# ALOHA: A(IBoM) tooL generatOr for Hugging fAce

Riccardo D'Avino*, Sabato Nocera*, Daniele Bifolco†, Federica Pepe†, Massimiliano Di Penta†, Giuseppe Scanniello*
*University of Salerno, Fisciano, Italy
†University of Sannio, Benevento, Italy
r.davino11@studenti.unisa.it, snocera@unisa.it, d.bifolco@studenti.unisannio.it,
f.pepe8@studenti.unisannio.it, dipenta@unisannio.it, gscanniello@unisa.it

***Abstract*—The increasing adoption of Artificial Intelligence (AI) in any kind of software has highlighted the need for greater transparency, security, and traceability within the AI supply chain. The AI Bill of Materials (AIBoM) extends the Software Bill of Materials (SBoM) concept by incorporating AI-specific components such as models, datasets, dependencies, and metadata. In this paper, we introduce ALOHA, a novel tool that automatically generates AIBoM from AI models hosted on Hugging Face (HF), leveraging the CycloneDX standard for software transparency and security. ALOHA extracts relevant metadata from model cards and maps them to a structured AIBoM format, ensuring compliance with existing SBoM frameworks. We conducted a preliminary empirical evaluation on a statistically significant sample of 312 AI models to assess ALOHA. Our initial findings indicate that while ALOHA successfully retrieves and structures essential AIBoM fields, challenges remain regarding metadata completeness and standardization of model cards. This work represents a step towards enhancing AI supply chain security and governance, providing a foundation for future advancements in AIBoM generation.**
**Tool link: https://doi.org/10.5281/zenodo.15052346**

***Index Terms*—Artificial Intelligence Bills of Material, Software Bills of Materials, Software Supply Chain**

## I. Introduction

Software Composition Analysis (SCA) aims at identifying the artifacts that compose a software system. These may include third-party libraries for which the system depends, directly or indirectly, but also source code elements (*e.g.,* code snippets) or other artifacts being included in the software system. The set of elements (libraries, tools, or other artifacts) required to build a software product is referred to as the "software supply chain."

SCA and software supply chain analysis have multiple purposes, including, for example, identifying defect-prone or insecure components, as well as performing a license compatibility compliance, *i.e.,* determining whether the license under which one is redistributing a software project is compatible with those of included components.

To properly document the inventory of components of a software project, owners can release Software Bills of Materials (SBoMs). An SBoM is an inventory of artifacts a software system depends upon, directly or indirectly. It can be considered something similar to the list of ingredients released with a food product (which one can use to assess the food against allergies), or the inventory of a mechanical system (*e.g.,* a car manufacturer can know exactly all components for a given car model so that if one of these components turns out to be defective, all models equipped with it are recalled for maintenance).

SBoMs are becoming relevant and crucial also because of governmental regulations, such as the US Executive Order [1] which requires SBoMs for any public administration software, or the EU Cyber Security Act [2].

While there exist standards for SBoMs, in particular, the SPDX [3] and CycloneDX [4] ones as well as tools for SBoM generation (*e.g.,* those provided by CycloneDX [5] and the one integrated into GitHub [6]), previous research has pointed out several challenges in SBoM generation and consumption [7], [8], [9]. Some of such challenges [8], [9] also emerge given the high pervasiveness, in many domains, of Artificial Intelligence (AI) enabled software systems. As the behavior of such systems is highly determined by AI models, including Machine Learning (ML) models, ensuring transparency for them requires SBoMs to also ensure transparency with respect to ML models (or AI components in general) on which the software depends.

On the one hand, an extension to SBoMs to provide AI transparency, the Artificial Intelligence Bills of Material (AIBoM) [7], has been conceived. AIBoM specifies various kinds of transparency pieces of information for AI models. These include, for example, the model version, dependencies, training information, attribution/licensing requirements, bias/fairness issues, and known vulnerabilities. On the other hand, and differently from SBoMs, to the best of our knowledge, AIBoM generation is not supported by suitable, automated tools. The only available support for "quasi-AIBoM" documents comes from MLOps tools such as DVC [10] or ML-Flow [11].

To fill this gap, we propose in this paper a tool, named ALOHA (AIBoM tooL generatOr for Hugging fAce). As its name suggests, ALOHA supports the automated generation of AIBoMs for ML-enabled projects that depend on models hosted on Hugging Face (HF) [12], a well-known hosting platform (and supporting framework) for different types of pre-trained models. Given the identifier (ID) of an AI model hosted on HF, ALOHA downloads and analyzes their model cards, extracting pieces of information including (i) the license,

(ii) the model's dependencies, (iii) the availability of a training dataset, and (iv) documentation about bias/fairness. After such information has been extracted and summarized, ALOHA generates an AIBoM file, that can be linked to the projects' SBoM.

We have conducted a preliminary empirical validation of ALOHA in generating AIBoMs on a sample of 312 AI models from HF. Our initial findings indicate that while ALOHA successfully retrieves and structures essential AIBoM fields, challenges remain regarding metadata completeness and standardization of model cards in HF.

**Paper Structure.** In Section II, we outline background and related work. In Section III, we present ALOHA, while our preliminary empirical evaluation of ALOHA is shown in Section IV. In this section, we also discuss the results and threats to the validity of these results. We conclude the paper with final remarks in Section V.

## II. Background and Related Work

In this section, we first present the role of AI in the software supply chain. Then, we describe HF and model cards used to share and document AI models. After providing the rationale for using BoMs to support the AI supply chain, we examine the state of the art in AIBoM generation. Finally, we discuss the SBoM standard chosen for AIBoM generation, specifically focusing on CycloneDX.

### A. AI in the Software Supply Chain

Integrating AI components into the software supply chain has improved security, efficiency, and reliability by identifying and fixing vulnerabilities, automating security tasks, and managing software development and deployment.

