A Comprehensive Study on Deep Learning Bug Characteristics

Md Johirul Islam
mislam@iastate.edu
Iowa State University
Ames, IA, USA

Giang Nguyen
gnguyen@iastate.edu
Iowa State University
Ames, IA, USA

Rangeet Pan
rangeet@iastate.edu
Iowa State University
Ames, IA, USA

Hridesh Rajan
hridesh@iastate.edu
Iowa State University
Ames, IA, USA

Md Johirul Islam

Data Scientist at Lowe's Companies, Inc.

夏洛特地铁 · 500+ 位好友

Lowe's Companies, Inc.

Iowa State University

Giang Nguyen

Graduate Student

PhD student

Fall 2019 Rotation Student

Rangeet Pan

Research Intern at Microsoft, Ph.D. student at Iowa State University

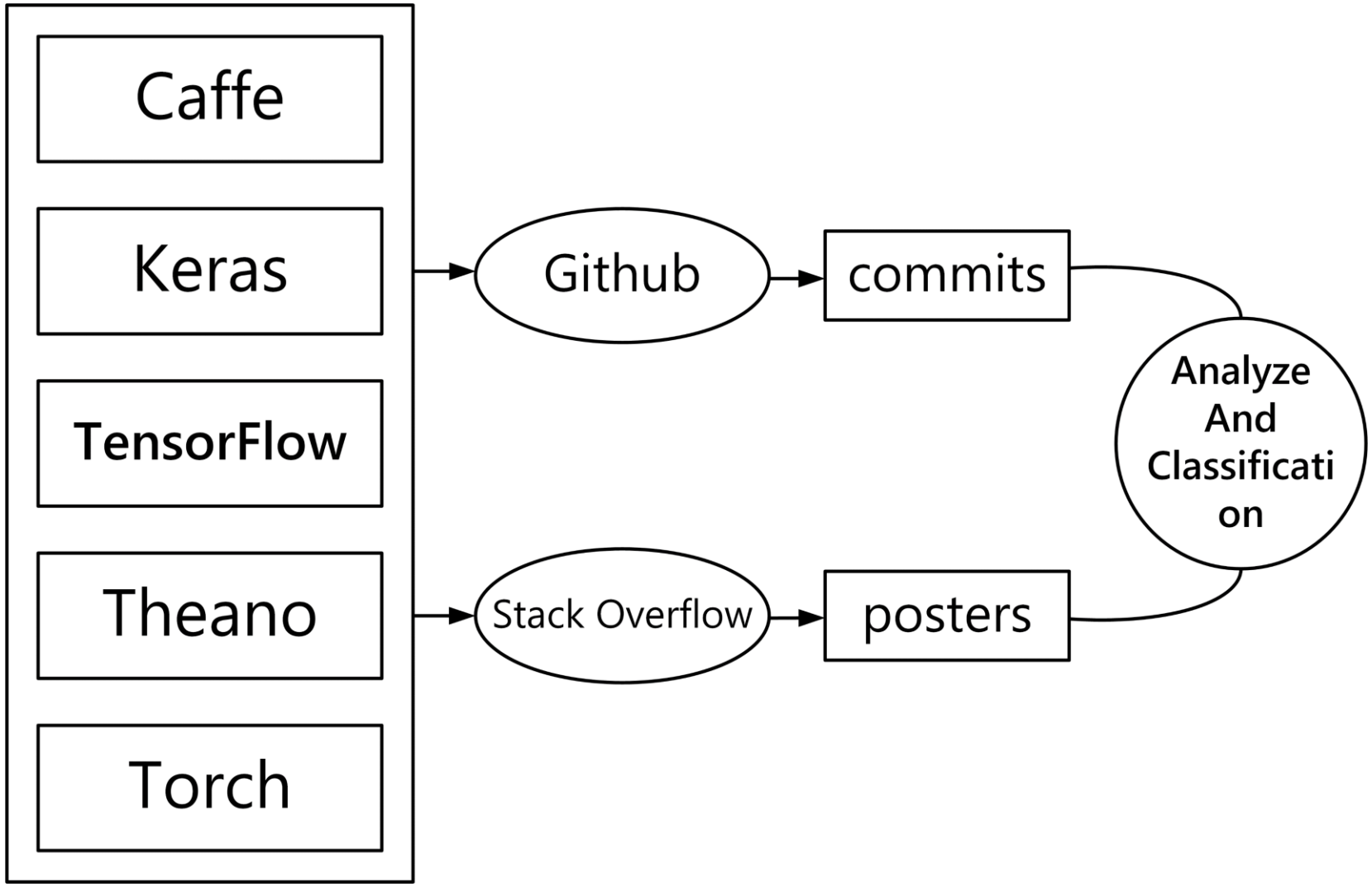
Iowa State University

Iowa State University

Hridesh Rajan

Professor

Research Objective



Motivation

- A class of machine learning algorithms known as deep learning has received much attention in both academia and industry.

Existing work

- bugs in the implementation of machine learning libraries themselves (Ferdian Thung, Shaowei Wang, David Lo, and Lingxiao Jiang. 2012. An empirical study of bugs in machine learning systems, International Symposium on Software Reliability Engineering)

- bugs in the usage of a specific deep learning library (Yuhao Zhang, Yifan Chen, Shing-Chi Cheung, Yingfei Xiong, and Lu Zhang. 2018. An empirical study on TensorFlow program bugs, ACM SIGSOFT International Symposium on Software Testing and Analysis)

In this paper

- Focuses on the characteristics of bugs in software that makes use of deep learning libraries.
- Caffe\Keras\Tensorflow\Theano\Torch

- RQ1: (Bug Type) What type of bugs are more frequent?
- RQ2: (Root cause) What are the root causes of bugs?
- RQ3: (Bug Impact) What are the frequent impacts of bugs?
- RQ4: (Bug prone stages) Which deep learning pipeline stages are
- more vulnerable to bugs?
- RQ5: (Commonality) Do the bugs follow a common pattern?
- RQ6: (Bug evolution) How did the bug pattern change over time?

Method: Data collection

Stackoverflow

- Searching for posts tagged with Caffe, Keras, Tensorflow, Theano, and Torch.
- Filter out posts that did not contain any source code because posts about bugs usually contain code snippets
- Grade more than 5
- Manually read: If the best-accepted answer was to fix the usages of the deep learning API(s) in the question, we considered that post as talking about deep learning bugs.

