Ineffective Right & Undone Science: the case of the access to administrative algorithms in France

Luc Pellissier* Noé Wagener†

January 29, 2026

Abstract

Specification of administrative algorithms, and tools for making them accountable to the public, represent both an undone science and an ineffective right. We aim at developping knowledge in this twin field —both in computer science, by studying the possible definitions, and in law science, by studying the extent to which such an accountability exists— by the way of the methodology of *strategic litigation*.

Programs cause many decisions having a great impact on our lives — be it by choosing which content to present to a user in a website, or by flagging an email as spam. In a more ominous way, programs are used by administrations[1] in a way that generates decisions which have legal standing, from the trivial (a program deciding whether someone can enter into a restricted area, such as a metro station enclosed by gates that require a badge) to the tedious (a program computing every month the different taxes to be paid) and the unnerving (a program deciding which welfare beneficiaries will be subjected to an extensive fraud investigation). As these programs get ubiquitous, the transparency of the administration —the fact that anyone can understand the decisions taken and possibly contest them—, one staple of the rule of law, seems to be in a state of flux.

An undone science. From the beginning of computer science (Turing 1949), the question of ensuring that a program is correct has been one of the drivers of many research programs, recognized of central importance by the whole community of theoretical computer scientists (as can be seen by the Turing awards to Pnueli in 1996, Clarke, Emerson and Sifakis in 2007, or the `Rocq` prover[2] development team receiving the ACM Software System Award in 2013). This subfield, known as *verification*, has, for the time being, mainly been applied to software in which a programming mistake could either result in human deaths (say, the fly-by-wire subsytem of a plane (Brière and Traverse 1993)) or in huge monetary losses (say, the design of microprocessors (Vardi 2014)), these two categories being dubbed *critical software*. In a perhaps surprising way, software used as a cog in an administrative machine, and on whose correction depends the legal situation of a state population, is not generally considered as critical, and its verification has remained largely undone: it focuses mainly on the impressive cases such as the use of Artificial Intelligence tools (Albarghouthi 2018; Richmond et al. 2023), as exemplified by the `Compas` controversy (Angwin et al. 2026), while ignoring the more mundane use of classical algorithms ((Monat, Fromherz, and Merigoux 2024) being one counterexample, concerned with date arithmetics). It must be remarked that, given the democratic component of these programs, their verification and specification should be able to be subjected to a democratic debate: we can guess that it will widely differ from the ones whose correction can be entrusted to experts[3].

*LACL (université Paris est Créteil) & IRIF (CNRS & université Paris Cité)

†MIL (université Paris est Créteil)

[1] Algorithmisation can actually be seen as a stage of bureaucratisation (Pégny 2024).

[2] At the time nown as the `Coq` prover.

[3] Leaving aside the debate on the possibility of democracy in a society characterized by its high degree of specialization of intellectual work.

An ineffective right. This situation is all the more surprising as the democratic problems originating in allowing decisions by programs were also foreseen before the generalization of computer in every administration. Indeed, the French law regulating the use of computers by public and private entities (1978's *loi informatique et liberté*, which could be translated as "computer science and freedom act") had, as its very name implies, an ambition much greater than just limiting the dissemination and reuse of personal data, but aimed at creating a general legal framework for automated procedures (Wagener 2023).

To keep the French example, the legal dispositions to have information on the software used by the administration multiplied over time: *source-codes* are considered administrative documents, and can thus be asked by anyone; for any "individual administrative decision founded on an algorithmic treatment", one can ask for "the rules defining the treatment as well as the main characteristics of its implementation"[4], and every person whose personal data has been used by an algorithmic treatment on which a decision (made by any entity, public or private) has been based can get "information allowing to know and contest the underlying logic of the treatment"[5].

Even if this legal framework can seem satisfactory, it remains completely ineffective (Ducros et al. 2025): it is indeed remarkable that very few people use this right in court, and when they do, how little the argument gets heard. Moreover, it is not even very clear what would constitute a standard of accountability for administrative algorithms (Merigoux et al. 2024).

Strategic litigation to bridge the gap. It is striking that an undone science stands next to a right that might be satisfying in the books but is ineffective in practice. We can, for the time being, only offer some hypotheses on the relationship between these two facts: the lack of a strong enough social force pushing for accountability of algorithmic administration may have caused the scientific community not to be interested in the subject; the lack of scientific research —and of scientific results— may have caused the civil society not to have enough resources to aim for this accountability.

It remains that to this day, the lack of research (on the computer science side) on specification for democratic accountability and the lack of litigation (by civil society in general) on the extent of the accountability of administrative algorithms leave us in a blind spot. We propose a method designed to both create standards of specification for administrative programs (which is both a computer science and legal science problem) and know the exact extent of current obligations of administrations: it consists of an agenda of *strategic litigations* in which we sue administrations so as to force courts to decide on the interpretation of the right to algorithmic transparency.

In the talk, we will give examples of such strategic litigations and update on their progress. Broadly speaking, we have probed two main subjects:

- the obligation for administrative documents to be disclosed in an "open standard, easily reusable and exploitable by an automated treatment system"[6], by asking different administrations for maps, executive orders, and pay slips (each querying different aspects of exploitation by an automated system);

- the specification of programs, by asking for the rules defining the treatment of the program that computes the remuneration of one of the authors. Indeed, the remuneration of French civil servants is entirely defined by legal texts, and then computed by a program, whose specification

[4] «*[U]ne décision individuelle prise sur le fondement d'un traitement algorithmique comporte une mention explicite en informant l'intéressé. Les règles définissant ce traitement ainsi que les principales caractéristiques de sa mise en œuvre sont communiquées par l'administration à l'intéressé s'il en fait la demande.*» (article L. 311-3-1 du Code des relations entre le public et l'administration).