In this scenario, Xia *et al.* [7] proposed using blockchain technology to securely share the SBoMs to ensure greater transparency and security in the handling of software and AI components. They further discussed the concept of AIBoM, which extends this approach to encompass the origins, dependencies, and all AI components within the supply chain.

Concerning the SBoM, Stalnaker *et al.* [8] explored the challenges in creating and using SBoMs, which are essential for managing software dependencies, vulnerabilities, and licenses. The authors surveyed 138 professionals from five stakeholder groups (SBoM-experienced developers, critical open-source project contributors, AI/ML experts, cyber-physical systems professionals, and legal experts). In addition, they interviewed eight participants to gain deeper insight into their experiences. The study showed 12 key challenges, including issues with SBoM content, tooling deficiencies, maintenance difficulties, and domain-specific challenges. The authors propose four actionable solutions to improve the adoption of SBoM and outline future research directions.

Beyond technical and security considerations, Widder and Nafus [9], through interviews with 27 AI experts, examined how these professionals struggle to address the ethical questions raised by AI guidelines. The feeling of "dislocated accountability" arises because the development process is split into smaller tasks. This breakdown can make it harder to see the big picture and can shift ethical duties to other members of the supply chain. The authors suggest that interventions such as ethical checklists and guidelines could be improved by adopting an approach of "situated accountability," recognizing the relationships and obligations that extend within and beyond the AI supply chain.

Charles *et al.* [13], through a comprehensive literature analysis, highlighted how the combined use of blockchain and AI could enhance the resilience of information and processes, enable faster and more cost-effective product delivery, and improve traceability. Additionally, the potential benefits of this integration are discussed in business environments characterized by volatility, uncertainty, complexity, and ambiguity.

In summary, these studies emphasize how AI and blockchain can improve security and traceability in the software supply chain while also raising new ethical concerns that require a rethinking of accountability among stakeholders.

### B. Studies About the Hugging Face Model Forge

For our AIBoM generation tool, we leverage model descriptions (known as model cards) from HF [12]. HF is a well-known hosting platform for pre-trained AI models as well as datasets, including those used to train models. To date, it features over 1.5M models and 300k datasets.

The main goal of an HF model card is to describe a reusable, pre-trained AI model, or a dataset. Model cards provide a structured approach to consistently detail the model, encompassing aspects such as the training dataset, evaluation metrics, and possible biases. Likewise, the AIBoM delineates the elements involved in constructing an AI model, which includes data, frameworks, and libraries. While several elements included in the AIBoM are already present in the model card, the AIBoM also encompasses software dependencies, versioning, and potential security risks. The goal of our research is to integrate the model card within the AIBoM framework, promoting model reproducibility more transparently and standardizing the process of AI model development, ultimately facilitating the work of developers.

Several studies investigated the content of HF model cards. Pepe *et al.* [14] conducted an empirical study to evaluate how HF models declare datasets, possible bias, and license incompatibilities. On the same line, Castaño *et al.* [15] examined the evolution and maintenance of ML models available on HF, analyzing 380,000 models based on their application domains, utilized frameworks, tag evolution, and associated datasets. Jiang *et al.* [16], instead, conducted interviews with 12 experts from HF to go deeper into the practices and challenges in reusing pre-trained models, emphasizing their provenance, reproducibility, and portability. The lack of transparency of ML models is also unveiled by Ait *et al.* [17], who focused on metadata availability, the quality of the documentation, and the datasets to assess if the HF platform is suitable for conducting empirical studies. Castaño *et al.* [18] analyzed the carbon dioxide emissions during the training phase to promote more sustainable ML-based development, analyzing

1,417 models presented on HF. Regarding the structured documentation of AI models, the concept of a model card was introduced by Mitchell *et al.* [19] to improve transparency in the documentation of the models. Bhat *et al.* [20] defined an ideal structure of a model card and developed a *DocML* tool to support developers in creating a model card. Crisan *et al.* [21] conducted a design study involving experts in ML and natural language processing to evaluate the effectiveness of interactive model cards. Their findings suggest that interactivity enhances model understandability, interpretability, and trust. In this direction, Yang *et al.* [22] examined the documentation of 7,433 datasets on HF to evaluate their compliance with standards of completeness and quality. They showed that the most widely used datasets tended to have more comprehensive documentation. The study explored various themes covered in these datasets, including technical aspects, social considerations, and limitations.

### C. Reseach and Tool Support About AIBoM

The research community has recently started investigating the need for complementing SBoMs with AIBoMs [23], [7], [8], [24]. In their qualitative study, Xia *et al.* [23] pointed out that the generation of AIBoMs is different from that of traditional SBoMs as AIBoMs include AI/ML-specific data. Xia *et al.* [7] acknowledged that a key difference between SBoMs and AIBoMs is the dynamic nature that the latter should have. While traditional software systems remain static in their behavior once deployed, the behavior of AI-enabled software systems could evolve as they process new data. This difference would require ad-hoc tools for the generation of AIBoMs. Stalnaker *et al.* [8] also conducted a qualitative study in which almost all respondents were unaware of any tool support for the generation of AIBoMs. Although no AIBoM generator was reported, the authors identified as "quasi-AIBoM" generators two tools: DVC [10] and ML-Flow [11]. In their work, Radanliev *et al.* [24] proposed an AIBoM schema and a tool to visually represent its instances. However, they did not propose any specific tool to generate AIBoMs. To the best of our knowledge, no AIBoM tool generators are publicly accessible online at the time of the submission of our paper. Despite extensive searches across platforms such as GitHub and Google, we were unable to identify any available solutions. This strongly suggests that ALOHA, the tool presented in this paper, is the first publicly available AIBoM generator, marking a significant advancement in the field.

### D. CycloneDX capabilities for AIBoM

Given the growth of AI systems usage and their complexity, traditional standards for SBoM generation need to be adapted to ensure transparency, traceability, and security. SPDX and CycloneDX have been widely adopted as standards for software supply chain management. Among the available standards, CycloneDX provides an optimal foundation for AIBoM generation due to its extensibility, AI-specific enhancements, and widespread tool support. The CycloneDX standard was released by OWASP. As highlighted by Xia *et al.* [23], it was initially designed to address issues of software license compliance. However, in its version 1.5, released in 2023, CycloneDX introduced AI and ML fields [25], facilitating the tracking and management of AI/ML models and their associated metadata. This extension, often referred to as ML-BOM [26], allows representing AI-specific components such as:

- **Machine Learning Models** documenting model architectures, weights, hyper-parameters, and configurations;
- **Datasets** capturing information about datasets used for training, validation, and testing, ensuring data provenance and integrity;
- **Algorithms and Frameworks** describing the underlying AI frameworks, libraries, and optimization techniques applied;
- **Dependencies and External Components** tracking third-party models, pre-trained embeddings, and associated dependencies, which are critical for vulnerability management and reproducibility.

The use of AIBoMs can help, for example, mitigate risks such as adversarial attacks, model poisoning, and data leakage.

For several reasons, we chose CycloneDX over other SBoM standards, such as SPDX. First, when we started our study, SPDX's AI-related functionalities were still in their early stages, whereas CycloneDX had already introduced AI-specific extensions, making it more suitable for our needs. Furthermore, according to Nocera *et al.* [27], CycloneDX is more widely adopted in open-source projects than SPDX, further confirming its stronger integration with security and compliance tool ecosystems.

Finally, CycloneDX's flexible and extensible JSON and XML schemas enable seamless integration with AI supply chain tools. The growing ecosystem of tools capable of generating, analyzing, and integrating SBoMs reinforces the choice of CycloneDX for AIBoM generation. By leveraging CycloneDX, we align with established best practices in software supply chain security while addressing the unique challenges of AI system transparency and governance.

## III. ALOHA: A Tool for AIBoM Generation

In this section, we first introduce the logical architecture of ALOHA, outlining its key components and design principles. Next, we describe the mapping process used to align the information provided by HF model cards with the corresponding data fields in CycloneDX, ensuring a structured and standardized representation as detailed in Section II-D. Finally, we present the ALOHA tool, showing its functionality and explaining and demonstrating its usage with an example of a generated AIBoM.

### A. ALOHA Underlying Workflow

ALOHA, developed in Python (it requires at least Python 3.10.2), takes as input a string containing the model ID from HF in the format *author/model*, producing an AIBoM in JSON format compliant with the CycloneDX standard.

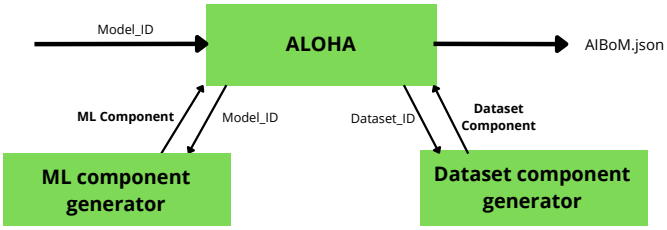

Fig. 1: Overview of the ALOHA Tool Workflow

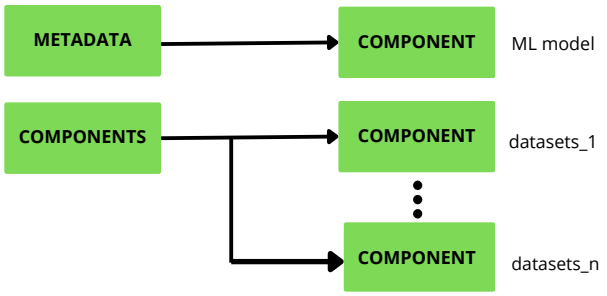

Fig. 2: AIBoM Structure

The overall process/workflow of ALOHA is illustrated in Figure 1. ALOHA first invokes the **ML Component Generator**, which retrieves model-related information and generates the ML component following the CycloneDX standard. Specifically, ALOHA performs the following steps to generate AIBoMs from HF:

- It performs a GET request to the HF API to obtain the model metadata and the README.md file, which contains the *Model description* and *Uses* sections from the text description of the model card.
- The extracted information is then mapped to the corresponding CycloneDX fields, forming the ML component.

Next, the ALOHA checks the model metadata to determine whether any datasets used for training or testing are specified. If datasets are present, the **Dataset Component Generator** is invoked for each dataset. This component:

- Performs a GET request to the HF API to obtain dataset metadata.
- Extracts relevant information and maps it to the CycloneDX fields, generating the dataset component.

Finally, the ML component and all dataset components are added to the base BoM structure, producing the AIBoM in JSON format, fully compliant with the CycloneDX standard (the final structure of the generated AIBoM is shown in Figure 2).

### *B. Mapping Hugging Face Model Card Sections to CycloneDX*

A key challenge in creating AIBoMs for models hosted on HF is the mapping of model card fields to their respective CycloneDX fields. The model cards are saved in the *README.md* file of each model repository and have two key parts, with overlapping information: **metadata** and **text description**. The metadata section is represented by a YAML block located at the top of the model card. It contains structured metadata about the model, such as tags, tasks, and licenses. The text description section is written in Markdown and provides a more detailed textual description of the model, including its purpose, performance, and limitations. HF provides templates for both metadata and textual descriptions. These templates allow straightforward mapping of model card fields to CycloneDX data fields. However, few models adhere to the text description template, making data extraction challenging. Some models introduce minor modifications to section titles, while others adopt a completely different structure. To address this issue, we analyzed the adherence of models to the templates recommended by HF. We implemented a Python script to automatically determine how many models adhere to the HF template and identify the most frequently used section titles. To ensure that the analysis focuses on models that are actually used, the 10,000 most downloaded model cards were selected. By prioritizing the most downloaded models, the study provides more meaningful and relevant insights into documentation practices within the HF ecosystem. In Table I, the occurrences of the template titles in the 10,000 most downloaded models are observed.

TABLE I: Occurrences of the template titles in the 10,000 most downloaded HF models

| Title | Count |
|---|---|
| Model Details | 1,808 |
| Model Description | 1,789 |
| Model Sources | 141 |
| Uses | 699 |
| Direct Use | 581 |
| Downstream Use | 77 |
| Out-of-Scope Use | 611 |
| Bias, Risks, and Limitations | 572 |
| Recommendations | 488 |
| How to Get Started with the Model | 600 |
| Training Details | 727 |
| Training Data | 1,118 |
| Training Procedure | 1,129 |
| Preprocessing | 187 |
| Training Hyperparameters | 811 |
| Speeds, Sizes, Times | 36 |
| Evaluation | 1,045 |
| Testing Data, Factors & Metrics | 452 |
| Testing Data | 440 |
| Factors | 421 |
| Metrics | 482 |
| Results | 606 |
| Summary | 422 |
| Model Examination | 18 |
| Environmental Impact | 506 |
| Technical Specifications | 73 |
| Model Architecture and Objective | 440 |
| Compute Infrastructure | 449 |
| Hardware | 506 |
| Software | 511 |
| Citation | 1,657 |
| Glossary | 17 |
| More Information | 59 |
| Model Card Authors | 83 |
| Model Card Contact | 474 |

Table II presents some of the most frequently used section titles that deviate from the official template. Among the

TABLE II: Most frequently used section titles in HF model cards that do not align with the official template.

| Title | Count |
|---|---|
| usage | 1,189 |
| how to use | 870 |
| uses | 699 |
| intended uses & limitations | 587 |
| license | 560 |
| training | 491 |
| framework versions | 463 |
| bibtex entry and citation info | 391 |
| intended use | 343 |
| description | 314 |
| limitations | 266 |
| quickstart | 177 |
| overview | 137 |

TABLE III: Mapping between CycloneDX and HF for the component data field in metadata.

| CycloneDX | Hugging Face |
|---|---|
| authors | author |
| tags | tags |
| description | #model description |
| name | modelId |
| id | cardData-> license |
| name | cardData-> license_name |
| task | pipeline_tag |
| architectureFamily | config-> model_type |
| modelArchitecture | config-> architectures |
| datasets | datasets |
| slice | dataset-> type, config, split |
| type | metric-> type |
| value | metric-> value |
| useCases | # Uses |
| environmentalConsiderations | emissions |
| environmentalConsiderations | source |
| environmentalConsiderations | training_type |
| environmentalConsiderations | geographical_location |
| environmentalConsiderations | hardware_used |
| properties | library_name |
| properties | base_model |
| properties | base_model_relation |

10,000 analyzed models, several alternative titles are used. For instance, 1,189 models use "Usage" instead of "Uses" to describe how the model is intended to be used, while 314 use "Description" instead of "Model Description."

To preserve the information from section titles that do not strictly follow the official template, we manually analyzed the section titles of the 10,000 most downloaded models. Titles with more than 20 occurrences were identified and mapped to their corresponding template sections. For example, "Description" and "Overview," which are frequently used by the community, are mapped to "Model Description," which is the template section for including information about the model's description. This analysis allowed us to extract the content of the model cards, identify the different sections using the template titles, retrieve the relevant information, and organize it within the AIBoM. Once the model information is retrieved, if datasets are specified, we extract their metadata

TABLE IV: Mapping between CycloneDX and HF for the component data field in components.

| CycloneDX | Hugging Face |
|---|---|
| contents-> URL | URL of the Model |
| contents-> properties | task_categories |
| contents-> properties | task_ids |
| contents-> properties | language |
| contents-> properties | language_details |
| contents-> properties | size_categories |
| contents-> properties | annotations_creators |
| contents-> properties | language_creators |
| contents-> properties | pretty_name |
| contents-> properties | source_datasets |
| contents-> properties | paperswithcode_id |
| contents-> properties | config_name |
| contents-> properties | data_files-> split |
| contents-> properties | data_files->path |
| contents-> properties | license |
| contents-> properties | license_name |
| contents-> properties | license_link |
| contents-> properties | license_details |
| description | description |
| owners-> organization-> name | author |
| owners-> organization-> url | URL to author page |

and map the relevant details into the AIBoM. We store the model's information within the component field of metadata and the dataset information within the component field of components. Figure 2 shows the final structure of the AIBoM. The final mapping is presented through Table III and Table IV, where one column lists the CycloneDX fields and the other shows the corresponding fields extracted from HF.

### C. ALOHA in Action

ALOHA can be executed via the command line using the following general syntax:

```
python ALOHA.py <model_ID> -o <output_-
dir_path>
```

- **¡model_ID¿**: It indicates the model ID from HF in the format *author/model*. This is a required parameter and must correspond to a valid model identifier.
- **¡output_dir_path¿**: It indicates the destination directory where the output files will be saved. This is an optional parameter; if not provided, the tool will use a default directory.

Figure 3 shows an example of an AIBoM for the HF model `bitnet_b1_58-3B` generated by ALOHA. The AIBoM follows the CycloneDX standard, starting with the fields **bomFormat** and **specVersion**, which indicate the format and the specific version of the CycloneDX. The Bill of Material (BoM) element has two key properties: **serialNumber** and **version**, which together define the unique identity of the AIBoM. While the serial number acts as a unique identifier, the version helps keep track of potential updates or changes to the BoM. Within the **metadata** field, we find the **timestamp** representing the ex-

act moment of the AIBoM's creation, along with a **component** object that provides essential information about the AI model described. Each component includes a field called **bom-ref**, a unique reference to connect the component to other elements in the BoM. The **external references** field provides additional relevant information not directly included in the BoM itself. In this case, it links to the official documentation of the model on HF, offering a direct source for further details. The **modelcard** section aims to describe specific aspects of the AI model, such as its **task**, **architecture**, and **datasets** used for training and testing. However, in this example, the model's author did not provide these details. Lastly, the **properties** section includes additional metadata that does not directly match the CycloneDX standard (*e.g.,* information about the used library).

## IV. Preliminary Empirical Evaluation

In this section, we present a preliminary empirical evaluation of ALOHA in the generation of AIBoMs. A replication package containing the artifacts used for this empirical evaluation (including the generated AIBoMs and the spreadsheet of the analysis) is available on the web [28].

### A. Study Design

The *goal* of this preliminary empirical evaluation is to study ALOHA with respect to the generated AIBoMs for the *purpose* of assessing their completeness in terms of data fields. The *perspective* is that of researchers interested in the software/AI supply chain. The *context* consists of AI models publicly hosted on HF.

Based on the aforementioned goal, we formulated and studied the following high-level Research Question (RQ):

**RQ.** *To what extent the AIBoMs generated by ALOHA report all the expected data fields?*

This RQ aims at assessing whether the AIBoMs generated by ALOHA report all the expected data fields and, if not, why. This is because AIBoMs generated through ALOHA may not report information for each data field in case such information is lacking on HF.

We leveraged the HF API Endpoints [29] and the associated Python wrapper, namely `huggingface_hub` [30], to search for the AI models for which to generate the AIBoMs. We queried HF to retrieve the list of all hosted AI models, along with information about their numbers of downloads and likes. To avoid the risk of selecting personal or inadequate AI models, we retrieved AI models with at least 100 downloads and at least 100 likes on HF. These inclusion criteria are commonly used to mitigate the perils of mining software repositories [31], [14]. Eventually, we retrieved a total of 1,643 AI models, which represent our study population. Due to the large size of this population, we built a statistically significant random sample with a confidence level equal to 95% and a margin of error equal to 5%. The size of this sample was equal to 312.


```
{
  "bomFormat": "CycloneDX",
  "specVersion": "1.6",
  "serialNumber": "urn:uuid:47a09005-e737-471b-8ab0-
      d65461fbd675",
  "version": 1,
  "metadata": {
    "timestamp": "2025-02-28T12:11:23.832568+00:00",
    "component": {
      "type": "machine-learning-model",
      "bom-ref": "1bitLLM/bitnet_b1_58-3B-9f518091
          -90ea-528e-9efe-15e2f9373a8",
      "name": "1bitLLM/bitnet_b1_58-3B",
      "licenses": [
        {
          "license": {
            "id": "MIT",
            "url": "https://spdx.org/licenses/MIT.
                html"
          }
        }
      ],
      "externalReferences": [
        {
          "url": "https://huggingface.co/1bitLLM/
              bitnet_b1_58-3B",
          "type": "documentation"
        }
      ],
      "modelCard": {
        "modelParameters": {
          "task": "text-generation",
          "architectureFamily": "llama",
          "modelArchitecture": "BitnetForCausalLM",
          "datasets": []
        }
      },
      "properties": [
        {
          "name": "library_name",
          "value": "transformers"
        }
      ]
    },
    "authors": [
      {
        "name": "1bitLLM"
      }
    ]
  }
}
```


Fig. 3: Example of an AIBoM for `bitnet_b1_58-3B` generated using ALOHA.

### B. Results

Table V shows the number and percentage of data fields that were reported (documented) and not reported (missing) in the AIBoMs generated from the AI models of our sample. The data fields `authors` and `tags` are always documented, while `license`, `task`, and `library_name` appear in over 90% of the AIBoMs. Following, `architectureFamily` and `modelArchitecture` appear almost in 70% of the AIBoMs, while `description` in about 40% of them. The data fields `datasets`, `performanceMetrics`, and `useCases` appear in between 10% and 30% of the AIBoMs. Lastly, `environmentalConsiderations` appears in less

TABLE V: Number and percentage of documented and missing data fields in the generated AIBoMs.

| Case | Documented Field | | Missing Field | |
|---|---|---|---|---|
| | **Sum** | **%** | **Sum** | **%** |
| authors | 312 | 100 | 0 | 0 |
| tags | 312 | 100 | 0 | 0 |
| description | 134 | 42.95 | 178 | 57.05 |
| license | 284 | 91.03 | 28 | 8.97 |
| task | 296 | 94.87 | 16 | 5.13 |
| architectureFamily | 217 | 69.55 | 95 | 30.45 |
| modelArchitecture | 213 | 68.27 | 99 | 31.73 |
| datasets | 88 | 28.21 | 224 | 71.79 |
| performanceMetrics | 41 | 13.14 | 271 | 86.86 |
| useCases | 77 | 24.68 | 235 | 75.32 |
| environmentalConsiderations | 2 | 0.64 | 310 | 99.36 |
| library_name | 298 | 95.51 | 14 | 4.49 |
| ***Total*** | *2,274* | *60.74* | *1,470* | *39.26* |

TABLE VI: Number and percentage of missing data fields in the generated AIBoMs grouped by codes.

| Case | Code 01 | | Code 02 | | Code 03 | |
|---|---|---|---|---|---|---|
| | **Sum** | **%** | **Sum** | **%** | **Sum** | **%** |
| description | 46 | 25.84 | 122 | 68.54 | 10 | 5.62 |
| license | 19 | 67.86 | 9 | 32.14 | 0 | 0.00 |
| task | 15 | 93.75 | 1 | 6.25 | 0 | 0.00 |
| architectureFamily | 89 | 93.68 | 6 | 6.32 | 0 | 0.00 |
| modelArchitecture | 89 | 89.90 | 10 | 10.10 | 0 | 0.00 |
| datasets | 194 | 86.61 | 30 | 13.39 | 0 | 0.00 |
| performanceMetrics | 170 | 62.73 | 101 | 37.27 | 0 | 0,00 |
| useCases | 200 | 85.11 | 33 | 14.04 | 2 | 0.85 |
| environmentalConsiderations | 305 | 98.39 | 5 | 1.61 | 0 | 0.00 |
| library_name | 14 | 100 | 0 | 0.00 | 0 | 0.00 |
| ***Total*** | *1,141* | *77.62* | *317* | *21.56* | *12* | *0.82* |

than 1% of AIBoMs.

These results indicate that the data fields that are mostly reported in the generated AIBoMs refer to the information of the model card that can be (i) reported by the author of the AI model or (ii) inferred directly by HF based on the information available in manifest/configuration files. The data fields that can be inferred by HF are `authors` [29], `tags` [32], `library_name` [33], `task` [34], `architectureFamily` [35], `modelArchitecture` [35], and `environmentalConsiderations` [36]. These data fields are among the most documented in the generated AIBoMs, except for `environmentalConsiderations`, which is the least documented data field. In fact, it might be inferred only for the AI models using the AutoTrain feature [37], but only one AI model in our sample used it (although we do not have `environmentalConsiderations` for that model).

We conducted a qualitative analysis of the missing data fields in the generated AIBoMs to better understand the reasons for which they were not reported. This analysis involved a manual inspection of the content of the model cards on HF and open coding to categorize the causes of missing data fields. That is, for each missing data field, we assigned a "code" explaining the reason why that data field was missing. We ended up classifying the causes of missing data fields with three codes:

- **Code 01 - Missing Information**. Information for the data field is not available anywhere in the model card.
- **Code 02 - Misplaced Information**. Information for the data field is available in a section of the model card that is not mapped with that data field.
- **Code 03 - Restricted Access to Information**. Direct access to the model card on HF, which is required to extract the information for generating the AIBoMs, is not allowed as there are conditions to be accepted.

Table VI shows the classification of missing data fields based on the aforementioned codes. The results indicate that the data field `library_name` is always missing due to Code 01, while `task`, `architectureFamily`, `environmentalConsiderations` are missing in over 90% of the AIBoMs due to the same code. Following, `modelArchitecture`, `datasets`, and `useCases` are missing between 80% and 90% of the AIBoMs due to Code 01. Due to the same code, the data fields `license` and `performanceMetrics` are missing between 60% and 70% of the AIBoMs, while `description` is missing in about 26% of them. As for Code 02, the data field `description` is missing in about 70% of the AIBoMs, while `license` and `performanceMetrics` are missing between 30% and 40% of them. Following, `modelArchitecture`, `datasets`, and `useCases` are missing between 10% and 20% of the AIBoMs due to Code 02. Due to the same code, all other data fields are missing in less than 10% of the generated AIBoMs. Only two data fields are missing due to Code 03, namely `description` and `useCases`, which happened in about 6% and 1% of the generated AIBoMs, respectively. Summing up, about 77.62% of the data fields are missing due to Code 01, 21.56% due to Code 02, and 0.82% due to Code 03. ALOHA was not able to report the expected data fields of all the generated AIBoMs because the needed information was not reported in the corresponding model cards.

The results of our investigation indicate that the missing data fields in the generated AIBoMs are mainly due to a lack of the associated information in the model cards. On the other hand, a portion of the data fields is missing because the associated information is documented in unexpected sections of the model cards; therefore, their model cards do not follow the template recommended by HF, and we were not capable of identifying a mapping with those sections. Lastly, some data fields in the generated AIBoMs were missing due to restrictions in accessing the associated information of the model cards; in these cases, the authors of the AI models limited the access upon acceptance of certain conditions (*e.g.,* sharing of contact information by those who want to access the information).

### C. Discussion and Limitations

The results of our preliminary investigation suggested that one of the most significant challenges in using ALOHA is that the "quality" of the generated AIBoMs is inherently dependent on the information provided by authors of AI models. Since ALOHA extracts data directly from model cards, any inconsistencies, omissions, or inaccuracies in the original documentation will be reflected in the resulting AIBoMs. For example, if authors do not provide detailed descriptions of datasets, architectures, or performance metrics, the AIBoM will lack critical information. This reliance on user-generated content introduces variability in the quality of the AIBoMs, as not all authors follow the same documentation standards or provide the same level of detail. Consequently, the tool's output is only as reliable as the input it receives, which may limit its effectiveness in generating comprehensive and accurate AIBoMs.

ALOHA was developed and tested specifically for models hosted on HF, which means its applicability to other platforms or ecosystems may be limited. Other platforms may have different standards or templates for model documentation, which could make it difficult to directly apply the tool without significant modifications.

### D. Threats to Validity

In this section, based on the classification provided by Wohlin *et al.* [38], we discuss the threats that might affect the validity of our preliminary empirical investigation.[1]

**Construct Validity**. Threats to construct validity concern the relation between theory and observation [38]. We did not take into account ALOHA in generating subfields of AIBoMs. That is, given a data field (*e.g.,* `environmentalConsiderations`) consisting of several subfields (*e.g.,* `emissions`, `source`, `training_type`, `geographical_location` and `hardware_used`), we considered ALOHA to have generated that data field whether it contained information for at least one of its subfields. This threat to validity is not a concern given the nature of the investigation, *i.e.,* a preliminary empirical investigation.

**Conclusion Validity**. Threats to conclusion validity concern statistical significance, including sample composition and size [38]. We built a statistically significant random sample of AI models with at least 100 downloads and at least 100 likes on HF, using a 95% confidence level and a 5% margin of error. The analysis of samples is common in empirical research to avoid analyzing the entire population of interest (*e.g.,* [39]). However, it might pose a threat to conclusion validity because the randomness of the sample selection process does not rule out selection bias, *i.e.,* the selected AI models may not fully represent the population under investigation. We also recognize that sampling popular AI models might not fully represent documentation practices across all of HF.

[1]Threats to internal validity are not discussed, as causality is not investigated.

**External Validity**. Threats to external validity concern the generalizability of results [38]. The results might not be generalizable to the generation of AIBoMs from AI models hosted on platforms other than HF. However, this is beyond the current features of ALOHA. Also, the results reflect the state of documentation of the model cards on HF at the time the investigation was conducted (*i.e.,* February 2025). Replications of this investigation at a later time might lead to observing different results, for example, due to changes to the model cards or the hosting of new AI models on HF.

## V. Conclusion and Future Work

The increasing complexity of AI-driven software necessitates robust mechanisms for ensuring transparency, traceability, and security. In this paper, we present ALOHA (AIBoM tooL generatOr for Hugging fAce). It represents a significant step forward by enabling the automated generation of AIBoM, leveraging the widely adopted CycloneDX standard. Our preliminary evaluation showed the ALOHA capability to extract and structure relevant data fields from the content of HF model cards (*i.e.,* metadata and free-text descriptions), though inconsistencies in documentation and missing metadata remain key challenges.

Future work will focus on refining ALOHA extraction techniques. This could include integrating natural language processing (NLP) techniques to infer missing metadata, provide semantic understanding of the extracted information, and improving compliance with evolving and different software supply chain standards. Moreover, incorporating validation mechanisms and stakeholder feedback loops can help the AIBoM generation. We also plan to conduct a more in-depth evaluation of ALOHA, focusing on its accuracy, the quality of the generated AIBoMs, and practical usefulness for practitioners. By addressing the aforementioned challenges, ALOHA has the potential to become a cornerstone in AI supply chain security, fostering greater accountability and trust in HF.

## Data and Tool Availability

ALOHA is available open-source under the Mozilla Public License 2.0 License on Zenodo [40]. The replication package of the empirical evaluation (which includes the generated AIBoMs and the spreadsheet of the analysis) is also available on the web [28].


## Acknowledgements

This research has been, in part, financially supported by the European Union NEXTGenerationEU project and by the Italian Ministry of the University and Research MUR, a Research Projects of Significant National Interest (PRIN) 2022 PNRR, project n. D53D23017310001 entitled "Mining Software Repositories for enhanced Software Bills of Materials (MSR4SBOM)".

## References


[1] Joe Biden. (2021) Executive Order on Improving the Nation's Cybersecurity. United States Government. Last Accessed: 21 March 2025. [Online]. Available: https://bidenwhitehouse.archives.gov/briefing-room/presidential-actions/2025/01/16/executive-order-on-strengthening-and-promoting-innovation-in-the-nations-cybersecurity/

[2] C. Skouloudi, A. Malatras, R. Naydenov, and G. Dede. (2020) Guidelines for Securing the Internet of Things - ENISA. European Union Agency for Cybersecurity. Last Accessed: 21 March 2025. [Online]. Available: https://www.enisa.europa.eu/publications/guidelines-for-securing-the-internet-of-things

[3] SPDX. (2024) SPDX - Linux Foundation Projects Site. The Linux Foundation. Last Accessed: 21 March 2025. [Online]. Available: https://spdx.dev

[4] CycloneDX. (2024) OWASP CycloneDX Software Bill of Materials (SBOM) Standard. OWASP Foundation. Last Accessed: 21 March 2025. [Online]. Available: https://cyclonedx.org

[5] ——. (2023) CycloneDX Tool Center. OWASP Foundation. Last Accessed: 21 March 2025. [Online]. Available: https://cyclonedx.org/tool-center/

[6] E. Tooley and C. Claessens. (2023) Introducing self-service SBOMs - The GitHub Blog. GitHub, Inc. Last Accessed: 21 March 2025. [Online]. Available: https://github.blog/2023-03-28-introducing-self-service-sboms

[7] B. Xia, D. Zhang, Y. Liu, Q. Lu, Z. Xing, and L. Zhu, "Trust in software supply chains: Blockchain-enabled sbom and the aibom future," in *Proceedings of the 2024 ACM/IEEE 4th International Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS) and 2024 IEEE/ACM Second International Workshop on Software Vulnerability*. New York, NY, USA: Association for Computing Machinery, 2024, pp. 12–19.

[8] T. Stalnaker, N. Wintersgill, O. Chaparro, M. Di Penta, D. M. German, and D. Poshyvanyk, "Boms away! inside the minds of stakeholders: A comprehensive study of bills of materials for software systems," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, ser. ICSE '24. New York, NY, USA: Association for Computing Machinery, 2024. [Online]. Available: https://doi.org/10.1145/3597503.3623347

[9] D. G. Widder and D. Nafus, "Dislocated accountabilities in the "ai supply chain": Modularity and developers' notions of responsibility," *Big Data & Society*, vol. 10, no. 1, p. 20539517231177620, 2023.

[10] DVC. (2025) Data Version Control. DVC. Last Accessed: 21 March 2025. [Online]. Available: https://dvc.org/

[11] MLflow Project. (2025) MLflow — MLflow — mlflow.org. MLflow Project. Last Accessed: 21 March 2025. [Online]. Available: https://mlflow.org/

[12] H. Face. (2025) Hugging face - the ai community building the future. Hugging Face Inc. Last Accessed: 21 March 2025. [Online]. Available: https://huggingface.co

[13] V. Charles, A. Emrouznejad, and T. Gherman, "A critical analysis of the integration of blockchain and artificial intelligence for supply chain," *Annals of Operations Research*, vol. 327, no. 1, pp. 7–47, 2023.

[14] F. Pepe, V. Nardone, A. Mastropaolo, G. Bavota, G. Canfora, and M. Di Penta, "How do hugging face models document datasets, bias, and licenses? an empirical study," in *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension*, ser. ICPC '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 370–381. [Online]. Available: https://doi.org/10.1145/3643916.3644412

[15] J. Castaño, S. Martínez-Fernández, X. Franch, and J. Bogner, "Analyzing the evolution and maintenance of ml models on hugging face," in *Proceedings of the 21st International Conference on Mining Software Repositories*, ser. MSR '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 607–618. [Online]. Available: https://doi.org/10.1145/3643991.3644898

[16] W. Jiang, N. Synovic, M. Hyatt, T. R. Schorlemmer, R. Sethi, Y.-H. Lu, G. K. Thiruvathukal, and J. C. Davis, "An empirical study of pre-trained model reuse in the hugging face deep learning model registry," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2023, pp. 2463–2475.

