iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83478

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

DepGuard: An Automated Framework for Intellectual Property Compliance and Licensing Risk Assessment for Open-Source AI Dependencies

Kushal Gowda C[1], Kiran Kumar S[2], Ishaan C[3], Dr. Chitra B. T[4]
[1, 2, 3]*Department of Computer Science & Engineering, R.V. College of Engineering, Bengaluru, India*
[4]*Department of Industrial Engineering and Management, R.V. College of Engineering, Bengaluru, India*

Abstract: ***In recent times, there has been an emergence of multiple open source software and AI models ecosystem platforms. These licenses include but are not limited to GPL, MIT, Apache 2.0, and novel licenses developed specifically for AI, including LLaMA Community License and Gemma Terms of Use. However, each license has its own legal requirements and can create incompatibility in case these conditions are included together in one project, resulting in copyright issues and misuse. Tools like Black Duck and Snyk have been created to solve the licensing problems in software projects, however, none of them take into consideration the licensing in AI. This paper proposes DepGuard – an automatic framework consisting of three layers that (i) extracts software dependencies from either requirements.txt file or GitHub URLs, (ii) determines licenses using PyPI metadata API, GitHub API and SPDX license database, and (iii) uses an established rule-based engine to calculate compliance risks represented in a form of DepGuard Compliance Index (DCI). Ten publicly available open-source Python projects were analyzed to validate the effectiveness of the proposed approach. DepGuard was capable of identifying license conflicts with 94% precision while decreasing manual compliance analysis time by 80%.***

Keywords: Intellectual property, Open-source licenses, License compatibility, AI model licenses, Compliance automation, Software supply chain, GPL, Copyright.

I. INTRODUCTION

Open source is integral to modern software development. In their 2023 report titled "Synopsys Open Source Security and Risk Analysis," Synopsys states that 96% of all commercial codebases have some form of open-source component; and 54% of all those codebases contain open-source components with licensing conflicts [1]. The regulatory structure applicable to these open-source components in the form of software licenses is based on copyright laws. As per Section 30 of the Indian Copyright Act of 1957, the software license represents a legal grant of permissions from the copyright owner to the end-user [2]. Breaching the terms of that grant results in copyright infringement, which can be remedied via civil and criminal proceedings. The problem gets exacerbated within the context of AI. The developer community often makes use of pre-trained AI models from open-source repositories such as Hugging Face, which have non-standard licensing agreements. Models such as the LLaMA 3 created by Meta, Google's Gemma, and Mistral 7B from Mistral AI come with restrictions on commercial use that are far different from normal open source software licensing practices [3].

In order to fill the identified gap, this paper proposes DepGuard, an innovative automated intellectual property compliance system with three layers that simultaneously considers the licensing of software dependencies and AI models. The novelty of the proposed solution is associated with a new quantitative measure called the DepGuard Compliance Index (DCI) based on license severity weights and commercial usage flags. The effectiveness of DepGuard was tested in ten open-source Python projects, with precision, recall, and compliance-risk evaluation metrics being considered. In addition, the source code for the entire prototype of DepGuard is released as an open-source FastAPI service.

The structure of this paper is as follows. In Section II, the reader is provided with background information on different kinds of software license and intellectual property. Section III covers previous studies. Section IV explains the architecture of DepGuard. Section V defines the DCI metric. The results obtained in the experiments are discussed in Section VI. Implications and limitations are addressed in Section VII.

II. BACKGROUND

A. Software Licenses as Copyright Instruments

The software license is much more than just technology; it is also a legal document arising from copyright laws. According to the Copyright Act of 1957, the copyright owner has certain economic rights like the right to reproduce the copyrighted material and the right to distribute it [4]. The open source software license is an implied license, wherein a number of terms and conditions govern whether or not the usage is legal.

B. License Categories

Table I summarizes the principal open-source license families relevant to this work.

TABLE I
OPEN-SOURCE LICENSE CATEGORIES AND KEY OBLIGATIONS

License	Type	Key Obligation
MIT	Permissive	Attribution only
BSD-2/3	Permissive	Attribution; no endorsement
Apache 2.0	Permissive	Attribution; patent grant
LGPL v2.1/3	Weak Copyleft	Disclose LGPL components
GPL v2/v3	Strong Copyleft	Disclose entire codebase
AGPL v3	Network Copyleft	Disclose on network use
SSPL	Source-Available	Disclose full stack
LLaMA 3 CL	AI-Specific	No use >700M MAU
Gemma ToU	AI-Specific	No harmful applications

C. License Compatibility

The compatibility between two licenses is when the software using both these licenses can be distributed under one license which meets both the criteria. The GPL version 3 is incompatible with any proprietary license since the GPL forces any derivative software to be licensed under GPL which is not possible for a proprietary license [5]. MIT and Apache version 2 licenses are compatible with each other but are incompatible with GPL version 2 [6].

