A Comparative Study of Open-Source Fuzzy Modelling Toolkit Licenses and Features

Felipe Arruda Pontes
Insight Centre for Data Analytics
University of Galway
Galway, Ireland
0000-0002-9062-7653

Edward Curry
Insight Centre for Data Analytics
University of Galway
Galway, Ireland
City, Country
0000-0001-8236-6433

Michael Schukat
School of Computer Science
University of Galway
Galway, Ireland
City, Country
0000-0002-6908-6100

***Abstract*—The practice of distributing software as open-source, which serves as a pillar for Open Science, can benefit future research in the field of fuzzy logic by enabling access and increasing the maintainability of free fuzzy logic frameworks alternatives to closed commercial solutions. Most open-source software uses a distribution license, with a wide range of restrictions and requirements on the distributions of these tools and their derived works. Therefore, understanding each license limitation and requirement is essential to ensure an ethical research process and avoid legal issues from breaching the open-source license. However, there is a lack of comparative research on the current state of open-source licenses for fuzzy modelling toolkits and frameworks. Previous studies are outdated or incomplete, missing some of the most popular open-source fuzzy frameworks or a detailed analysis of the licenses used. Our research analysed 20 of the most prominent open-source toolkits and categorised them based on their licenses, with qualitative comparisons of their characteristics and limitations. In our study, we note a concerning scenario where a potential license violation may impede the publication and dissemination of academic works derived from fuzzy toolkits using the GPL license. Additionally, we identify a gap in open-source fuzzy toolkits with support for Fuzzy Control Language (FCL) and Type-2 fuzzy sets. Finally, this study aims to support and guide future researchers in selecting the fuzzy toolkit that best matches their research problem without infringing the open-source licenses used, with an intuitive and up-to-date overview of the licenses and features of open-source fuzzy modelling frameworks available.**

I. Introduction

Fuzzy Set theory (FST) was first introduced in Zadeh's classical article [1] as a tool for handling the uncertainties and different ranges of values and meanings associated with natural language and human logical reasoning. At its core, FST defines human reasoning as fuzzy and imprecise, thus requiring a new method to retain value from this so-called *"fuzziness"*. Together with Neural Networks and evolutionary computing as pillars of the Soft Computing research area, FST can be applied to multiple problems dependent on variables with imprecise definitions and measurements and can be used in complex and dynamic environments [2], [3]. Recent years have seen increased growth in open-source software and tools in academia and industry [4], which presents many advantages for future research in the fuzzy logic field [5]. These open-source software are commonly distributed according to specific licenses. Violating these licenses can be seen as unethical and may lead to multiple legal issues, as happened in the SCO Group (2003) case [6]. These licenses have different requirements and restrictions, which can be hard to understand and may even pose concerning limitations on the dissemination of academic works. Therefore, to avoid legal issues and ensure an ethical research methodology, researchers need to understand their responsibilities and the impact different licenses can have on the artefacts produced by their research. However, when working with fuzzy logic, there are no recent studies on the differences between open-source licenses for the available toolkits and frameworks. In [7], multiple fuzzy toolkits are compared regarding their license, release date, programming language, user interface, and use of the Fuzzy Control Language (FCL) standard [8]. However, they do not consider fuzzy set types in their comparative study. [5] conducts an extensive survey of fuzzy logic software. However, its analysis of the open-source licenses is very brief, and no software engineering metrics for open-source projects are analysed (e.g., presence of unit tests, extensive documentation, and sustained maintainability) [9]. Additionally, both studies are potentially outdated and do not include recent and prominent fuzzy logic frameworks such as Scikit-Fuzzy [10] or eFLL [11]. In response to these issues, our research proposes an up-to-date qualitative comparison of the most prominent open-source fuzzy logic frameworks and toolkits regarding the type of open-source license, popularity, programming language, good software engineering practices, and some fuzzy logic modelling features and limitations. As a result, we note that there is a low number of open-source toolkits with support for Type-2 fuzzy sets and a gap in fuzzy toolkits that can support Type-2 sets and FCL without the need for third-party solutions. Additionally, we identify a possible license issue researchers may face when publishing an article on extensions of fuzzy toolkits using the GPL license. This possible breach of license only reinforces the importance of

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2, co-funded by the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

choosing a fuzzy toolbox with a license compatible with the researcher's expected dissemination of academic work.

The structure of this paper is as follows: First, we present a background of Fuzzy Set Theory and open-source licenses. Next, we formalise our methodology for conducting our research and qualitative analysis. Then, we categorise the open-source projects based on their licenses and analyse their differences. Finally, we offer our concluding remarks and what are the main issues and gaps in the available open-source fuzzy modelling frameworks.