Method: Data collection

Github

- find the repositories that contain the keywords related to the libraries
- commits whose title contains the word ”fix”
- manually check the import statements in the program
- randomly select 100 commits for each library
- manually read

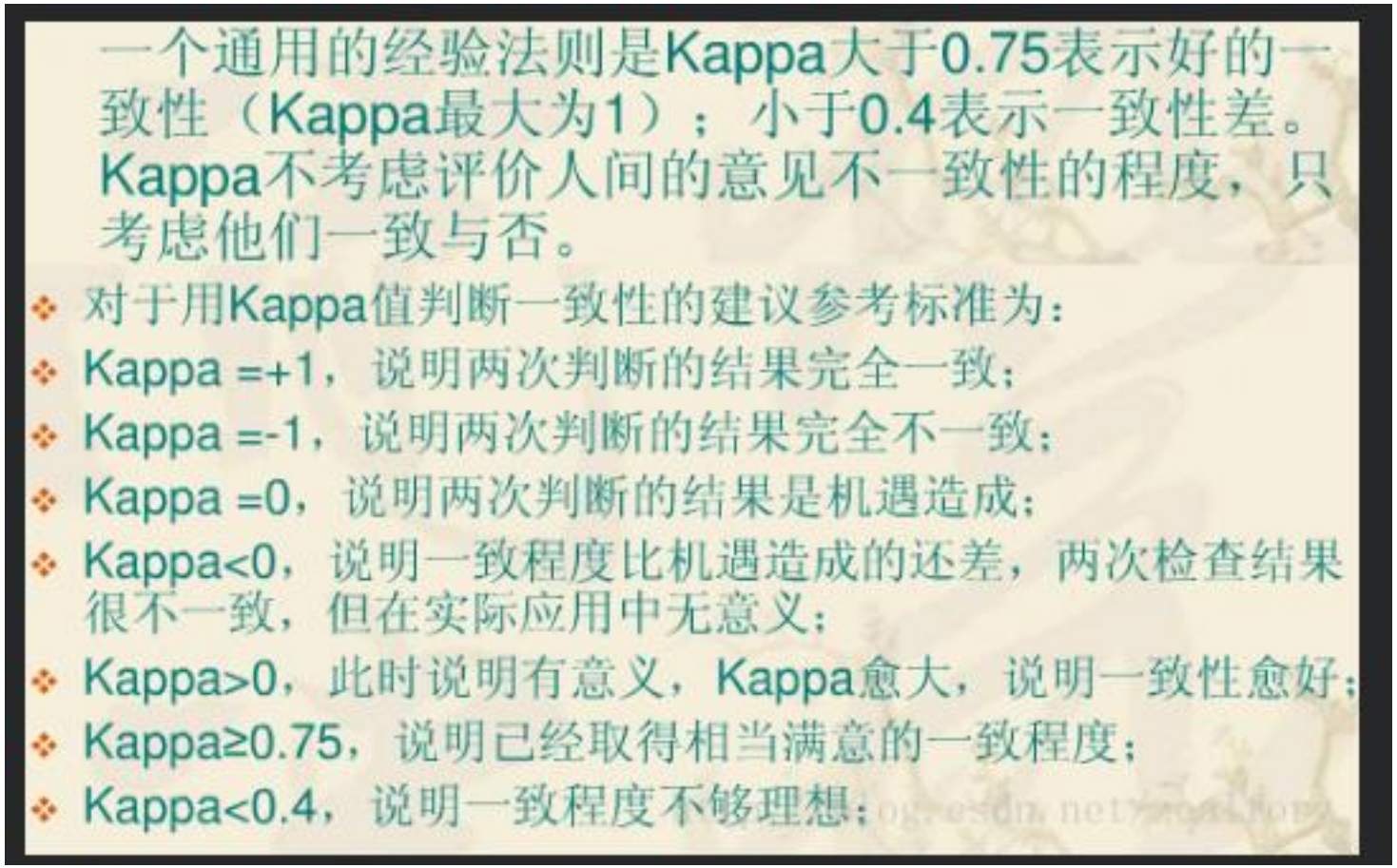
Table 1: Summary of the dataset used in the Study

Library	*Stack Overflow*		*Github*	
	# Posts	# Bugs	# Commits	# Bugs
Caffe	183	35	100	26
Keras	567	162	100	348
Tensorflow	1558	166	100	100
Theano	231	27	100	35
Torch	177	25	100	46
Total	2716	415	500	555

