa.eu/eli/dir/1995/46/oj/eng

[8] ——, "General Data Protection Regulation," May 2016. [Online]. Available: http://data.europa.eu/eli/reg/2016/679/oj/eng

[9] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness Through Awareness," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ser. ITCS '12. New York, NY, USA: ACM, 2012, pp. 214–226. [Online]. Available: http://doi.acm.org/10.1145/2090236.2090255

[10] W. d. Santos, L. F. M. Carvalho, G. de P. Avelar, A. Silva, Jr., L. M. Ponce, D. Guedes, and W. Meira, Jr., "Lemonade: A scalable and efficient spark-based platform for data analytics," in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, ser. CCGrid '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 745–748.

[11] P. E. Verssimo, N. F. Neves, and M. P. Correia, "Intrusion-Tolerant Architectures: Concepts and Design," in *Architecting Dependable Systems*, ser. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2003, pp. 3–36.

[12] R. Badia, J. Conejero, C. Diaz, J. Ejarque, D. Lezzi, F. Lordan, C. Ramon-Cortes, and R. Sirvent, "Comp superscalar, an interoperable programming framework," *SoftwareX*, vol. 3-4, pp. 32–36, Dec 2015. [Online]. Available: http://hdl.handle.net/2117/89874

[13] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache spark: A unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, Oct. 2016.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[16] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," *CoRR*, vol. abs/1703.00810, 2017. [Online]. Available: http://arxiv.org/abs/1703.00810

[17] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," *arXiv preprint arXiv:1610.01644*, 2016.

[18] T. Chen, T. He, M. Benesty *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, pp. 1–4, 2015.

[19] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017.

[20] M. Tulio Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," in *ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*, 06 2016.

[21] A. Fisher, C. Rudin, and F. Dominici, "Model Class Reliance: Variable Importance Measures for any Machine Learning Model Class, from the "Rashomon" Perspective," *ArXiv e-prints*, Jan. 2018.

[22] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 1135–1144. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939778

[23] H. Daumé III, "A course in machine learning," *Publisher, ciml. info*, pp. 5–73, 2012.

[24] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, "Fair clustering through fairlets," in *Advances in Neural Information Processing Systems*, 2017, pp. 5036–5044.

[25] R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, and A. Roth, "A convex framework for fair regression," *arXiv preprint arXiv:1706.02409*, 2017.

[26] J. Su, D. V. Vargas, and S. Kouichi, "One pixel attack for fooling deep neural networks," *arXiv:1710.08864 [cs, stat]*, 2017, arXiv: 1710.08864. [Online]. Available: http://arxiv.org/abs/1710.08864

[27] D. Ensign, S. A. Friedler, S. Neville, C. Scheidegger, and S. Venkatasubramanian, "Runaway feedback loops in predictive policing," *arXiv preprint arXiv:1706.09847*, 2017.