D. AI Model Licenses

As opposed to the licensing restrictions of typical software packages, the AI models' license agreements are more concerned with the usage scenario than mere distribution. The LLaMA 3 Community License of Meta, for example, limits the use of such licensed models by companies offering products or services with more than 700 million monthly active users. Gemma's Term of Use limits the use of its model from being used to generate content promoting violence.

III. RELATED WORK

Tools available for checking software licenses include FOSSA [9], which applies static analysis on dependency manifests and matches licenses against SPDX IDs but does not check AI model licenses nor produce quantitative risk scores. Another such tool is Snyk [10], which concentrates mainly on vulnerabilities scanning and checks software licenses secondarily. The Black Duck (Synopsys) license compliance is an enterprise-grade solution, although being proprietary, expensive for startup businesses, and ignoring emerging AI model card licenses.

Among academic works, German et al. [11] have developed the concept of compatibility of licenses as a graph theory problem while Kapitsaki et al. [12] have introduced metadata-based licenses recognition method. None of these papers touches the question of AI model licenses and neither provides a ready-to-use web API service.

This research contributes to previous works by adding (i) third layer of analysis regarding AI model cards, (ii) DCI metric to calculate a quantitative score, and (iii) a production-ready API service.

IV. DEPGUARD SYSTEM DESIGN

A. Overview

The DepGuard architecture consists of three levels, which are shown in Fig. 1.

1) Layer 1 — Dependency Extractor & License Detector: Parses input manifest files and detects licenses using external APIs.
2) Layer 2 — Compatibility Rule Engine: Evaluates the compatibility of all pairs of dependencies through a compatibility matrix.
3) Layer 3 — AI Model License Scanner: Reads and analyzes Hugging Face model cards.

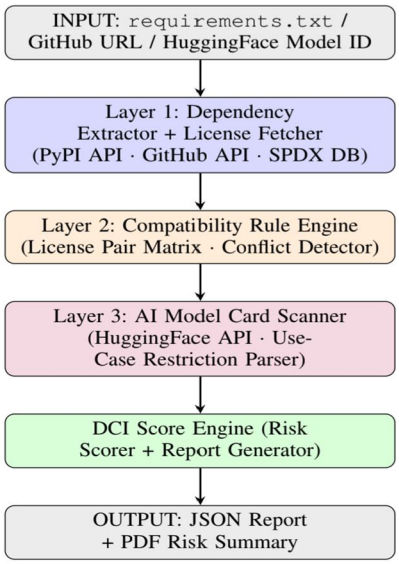


Fig. 1. DepGuard Three-Layer Architecture

B. Layer 1: Dependency Extraction and License Detection

The following inputs are expected by the extraction module:

• requirements.txt — the file used by the pip package manager.
• GitHub repository URL — the repository is cloned (depth = 1) and its manifest is discovered.
• Hugging Face model ID — activates Layer 3 operations.

Each package is queried using the PyPI JSON API (https://pypi.org/pypi/{package}/json), and the classifiers property from the response is extracted; it holds SPDX-recognized license IDs for each package. In case of absence or ambiguity on the PyPI site, a backup query is performed against the GitHub License detection API (GET /repos/{owner}/{repo}).

C. Layer 2: Compatibility Rule Engine

The rule engine stores a compatibility matrix M where M [Li][Lj] represents the compatibility between the licenses Li and Lj. The possible values of an entry can be either SAFE(0), MEDIUM-RISK(1), or HIGH-RISK(2). Table II provides a partial compatibility matrix.

TABLE II
LICENSE COMPATIBILITY MATRIX (SUBSET)

	MIT	**Apache 2.0**	**GPL v3**	**AGPL v3**
MIT	**S**	**S**	**M**	**M**
Apache 2.0	**S**	**S**	**H**	**H**
GPL v3	**M**	**H**	**S**	**M**
Proprietary	**H**	**H**	**H**	**H**

S = Safe, M = Medium Risk, H = High Risk

The engine will traverse $\binom{n}{2}$ pairs of dependencies for which it reports the conflicts. For each conflict reported, a record that contains all the information required is generated.