Method: Labeling the Bugs

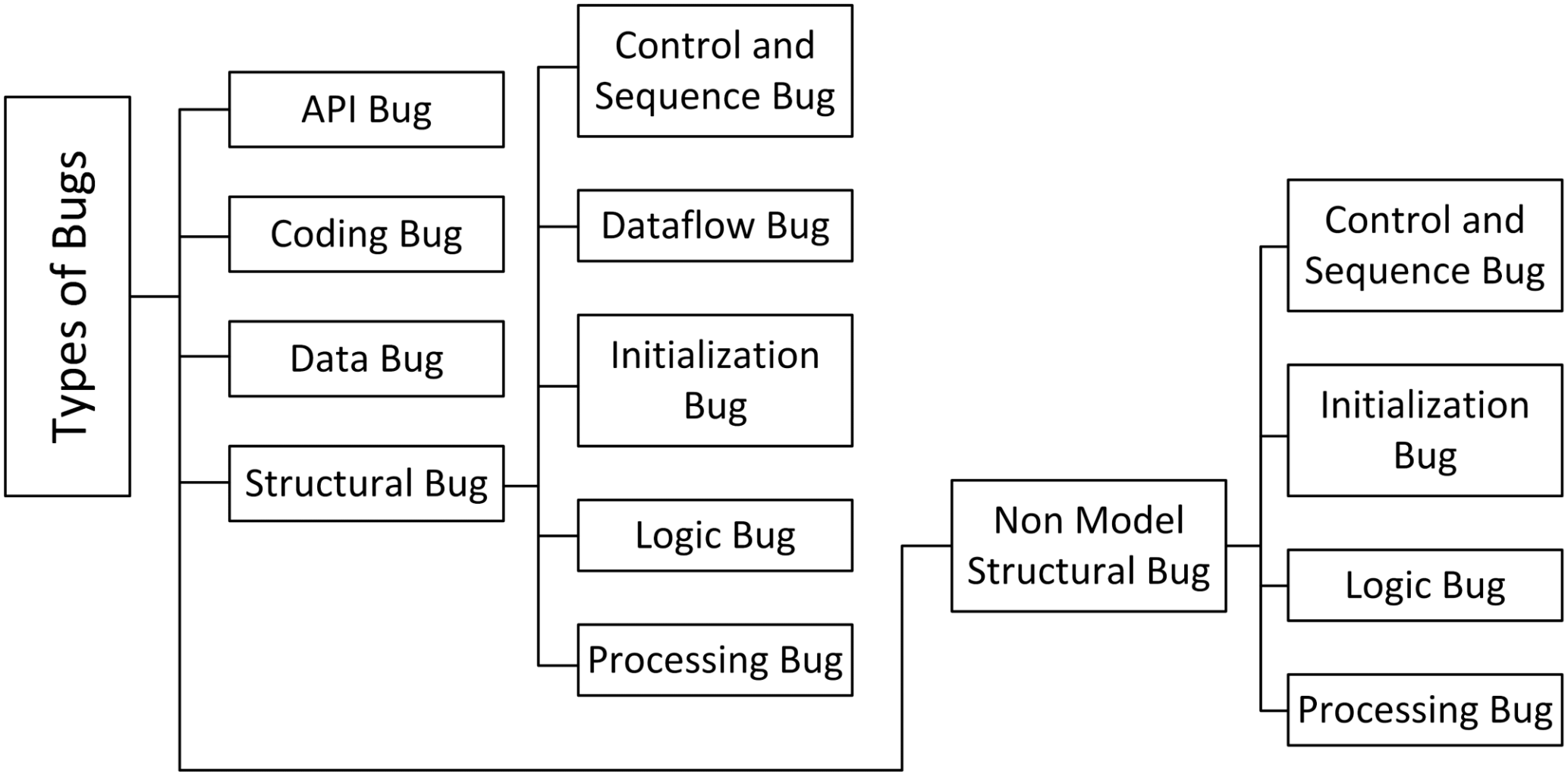
- Cohen's Kappa coefficient

- 5% : 0
- 10% : 82%
- >90%



Classification (Types of Bugs)

Boris Beizer. 1984. Software system testing and quality assurance



Data bugs

```
def _read32(bytestream):
  dt = numpy.dtype(numpy.uint32).newbyteorder('>')
  return numpy.frombuffer(bytestream.read(4), dtype=dt)
```

```
TypeError: only integer scalar arrays can be converted to a scalar
      index
```

```
return numpy.frombuffer(bytestream.read(4), dtype=dt)[0]
```

API bugs

```
model.fit(tX, tY, epochs=100, batch_size=1, verbose=2)
```

The developer will get the error because epochs keyword does not exist in version 2+ of *Keras*.

```
model.fit(tX, tY, batch_size=1, verbose=2, epochs = 100) File
"keras/models.py", line 612, in fit str(kwargs))
Exception: Received unknown keyword arguments: {'epochs': 100}
```

To fix this error, the developer needs to change the keyword parameter from epochs to nb_epoch.

```
model.fit(tX, tY, nb_epoch=100, batch_size=1, verbose=2)
```

