Anything to Hide? Studying Minified and Obfuscated Code in the Web

Philippe Skolka **Cristian-Alexandru Staicu** Michael Pradel

TU Darmstadt

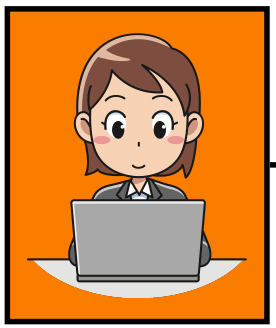
www.software-lab.org

15th of May 2019

TECHNISCHE
UNIVERSITÄT
DARMSTADT

SOLA
SoftwareLab

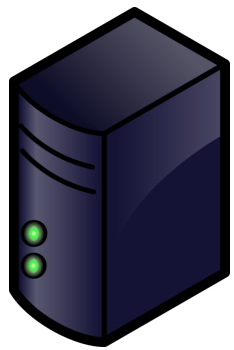
Code Distribution on the Web



write

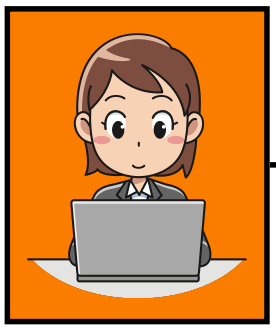
```
function sum(first, second) {
  return first + second;
}
```

deploy



serve

Code Distribution on the Web



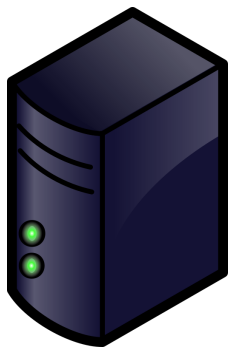
write

```
function sum(first, second) {
  return first + second;
}
```

transform

```
function sum(n,u){return n+u}
```

deploy



serve

Common Transformations: Minification and Obfuscation

Hand-written

```
function hi(name) {
  console.log("Hi" + " " + name);
}
hi();
```

Objective: ↑ maintainability

Common Transformations: Minification and Obfuscation

Hand-written

```
function hi(name) {
  console.log("Hi" + " " + name);
}
hi();
```

Objective: ↑ maintainability

Minified

```
function hi(i){console.log("Hi "+i)}hi();
```

Objective: ↓ code size

Common Transformations: Minification and Obfuscation

Hand-written

```
function hi(name) {
  console.log("Hi" + " " + name);
}
hi();
```

Objective: ↑ maintainability

Minified

```
function hi(i){console.log("Hi "+i)}hi();
```

Objective: ↓ code size

Obfuscated

```
var a=['\x6c\x6f\x67'];var b=function(c,d){
c=c-0x0;var e=a[c];return e;};function
c(d){console[b('0x0')]('\x48\x69'+'\x20'+d);}c();
```

Objective: ↓ understandability

Research Questions

RQ1: **How prevalent** is transformed code on the web?

Research Questions

RQ1: **How prevalent** is transformed code on the web?

RQ2: **Which tools** are used for obfuscation on the web?

Research Questions

RQ1: **How prevalent** is transformed code on the web?

RQ2: **Which tools** are used for obfuscation on the web?

RQ3: Does prevalence differ among website **categories**?

Research Questions

RQ1: **How prevalent** is transformed code on the web?

RQ2: **Which tools** are used for obfuscation on the web?

RQ3: Does prevalence differ among website **categories**?

RQ4: What **behavior is hidden** using obfuscation?

Research Questions

RQ1: **How prevalent** is transformed code on the web?

RQ2: **Which tools** are used for obfuscation on the web?

RQ3: Does prevalence differ among website **categories**?

RQ4: What **behavior is hidden** using obfuscation?

RQ5: How do transformations impact **performance?**

Research Questions

RQ1: **How prevalent** is transformed code on the web?

RQ2: **Which tools** are used for obfuscation on the web?

RQ3: Does prevalence differ among website **categories**?

RQ4: What **behavior is hidden** using obfuscation?

RQ5: How do transformations impact **performance?**

RQ6: How do transformations impact **correctness?**

Research Questions

RQ1: **How prevalent** is transformed code on the web?

RQ2: **Which tools** are used for obf… the web?

RQ3: Does prevalen… …ong website **categories**?

RQ4: Wha… **…avior is hidden** using obfuscation?

RQ5: How do transformations impa… **?**

RQ6: How… …ations impact **correctness?**

How are the transformations used?

What is their cost?

Why Machine Learning?

Why Machine Learning?

- Large scale study, expensive to do manual

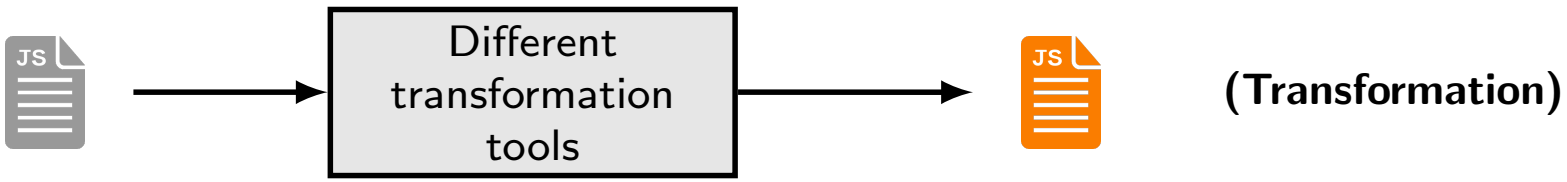
Why Machine Learning?

- Large scale study, expensive to do manual
- Heuristics hard to get right

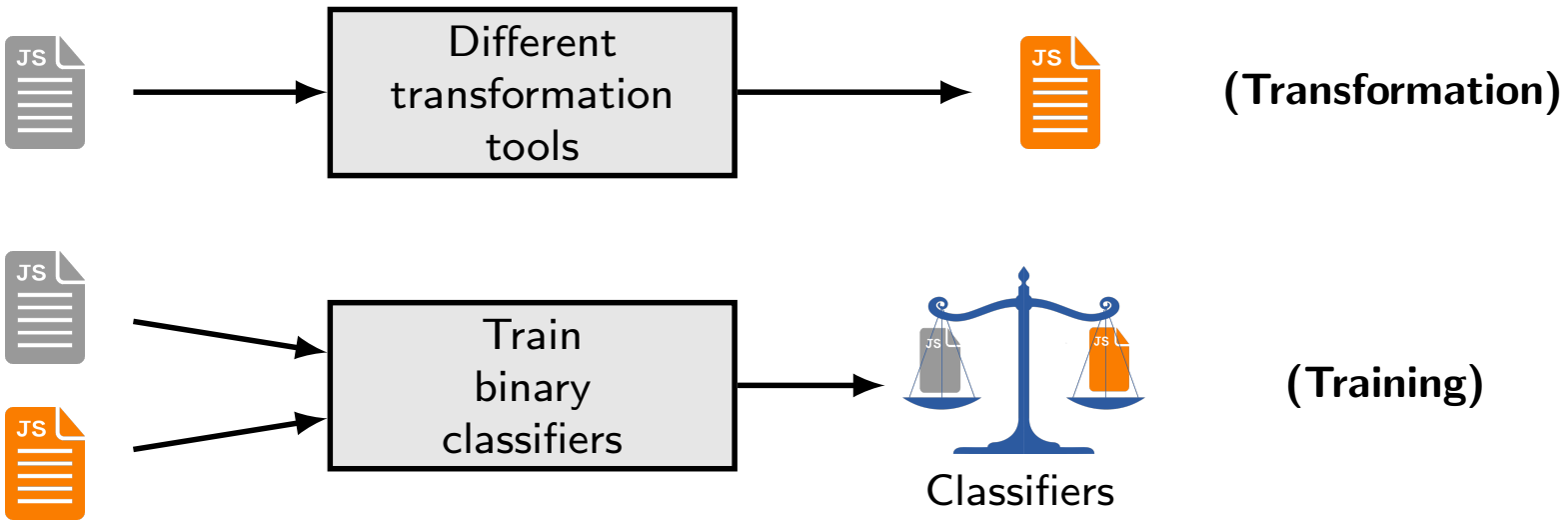
Why Machine Learning?

- Large scale study, expensive to do manual
- Heuristics hard to get right
- Training data easy to acquire

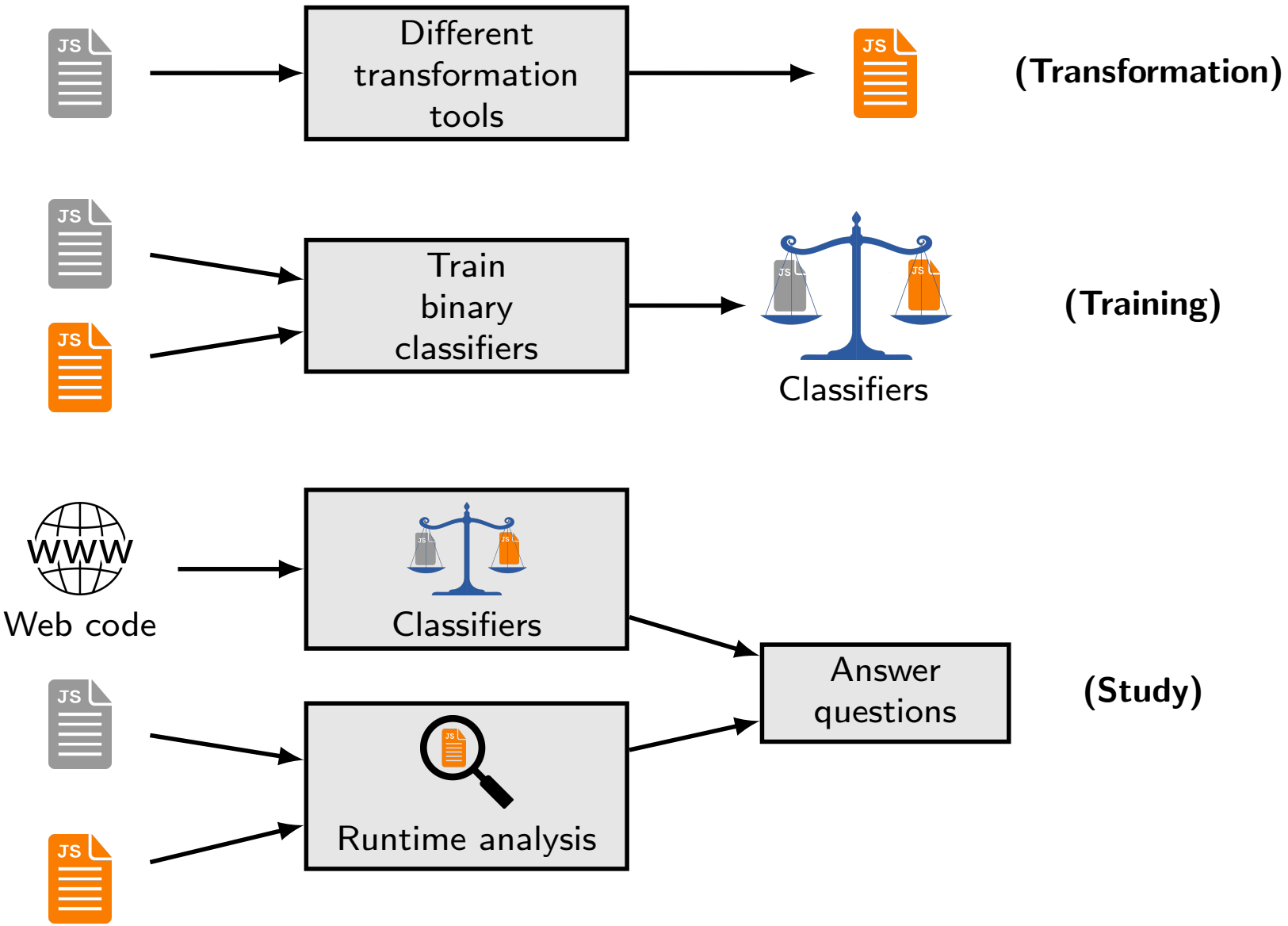
Methodology



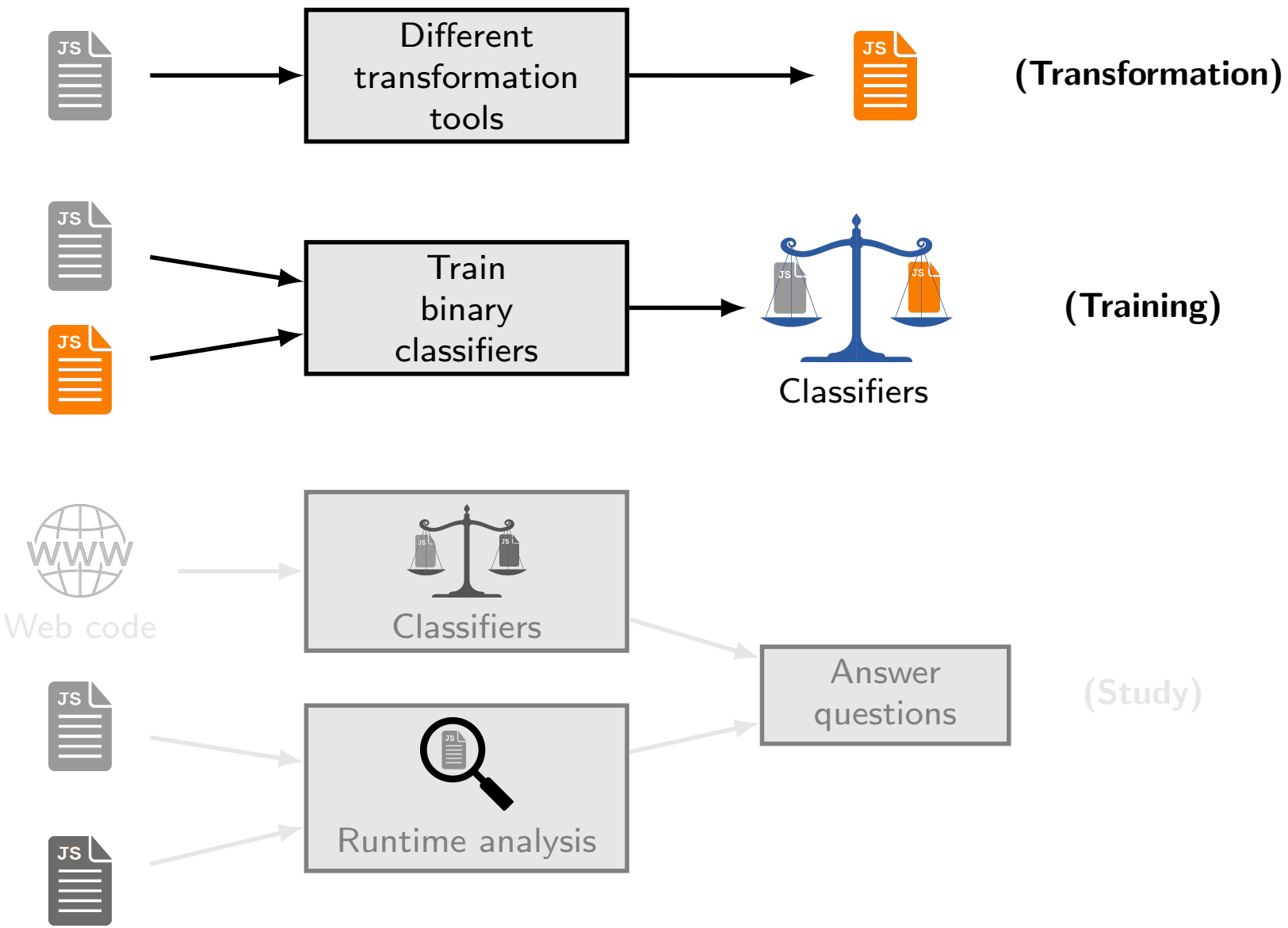
Methodology



Methodology



Methodology



Transformation Tools

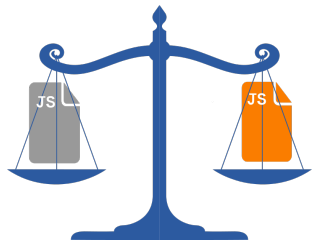
Minifiers	**Obfuscators**
UglifyJS	javascript-obfuscator
babel-minify	javascriptobfuscator.com
Google Closure Compiler	DaftLogic Obfuscator
javascript-minifier.com	jfogs
Matthias Mullie Minify	JSObfu
YUI Compressor	

- **11 tools** with a total of 46 different configuration
- transform files from the "150k Javascript Dataset"[1]

[1] Raychev, V., Bielik, P., Vechev, M. and Krause, A., *Learning Programs from Noisy Data*, POPL '16

Classification Tasks

Seven binary classifiers:



Classification Tasks

Seven binary classifiers:

- **TRANSFORMATION** classifier
  Is the code transformed?

Classification Tasks

Seven binary classifiers:

- **TRANSFORMATION** classifier
  Is the code transformed?
- **OBFUSCATION** classifier
  Is the code obfuscated?

Classification Tasks

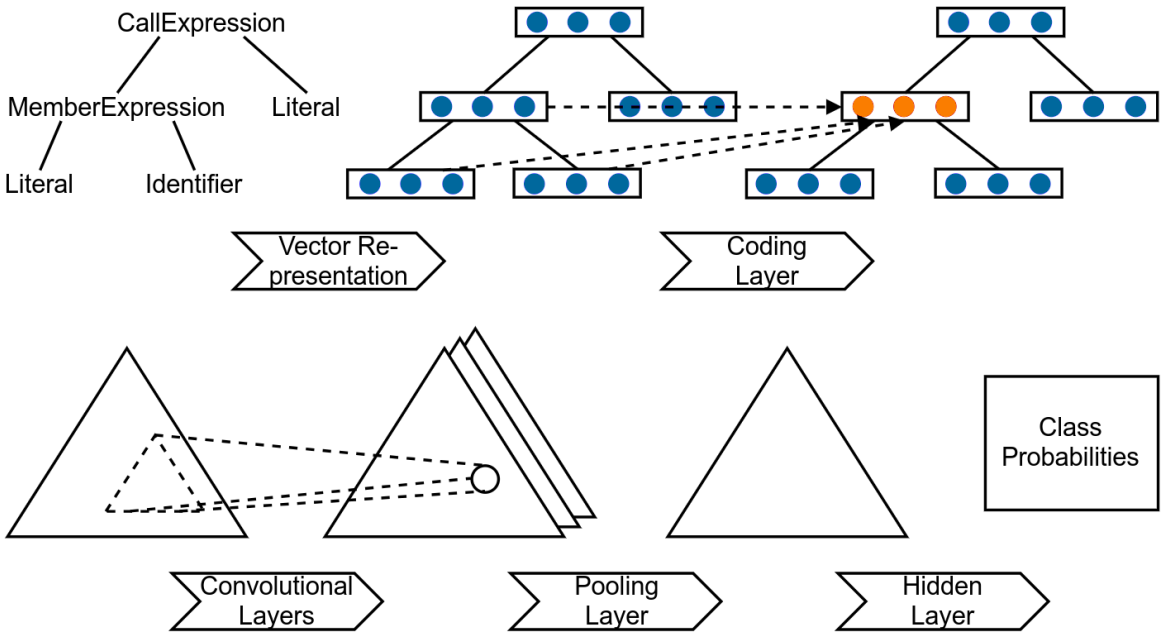
Seven binary classifiers:

- **TRANSFORMATION** classifier
  Is the code transformed?
- **OBFUSCATION** classifier
  Is the code obfuscated?
- **TOOL-X** classifier $*$ 5
  Is the code produced by a given obfuscation tool?

Binary Classifiers

Convolutional neural network

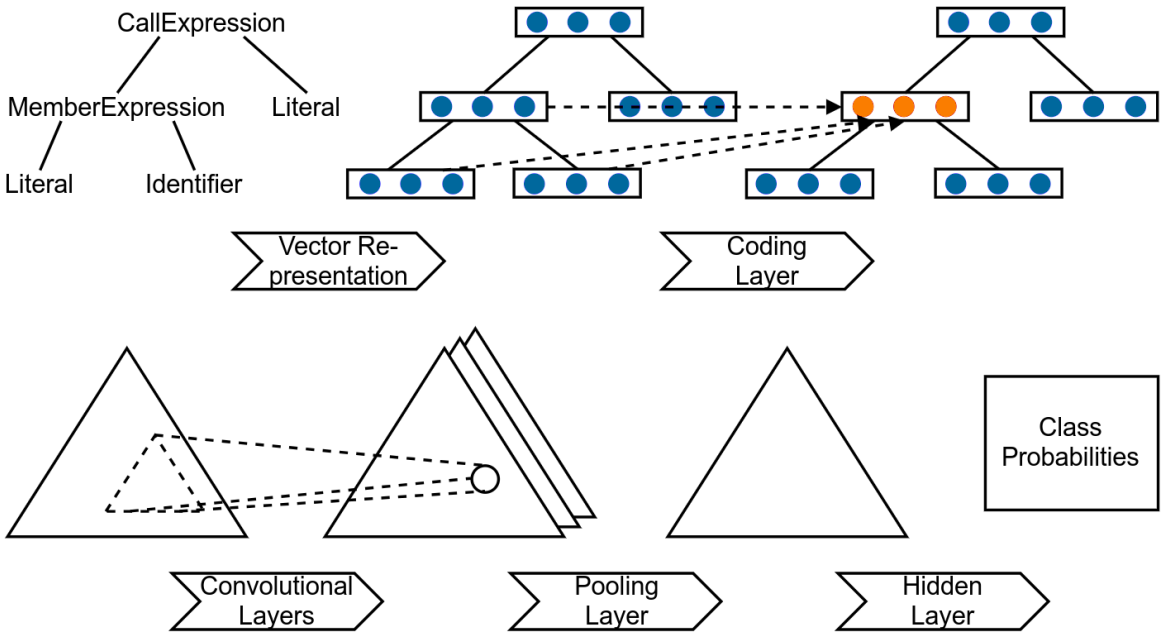
- reuse existing architecture [Mou et al., AAAI, 2016]



Binary Classifiers

Convolutional neural network

- reuse existing architecture [Mou et al., AAAI, 2016]

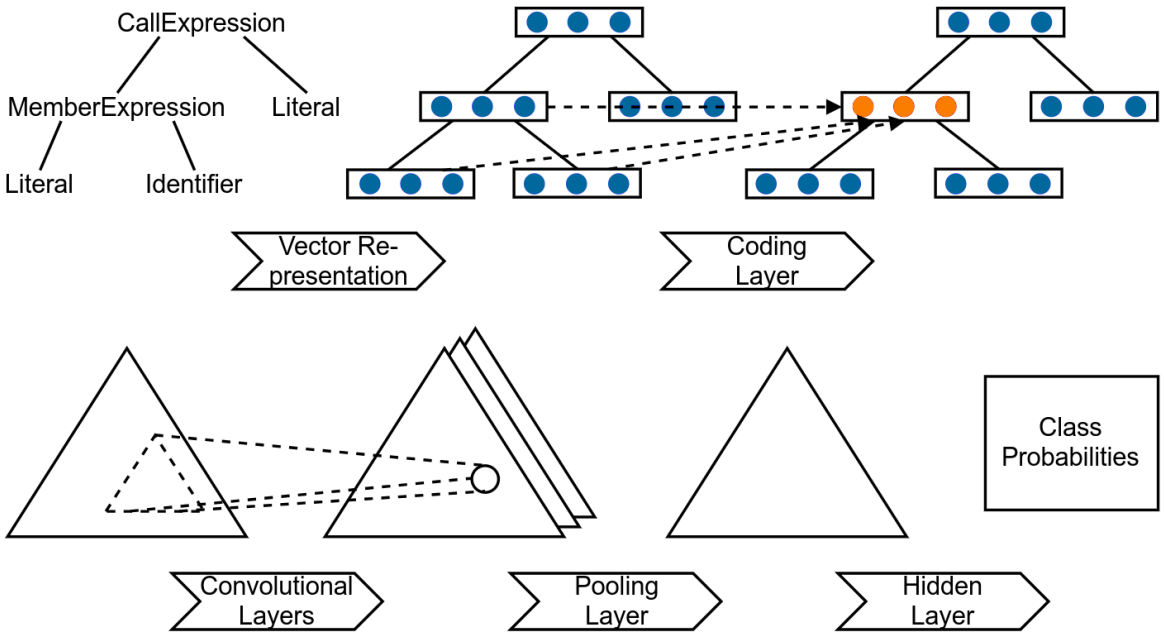


- input to the network: **simplified abstract syntax tree (AST)** representation of code

Binary Classifiers

Convolutional neural network

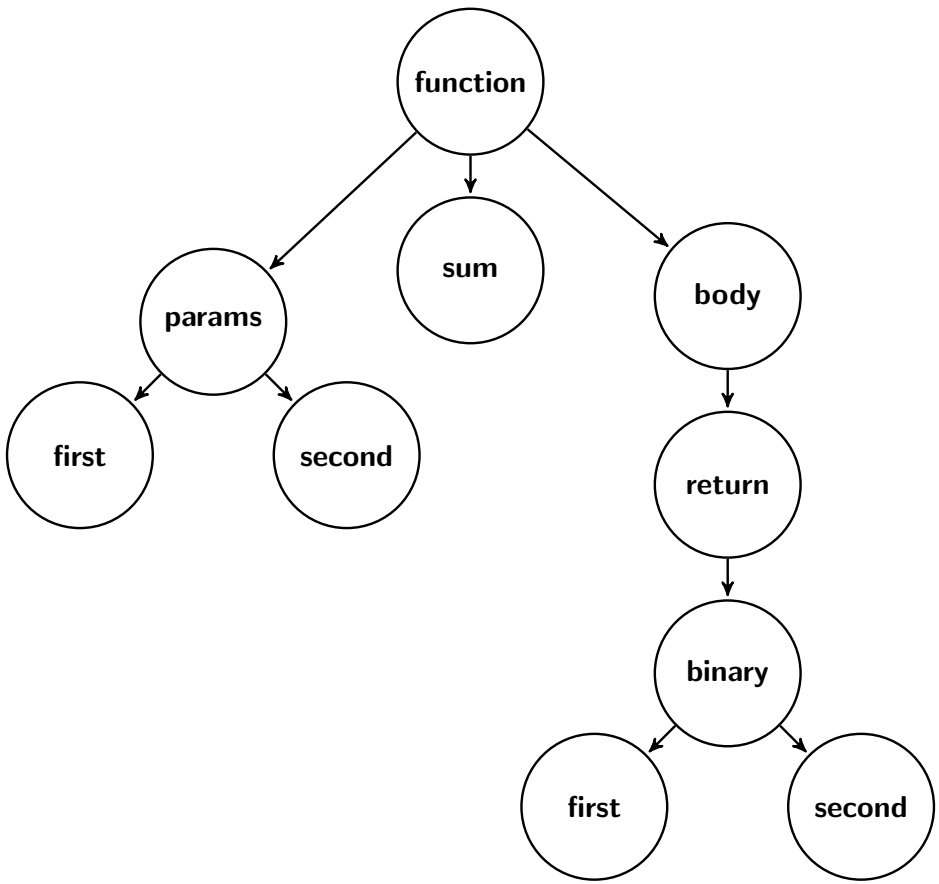
- reuse existing architecture [Mou et al., AAAI, 2016]



- input to the network: **simplified abstract syntax tree (AST)** representation of code
- 30 feature vector size, 50 epochs, batch size 1

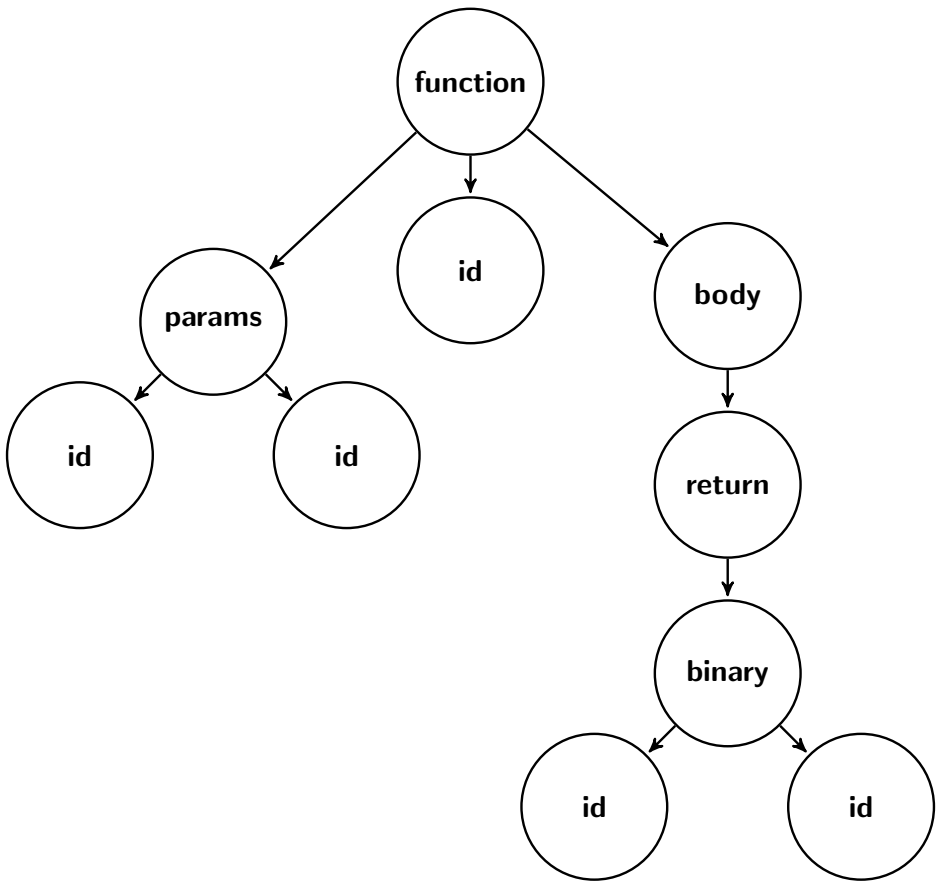
AST Representation for Convolutional Neural Network

```
function sum(first,second) {
  return first + second;
}
```



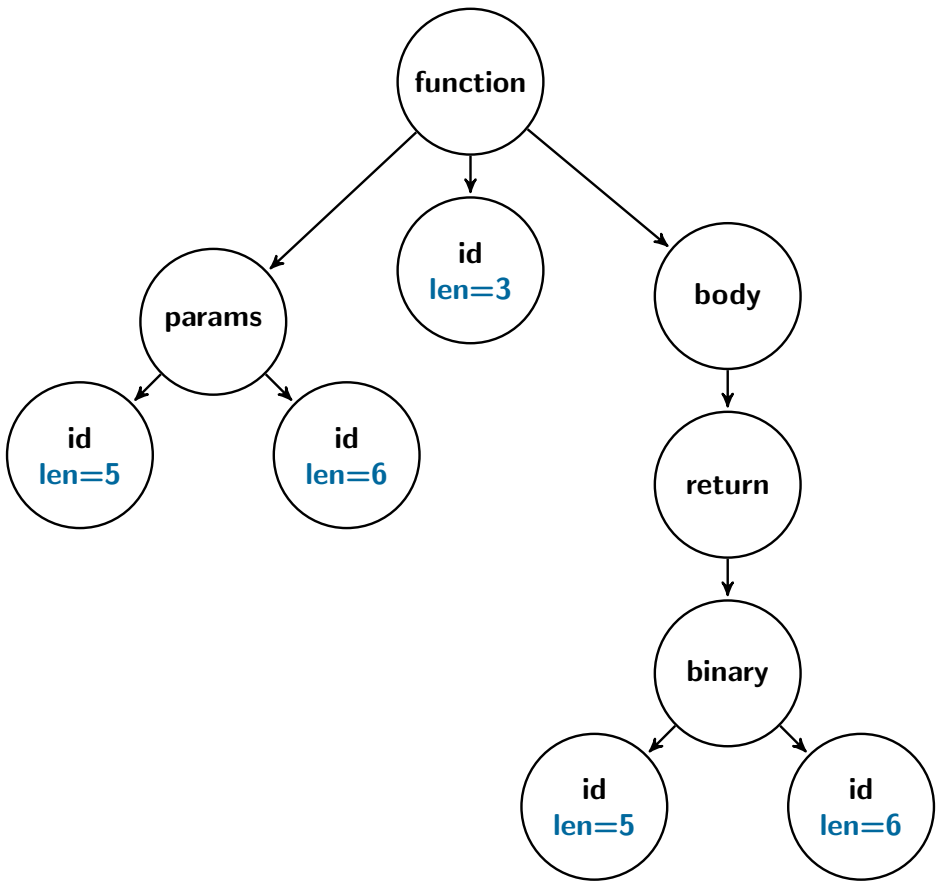
AST Representation for Convolutional Neural Network

```
function sum(first,second) {
  return first + second;
}
```



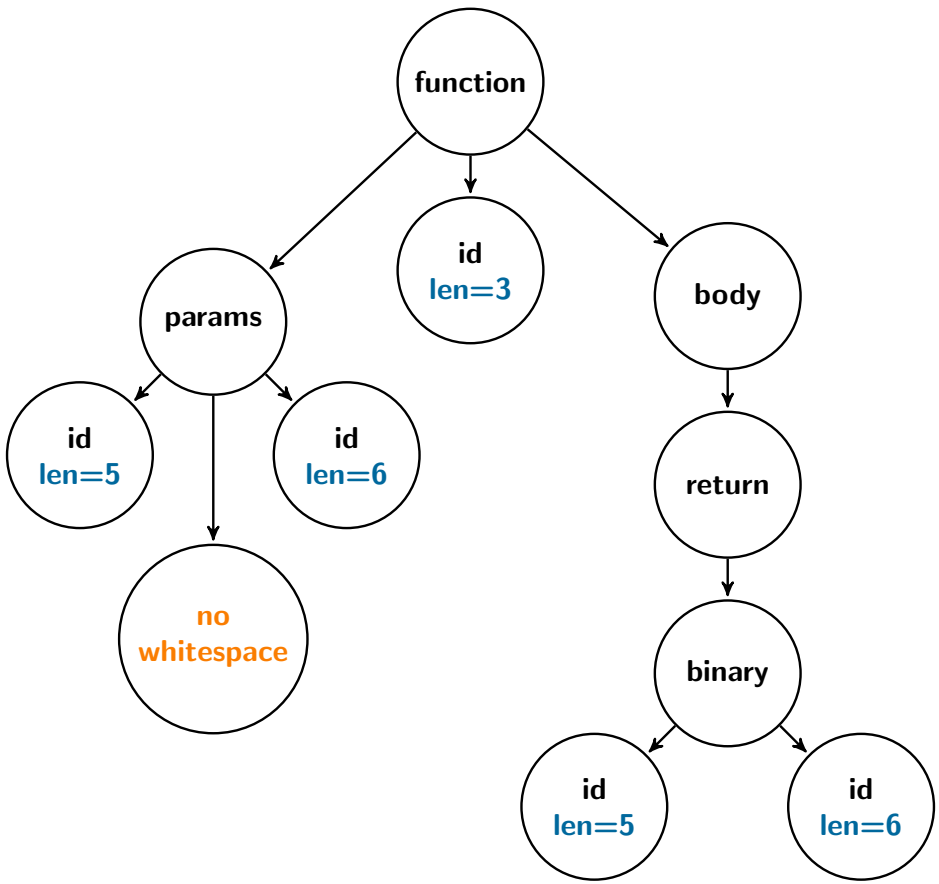
AST Representation for Convolutional Neural Network

```
function sum(first,second) {
  return first + second;
}
```



AST Representation for Convolutional Neural Network

```
function sum(first,second) {
  return first + second;
}
```



Accuracy of the Classifiers

validation set =

2,500 files from the corpus and their transformed versions

Classifier	Accuracy
TRANSFORMATION (no spaces info)	85.58%
TRANSFORMATION	95.06%
OBFUSCATION (no spaces info, no identifiers length)	75.43%
OBFUSCATION (no spaces info)	99.83%
OBFUSCATION	99.95%
TOOL-JSObfu	100%
TOOL-jsobfcom	100%
TOOL-jfogs	99.56%
TOOL-daft-logic	100%
TOOL-jsobf	100%

Accuracy of the Classifiers: User Study

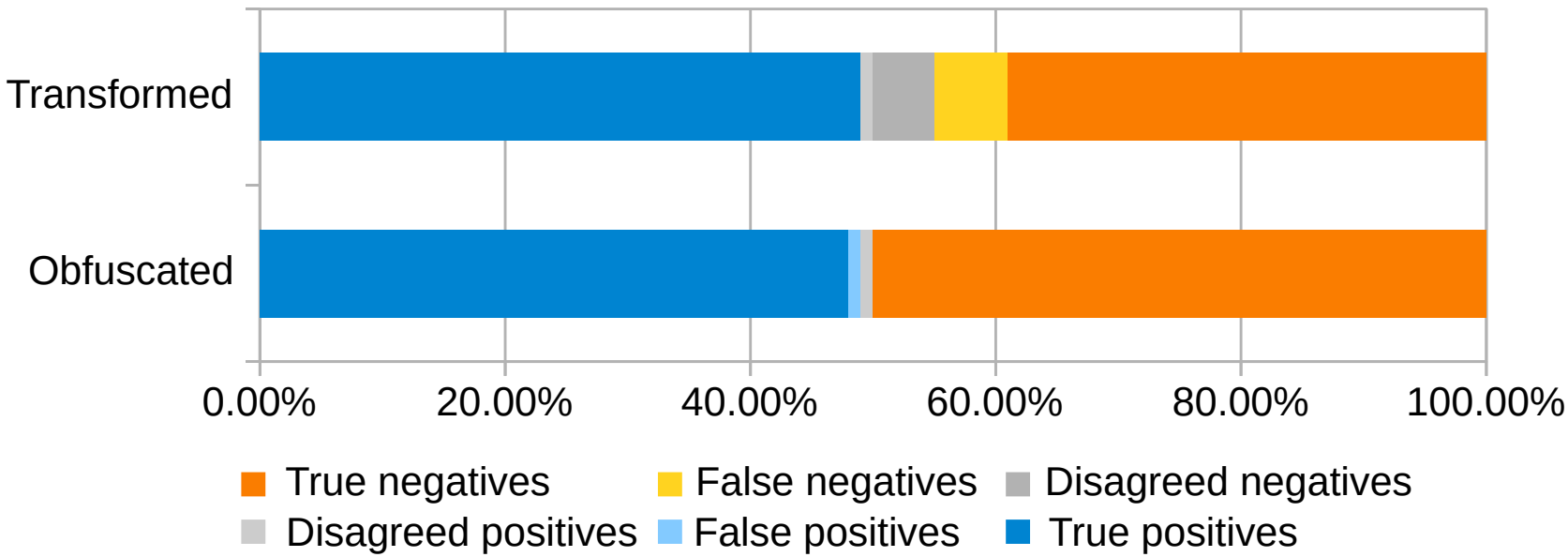
- **five users** and **200 scripts** from the web:
  - 50 positive and 50 negative classified by TRANSFORMATION
  - 50 positive and 50 negative classified by OBFUSCATION

Accuracy of the Classifiers: User Study

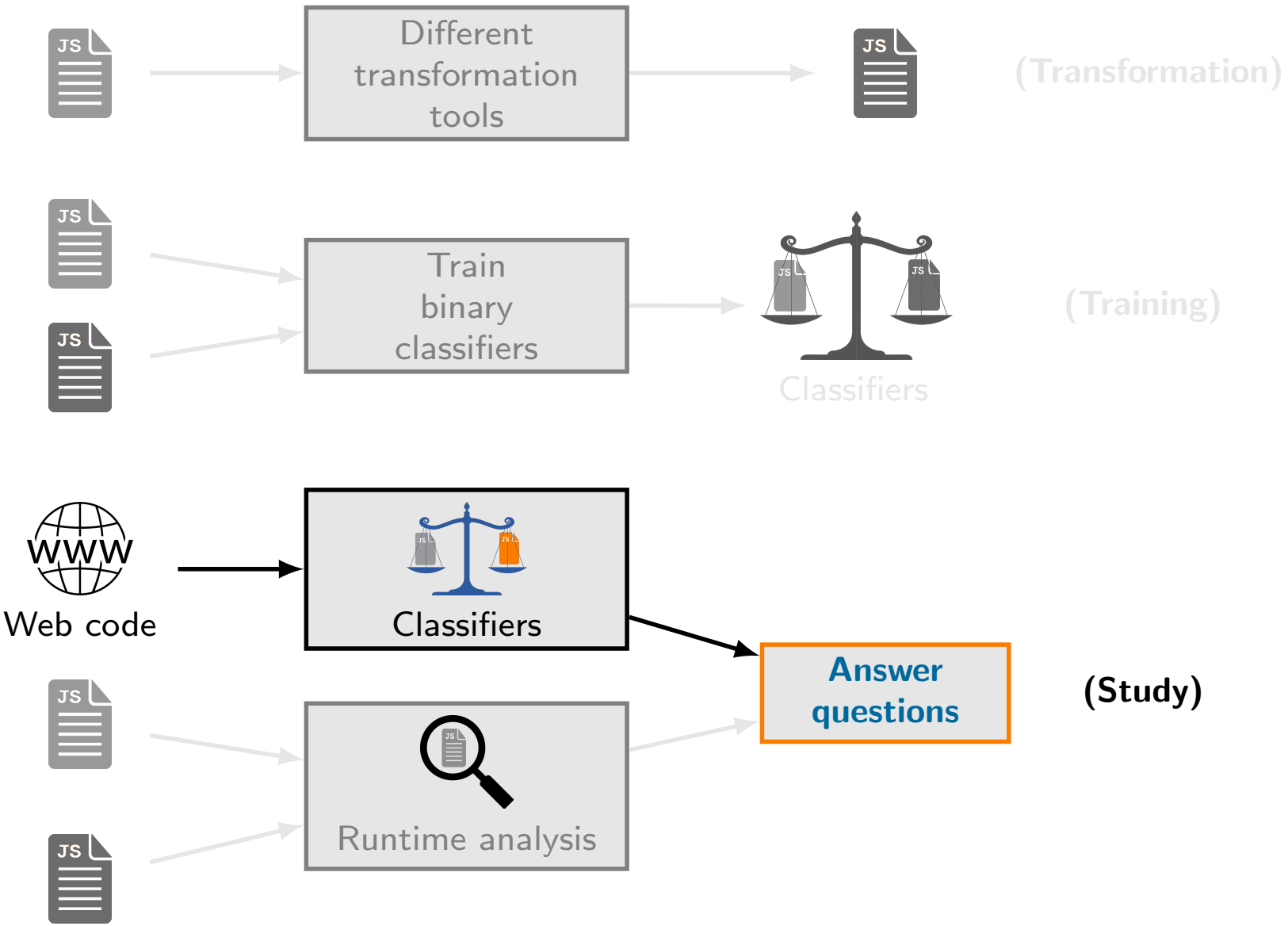
- **five users** and **200 scripts** from the web:
  - 50 positive and 50 negative classified by TRANSFORMATION
  - 50 positive and 50 negative classified by OBFUSCATION
- 0.81 inter-rater pairwise agreement

Accuracy of the Classifiers: User Study

- **five users** and **200 scripts** from the web:
  - 50 positive and 50 negative classified by TRANSFORMATION
  - 50 positive and 50 negative classified by OBFUSCATION
- 0.81 inter-rater pairwise agreement

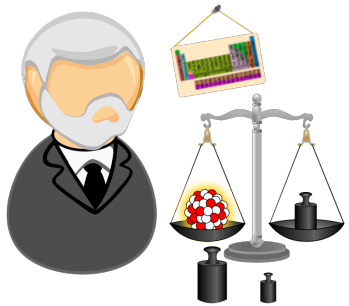


Methodology



Experimental Setup

- top **100,000** most popular websites



Experimental Setup

- top **100,000** most popular websites
- both inlined and included scripts

Experimental Setup

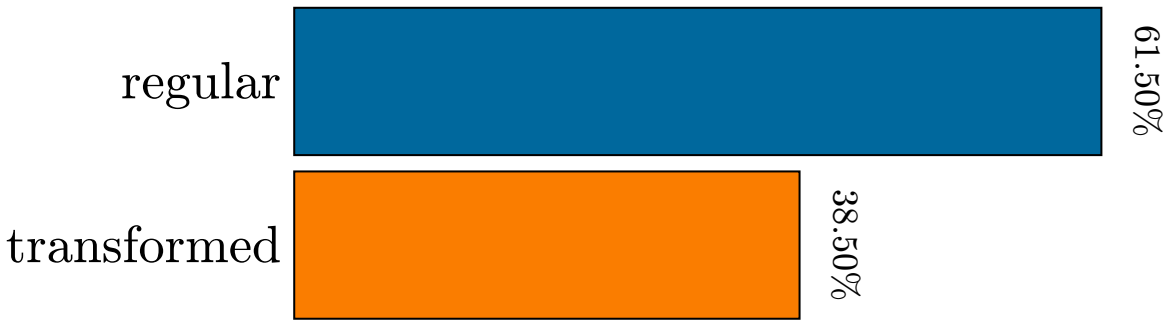
- top **100,000** most popular websites
- both inlined and included scripts
- more than **400,000 unique scripts**

Experimental Setup

- top **100,000** most popular websites
- both inlined and included scripts
- more than **400,000 unique scripts**
- each script mapped to a category, e.g., "news"

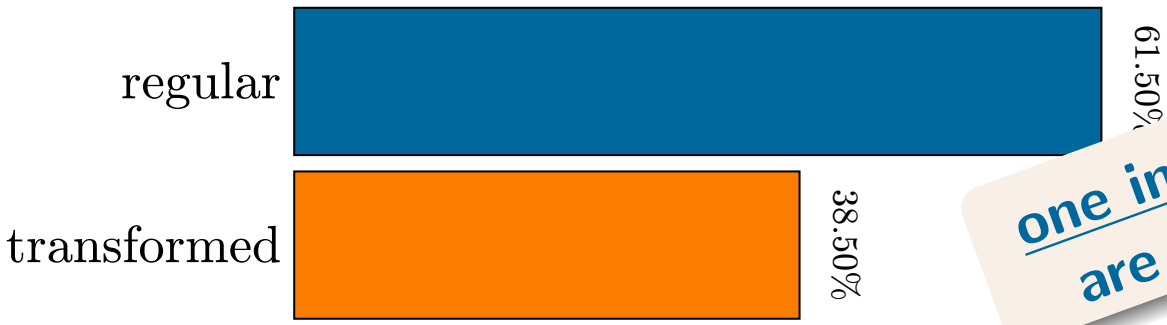
RQ1: How prevalent is transformed code on the web?

web scripts judged by **TRANSFORMATION** classifier



RQ1: How prevalent is transformed code on the web?

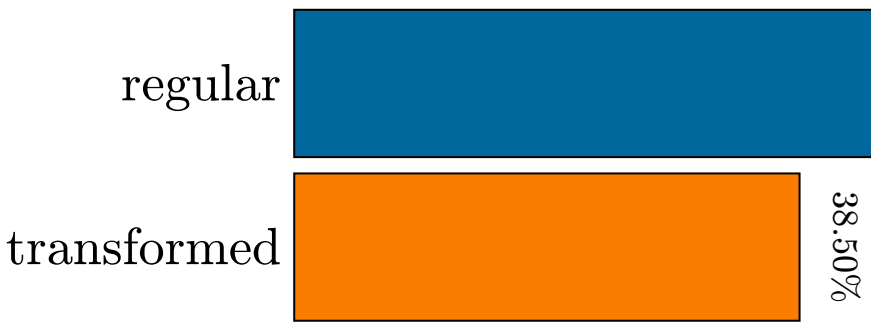
web scripts judged by **TRANSFORMATION** classifier



one in three scripts are transformed

RQ1: How prevalent is transformed code on the web?

web scripts judged by **TRANSFORMATION** classifier

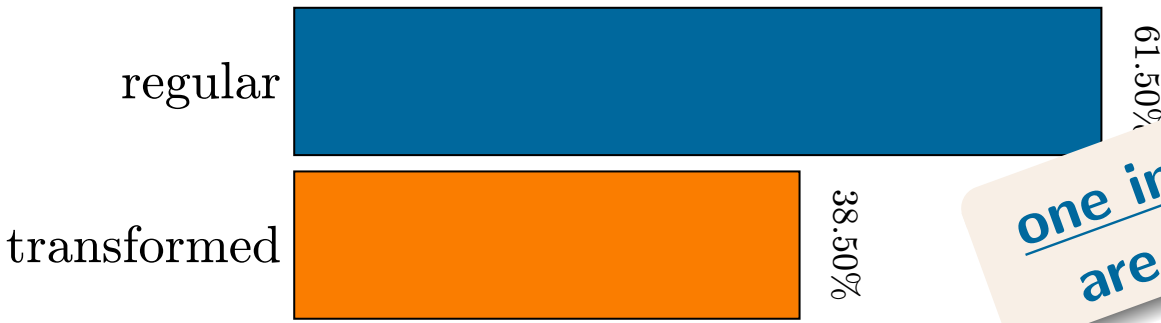


one in three scripts are transformed

web scripts judged by **OBFUSCATION** classifier

RQ1: How prevalent is transformed code on the web?

web scripts judged by **TRANSFORMATION** classifier

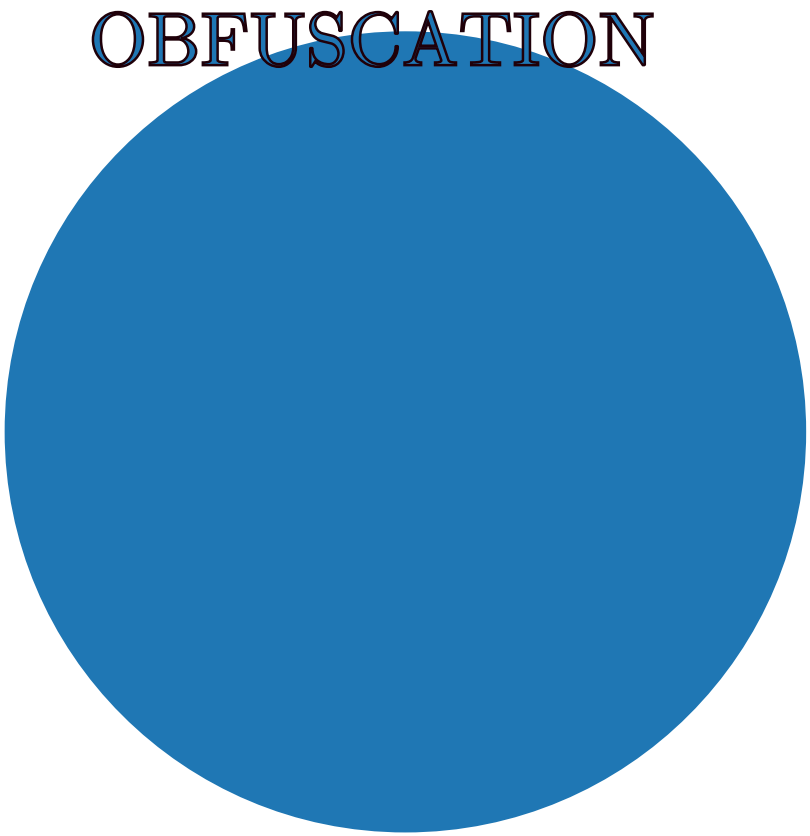


one in three scripts are transformed

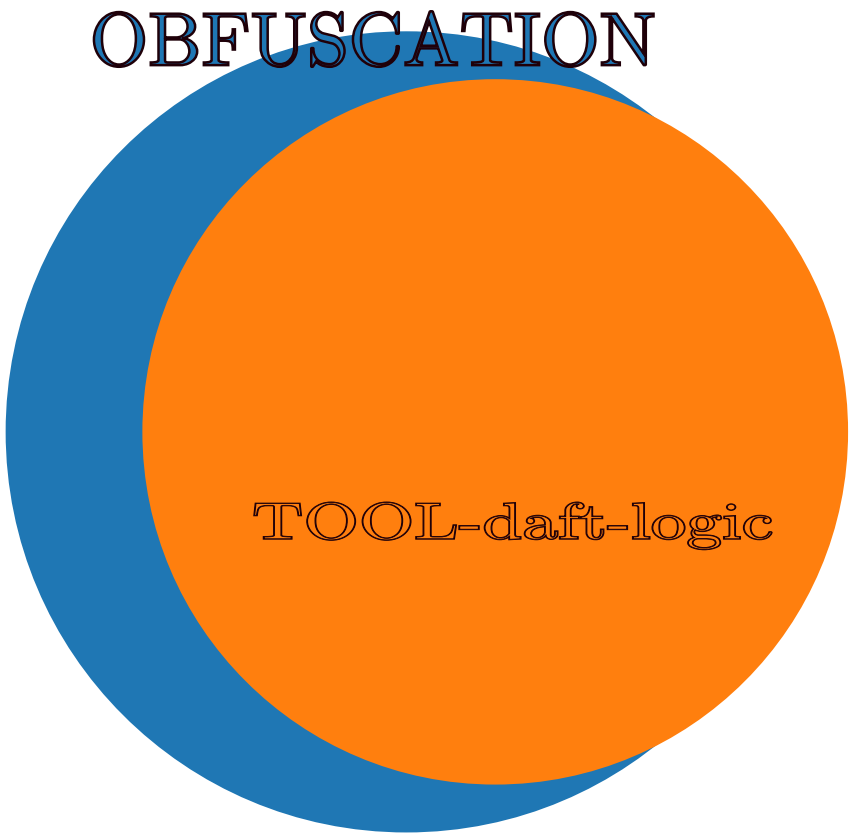
web scripts judged by **OBFUSCATION** classifier

>2,500 scripts are obfuscated

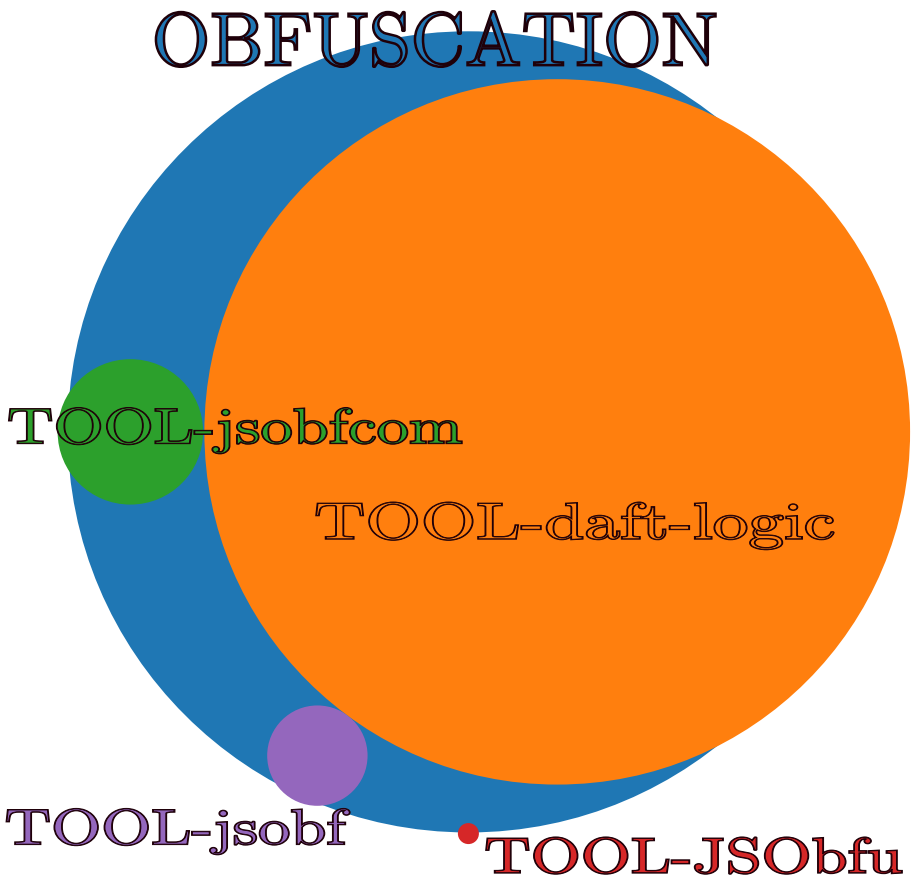
RQ2: Which tools are used for obfuscation on the web?



RQ2: Which tools are used for obfuscation on the web?

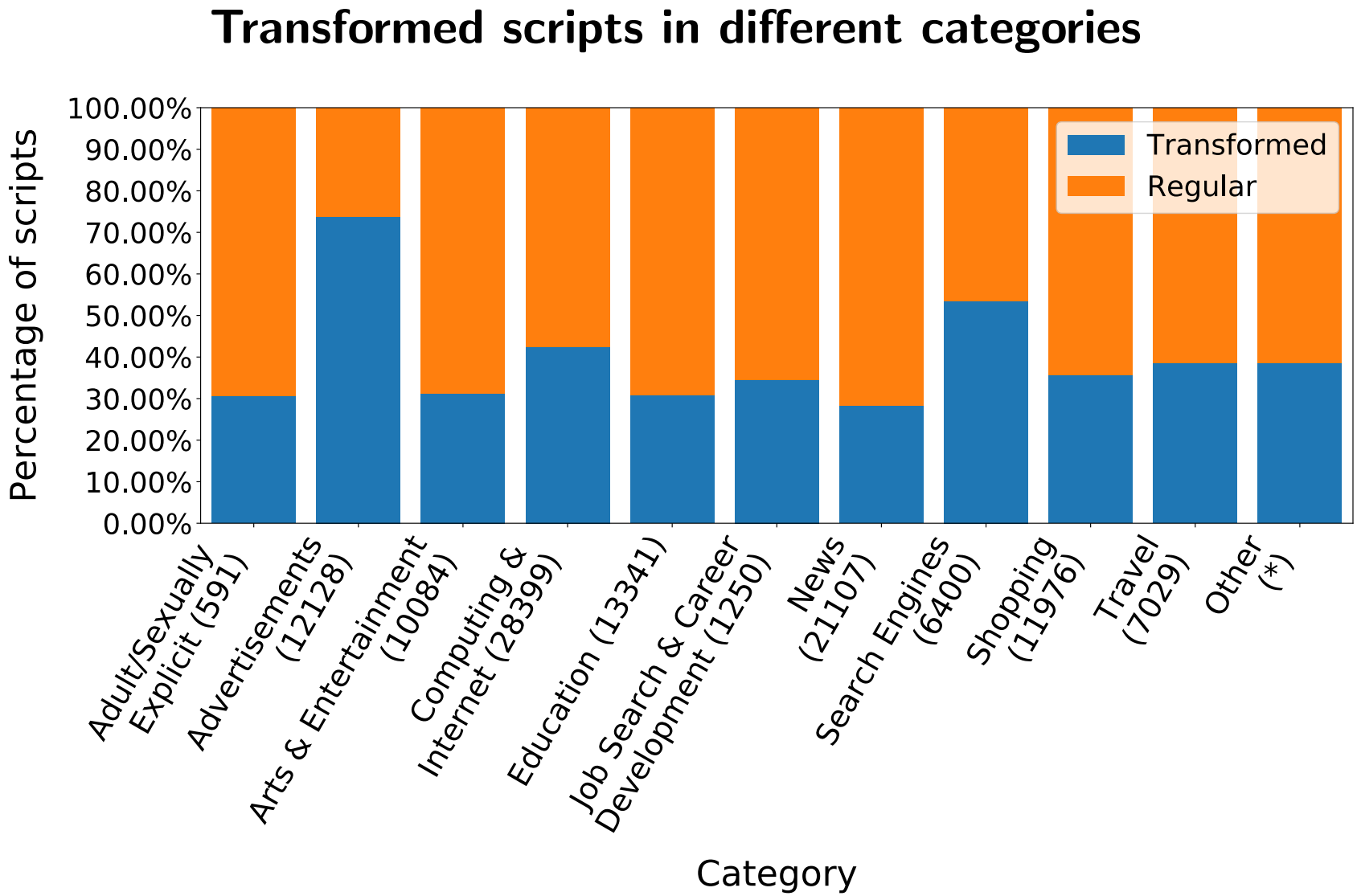


RQ2: Which tools are used for obfuscation on the web?

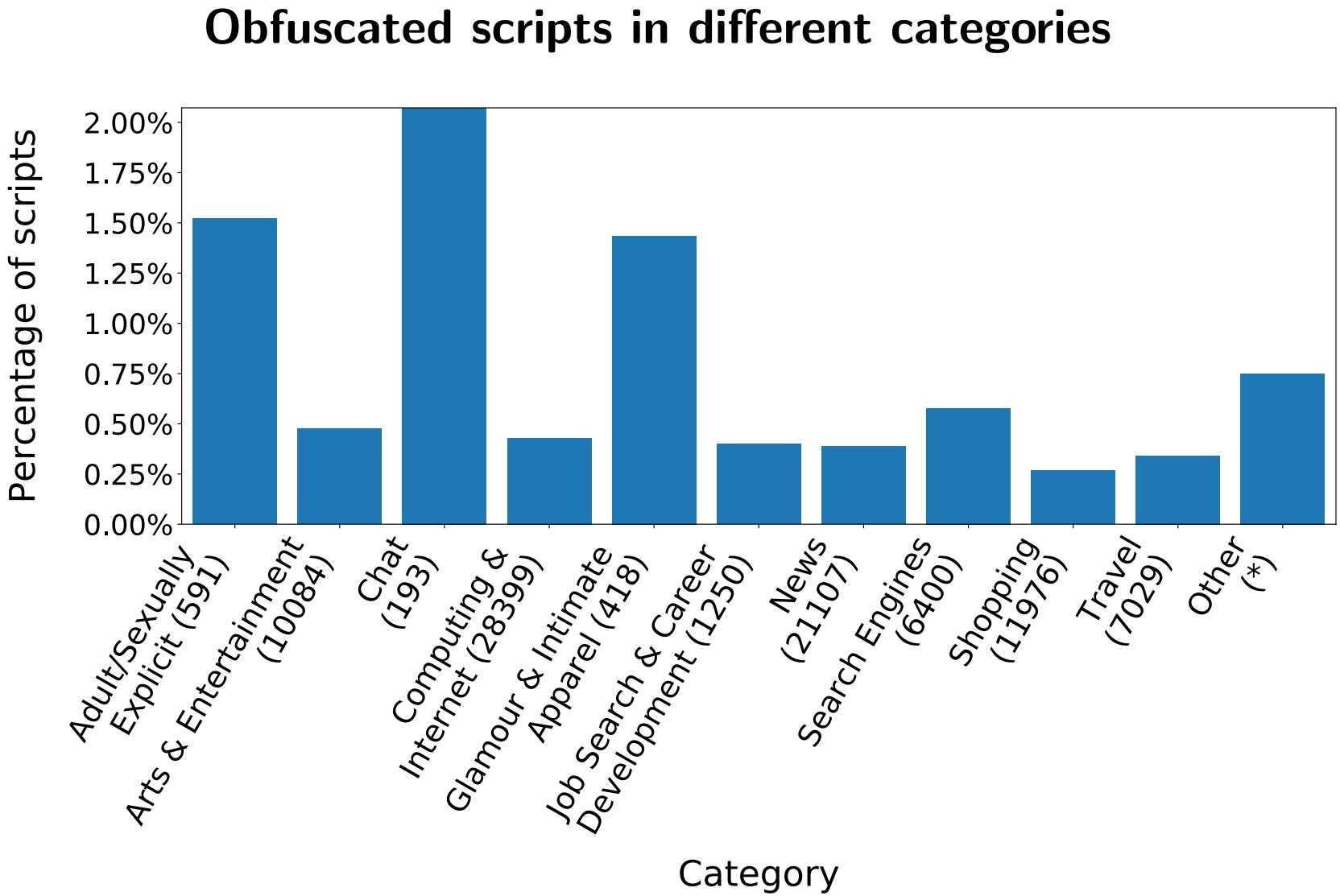


2,842 unique obfuscated scripts

RQ3: Does prevalence differ among website categories?



RQ3: Does prevalence differ among website categories?

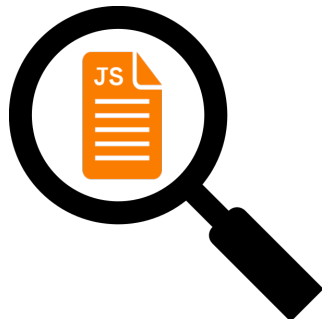


RQ4: What behavior is hidden using obfuscation?

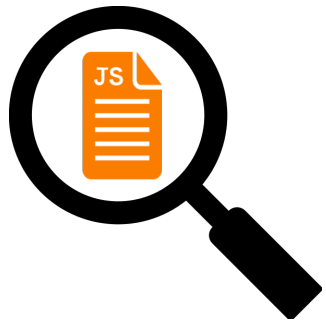
- perform lightweight **dynamic analysis** in Node.js

RQ4: What behavior is hidden using obfuscation?

- perform lightweight **dynamic analysis** in Node.js
- collect and analyze traces with accessed properties

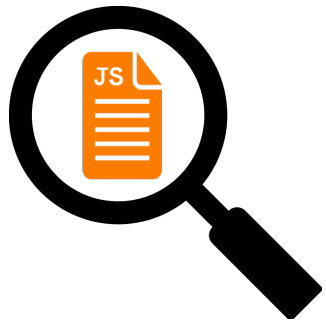


RQ4: What behavior is hidden using obfuscation?



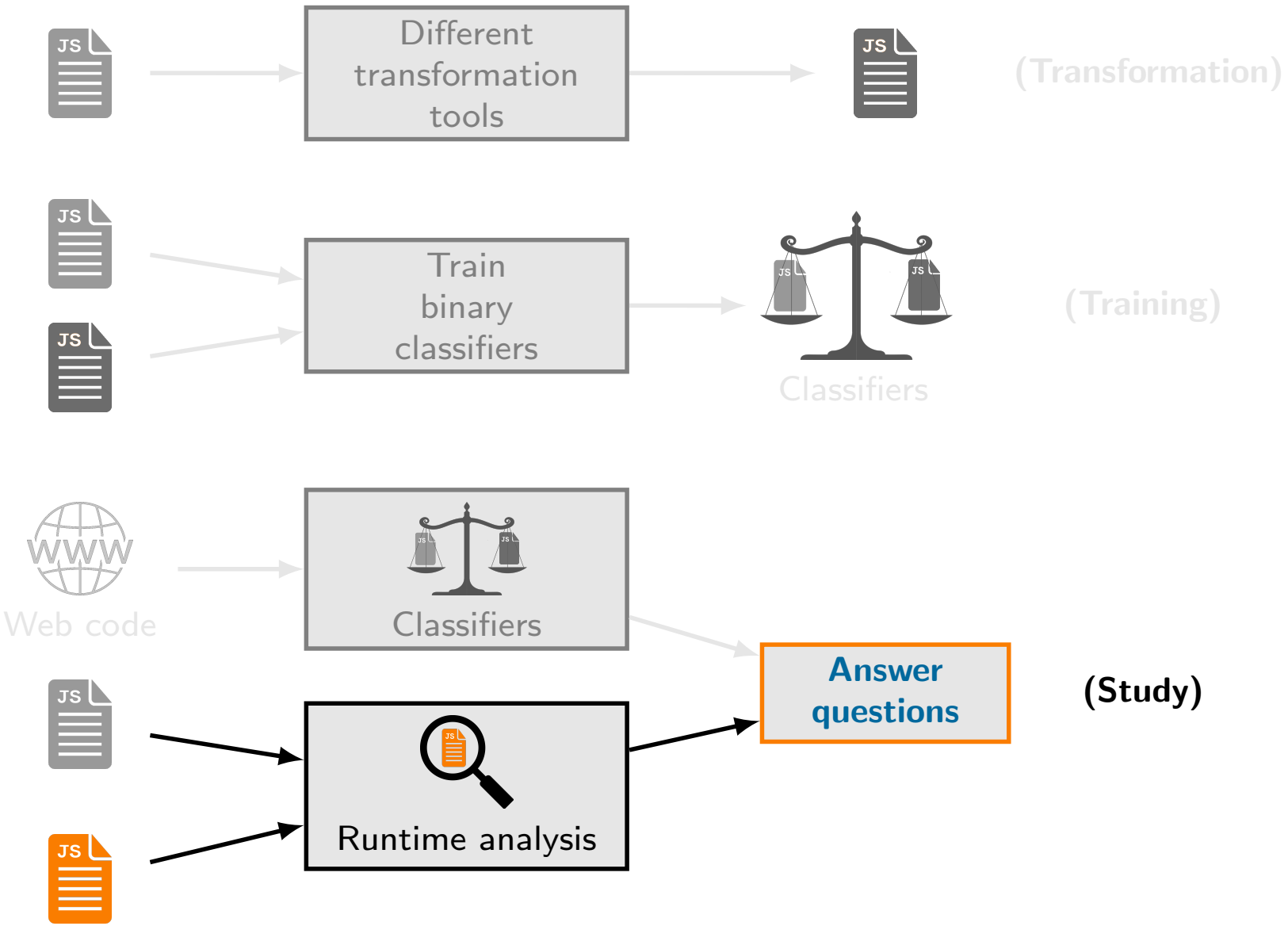
- perform lightweight **dynamic analysis** in Node.js
- collect and analyze traces with accessed properties
- multiple scripts access **privacy sensitive APIs**:
  - 11% read the `cookie`
  - 10% access the `userAgent`
  - 3% read the `referrer`
  - 10% inject additional JavaScript code

RQ4: What behavior is hidden using obfuscation?



- perform lightweight **dynamic analysis** in Node.js
- collect and analyze traces with accessed properties
- multiple scripts access **privacy sensitive APIs**:
  - 11% read the `cookie`
  - 10% access the `userAgent`
  - 3% read the `referrer`
  - 10% inject additional JavaScript code
- several scripts seem to perform browser fingerprinting

Methodology



Experimental Setup

- **10 libraries** with more than 400 tests each

Experimental Setup

- **10 libraries** with more than 400 tests each
- 46 transformed versions of the libraries

Experimental Setup

- **10 libraries** with more than 400 tests each
- 46 transformed versions of the libraries
- for each version, run the tests **20 times**

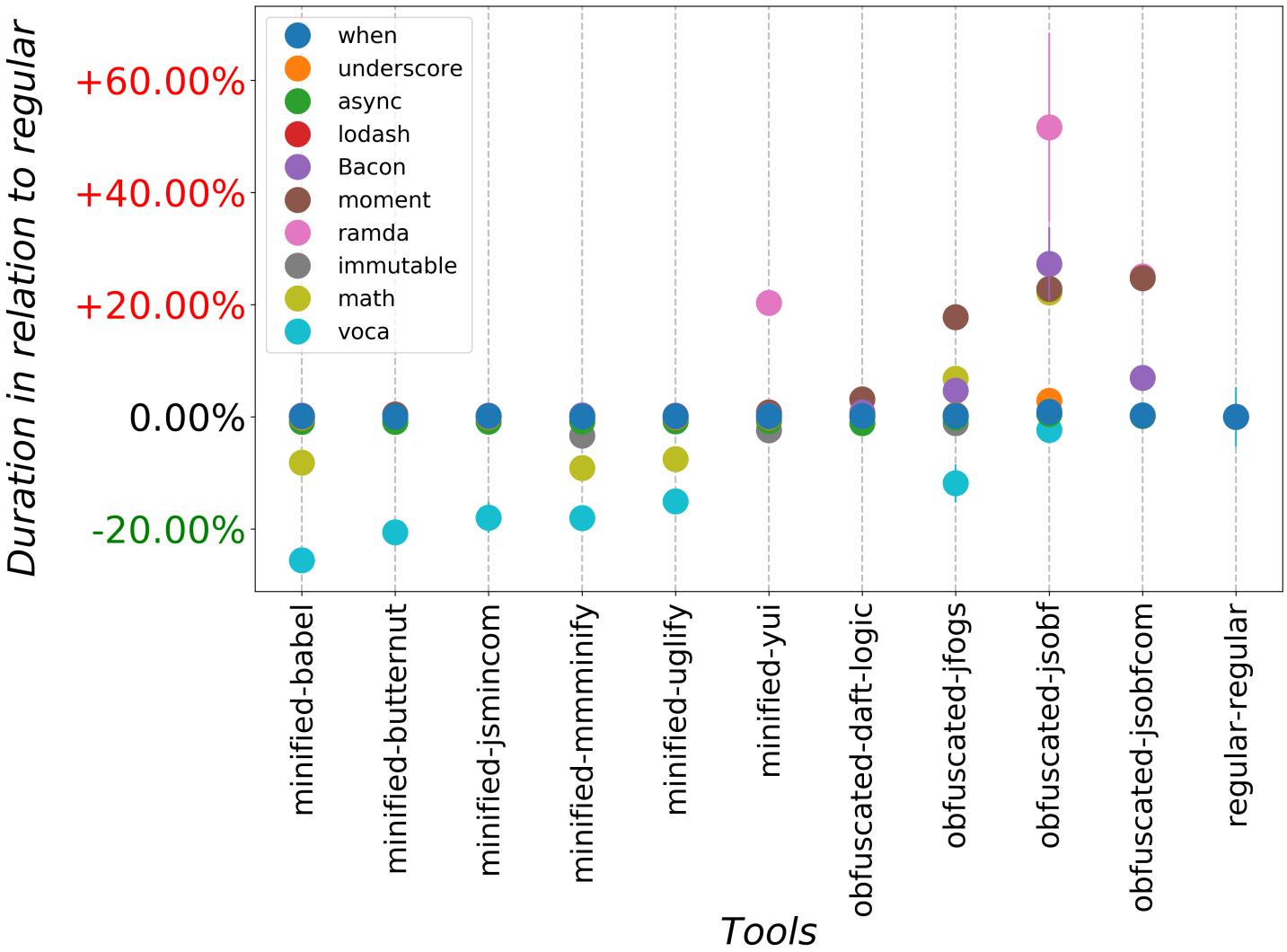
Experimental Setup

- **10 libraries** with more than 400 tests each
- 46 transformed versions of the libraries
- for each version, run the tests **20 times**
- run tests on a machine with 6 cores and 16GB RAM

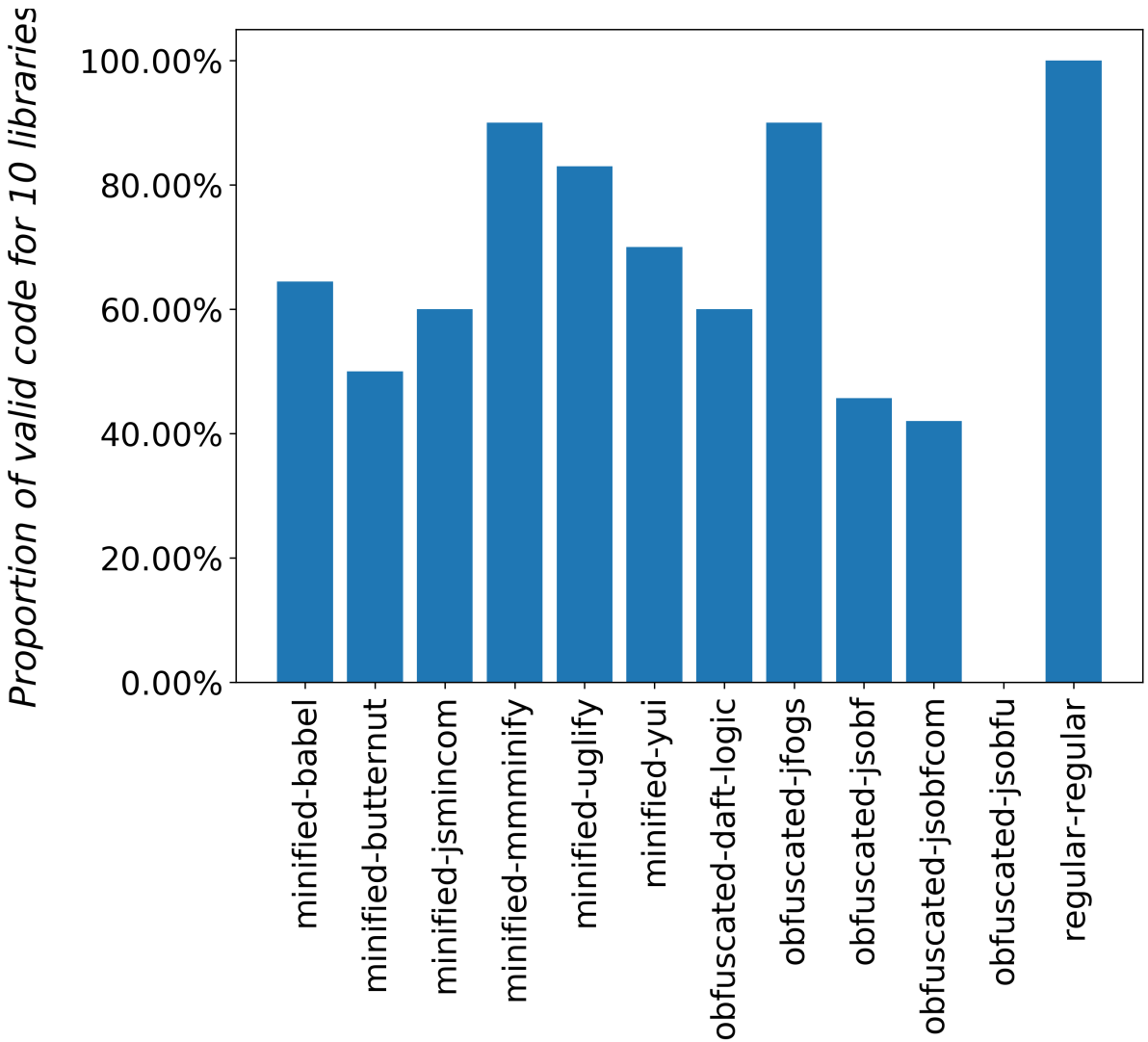
Experimental Setup

- **10 libraries** with more than 400 tests each
- 46 transformed versions of the libraries
- for each version, run the tests **20 times**
- run tests on a machine with 6 cores and 16GB RAM
- compare number of failing tests and performance of **transformed vs. original code**

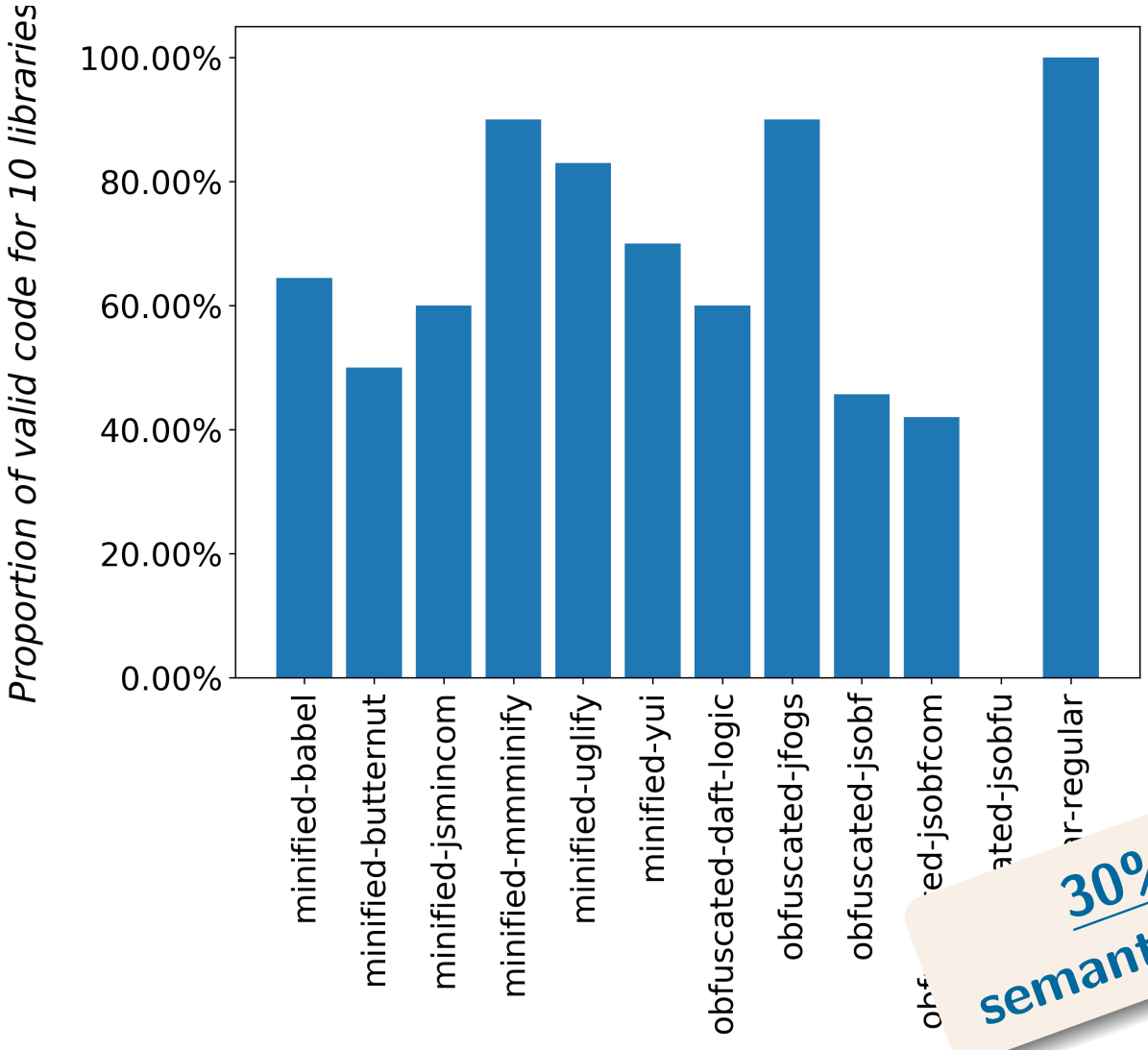
RQ5: How do transformations impact performance?



RQ6: How do transformations impact correctness?



RQ6: How do transformations impact correctness?



30% are not semantics-preserving

Conclusions

100,000

most popular websites

Conclusions

100,000

most popular websites

1 in 3

scripts are transformed

Conclusions

100,000
most popular websites

1 in 3
scripts are transformed

>2,500
obfuscated scripts

Conclusions

100,000
most popular websites

1 in 3
scripts are transformed

>2,500
obfuscated scripts

- transformations are prevalent on the web

Conclusions

100,000
most popular websites

1 in 3
scripts are transformed

>2,500
obfuscated scripts

- transformations are prevalent on the web
- obfuscation is seldom

Conclusions

100,000
most popular websites

1 in 3
scripts are transformed

>2,500
obfuscated scripts

- transformations are prevalent on the web
- obfuscation is seldom
- ML models effective at analyzing web code

Conclusions

100,000
most popular websites

1 in 3
scripts are transformed

>2,500
obfuscated scripts

- transformations are prevalent on the web
- obfuscation is seldom
- ML models effective at analyzing web code

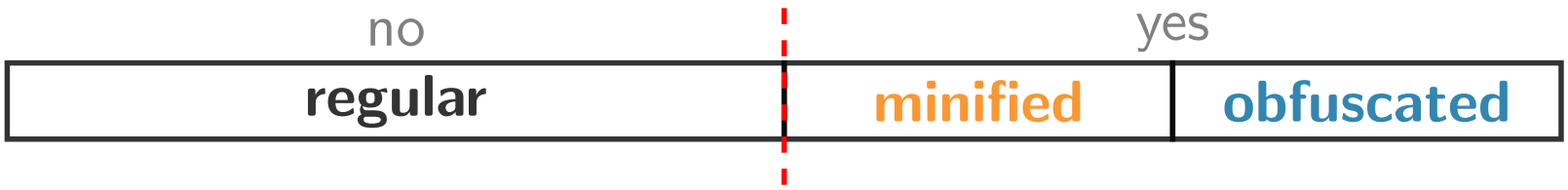
Obfuscation Techniques

Transformation techniques	Obfuscation tools				
	jsobf	jsobfcom	jfogs	JSObfu	daft-logic
String splitting	✓				✓
Keyword substitution					
String concatenation				✓	
Encoding the entire code					✓
Encrypting the entire code					
Identifier encoding	✓	✓	✓	✓	✓
String encoding	✓	✓		✓	
Dead code injection	✓				
Control flow flattening	✓				
String array	✓	✓	✓		✓
Code protecting techniques	✓				

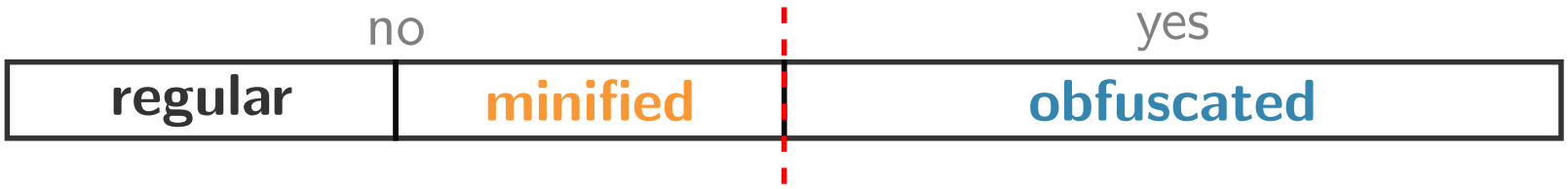
Classification Tasks

Seven binary classifiers:

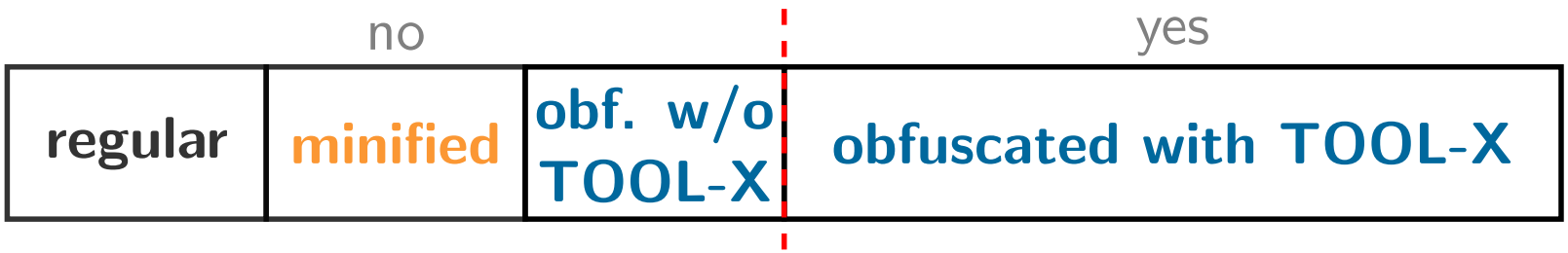
- **TRANSFORMATION** classifier



- **OBFUSCATION** classifier



- **TOOL-X** classifier $*$ 5



Examples of Obfuscated Code

```
$axure.loadDocument(
(function() {
    var _ = function() { var r={},a=arguments; for(var i=0; i<a.length; i+=2) r[a[i
]]=a[i+1]; return r; }
    var _creator = function() { return _(b,_(c,d,e,f,g,f,h,f,i,d,j,k,l,d,m,d,n,f,o,
f,p,f,q,[],r,d,s,t,u,d),v,_(w,[_(x,y,z,A,B,C,D,[_(x,E,z,A,B,F),_(x,G,z,A,B,H),_(x,I
,z,A,B,J),_(x,K,z,A,B,L),_(x,M,z,N,B,O,D,[_(x,P,z,A,B,Q),_(x,R,z,A,B,S),_(x,T,z,A,B
,U),_(x,V,z,A,B,W),_(x,X,z,N,B,O,D,[_(x,Y,z,A,B,Z)])]),_(x,ba,z,N,B,O,D,[_(x,bb,z,A
,B,bc,D,[_(x,bd,z,A,B,be)]),_(x,bf,z,A,B,bg)]),_(x,bh,z,N,B,O,D,[_(x,bi,z,A,B,bj,D,
[_(x,bk,z,A,B,bl)]),_(x,bm,z,A,B,bn),_(x,bo,z,A,B,bp),_(x,bq,z,A,B,br,D,[_(x,bs,z,A
,B,bt)]),_(x,bu,z,A,B,bv),_(x,bw,z,A,B,bx),_(x,by,z,A,B,bz),_(x,bA,z,A,B,bB),_(x,bC
,z,A,B,bD)]),_(x,bE,z,N,B,O,D,[_(x,bF,z,A,B,bG,D,[_(x,bH,z,N,B,O,D,[_(x,bI,z,A,B,bJ
)])]),_(x,bK,z,A,B,bL,D,[_(x,bM,z,A,B,bN)]),_(x,bO,z,A,B,bP),_(x,bQ,z,A,B,bR),_(x,bT
S,z,A,B,bT),_(x,bU,z,A,B,bV),_(x,bW,z,N,B,O,D,[_(x,bX,z,A,B,bY),_(x,bZ,z,A,B,ca),_(
x,cb,z,A,B,cc),_(x,cd,z,A,B,ce)]),_(x,cf,z,N,B,O,D,[_(x,cg,z,A,B,ch),_(x,ci,z,A,B,cj
j),_(x,ck,z,A,B,cl),_(x,cm,z,A,B,cn),_(x,co,z,A,B,cp)]),_(x,cq,z,N,B,O,D,[_(x,cr,z,
A,B,cs),_(x,ct,z,A,B,cu),_(x,cv,z,A,B,cw),_(x,cx,z,A,B,cy),_(x,cz,z,A,B,cA),_(x,cB
```

```
{document.write(String.fromCharCode(115,117,112,112,111,114,116,64,104,111,115,116,46,98,103))}
```

```
(function(_){_[(function(_){return(_=_[/fromCharCode/.source],_(0X62,7.5e+1,0X74,9.
9E+1,0122,0153)+_(0117,0112,1.12e+2,0106,120,116,0X36,5.6E+1,5.1e+1,0x36,0163,45,0X
57,8.3e+1))})(String)]=(function(_){return(_=_[/fromCharCode/.source],_(7.8E+1,109,
0137,0146,0X75,0163,71,0156,8.8e+1,0X47,104,119,0X45,9.7E+1,99,8.1e+1,6.6E+1,109)+_
(5.7e+1,0131,5.3e+1,0142,1.07e+2,52,49))})(String);})(window);
```

SVM Classifier

- consider most popular 30,000 tokens in our dataset
- identifiers embedding

```
function sum(first, second) {
  return first + second;
}
```

$$\{\text{sum} \rightarrow 1, \text{first} \rightarrow 2, \text{second} \rightarrow 2\}$$

foo	...	first	...	sum	...	second
0	0	2	0	1	0	2

- we use *tf-idf* values to compute the vector entries