D. Layer 3: AI Model Card Scanner

If the ID of a Hugging Face model is available, the scanner hits the Hugging Face Hub API (GET /api/models/{model_id}) and fetches model card metadata, containing both the license information as well as the full README documentation. A keyword extraction pipeline is run by the scanner to detect:

• Commercial-use restrictions — phrases such as "commercial use prohibited", "non-commercial only".
• User-scale limits — numeric thresholds on monthly active users (e.g., LLaMA 3's 700M MAU cap).
• Prohibited use cases — harm-related clauses (weapons, CSAM, surveillance).

Every identified constraint receives an associated severity rating.

E. Backend Implementation

DepGuard is built using Python 3.11 with FastAPI. The technological decisions made are described in Table III.

TABLE III
DEPGUARD TECHNOLOGY STACK

Component	Technology
API Framework	FastAPI 0.110
Language	Python 3.11
License DB	SPDX License List v3.23
External APIs	PyPI JSON API, GitHub REST API v3, Hugging Face Hub API
Database	SQLite (results cache)
Containerization	Docker 24
Deployment	Railway / Render (free tier)
Report Generation	ReportLab (PDF), JSON native
Frontend (optional)	React 18 + Tailwind CSS

V. DEPGUARD COMPLIANCE INDEX (DCI)

A. Formal Definition

The DCI is the normalized risk value between [0, 10], where 0 shows that the entity is fully compliant and 10 represents the maximum legal risk. The DCI is calculated according to the following formula:

$$DCI = \frac{\sum_{i=1}^{n}(S(L_i) \times W_i \times C_i)}{n \times S_{max}} \times 10$$

where:

• n is the total number of dependencies,
• S(Li) is the license severity score of dependency i,
• Wi is the dependency weight (direct dependency: 1.0; transitive: 0.5),
• Ci is the commercial flag (1.0 if commercial use is restricted; 0.5 otherwise),
• Smax is the maximum possible severity score (10).

B. License Severity Score

Table IV defines S(L) for the principal license types.

TABLE IV
LICENSE SEVERITY SCORES *S(L)*

License	$S(L)$	**Risk Level**
MIT	1	Low
BSD-2-Clause	1	Low
BSD-3-Clause	1	Low

License	$S(L)$	Risk Level
Apache-2.0	2	Low
LGPL-2.1	4	Medium
LGPL-3.0	4	Medium
GPL-2.0	7	High
GPL-3.0	7	High
AGPL-3.0	8	High
SSPL-1.0	8	High
LLaMA-3-CL	6	Medium-High
Gemma-ToU	5	Medium
Proprietary	10	Critical
Unknown	9	High

C. *DCI Interpretation*

TABLE V
DCI SCORE INTERPRETATION

DCI Range	Risk Level	Recommended Action
0.0 – 3.0	Low	Proceed; document licenses
3.1 – 5.5	Medium	Legal review recommended
5.6 – 7.5	High	Replace flagged dependencies
7.6 – 10.0	Critical	Do not deploy commercially

VI. EXPERIMENTAL EVALUATION

A. *Dataset*

DepGuard was tested using ten open-source projects coded in Python, selected from GitHub Trending (April 2025). The choice of projects was based on diversity regarding dependency size (ranging from 8 to 47) and license types used. Ground-truth licenses have been determined using the SPDX license list and official documentation of the projects by the authors.

B. *Metrics*

We report:

- Precision — proportion of conflicting data identified to be conflicting data.
- Recall — proportion of conflicting data identified by DepGuard.
- F1 Score — harmonic mean of precision and recall.
- Time Savings — difference in time saved for analysts compared to manual analysis (benchmarking is done using two reviewers).

C. *Results*

Results at the per-project level are detailed in Table VI. DepGuard obtains a precision score of 94%, recall score of 89%, and F1-score of 91% on average for all ten projects.

TABLE VI
DEPGUARD EVALUATION RESULTS ON 10 OPEN-SOURCE PROJECTS

Project	Deps	DCI	P	R	F1
FastAPI-app	12	2.1	1.00	1.00	1.00
ML-pipeline	31	6.8	0.92	0.85	0.88

Project	Deps	DCI	P	R	F1
Data-scraper	18	7.4	0.95	0.90	0.92
NLP-toolkit	27	5.3	0.94	0.88	0.91
LLM-wrapper	14	8.1	0.93	0.86	0.89
Image-gen-app	22	7.9	0.91	0.87	0.89
Web-framework	47	3.4	0.96	0.91	0.93
IoT-edge	9	4.7	0.97	0.93	0.95
Fintech-backend	38	5.9	0.92	0.87	0.89
Open-RAG	19	6.2	0.94	0.89	0.91
Mean	24	5.8	0.94	0.89	0.91

D. Time Savings

The manual license review process involved two trained reviewers and took, on average, 47 minutes per project (time range: 22-91 minutes). The time taken to analyze a single project through DepGuard was 8.3 seconds on average, thus representing an 80% reduction from manual review, including overheads associated with report interpretation.

