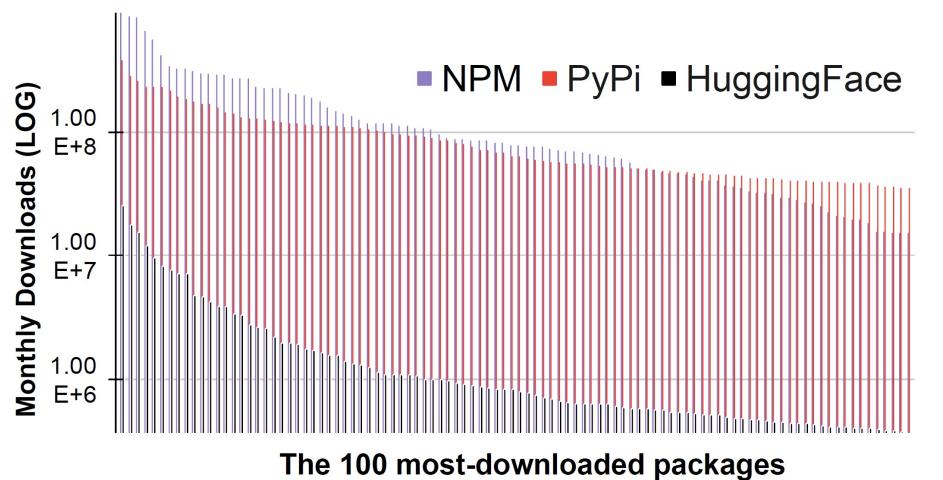
# CERIAS

The Center for Education and Research in Information Assurance and Security

## Trustworthiness Re-use of Pre-trained Neural Networks (ICSE'23)

Wenxin Jiang, Taylor R. Schorlemmer, James C. Davis

### **Motivation**



**Figure 1**. Package download rates comparing two software package registries, NPM and PyPi, with the leading DL model registry, Hugging Face.

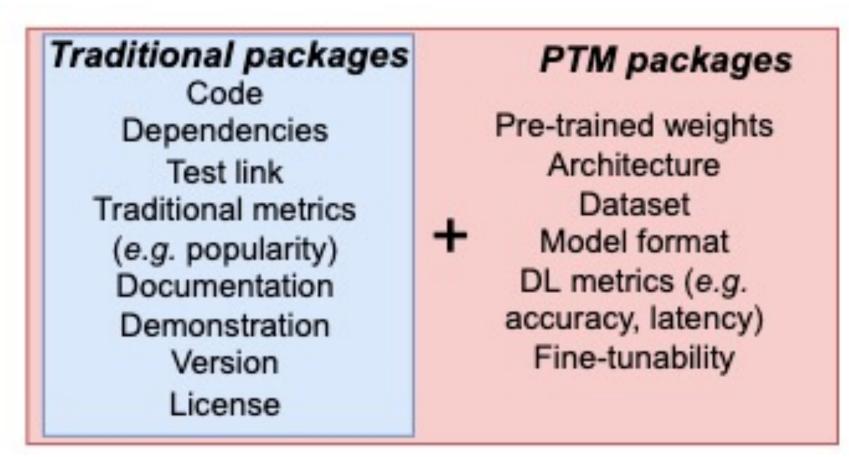


Figure 2. Components of traditional packages and PTM packages

### Methodology

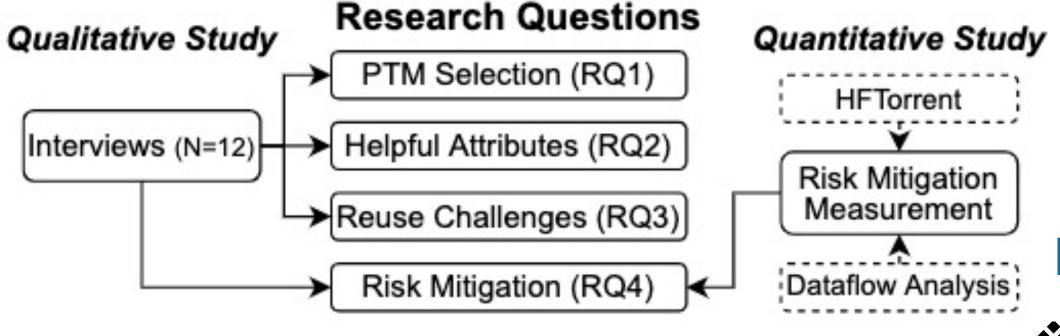


Figure 2. Relationship between RQs and methodology.