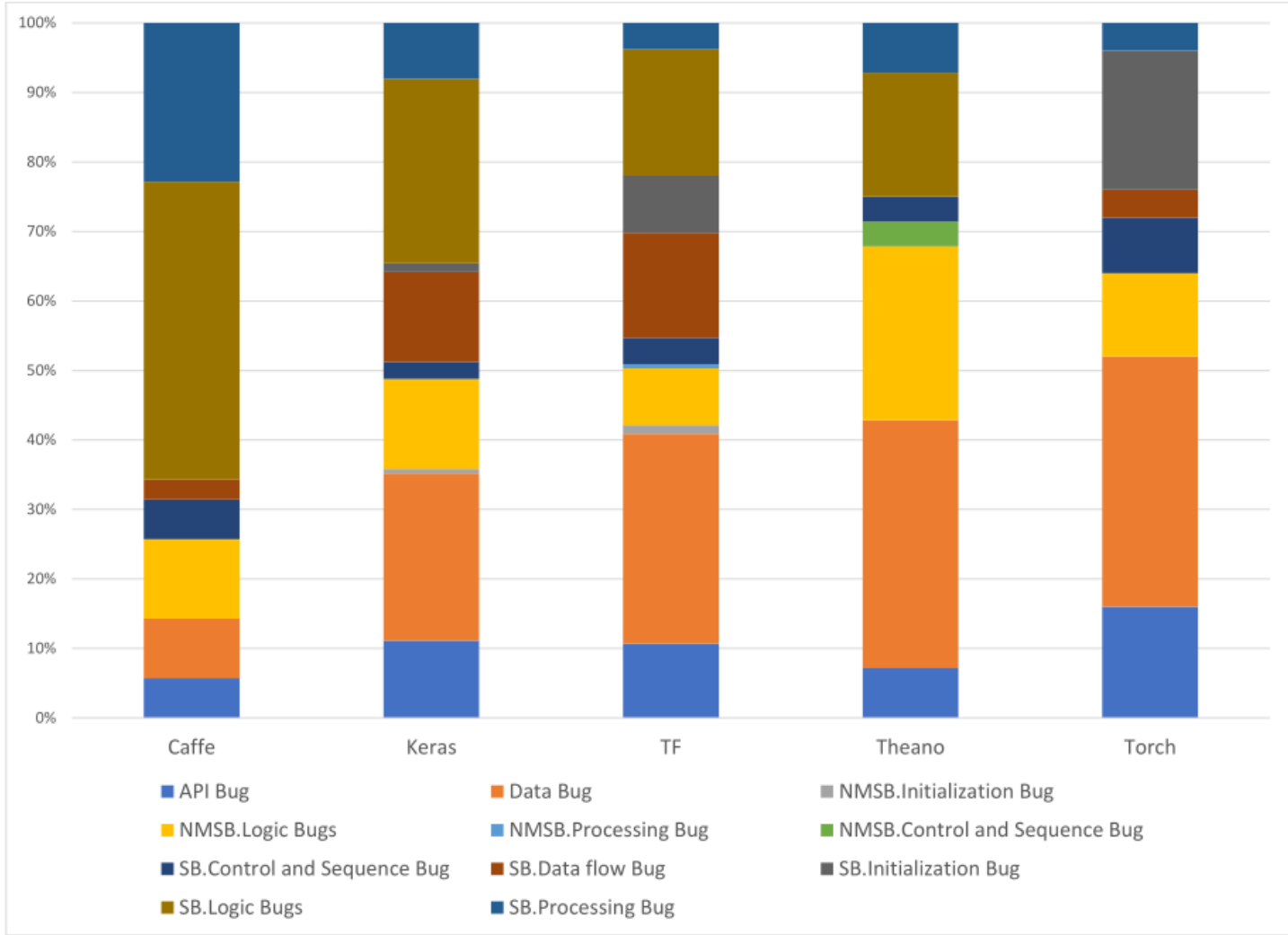


Figure 1: Distribution of Bug Types in *Stack Overflow*

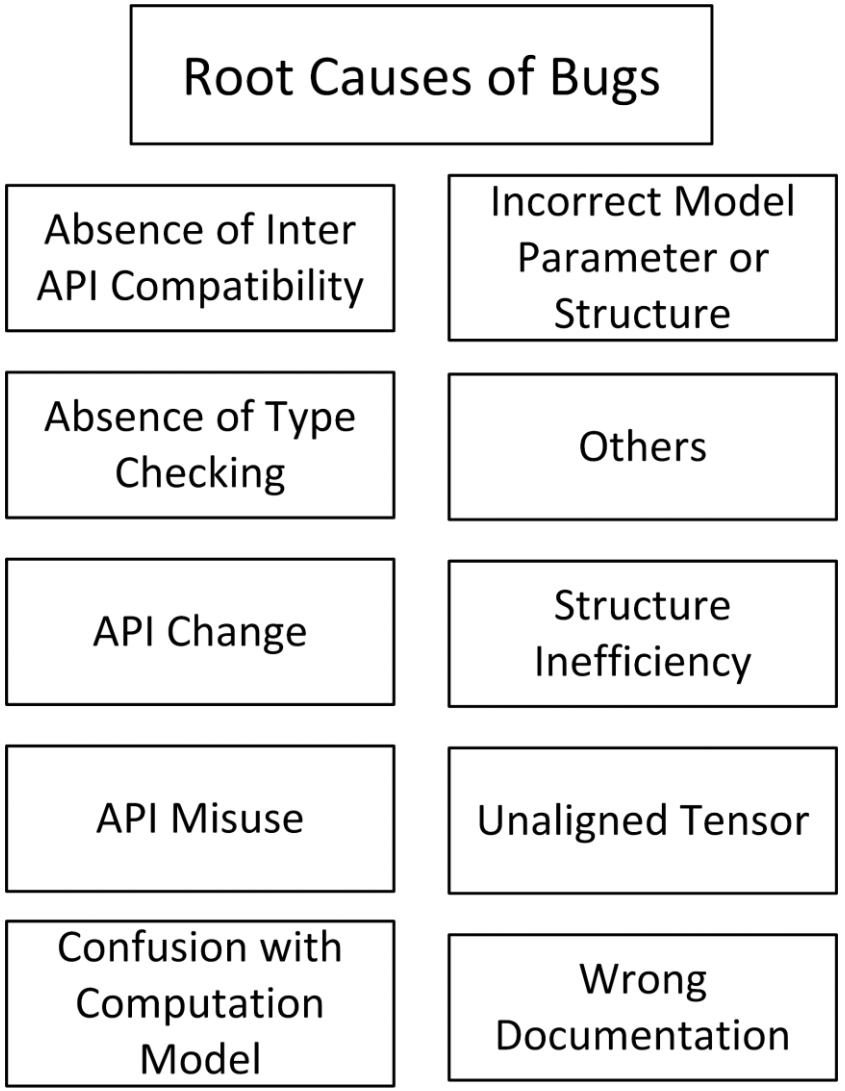
Table 2: Statistics of Bug Types in *Stack Overflow* and *Github*

	Caffe		Keras		TF		Theano		Torch		P value
	SO	GitHub	SO	GitHub	SO	GitHub	SO	GitHub	SO	GitHub	
API Bug	6%	0%	11%	57%	11%	72%	7%	3%	16%	2%	0.3207
Data Bug	9%	49%	24%	8%	30%	0%	35%	17%	36%	15%	0.3901
NMSB.Control and Sequence Bug	0%	8%	0%	0%	0%	0%	4%	0%	0%	7%	0.3056
NMSB.Initialization Bug	0%	0%	1%	0%	1%	0%	0%	3%	0%	0%	0.7655
NMSB.Logic Bugs	11%	0%	13%	2%	8%	0%	25%	6%	12%	7%	0.0109
NMSB.Processing Bug	0%	0%	0%	0%	1%	0%	0%	3%	0%	7%	0.2323
SB.Control and Sequence Bug	6%	12%	2%	0%	4%	0%	4%	3%	8 %	9%	1.0000
SB.Data flow Bug	3%	8%	13%	26%	15%	0%	0%	14%	4%	16%	0.2873
SB.Initialization Bug	0%	0%	1%	0%	8%	1%	0%	23%	20%	11%	0.8446
SB.Logic Bugs	42%	15%	27%	3%	18%	23%	18%	14%	0%	13%	0.3442
SB.Processing Bug	23%	8%	8%	4%	4%	4%	7%	14%	4%	13%	0.8535

RQ1: (Bug Type) What type of bugs are more frequent?

- Finding1: Data Bugs appear more than 26% of the times
- Finding 2: Caffe has 43% Structural Logic Bugs
- Finding 3: Torch, Keras, Tensorflow have 16%, 11% and 11% API bugs respectively
- Finding 4: All the bug types except Non Model Structural Logic Bug have a similar pattern in Github and Stack Overflow for all the libraries

Classification (Root Causes of Bugs)



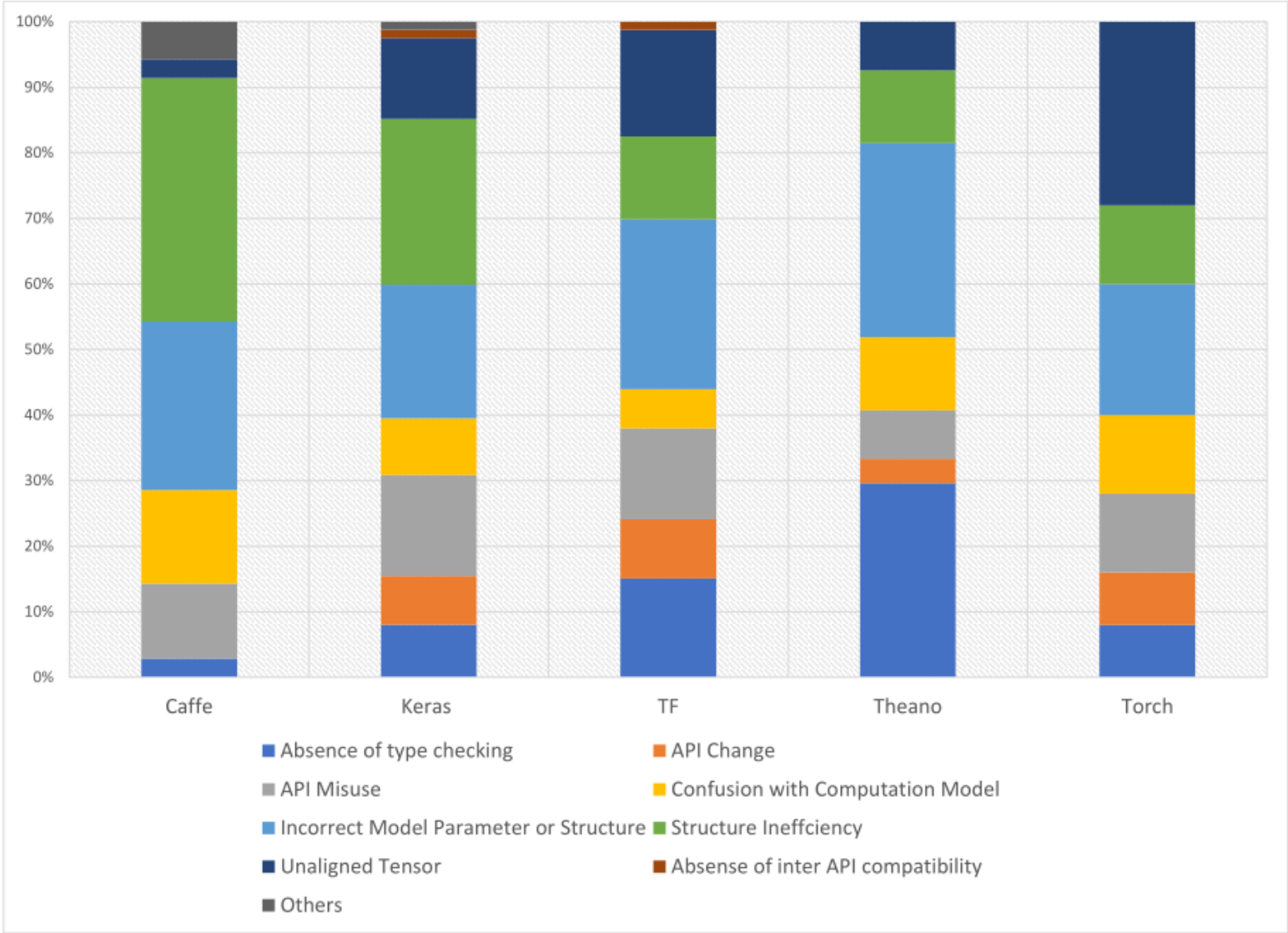


Figure 2: Stack Overflow Root Cause Classification

Table 3: Statistics of the Root Causes of Bugs

	Caffe		Keras		TF		Theano		Torch		P value
	SO	GitHub	SO	GitHub	SO	GitHub	SO	GitHub	SO	GitHub	
Absense of inter API compatibility	0%	0%	1%	0%	1%	0%	0%	0%	0%	0%	0.1411
Absence of type checking	3%	12%	8%	3%	15%	15%	30%	20%	8%	13%	0.9717
API Change	0%	0%	7%	51%	9%	58%	4%	0%	8%	2%	0.2485
API Misuse	11%	0%	15%	4%	14%	0%	7%	3%	12%	2%	0.0003
Confusion with Computation Model	14%	28%	9%	1%	6%	10%	11%	3%	12%	4%	0.7839
Incorrect Model Parameter or Structure	26%	31%	21%	30%	26%	16%	30%	14%	20%	19%	0.5040
Others	0%	0%	0%	0%	0%	0%	0%	0%	0%	2%	0.3466
Structure Ineffciency	37%	12%	26%	5%	13%	1%	11%	26%	12%	38%	0.7170
Unaligned Tensor	3%	19%	12%	5%	16%	0%	7%	34%	28%	20%	0.7541
Wrong Documentation	6%	0%	1%	1%	0%	0%	0%	0%	0%	0%	0.3402