II. Background

A. Fuzzy Set Theory

FST is a powerful mathematical tool for modelling real-world problems where uncertainties are present. As defined by its creator, Zadeh, fuzzy sets' core concept stands from the fact that real-world sets are *"fuzzy"* and not precisely defined. It extends classical set theory by having a membership of elements in a fuzzy set defined in degrees of truth instead of binary terms. This difference means that in classical sets (a.k.a. crisp sets), an element either belongs or does not to a set (i.e., represented by 0 or 1). In contrast, in fuzzy sets, an element membership can be partial (i.e., represented by a real number in the interval of $[0, 1]$) [1].

As they were initially defined, fuzzy sets are also called Type-1 fuzzy sets, which early on faced some criticism concerning the level of uncertainty they could handle. As a response, Zadeh defined the Type-2 fuzzy sets, incorporating a more elaborate and comprehensive uncertainty representation in the FST [12].

The fuzzy linguistic variables and their modifiers are other central elements in fuzzy logic. The first represents words and terms instead of numerical variables. The modifiers are similar to adjectives and adverbs, in which a modifier changes the fuzzy membership functions related to a fuzzy linguistic variable [1].

Fuzzy Inference Systems (FIS) are systems based on the FST, employing an inference process known as approximate reasoning through fuzzy *if-then* rules and known facts [13]. Moreover, the only globally accepted standard for modelling FIS is the FCL [5], as published in the IEC 61131-7 [8]. Although, more recent standards such as the Fuzzy Markup Language (FML) [14], have surfaced with the promise to improve and substitute FCL [5], having already received the support of the IEEE Computational Intelligence Society [15]. FIS are generally broken into the following steps:

- **Fuzzification**: Where the system maps the precise numeral (crisp) input values into fuzzy sets outputs, according to the fuzzy set's membership functions definitions;
- **Inference**: Where the fuzzy rules are used to transform the crisp variables into fuzzy variables according to specific inference mechanisms;
- **Defuzzification**: Step where the system maps the outputs of the fuzzy set into a crisp value.

B. Open-Source Licenses

The choice of the open-source license is an essential factor when choosing a fuzzy logic framework to not cause any breach in the license and to ensure the author is aware of the liabilities and requirements imposed based on the open-source license. These licenses are usually divided into two main groups: *copyleft* and *permissive licenses*. Copyleft licenses, such as the GNU General Public License (GPL), usually have a *"share-alike"* clause that requires derivative works of a project/tool to be distributed using a copyleft license compatible with the license from the original project. Meanwhile, the permissive licenses (e.g., MIT license) have no such copyleft *share-alike* requirements, are less restrictive on the use and rules of distribution of derivative works, and tend to be shorter and more straightforward to interpret.

There is also a third, though smaller category, denominated as *weak copyleft*, which stands in the middle ground between copyleft and permissive licenses in terms of obligations, liabilities and simplicity.

Moreover, when choosing an open-source fuzzy toolkit, it is essential to note that, in many cases, it is possible to ask the author of the toolkit for a version to be released under a different open-source license. Therefore, this communication is an excellent first action a researcher should explore before giving up on using a toolkit because of its license.

III. Methodology

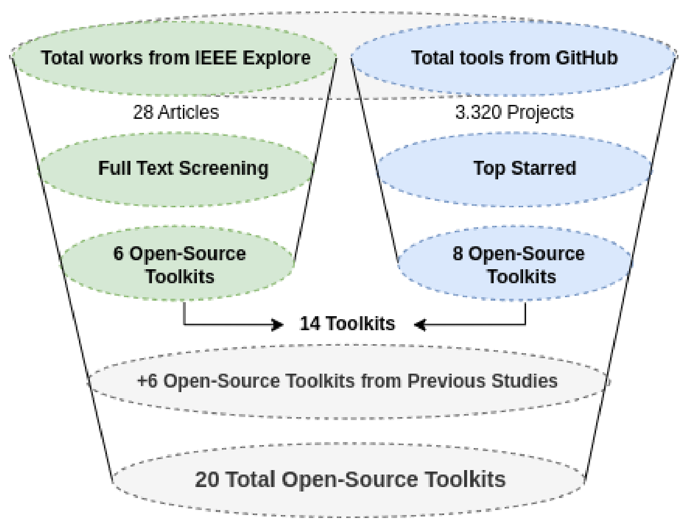

Fig. 1. Overview of the methodology used, in which the final results on the selected database where extended with the 8 most popular projets from Github. Finally, we also add toolkits from previous fuzzy toolbox comparative studies.