### **Results**

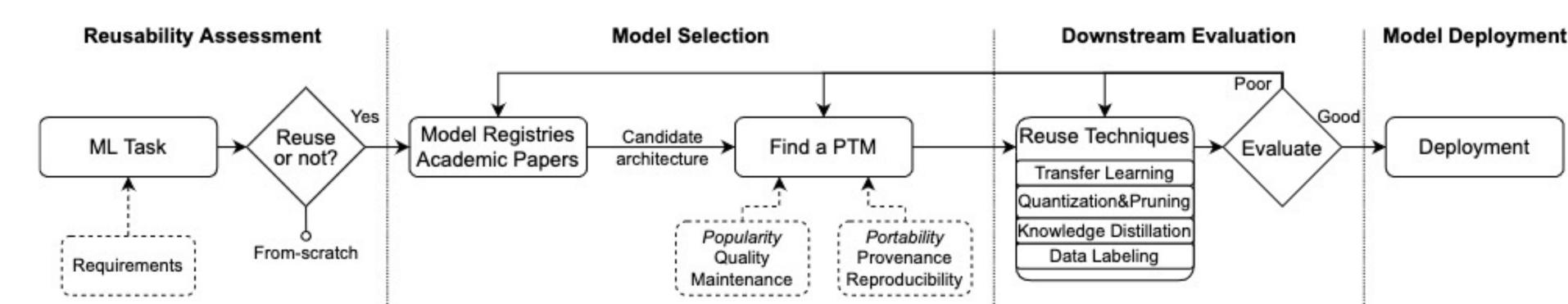
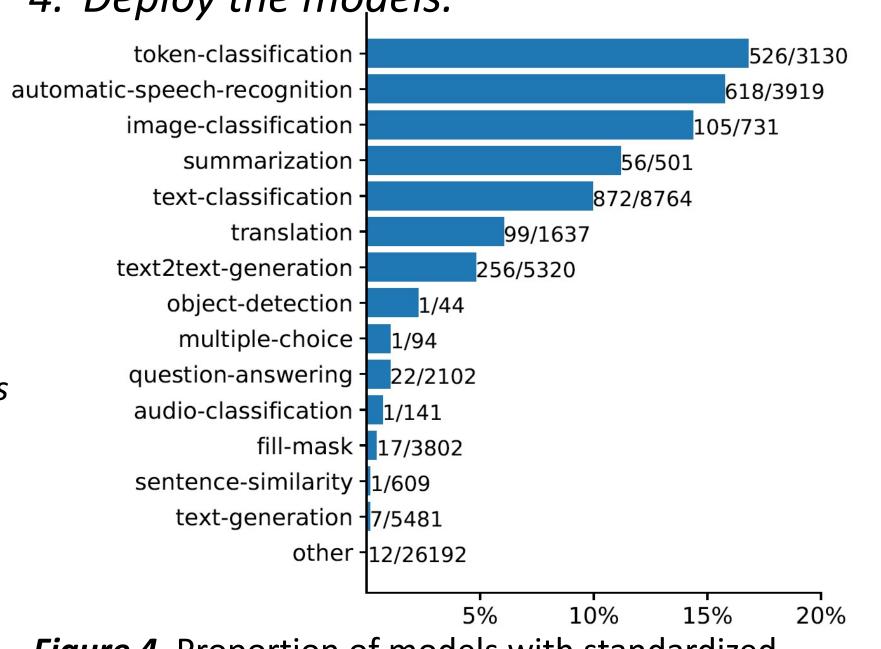


Figure 3. Diagram of the four-stage decision making model for PTM reuse.

The software engineers we interviewed broadly followed four steps.

- 1. Identify a task and determine if reuse is appropriate.
- 2. Select an architecture and a particular PTM.
- 3. Apply reuse techniques and evaluate model performance a potentially repetitive stage.
- 4. Deploy the models.



**Figure 4.** Proportion of models with standardized machine-readable performance claims. Organized by PTM task type.

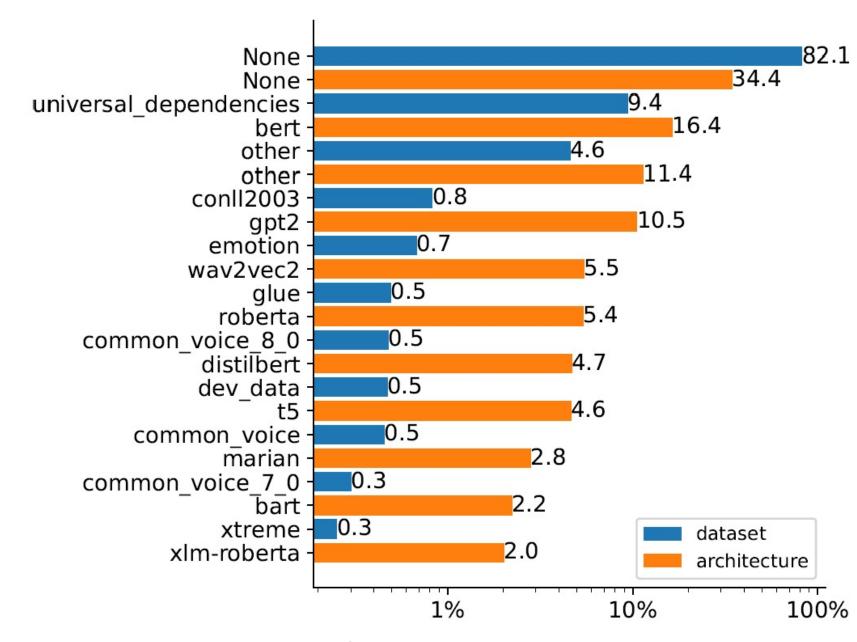


Figure 5. Percent of models using a particular architecture or trained on a specific dataset.