[5] «*Les informations permettant de connaître et de contester la logique qui sous-tend le traitement automatisé en cas de décision prise sur le fondement de celui-ci et produisant des effets juridiques à l'égard de l'intéressé.* (article 119 de la loi informatique et libertés).

[6] «*[S]tandard ouvert, aisément réutilisable et exploitable par un système de traitement automatisé*» (article L. 300-4 du Code des relations entre le public et l'administration).

should obviously be included in the body of applicable law[7]. It is thus a situation in which a program is used to take administrative decisions, that is supposedly well-specified.

At this stage, we have no judgments and only various answers of administrations, which already give fascinating glimpses as to how they construe the programs they use.

As a preliminary finding, we remarked that, when they are aware of the implications of their use of computer programs, the administrations tend to use computer science arguments —such as security concerns if the system was to be opened too much[8]— to completely shut down the judicial debate, raising important questions on the relationship between technical expertise and the ability by civil society to contest decisions, and more broadly, giving a description of a new frontier of technocracy.

If our hypothesis that the right to program accountability is ineffective in part because the science of democratic verification is undone, and that the science is undone in part because there is no social force asking for it, our use of strategic litigation is to be understood as trying to broaden the questions that Society is even able ask, and thus create such a social force.

This work takes place in the broader setting of an interdisciplinary research program in computer science and law, centered around the a masters and doctoral program at the author's university[9], as well as a program with geographers, specifically on the legal representation of geographic data.

Bibliography

Aws Albarghouthi (2018). "Fairness: A Formal-Methods Perspective". In: *Static Analysis*. Ed. by Andreas Podelski. Cham: Springer International Publishing, pp. 1–4.

Julia Angwin, Jeff Larson, Surya Mattu, and Kirchner Lauren (May 2026). *Machine Bias. There's software used across the country to predict future criminals. And it's biased against blacks*. ProPublica. URL: `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`.

Dominique Brière and Pascal Traverse (1993). "Airbus A320/A330/A340 Electrical Flight Controls: A Family of Fault-Tolerant Systems". In: *Digest of Papers: FTCS-23, The Twenty-Third Annual International Symposium on Fault-Tolerant Computing, Toulouse, France, June 22-24, 1993*. IEEE Computer Society, pp. 616–623.

Philippine Ducros, Maxime Zimmer, Luc Pellissier, and Noé Wagener (2025). "L'ineffectivité du droit d'accès à l'information sur les algorithmes: une étude empirique". In: *Revue des droits et libertés fondamentaux* **2025**.Constitutionnalisme numérique et approches critiques en droit et technologie. Ed. by Manon Altwegg-Boussac and Afroditi Marketou, Chronique n° 45.

Denis Merigoux, Marie Alauzen, Justine Banuls, Louis Gesbert, and Émile Rolley (2024). *De la transparence à l'explicabilité automatisée des algorithmes : comprendre les obstacles informatiques, juridiques et organisationnels*. Tech. rep. 9535. INRIA.

Raphaël Monat, Aymeric Fromherz, and Denis Merigoux (Apr. 2024). "Formalizing Date Arithmetic and Statically Detecting Ambiguities for the Law". In: *Lecture Notes in Computer Science*. Ed. by Stephanie Weirich. Vol. 14577. Lecture Notes in Computer Science. Luxembourg City, Luxembourg: Springer Nature Switzerland, pp. 421–450.

Maël Pégny (2024). *Éthique des algorithmes et de l'Intelligence Artificielle*. Vrin, p. 322.

[7] Of course, it is not true when looking at the finer details, such as the floating point computations.

[8] The most remarkable such example we know of being the *Cour de cassation* refusing to publicize its machine learning model used to anonymize judicial decisions, because "in the current state of scientific knowledge" there is no proof that it could not be used to de-anonymize the decisions used to train the model, which would be a privacy violation, see (CADA, n° 20230314, 30/03/2023). While entirely plausible a concern, this is very problematic regarding the burden of proof: how can anyone prove that a model to which they have no access cannot possibly be used in a certain way? This can also be generalized to any machine learning model: any administration can allege the presence of sensitive data in the training set and the possibility to recover it from the model to escape from its transparency requirements.

[9] LIEN VERS LE SITE DU MASTER

Karen McGregor Richmond, Satya M. Muddamsetty, Thomas Gammeltoft-Hansen, Henrik Palmer Olsen, and Thomas B. Moeslund (2023). "Explainable AI and Law: An Evidential Survey". In: *Digital Society* **3** (1).

Alan Turing (1949). "Checking a Large Routine". In.

Moshe Y. Vardi (2014). "From Löwenheim to PSL and SVA". In: *Language, Culture, Computation. Computing - Theory and Technology - Essays Dedicated to Yaacov Choueka on the Occasion of His 75th Birthday, Part I*. Ed. by Nachum Dershowitz and Ephraim Nissan. Vol. 8001. Lecture Notes in Computer Science. Springer, pp. 78–102.

Noé Wagener (2023). "L'ambition déçue de la loi Informatique & Libertés de réglementer les traitements algorithmiques". In: *1024. Bulletin de la Société informatique de France* 22, pp. 21–31.