For this study, we apply a similar methodology of querying an academic research database to that of [5], [16] but extend the search query with results for open-source works that are publicly available but not necessarily published. The frameworks included in this paper have been chosen based on literature reviews. We consider the most prominent fuzzy frameworks based on the perceived industry and academic acceptance, whereby we query a major online open-source software repository (GitHub) for the most popular fuzzy logic projects. The frameworks, toolkits and toolboxes were only included in this research if considered generic, reusable, and not use-case specific, and with some measure of fuzzy modelling capability. We also limit the inclusion to works that still have an accessible open-source code at the moment of writing this study and limit to the fuzzy frameworks themselves without including works that are extensions of existing fuzzy frameworks. An overview of this methodology can be seen in Figure 1.

First, we used GitHub's search API to query for projects (repositories) with the string *"fuzzy logic"*, ordered by the number of stars (i.e., the popularity). Given the scope and limit of text space for this study, we selected only top 8 fuzzy frameworks from this query using previously defined inclusion criteria. Then, we expanded the search using the IEEE Xplore database for works containing the word *"fuzzy"* closely followed by the string *"tool*"* or *"framework"* and that contain the word *"open-source"*, using the following query:

*("All Metadata":**fuzzy**) NEAR/3 (("All Metadata":**tool***) OR ("All Metadata":**framework**)) AND (("All Metadata":**open-source**) OR ("All Metadata":**open source**)).*

Finally, we include fuzzy logic frameworks mentioned in previous comparative studies [5], [7], while following the same inclusion criteria.

The open-source tools were categorised according to their different types of licenses, and then analysed according to the following criteria, which cover both their popularity, good software engineering practices, and fuzzy features:

- **License**: What kind of open-source distribution license is used;
- **Language**: What programming language the framework is built on;
- **Number of Stars**: In GitHub, any registered user may give a single star to any project. Borges et al. study [17] show that the number of stars in a GitHub project is a good representation of the perceived acceptance and popularity of an open-source project. Therefore, the higher the number of stars, the higher the perceived acceptance of the fuzzy framework;
- **Last Activity Date**: Date of the last change in the tool, used here to indicate the current state of support of the project and as a measure of more sustained solution [9];
- **Uses Tests**: If the framework makes use of tests (e.g., unit tests) to ensure a more reliable quality-driven solution [9];
- **Documentation**: To what extent is the framework documented (as a measurement of maintainability), whereby we check if it has examples, API documentation (i.e., classes and methods documented) and a User Guide [9];
- **Fuzzy Set Types**: Based on the study of [5] we note that there are not many tools for Type-2 fuzzy sets, therefore we choose to analyse if frameworks in our study have support for both Type-1 and Type-2 fuzzy sets;
- **FCL Support**: As noted in [5], an universal Fuzzy Control Language is highly important for the interoperability of FISs, therefore decided to analyse what kind of support (if any) does the framework have for a Fuzzy Control Language?

Using this methodology, we found a total of 3.320 projects on GitHub, from which we selected the top 8 (with the most stars). Moreover, we see that the most used programming language was Python, with 722 projects; Java, with 362 projects; and C++, with 263 projects. The top 8 projects with the most stars from the GitHub search were: Scikit-Fuzzy [10], FuzzyLite [18], eFLL [11], fuzzylogic [19], simpful [20], FuzzyCLIPS [21], jFuzzyLogic [22], PyIT2FLS [23]. From the IEEE Xplore database query, we obtained a total of 28 results, of which six were included after filtering out results that did not meet our inclusion criteria. From this list, JFML [24] and FuzzyR [25] had projects on GitHub, while Guaje [26], Octave Fuzzy Toolbox [27], Juzzy [28] and fuzzycreator [29] are hosted in other locations. Finally, we included six more projects from previous fuzzy toolbox comparative studies [5], [7]: FisPro [30], XFuzzy [31], FLT [32], AForge.NET [33], DotFuzzy [34] and Pyfuzzy [35]. Of these last six, only XFuzzy was not available on GitHub. For Pyfuzzy it was used a mirror (copy of the repository) available on GitHub. As a final count, 20 fuzzy logic modelling frameworks were selected for the purpose of this comparative study. Table I shows the list of these frameworks with their open-source code repository locations and from where it was retrieved during our search process (i.e., GitHub API, IEEE Xplore or previous research).

A. Overview of the Fuzzy Frameworks Comparison

We present in this section (see Table II) an overview of all the fuzzy frameworks analysed in this study. The projects were grouped by their open-source license type, and compared in regards to their: number of stars (stars); the last activity date (Updated); programming language (Lang.); use of unit tests (Test); extent of documentation (Doc); support fuzzy set types (Types); and support for FCL. It is important to note that the projects not available on GitHub were also included in this table with their stars count set to "-". Therefore, a project with zero stars in the table signifies that the project is available on GitHub, but has a total of zero stars.

IV. Open-Source Fuzzy Modelling Toolkits Classification

To better understand the most popular open-source fuzzy toolboxes and frameworks, we classify them according to their type of distribution license (i.e., permissive or copyleft/weak-copyleft licenses). We analyse each group according to each toolkit's popularity (above and below the average number of stars), maintainability status, good practices of software engineering (documentation and tests), support of FCL, and the types of fuzzy sets supported. As mentioned before, most of the frameworks and toolboxes analysed had their source repository (or a mirror) hosted on GitHub, which allowed us to

TABLE I
ORIGIN OF THE FUZZY LOGIC FRAMEWORKS COMPARED IN THIS STUDY

Name	Retrieved From	Repository
Scikit-Fuzzy	GitHub API	https://github.com/scikit-fuzzy/scikit-fuzzy
FuzzyLite	GitHub API	https://github.com/fuzzylite/fuzzylite
eFLL	GitHub API	https://github.com/alvesoaj/eFLL
fuzzylogic	GitHub API	https://github.com/amogorkon/fuzzylogic
simpful	GitHub API	https://github.com/aresio/simpful
FuzzyCLIPS	GitHub API	https://github.com/rorchard/FuzzyCLIPS
jFuzzyLogic	GitHub API	https://github.com/pcingola/jFuzzyLogic
PyIT2FLS	GitHub API	https://github.com/Haghrah/PyIT2FLS
JFML	IEEE Xplore	https://github.com/sotillo19/JFML
FuzzyR	IEEE Xplore	https://github.com/cran/FuzzyR
Guaje	IEEE Xplore	https://gitlab.citius.usc.es/jose.alonso/guaje
Octave Fuzzy	IEEE Xplore	https://sourceforge.net/p/octave/fuzzy-logic-toolkit
Juzzy	IEEE Xplore	http://juzzy.wagnerweb.net/
fuzzycreator	IEEE Xplore	https://bitbucket.org/JosieMcCulloch/fuzzycreator
FisPro	Prev. Res.	https://github.com/cran/FisPro
XFuzzy	Prev. Res.	http://www2.imse-cnm.csic.es/Xfuzzy/ †
FLT	Prev. Res.	https://github.com/ajavibp/FLT
AForge.NET	Prev. Res.	https://github.com/andrewkirillov/AForge.NET
DotFuzzy	Prev. Res.	https://github.com/MicheleBertoli/DotFuzzy
Pyfuzzy	Prev. Res.	https://pyfuzzy.sourceforge.net ‡

† Requires registration to access source code.
‡ A mirror repository on GitHub was used instead: https://github.com/arruda/pyfuzzy

TABLE II
OVERVIEW OF THE OPEN-SOURCE FUZZY LOGIC FRAMEWORKS

Name	Stars	Updated	Lang.	License	Test	Doc*	Types	FCL
Fuzzy Modelling Frameworks Using Permissive License								
Scikit-Fuzzy	631	2022	Python	BSD(3-c)	Yes	Eg/API/UG	1	None†
eFLL	170	2021	C++**	MIT	Yes	Eg/UG	1	None
fuzzylogic	70	2022	Python	MIT	Yes	Eg	1	None
PyIT2FLS	44	2021	Python	MIT	No	Eg/API/UG	1 & 2	None
DotFuzzy	5	2017	C#	MIT	No	API/UG	1	FCL
XFuzzy	-	2020	Java	BSD(3-c)	Yes	Eg/API/UG	1	XFL
Juzzy	-	2014	Java	BSD(3-c)	No	Eg	1 & 2	None
Fuzzy Modelling Frameworks Using Copyleft or Weak-Copyleft License								
AForge .NET	923	2020	C#	LGPL-3 GPL-3	Yes	Eg/API/UG	1	None
FuzzyLite	210	2021	C++	GPL-3	Yes	Eg/API/UG	1	FCL
simpful	61	2022	Python	GPL-3	Yes	Eg/API	1	None
jFuzzyLogic	56	2016	Java	LGPL-3	Yes	Eg/API/UG	1	FML
JFML	23	2021	Java	GPL-3	Yes	Eg/API/UG	1	FML
Pyfuzzy	10	2012	Python	LGPL-3	No	Eg/API/UG	1	FCL
FuzzyR	1	2021	R	GPL-2	No	Eg/API/UG	1 & 2	None
FisPro	0	2022	R	CeCILL-2.1	Yes	Eg/API/UG	1	FIS
FLT	0	2021	C++	GPL-3	No	Eg/API/UG	1	FIS
OctaveFuzzy	-	2021	Octave	GPL-3	Yes	Eg/API/UG	1	FIS, M
fuzzy creator	-	2020	Python	GPL-3	Yes	Eg/API/UG	1 & 2	None
Guaje	-	2020	Java	GPL-3	Yes	Eg/API/UG	1	XFL, FIS, PLA, KB, XML
Fuzzy Modelling Frameworks Without License								
FuzzyCLIPS	62	2015	CLIPS	E/R‡	No	Eg/API/UG	1	None

* Eg, API, and UG are: Examples, API documentation, and User Guide, respectively.
** Specifically for Embedded Systems, such as Arduino and micro-controllers
† FCL support is mostly implemented and planned for future releases.
‡ Free for education and research purposes.

compare their popularity based on their number of stars [17]. The exceptions were the frameworks: Octave Fuzzy Toolbox, XFuzzy, fuzzycreator, Guaje and Juzzy. The number of stars of the GitHub-hosted fuzzy frameworks was, on average, 151 stars. Finally, it is also worth mentioning that the FuzzyCLIPS framework does not have a license. However, the author of the toolbox commented that they had allowed it to be used for free for research or educational purposes by other organisations[1].

[1]FuzzyCLIPS's Author comments on the license: https://github.com/rorchard/FuzzyCLIPS/pull/2/#issuecomment-30904479

A. *Fuzzy Toolkits with Permissive Licenses*

There are seven projects with permissive licenses, in which the eFLL, fuzzylogic, PyIT2FLS, and DotFuzzy use the MIT license. Using the BSD license, we have Scikit-Fuzzy, XFuzzy, and Juzzy, which indicates a good balance of projects with MIT and BSD licenses.

In this study, only projects with the BSD with 3 Clauses are present (i.e., BSD-3c). Some of the main differences between the MIT and the BSD-3c are in the clauses regarding advertising and promotion, in which the MIT explicitly allows publishing, sub-licensing and selling, while BSD-3c has this implicitly. Both licenses are compatible, meaning it is possible to use code from a project under one license and use it in a project with the other.

Regarding popularity, the most popular projects of each permissive license are Scikit-Fuzzy and eFLL, with 631 and 170 stars, respectively. Python is the most common language used on fuzzy toolbox with permissive licenses, with three projects. Meanwhile, most toolkits with permissive licenses have good maintainability status with recent updates (last three years) in their code base, with only two projects having a lower maintainability status: Dotfuzzy and Juzzy, with the last update in the toolboxes in the years 2017 and 2014 respectively. However, only three of these projects (i.e., Scikit-Fuzzy, PyIT2FLS and XFuzzy) have extensive documentation, with examples, API documentation and a user guide, which are critical when extending or using these toolboxes in future research. Additionally, almost half of the projects lack unit tests, which is an important measure of code quality.

Concerning fuzzy modelling features, we see that the support of FCL is not covered by most of these toolboxes, with only the DotFuzzy and XFuzzy having this feature, in which the first implements a more strict representation of the FCL standard. In contrast, the latter implements its own XFL format. Meanwhile, the Scikit-Fuzzy support for FCL is being finalised and planned for future releases. Moreover, most projects using permissive licenses implement only Type-1 fuzzy sets, with PyIT2FLS and Juzzy being the only two to support both Type-1 and Type-2. Finally, we also note that no projects with permissive licenses support FLC in addition to both types of fuzzy sets.

Issues and Considerations of Permissive Licenses: One of the most important points one must understand when choosing a fuzzy toolkit using a permissive license is that these licenses do not guarantee derivative works or future versions from being released as closed-source. In terms of academic works, this fact could make future improvements on the toolkits no longer being freely available to researchers if the developers of such features desired so. The Free Software Foundation (FSF) notes that using such licenses may lead to the risk of having to compete with a closed-source version of your own work. Finally, these licenses are commonly considered low-risk in terms of legal issues. The authors of this paper could not find any case where the use of a permissive license limited the dissemination of academic works.

B. *Fuzzy Toolkits with Copyleft Licenses*

The copyleft (i.e., GNU GPL and CeCILL) licensed projects cover about half of the available software (10 projects). The GNU GPL version 3 (GPL-3) is the most common among them, with only the FuzzyR and FisPro frameworks using the older GNU GPL version 2 (GPL-2) and CeCILL version 2.1 (CeCILL-2.1), respectively. The CeCILL license is mostly an extension of the GPL adapted to French legal matters. Of weak copyleft licenses (i.e., LGPL), we observed three frameworks using the GNU LGPL version 3 (LGPL-3). The AForge.NET dual licensing is noteworthy, in which the project is licensed under LGPL-3 and GPL-3 since they are obliged to add a GPL-2 compatible license. This dual licensing happens because of a sub-component that uses a GPL-2 licensed library (FFMPEG). It is important to note that, not all of these licenses are compatible with each other (e.g., GPL-2 is not compatible with GPL-3 or LGPL). Additionally, the researcher must first understand exactly what constitute a software, a library or a statically linked library according to these licenses, since these definitions may also apply restrictions of use and compatibility with other licenses[2].

Even with these restrictions, the fuzzy toolboxes using copyleft licenses are the most common and the most popular of all the projects explored in this study. AForge.Net is the highest-starred project, with a total of 923 stars, while FuzzyLite is the third most popular project, with 210 stars. These projects also cover a wide range o programming languages, with Python and Java being the most common. Moreover, most of these projects with copyleft (and weak-copyleft) licenses have a good maintainability status, with the last update on their code made in the last three years, with only two projects with a lower maintainability status: jFuzzyLogic, and pyfuzzy, with last the update on 2016, and 2012, respectively. Most projects have extensive documentation with examples, API documentation, and a user guide. The only exception is the project simpful, which still has good documentation coverage and lacks only a user guide. Similarly, most projects have unit tests to improve the quality of the code and ensure the correctness of the fuzzy solutions available, the only exceptions being FLT, FuzzyR, and pyfuzzy. This is a high contrast to the good software engineering practices that are not so commonly observed in the toolboxes using permissive licenses.

Additionally, there are many types of FCL supported by these toolboxes. The most common fuzzy control language is the FIS (similar to the Matlab format). Guaje is the most comprehensive solution for working with different FCLs, since it can work with and translate up to five different fuzzy language formats, including formats from other tools such as XFuzzy. The FuzzyLite project implements the strict IEC 61131-7 FCL standard, while JFuzzy-Logic and JFML both use a new standard, the Fuzzy Modeling Language (FML). The

[2]GPL licenses compatibility matrix: https://www.gnu.org/licenses/gpl-faq.en.html#AllCompatibility

only copyleft tools that do not support FCL are AForge.Net, simpful, FuzzyR, and fuzzycreator.

Moreover, most copyleft tools focus only on Type-1 fuzzy sets, with only the FuzzyR and fuzzycreator supporting Type-1 and Type-2. We see a similarity to the toolkits with a permissive license, in which no copyleft (or weak-copyleft) projects support FCL and both fuzzy set types.

Issues and Considerations of Copyleft Licenses: Copyleft licenses are generally considered high-risk in legal issues, given their extensive length and use of legal jargon and complex definitions. On the one hand, the *"share-alike"* clause of copyleft licenses can benefit academic works by ensuring that derived works are always free and open-source. On the other hand, this also may give rise to specific scenarios in which the publication of academic articles may lead to a breach of the distribution license. The authors of this paper found that there is a grey area in which this breach may occur when a journal reviewer requests the source code of an article concerning a derived work of a fuzzy toolkit using the GPL license and when the authors do not wish to make publicly available before publication of the article. The synthesis of this complex scenario[3] is that most journals implement good privacy policies that ensure the confidentiality of the research artefacts during the peer review. These policies are a vital aspect of the peer-review process. However, with the code being a GPL-derived work, the same license should apply, which implies that when the author shares the code with the reviewer, they cannot impose any limitations on their freedom of sharing their research source code, and to do so would constitute a violation of the license. Given the complex nature of the GPL license, it would require a legal expert to affirm who is breaching the license, if it is the researcher submitting the work or the journal publisher, and to what extent this is considered a violation of the GPL license.

As mentioned before, although it is possible to contact an author of a GPL-based toolkit in order to ask for the toolkit to be released in a different open-source license, the author may not be able to comply with that request, even if they want to. This can happen if the toolkit itself makes use of another GPL-based software, and therefore it has to maintain a compatible copy-left license. This is the case with AForge.Net, in which the tool is bound to use GPL because of the use of a GPL library in one of its modules.

Finally, given the ethical concerns of a possible license violation when disseminating academic works of fuzzy toolkits using copyleft licenses, we recommend future research on copyleft-licensed fuzzy toolkits to carefully investigate and fully understand the risks and requirements before working with these licenses.

V. Conclusion

In this paper, we presented an up-to-date qualitative comparison of the open-source licenses of popular open-source fuzzy modelling frameworks and toolkits. We analysed the toolkits according to their popularity, programming language, good software engineering practices, and fuzzy logic modelling features and limitations. However, a limitation of our study is that the popularity metric penalises toolkits not hosted on Github or equivalent public open-source social platforms. In the future, alternative popularity metrics could be explored to mitigate this issue. We are glad to report that toolkit alternatives are available in multiple programming languages, some of which were not listed in previous studies, and that most of the frameworks analysed use good software engineering practices with unit tests and extensive documentation. Moreover, when analysing the open-source distribution license used by the frameworks, we noted that fewer open-source alternatives are using permissive than copyleft licenses. We understand that there is a need for more recently maintained tools with permissive licenses that also have FCL support. Furthermore, we conclude that, without the use of external modules, there is a low number of open-source toolkits with support for Type-2 fuzzy sets and a gap in open-source toolkits that can support both Type-2 sets and FCL. Additionally, we raise awareness of a possible license violation that may impede researchers from publishing works extending open-source fuzzy toolkits when using the GPL license. Finally, we believe that our study can help future researchers understand the requirements and ethical obligations of the open-source distribution licenses that best fit their needs and academic work dissemination expectations when selecting an open-source fuzzy modelling framework.

[3] An extensive discussion on this specific scenario: https://opensource.stackexchange.com/questions/13473/does-this-constitute-a-breach-in-gpl-license-article-with-gpl-licensed-software

References

[1] L. Zadeh, "Zadeh, fuzzy sets," *Inform Control*, vol. 8, pp. 338–353, 1965.

[2] L. A. Zadeh, "Fuzzy logic, neural networks, and soft computing," in *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*. World Scientific, 1996, pp. 775–782.

[3] T. J. Ross, *Fuzzy logic with engineering applications*. John Wiley & Sons, 2005.

[4] J. Willinsky, "The unacknowledged convergence of open source, open access, and open science," 2005.

[5] J. Alcalá-Fdez and J. M. Alonso, "A survey of fuzzy systems software: Taxonomy, current research trends, and prospects," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 1, pp. 40–56, 2015.

[6] R. Kemp, "Current developments in open source software," *Computer Law & Security Review*, vol. 25, no. 6, pp. 569–582, 2009.

[7] D. M. Arruda, G. M. D. Abud, F. A. Pontes, R. M. Pontes, and B. B. F. de Oliveira, "Análise comparativa de ferramentas computacionais para modelagem de lógica fuzzy." *X Simpósio de Excelência em Gestao e Tecnologia, Resende*, 2013.

[8] I. E. Commission *et al.*, "IEC 1131-programmable controllers. part 7-fuzzy control programming," *Committee Draft CD*, vol. 1, 1997.

[9] N. Munaiah, S. Kroh, C. Cabrey, and M. Nagappan, "Curating github for engineered software projects," *Empirical Software Engineering*, vol. 22, pp. 3219–3253, 2017.

[10] J. Warner, J. Sexauer, Twmeggs, alexsavio, A. Unnikrishnan, G. Castelão, F. A. Pontes, T. Uelwer, Laurazh, F. Batista, Alexbuy, W. V. den Broeck, W. Song, R. A. M. Pérez, J. F. Power, H. Mishra, G. O. Trullols, and A. Hörteborn, "Jdwarner/scikit-fuzzy: Scikit-Fuzzy version 0.4.2," Nov. 2019. [Online]. Available: https://github.com/scikit-fuzzy/scikit-fuzzy

[11] A. J. Alves, R. Lira, M. Lemos, D. S. Kridi, and K. Leal, "eFLL (embedded fuzzy logic library)," last accessed 30 Sep 2022. [Online]. Available: https://github.com/zerokol/eFLL

[12] L. A. Zadeh, "The concept of a linguistic variable and its application to approximate reasoning—i," *Information sciences*, vol. 8, no. 3, pp. 199–249, 1975.

[13] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [book review]," *IEEE Transactions on automatic control*, vol. 42, no. 10, pp. 1482–1484, 1997.