[17] A. Ait, J. L. Cánovas Izquierdo, and J. Cabot, "On the suitability of hugging face hub for empirical studies," *Empirical Software Engineering*, vol. 30, no. 2, pp. 1–48, 2025.

[18] J. Castaño, S. Martínez-Fernández, X. Franch, and J. Bogner, "Exploring the carbon footprint of hugging face's ml models: A repository mining study," in *2023 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. IEEE, 2023, pp. 1–12.

[19] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru, "Model cards for model reporting," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, ser. FAT* '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 220–229. [Online]. Available: https://doi.org/10.1145/3287560.3287596

[20] A. Bhat, A. Coursey, G. Hu, S. Li, N. Nahar, S. Zhou, C. Kästner, and J. L. Guo, "Aspirations and practice of ml model documentation: Moving the needle with nudging and traceability," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, ser. CHI '23. New York, NY, USA: Association for Computing Machinery, 2023. [Online]. Available: https://doi.org/10.1145/3544548.3581518

[21] A. Crisan, M. Drouhard, J. Vig, and N. Rajani, "Interactive model cards: A human-centered approach to model documentation," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 427–439. [Online]. Available: https://doi.org/10.1145/3531146.3533108

[22] X. Yang, W. Liang, and J. Zou, "Navigating dataset documentations in AI: A large-scale analysis of dataset cards on huggingface," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=xC8xh2RSs2

[23] B. Xia, T. Bi, Z. Xing, Q. Lu, and L. Zhu, "An empirical study on software bill of materials: Where we stand and the road ahead," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2023, pp. 2630–2642.

[24] P. Radanliev, O. Santos, and A. Brandon-Jones, "Capability hardware enhanced instructions and artificial intelligence bill of materials in trustworthy artificial intelligence systems: analyzing cybersecurity threats, exploits, and vulnerabilities in new software bills of materials with artificial intelligence," *The Journal of Defense Modeling and Simulation*, p. 15485129241267919, 2024.

[25] J. F. Daniel Bardenstein, Nitish kulkarni. (2024) Driving ai transpareny the ai bill of materials. The Linux Foundation. Last Accessed: 21 March 2025. [Online]. Available: https://www.linuxfoundation.org/hubfs/LF%20Research/lfr_spdx_aibom_102524a.pdf

[26] CycloneDX. (2025) Machine Learning Bill of Materials (ML-BOM). OWASP Foundation. Last Accessed: 21 March 2025. [Online]. Available: https://cyclonedx.org/capabilities/mlbom/

[27] S. Nocera, S. Romano, M. D. Penta, R. Francese, and G. Scanniello, "Software bill of materials adoption: A mining study from github," in *2023 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2023, pp. 39–49. [Online]. Available: https://doi.org/10.1109/ICSME58846.2023.00016

[28] R. D'Avino, S. Nocera, D. Bifolco, F. Pepe, M. Di Penta, and G. Scanniello. (2025) Aloha: A(ibom) tool generator for hugging face - replication package of the empirical evaluation. DOI: 10.6084/m9.figshare.28846169. [Online]. Available: https://doi.org/10.6084/m9.figshare.28846169

[29] H. Face. (2025) Hub api endpoints. Hugging Face Inc. Last Accessed: 21 March 2025. [Online]. Available: https://huggingface.co/docs/hub/api

[30] ——. (2025) Hub client library. Hugging Face Inc. Last Accessed: 21 March 2025. [Online]. Available: https://huggingface.co/docs/huggingface_hub/en/index

[31] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian, "The promises and perils of mining github," in *Proceedings of the 11th Working Conference on Mining Software Repositories*, ser. MSR 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 92–101. [Online]. Available: https://doi.org/10.1145/2597073.2597074

[32] H. Face. (2025) Model Cards — how are model tags determined? Hugging Face Inc. Last Accessed: 21 March 2025. [Online]. Available: https://huggingface.co/docs/hub/model-cards#how-are-model-tags-determined

[33] ——. (2025) Model Cards — specifying a library. Hugging Face Inc. Last Accessed: 21 March 2025. [Online]. Available: https://huggingface.co/docs/hub/model-cards#specifying-a-library

[34] ——. (2025) Model Cards — specifying a task ( pipeline_tag ). Hugging Face Inc. Last Accessed: 21 March 2025. [Online]. Available: https://huggingface.co/docs/hub/model-cards#specifying-a-task--pipelinetag-

[35] ——. (2025) configs. Hugging Face Inc. Last Accessed: 21 March 2025. [Online]. Available: https://huggingface.co/docs/transformers.js/en/api/configs
[36] ——. (2025) Displaying carbon emissions for your model. Hugging Face Inc. Last Accessed: 21 March 2025. [Online]. Available: https://huggingface.co/docs/hub/model-cards-co2#how-is-the-carbon-footprint-of-my-model-calculated-
[37] ——. (2025) AutoTrain. Hugging Face Inc. Last Accessed: 21 March 2025. [Online]. Available: https://huggingface.co/autotrain
[38] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, and A. Wessln, *Experimentation in Software Engineering*. Springer, 2012.
[39] D. Bifolco, S. Nocera, S. Romano, M. Di Penta, R. Francese, and G. Scanniello, "On the accuracy of github's dependency graph," in *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, ser. EASE '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 242–251. [Online]. Available: https://doi.org/10.1145/3661167.3661175
[40] R. D'Avino, S. Nocera, D. Bifolco, F. Pepe, M. Di Penta, and G. Scanniello, "Aloha: A(ibom) tool generator for hugging face," Mar. 2025. [Online]. Available: https://doi.org/10.5281/zenodo.15052347