RQ2: (Root cause) What are the root causes of bugs?

- Finding 5: Incorrect Model Parameter (IPS) is the most common root cause resulting in average 24% of the bugs across the libraries.
- Finding 6: Keras, Caffe have 25% and 37% bugs that arise from Structural Inefficiency (SI)
- Finding 7: Torch has 28% of the bugs due to Unaligned Tensor (UT)
- Finding 8: Theano has 30% of the bugs due to the absence of type checking
- Finding 9: Tensorflow and Keras have 9% and 7% bugs due to API change
- Finding 10: Except API Misuse all other root causes have similar patterns in both Github and Stack Overflow root causes of bugs
- Finding 11: Structural Inefficiency (SI) contributes 3% - 53% and IPS contributes 24% - 62% of the bugs related to model

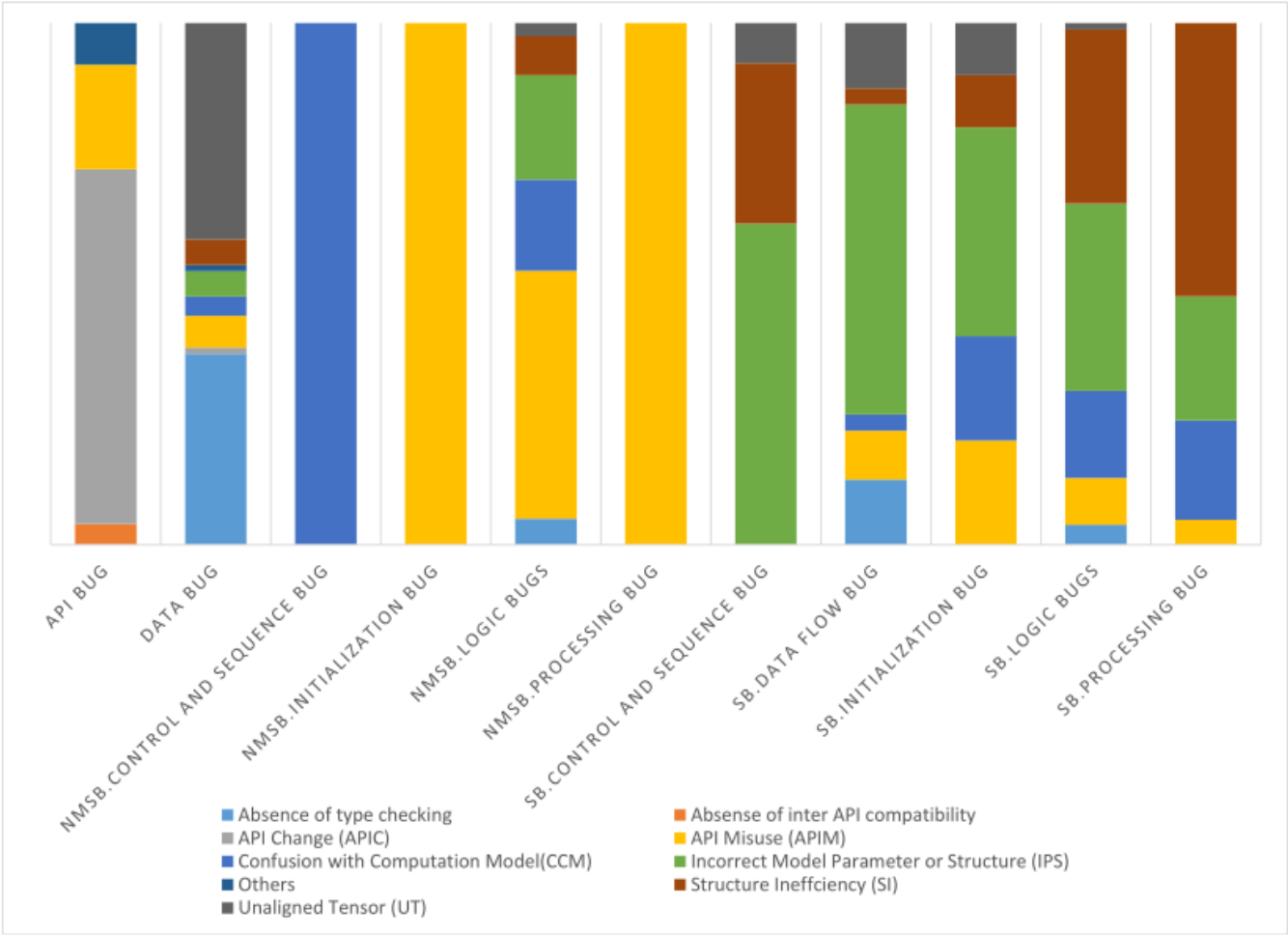
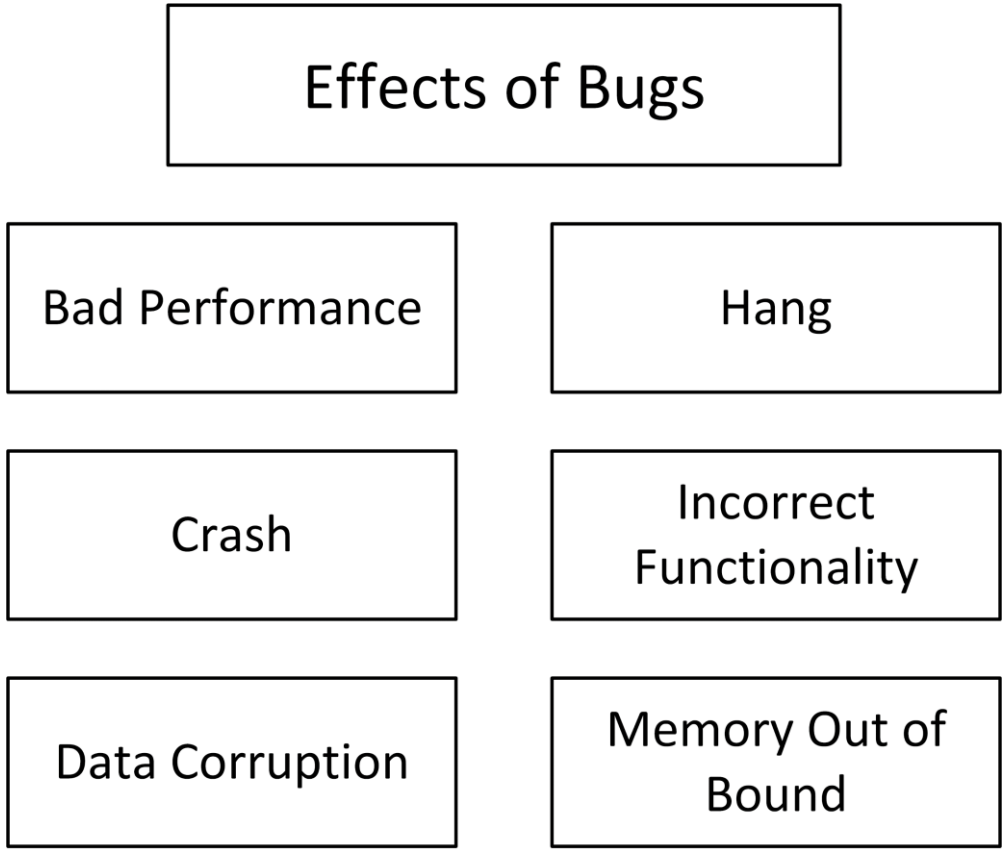


Figure 3: Relation between Root Causes and Types of Bugs

Classification (Effects of Bugs)



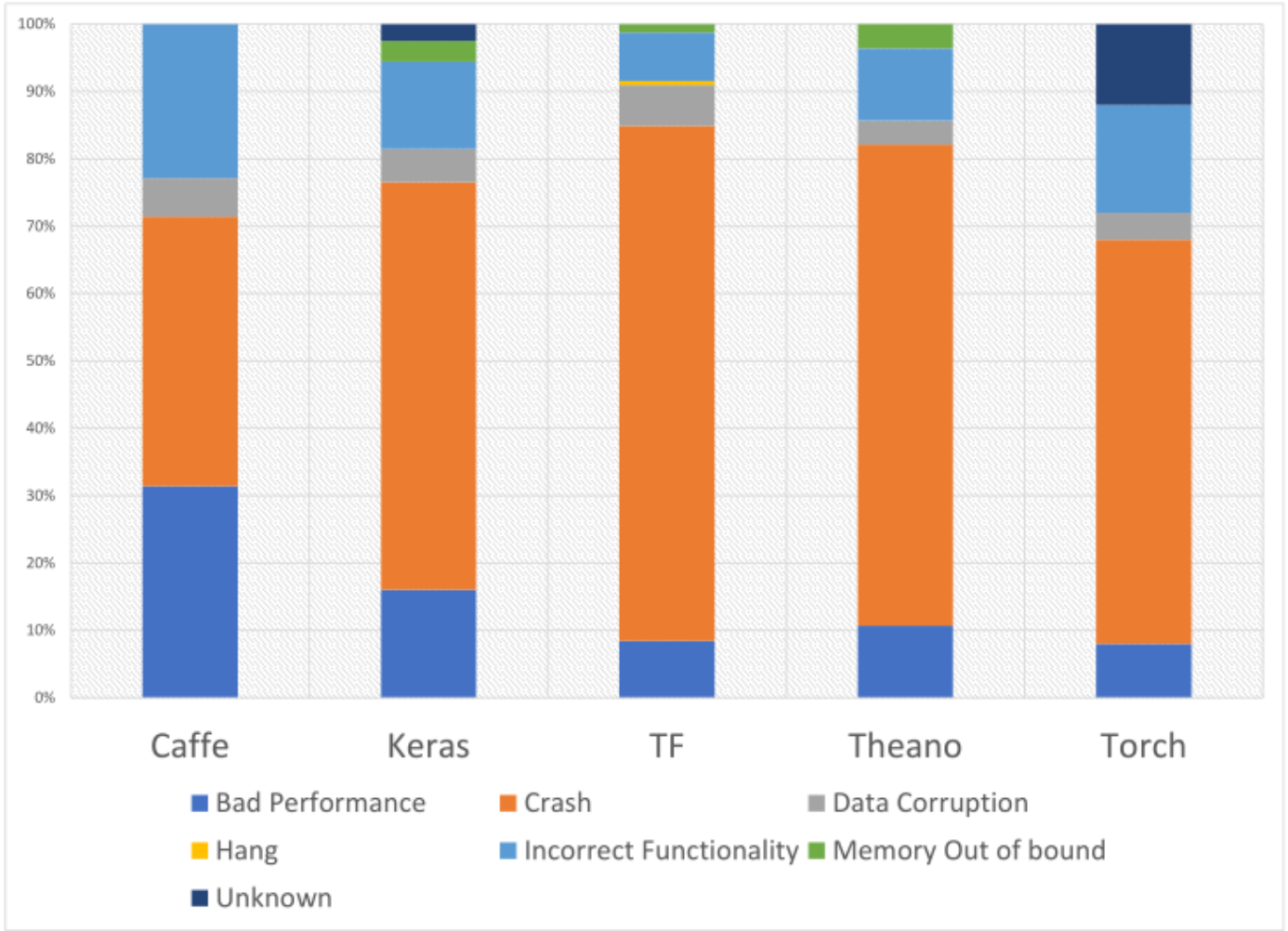


Figure 4: Distribution of Bug Effects in *Stack Overflow*

Table 4: Effects of Bugs in *Stack Overflow* and *Github*

	Caffe		Keras		TF		Theano		Torch		P value
	SO	GitHub	SO	GitHub	SO	GitHub	SO	GitHub	SO	GitHub	
Bad Performance	31%	19%	16%	14%	8%	8%	11%	6%	8%	24%	0.9152
Crash	40%	69%	61%	86%	77%	92%	70%	20%	60%	16%	0.7812
Data Corruption	6%	4%	5%	0%	6%	0%	4%	6%	4%	16%	0.948
Hang	0%	0%	0%	0%	1%	0%	0%	0%	0%	0%	0.3466
Incorrect Functionality	23%	8%	13%	0%	7%	0%	11%	59%	16%	42%	0.5418
Memory Out of bound	0%	0%	3%	0%	1%	0%	4%	0%	0%	0%	0.0844
Unknown	0%	0%	2%	0%	0%	0%	0%	9%	12%	2%	0.8419

RQ3: (Bug Impact) What are the frequent impacts of bugs?

- Finding 12: More than 66% of the bugs cause crash.
- Finding 13: In Caffe, Keras, Tensorflow, Theano, Torch 31%, 16%, 8%, 11%, and 8% bugs lead to bad performance respectively.
- Finding 14: 12% of the bugs cause Incorrect Functionality.
- Finding 15: For all the libraries the P value for Stack Overflow and Github bug effects reject the null hypothesis to confirm that the bugs have similar effects from Stack Overflow as well as Github bug.

RQ4: (Bug prone stages) Which deep learning pipeline stages are more vulnerable to bugs?

- Finding 16: 32% of the bugs are in the data preparation stage.
- Finding 17: 27% of the bugs are seen during the training stage.
- Finding 18: Choice of model stage shows 23% of the bugs

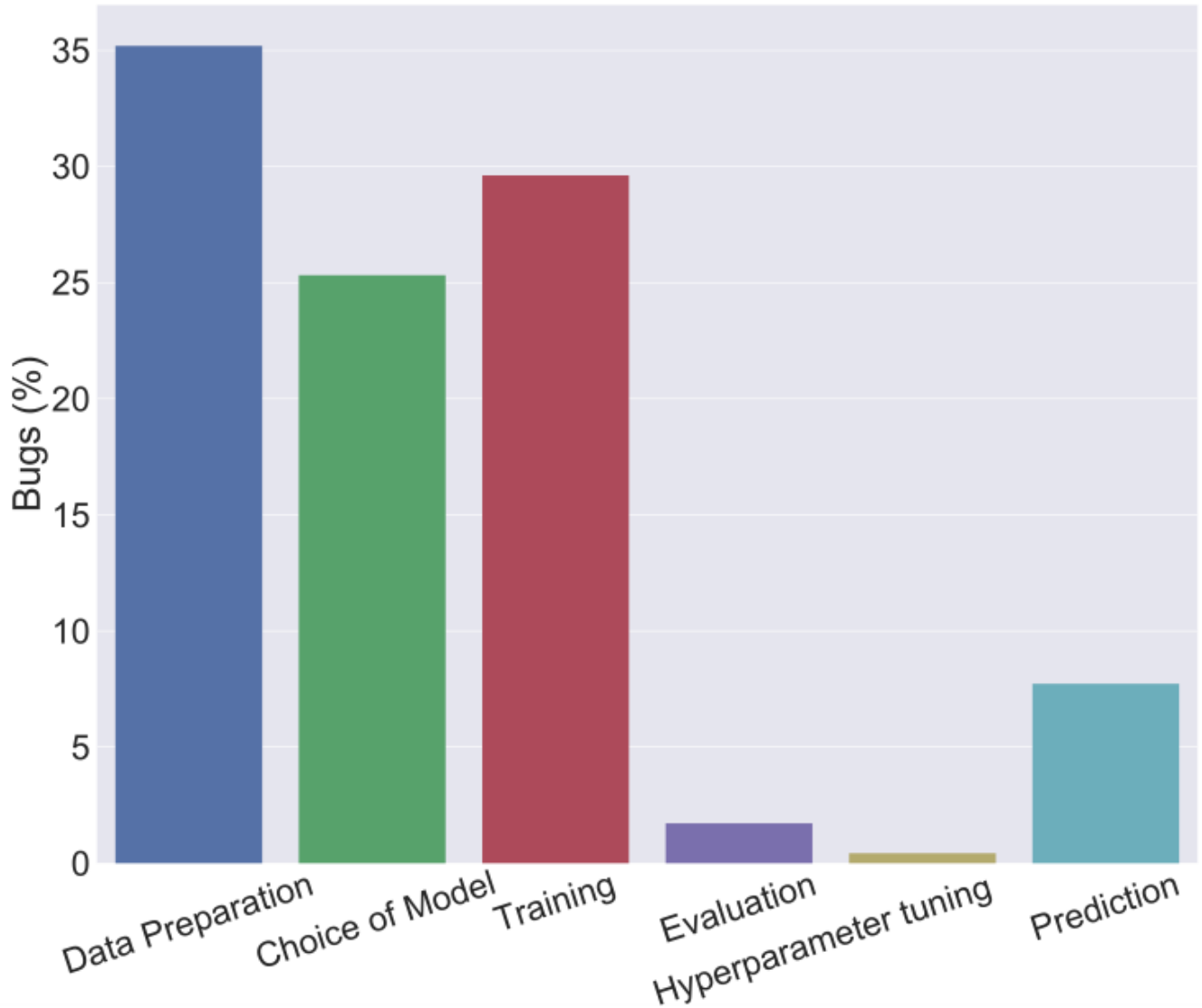


Figure 5: Bugs across stages of the Deep Learning pipeline

RQ5: (Commonality) Do the bugs follow a common pattern?