[14] G. Acampora, V. Loia, C.-S. Lee, and M.-H. Wang, *On the power of fuzzy markup language*. Springer, 2013.

[15] I. S. Association *et al.*, "Ieee standard for fuzzy markup language," *The Institute of Electrical and Electronics Engineers, Inc., New York, USA, Standard IEEE Std*, vol. 2016, 1855.

[16] M. Loor and G. De Tre, "An open-source software package to assess similarity measures that compare intuitionistic fuzzy sets," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–6.

[17] H. Borges and M. T. Valente, "What's in a github star? understanding repository starring practices in a social coding platform," *Journal of Systems and Software*, vol. 146, pp. 112–129, 2018.

[18] J. Rada-Vilela, "The FuzzyLite libraries for fuzzy logic control," last accessed 30 Sep 2022. [Online]. Available: https://www.fuzzylite.com/

[19] A. Kiefner, "FuzzyLogic for Python," Feb. 2022.

[20] S. Spolaor, C. Fuchs, P. Cazzaniga, U. Kaymak, D. Besozzi, and M. Nobile, "Simpful: fuzzy logic made simple," *Int J Comput Intell Syst*, vol. 13, no. 1, pp. 1687–1698, 2020.

[21] R. Orchard, "FuzzyCLIPS version 6.04 a user's guide," 1998.

[22] P. Cingolani and J. Alcalá-Fdez, "jFuzzyLogic: a java library to design fuzzy logic controllers according to the standard for fuzzy control programming," *International Journal of Computational Intelligence Systems*, vol. 6, no. sup1, pp. 61–75, 2013.

[23] A. A. Haghrah and S. Ghaemi, "PyIT2FLS: A new python toolkit for interval type 2 fuzzy logic systems," *arXiv preprint arXiv:1909.10051*, 2019.

[24] J. M. Soto-Hidalgo, J. M. Alonso, G. Acampora, and J. Alcalá-Fdez, "JFML: a java library to design fuzzy logic systems according to the IEEE std 1855-2016," *IEEE Access*, vol. 6, pp. 54 952–54 964, 2018.

[25] T. R. Razak, C. Chen, J. M. Garibaldi, and C. Wagner, "Designing the hierarchical fuzzy systems via FuzzyR toolbox," in *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2021, pp. 1–6.

[26] D. P. Pancho, J. M. Alonso, and L. Magdalena, "Quest for interpretability-accuracy trade-off supported by fingrams into the fuzzy modeling tool GUAJE," *International Journal of Computational Intelligence Systems*, vol. 6, no. sup1, pp. 46–60, 2013.

[27] L. Markowsky and B. Segee, "The octave fuzzy logic toolkit," in *2011 IEEE International Workshop on Open-source Software for Scientific Computation*. IEEE, 2011, pp. 118–125.

[28] C. Wagner, "Juzzy-a java based toolkit for type-2 fuzzy logic," in *2013 IEEE Symposium on Advances in Type-2 Fuzzy Logic Systems (T2FUZZ)*. IEEE, 2013, pp. 45–52.

[29] J. McCulloch, "Fuzzycreator: A python-based toolkit for automatically generating and analysing data-driven fuzzy sets," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–6.

[30] S. Guillaume and B. Charnomordic, "Fuzzy inference systems: An integrated modeling environment for collaboration between expert knowledge and data using FisPro," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8744–8755, 2012.

[31] I. Baturone, F. J. Moreno-Velo, S. Sánchez-Solano, Á. Barriga, P. Brox, A. A. Gersnoviez, and M. Brox, "Using xfuzzy environment for the whole design of fuzzy systems," in *2007 IEEE International Fuzzy Systems Conference*. IEEE, 2007, pp. 1–6.

[32] J. M. A. Márquez, A. J. B. Piña, and M. E. G. Arias, "A general and formal methodology to design stable nonlinear fuzzy control systems," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 5, pp. 1081–1091, 2009.

[33] A. Kirillov, A. Gustafsson, S. J., and C. Millard, "AForge.NET," last accessed 30 Sep 2022. [Online]. Available: https://github.com/andrewkirillov/AForge.NET

[34] M. Bertoli, "DotFuzzy," last accessed 30 Sep 2022. [Online]. Available: https://github.com/MicheleBertoli/DotFuzzy

[35] R. Liebscher, M. Vollmer, and P. Lafoucrière, "Pyfuzzy," last accessed 30 Sep 2022. [Online]. Available: https://github.com/arruda/pyfuzzy