### **Implications**

- \* Model audit: large-scale measurements
- Infrastructure: recommendation systems
- PTM standardization: Information extraction, ONNX
- Adversarial attack detection: detection tools

### PTMTorrent Dataset for Mining Open-source Pre-trained Model Packages (MSR'23)

Sibling

ModelHubURL: string

ModelHubName: string

MetadataFilePath: string

MetadataObjectID: integer

**Data Schema** 

**HFTorrent** 

### **Data Collection**

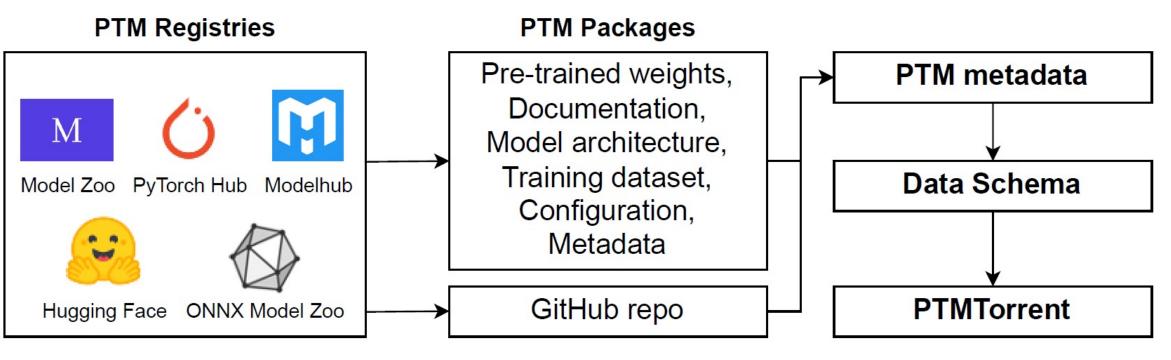


Figure 6. Data collection and processing workflow for PTMTorrent. We standardize data and metadata based on a common PTMTorrent data schema.

Name	# Models	Data Size
Hugging Face [24]	12,401	61TB
Model Zoo [20]	3,245	115GB
PyTorch Hub [21]	49	1.5GB
ONNX Model Zoo [22]	185	441MB
Modelhub [23]	33	721MB
PTMTorrent	15,913	~61TB

Table 1.

Number of
models and
storage size for
each collected
model registry.

#### Category modelld: string slug: string rfilename: string title: string sha: string size: null slug: string enum lastModifed: string description: string blob id: null title: string enum stars: intege tags: string∏ Ifs: null short title: string enum link: string pipeline tag: string framework: string enum siblings: object[] Config private: boolean categories: object[] architectures: string[] ONNXTorrent (Model) author: null model type: string config: object ld: integer ONNXTorrent (Hub) CardData securityStatus: null Model: string ld: integer tags: string[] id: string ModelClass: string language: string id: string ModelPath: string ModelREADMEPath: string license: string cardData: object Paper: string RepoREADMEPath: string datasets: string[] likes: integer ONNXVersion: string Paper: string downloads: integer Accuracy: string Description: string **PTTorrent** library\_name: string Dataset: string enum HFSpaceURL: string Id: integer OpsetVersion: string Category: string Category: string enum ModelAuthor: string Github URL: string **PTM Torrent** ModelDescription: string ModelSampleSize: string ld: integer MHTorrent ModelURL: string ModelSamplePath: string Modelhub: object ld: string GitHubURL: string ModelName: string name: string ColabURL: string ModelURL: string type: string DemoURL: string Dataset ModelOwner: string github: string ModelOwnerURL: string DatasetOwner: string github\_branch: string enum Modelhub DatasetURL: string LatestGitCommitSHA: string backend: string[]

MZTorrent

**Figure 7.** PTMTorrent's data schema. *Grey boxes*: general schema shared by each model hub. *Colored boxes*: customized schema for hub-specific data.

gpu: boolean

### **Future Work**

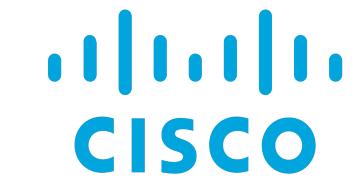
- Supporting future PTM supply chain analysis.
- Expanding PTM model registry analysis.
- Furthering the state of mining tool development.

#### **More Information**









ModelPaperDOIs: string[] [0..1]

ModelArchitecture: string [0..1]

ModelTask: string [0..1]

Datasets: object[] [0..\*]



DatasetOwnerURL: string

DatasetPaperDOI: string [0..1]

DatasetUsages: string[] [0..1]

DatasetName: string