- Finding 19: Tensorflow and Keras have a similar distribution of antipatterns while Torch has different distributions of antipatterns

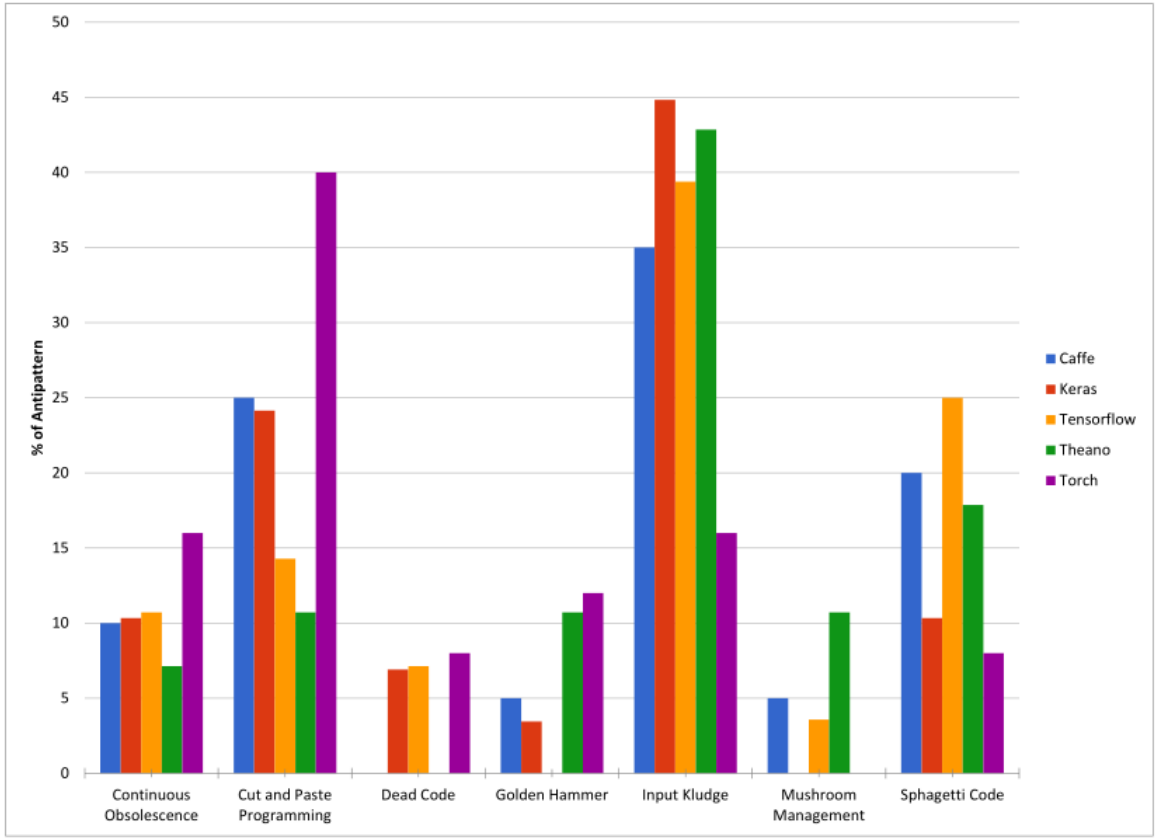


Figure 7: Distribution of different antipatterns

Antipatterns:

https://sourcemaking.com/antipatterns/software-development-antipatterns

RQ6: (Bug evolution) How did the bug pattern change over time?

- Finding 20: In Keras, Caffe, Tensorflow Structural logic bugs are showing increasing trend.

- Finding 21: Data Bugs slowly decreased since 2015 except Torch.

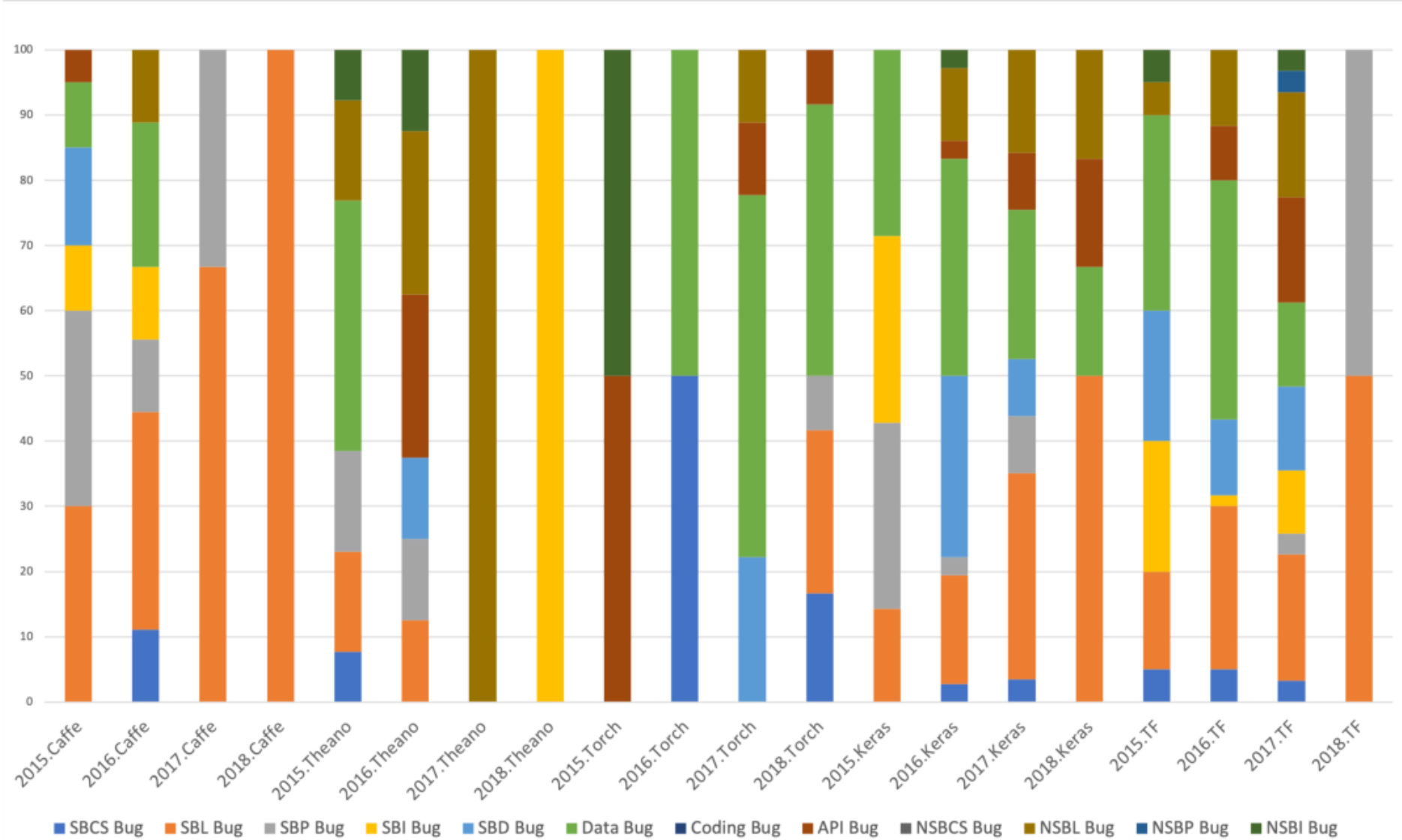


Figure 9: Timeline of Evolution of Bugs