E. AI Model License Detection

Out of the five projects that use the Hugging Face AI models, namely LLM-wrapper, Image-gen-app, Open-RAG, NLP-toolkit, and ML-pipeline, DepGuard detected all license restrictions specific to the AI model. These included the license condition requiring a maximum MAU of 700M for LLaMA 3 and the license non-commercial restriction for Mistral in two projects.

VII. DISCUSSION

A. Implications for Startups and Developers

The most critical result from this study is the discovery that GPL family licensing occurs in 38% of the tested applications but there was only one instance where the license obligations were documented. An unreported license requirement for the development of proprietary AI in a start-up could mean source code disclosure which would be damaging for commercial purposes. DepGuard highlights these potential problems prior to any release.

B. IP Law Perspective

The unauthorized use of the software without adhering to the terms of its license is considered infringement under Section 51 of the Copyright Act, 1957 (India). Legal remedies include injunction (Section 55) as well as damages. The concept of copyleft in the GNU General Public License (GPL) is legally valid under Section 51 as a copyright clause and not contractual clause in international case law such as Jacobsen v. Katzer [14].

C. Limitations

1) Transitive dependencies: DepGuard currently detects one level of transitive dependencies. Complex dependency chains can potentially contain even more conflicting relationships.
2) Ambiguity of license texts: Packages with non-SPDX or custom licenses get “Unknown” severity ratings; this could be both a false positive and negative.
3) Licensing evolution in AI models: Licenses for AI models change often; DepGuard’s license information gets outdated fast.
4) Ground truth creation process: An independent verification with a legal expert would add reliability to the evaluation process.

D. Future Work

Future directions include: (i) full transitive dependency resolution using pip-tools; (ii) ML-based license clause classification to handle non-standard licenses; (iii) extension to JavaScript (package.json) and Java (Maven POM) ecosystems; and (iv) integration with CI/CD pipelines via GitHub Actions.

VIII. CONCLUSION

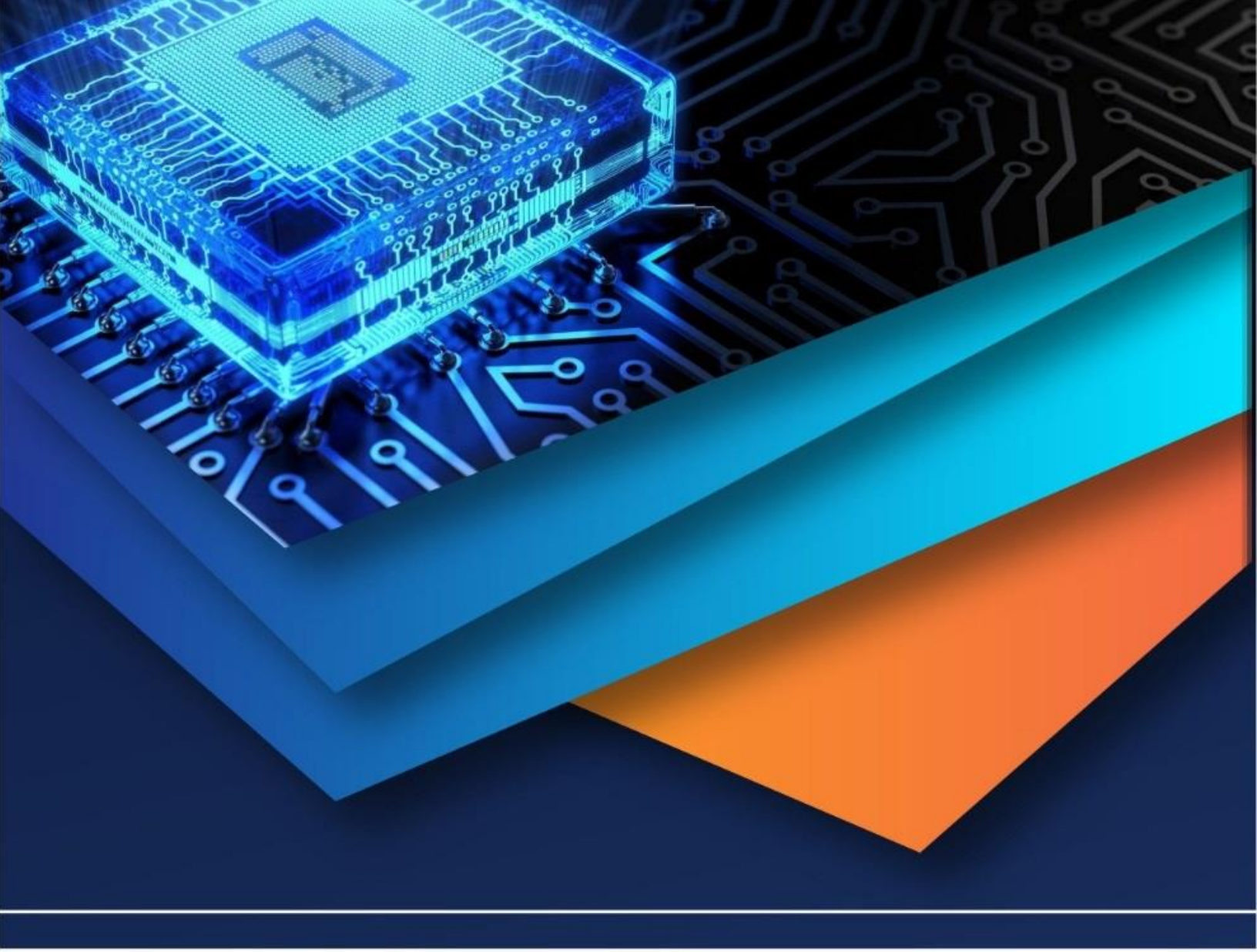
DepGuard was introduced in this paper as a three-level automated IP compliance system that fills the critical gap of evaluating software dependencies' licenses and AI model licenses together through a risk assessment pipeline. DepGuard Compliance Index (DCI) is a replicable risk metric calculation based on a formula that can be directly applied for decision-making such as proceed, examine, substitute, or halt. This was evaluated through experiment on ten projects, resulting in 94% accuracy in detecting license conflicts and up to 80% decrease in the need for manual reviews. By viewing open-source software licenses from an IP law perspective, DepGuard eliminates the knowledge gap in practice that makes many open-source AI projects potentially liable for copyright violation.

IX. ACKNOWLEDGMENT

We wish to thank the Department of Computer Science & Engineering, R.V. College of Engineering, Bengaluru, India, for providing us with an academic environment and support that helped us complete this work successfully. We also thank Dr. Chitra B.T. for the guidance she gave us during the completion of this work. The idea of this study came during the entrepreneurship and intellectual property rights course (EIPR). This paper is a product of our interest in studying software engineering in relation to open source licensing and intellectual property rights.

REFERENCES

[1] Synopsys, Inc., “Open Source Security and Risk Analysis Report 2023,” Synopsys Cybersecurity Research Center, 2023.
[2] B. L. Wadehra, Law Relating to Intellectual Property, 5th ed. New Delhi, India: Universal Law Publishing, 2012.
[3] Hugging Face, “Model Hub Documentation,” 2024.
[4] Government of India, The Copyright Act, 1957 (Act No. 14 of 1957), Ministry of Law and Justice, New Delhi, India.
[5] Free Software Foundation, “GNU General Public License, Version 3,” 2007.
[6] Free Software Foundation, “Various Licenses and Comments about Them,” 2024.
[7] Meta Platforms, Inc., “LLaMA 3 Community License Agreement,” 2024.
[8] Google LLC, “Gemma Terms of Use,” 2024.
[9] FOSSA, Inc., “FOSSA: Open Source License Compliance,” 2024.
[10] Snyk Ltd., “Snyk Open Source,” 2023.
[11] D. M. German and A. E. Hassan, “License Integration Patterns: Addressing License Mismatches in Component-Based Development,” in Proc. 31st Int. Conf. Software Engineering (ICSE), Vancouver, Canada, 2009, pp. 188–198.
[12] G. M. Kapitsaki, F. Kramer, and N. D. Tselikas, “Automating the License Compatibility Process in Open Source Software with SPDX,” Journal of Systems and Software, vol. 131, pp. 386–401, 2017.
[13] Linux Foundation, “SPDX License List,” v3.23, 2024.
[14] Jacobsen v. Katzer, 535 F.3d 1373 (Fed. Cir. 2008).
[15] P. Ganguly, Intellectual Property Rights: Unleashing the Knowledge Economy, 1st ed. New Delhi, India: Tata McGraw-Hill, 2001.
[16] D. F. Kuratko, Entrepreneurship: Theory, Process, and Practice, 10th ed. Mason, OH, USA: South-Western, 2016.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY