Better C++ using Machine Learning on Large Projects

Nicolas Fleury & Mathieu Nayrolles

Ubisoft Montreal

Trying to Prevent Bugs

- Code reviews
- Unit tests and functional tests
- Clang-Tidy, Coverity, etc.

Trying to Prevent Bugs

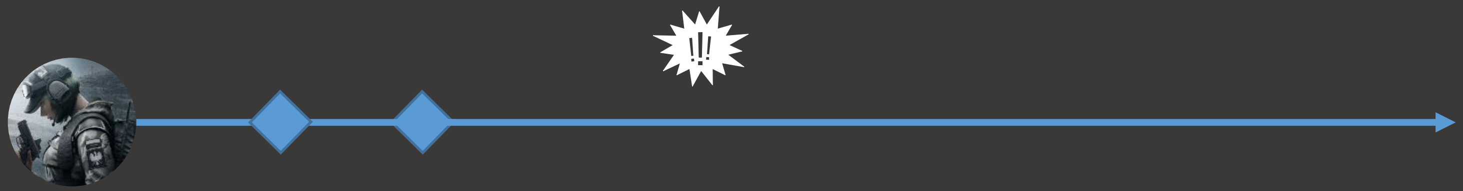
- Code reviews
- Unit tests and functional tests
- Clang-Tidy, Coverity, etc.
- Testers

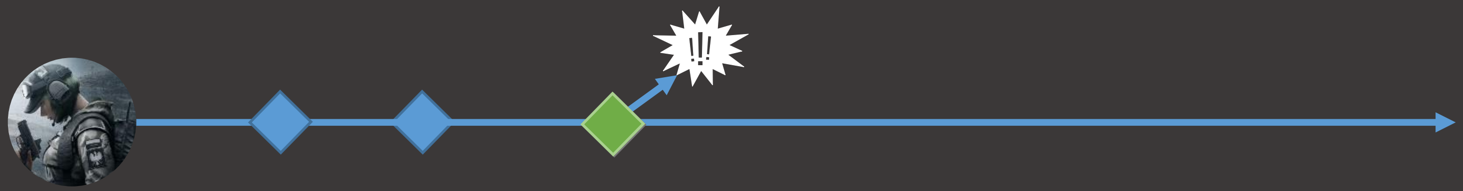
Trying to Prevent Bugs

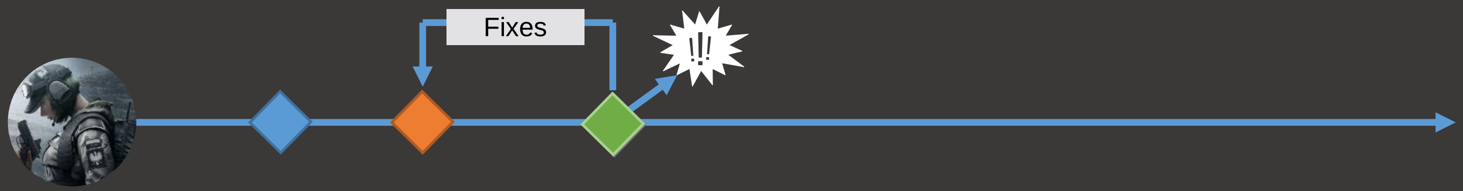
- Code reviews
- Unit tests and functional tests
- Clang-Tidy, Coverity, etc.
- Testers
- Beta versions, test servers

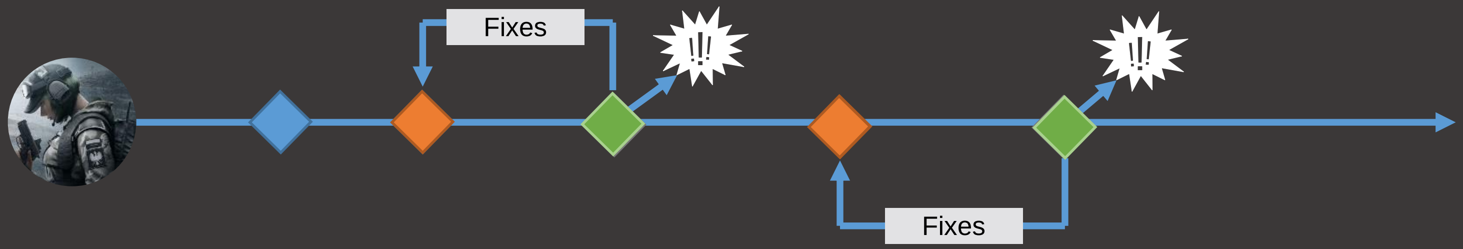
Outline

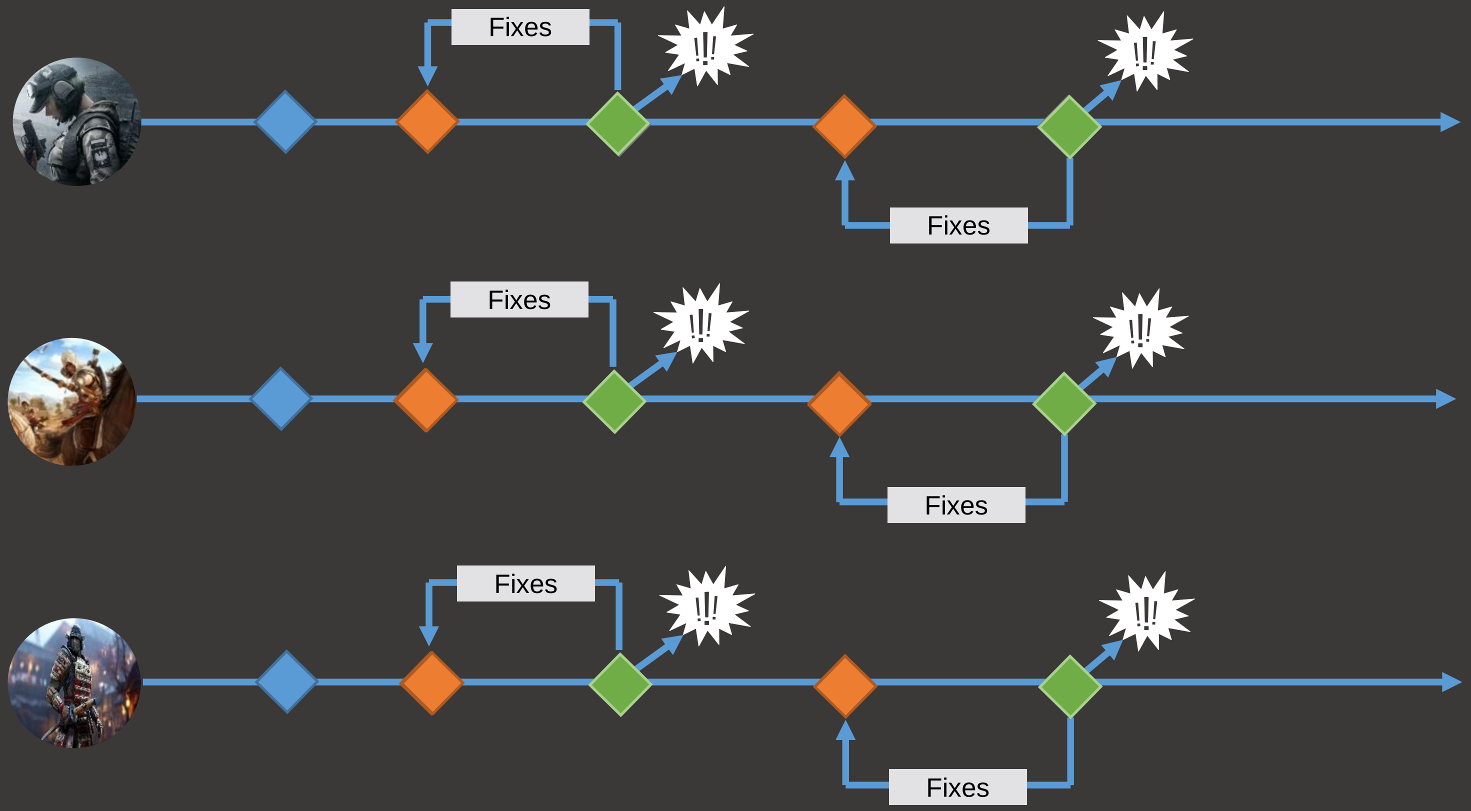
- Mathieu – How it works
  - Identify buggy commits from the past
  - Match some code contributions with past bugs
  - Use ML to predict the riskiness of a commit
- Nicolas – Application on Rainbow Six Siege
  - Bug introduction rate
  - Tools and code suggestions

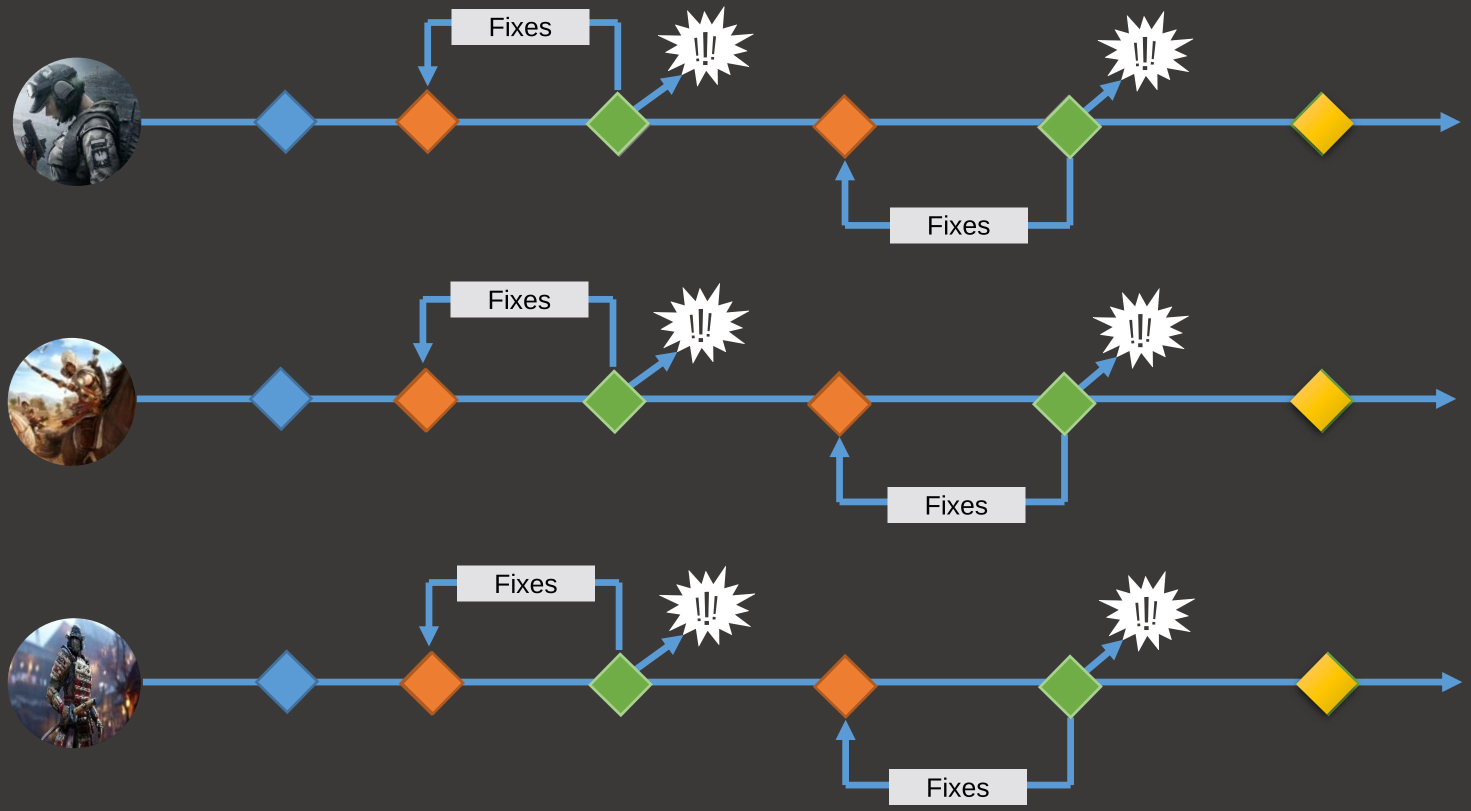


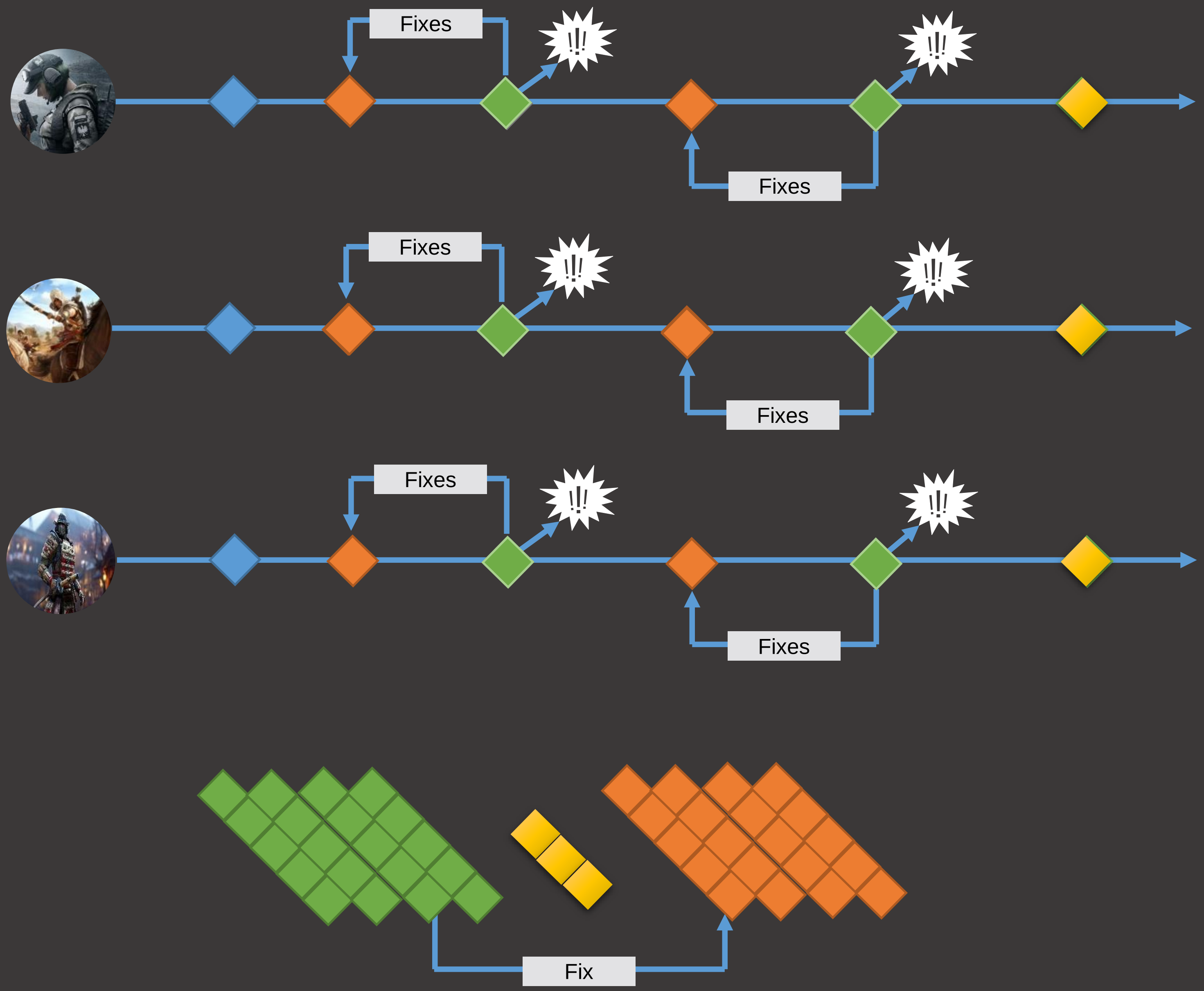


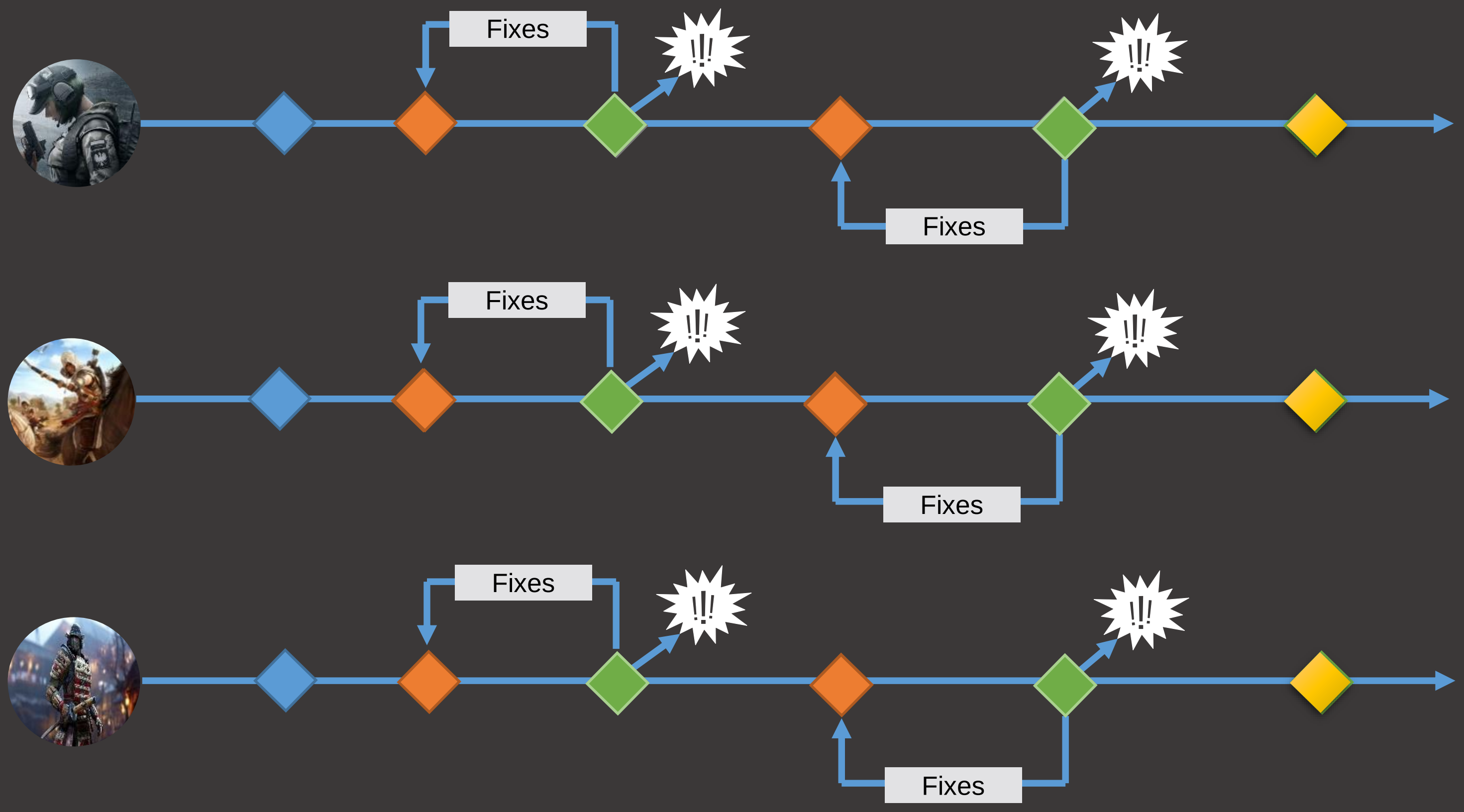

Fixes
!!!


Fixes
!!!
!!!
Fixes


Fixes
!!!
!!!
Fixes
Fixes
!!!
!!!
Fixes
Fixes
!!!
!!!
Fixes


Fixes
!!!
Fixes
!!!
Fixes
!!!
Fixes
!!!
Fixes
!!!
Fixes
!!!


Fixes
!!!
!!!
Fixes
Fixes
!!!
!!!
Fixes
Fixes
!!!
!!!
Fixes
Fix



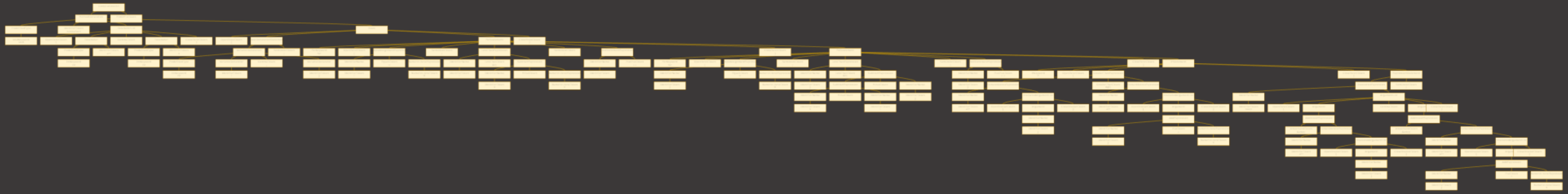
23% of Bug Introducing Commits OSS

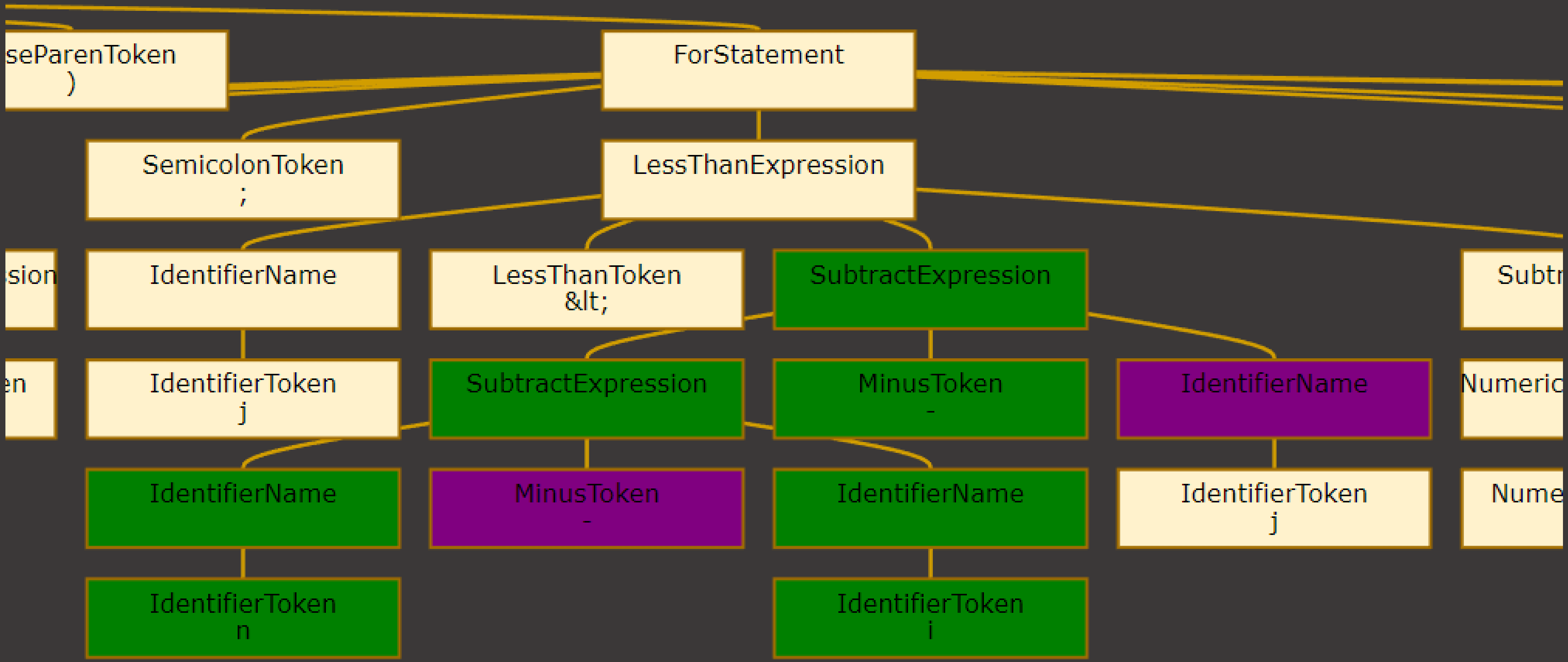
```
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)
        for (j = 0; j < 1-j; j++)
            if (arr[j] > arr[j+1])
                swap(&arr[j], &arr[j+1]);
}
```

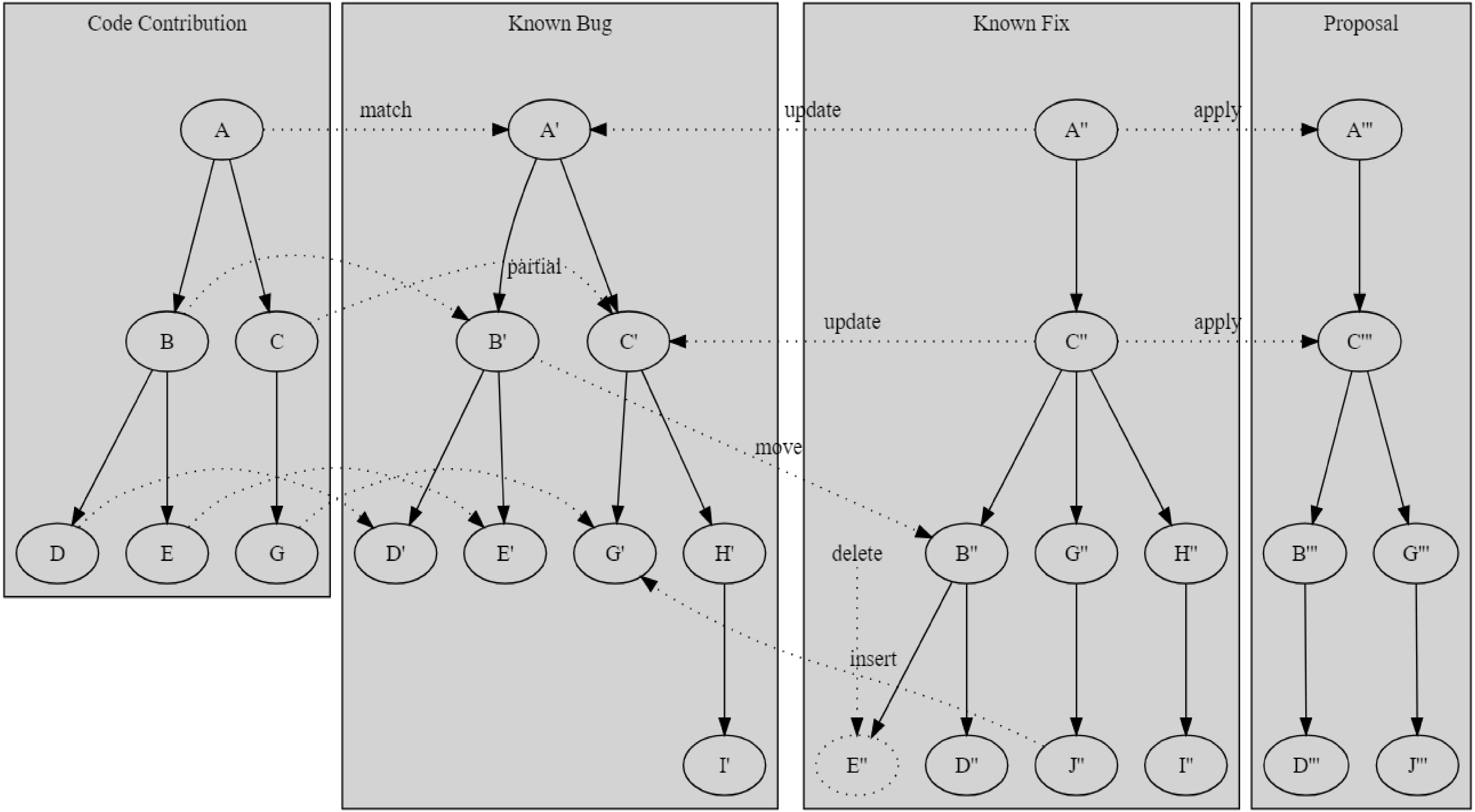
```
while (swapped) {
   swapped = false;
   j++;
   for (int i = 0; i < n - j; i++) {
      if (arr[i] > arr[i + 1]) {
         swap(&arr[i], &arr[i+1]);
         swapped = true;
      }
   }
}
```

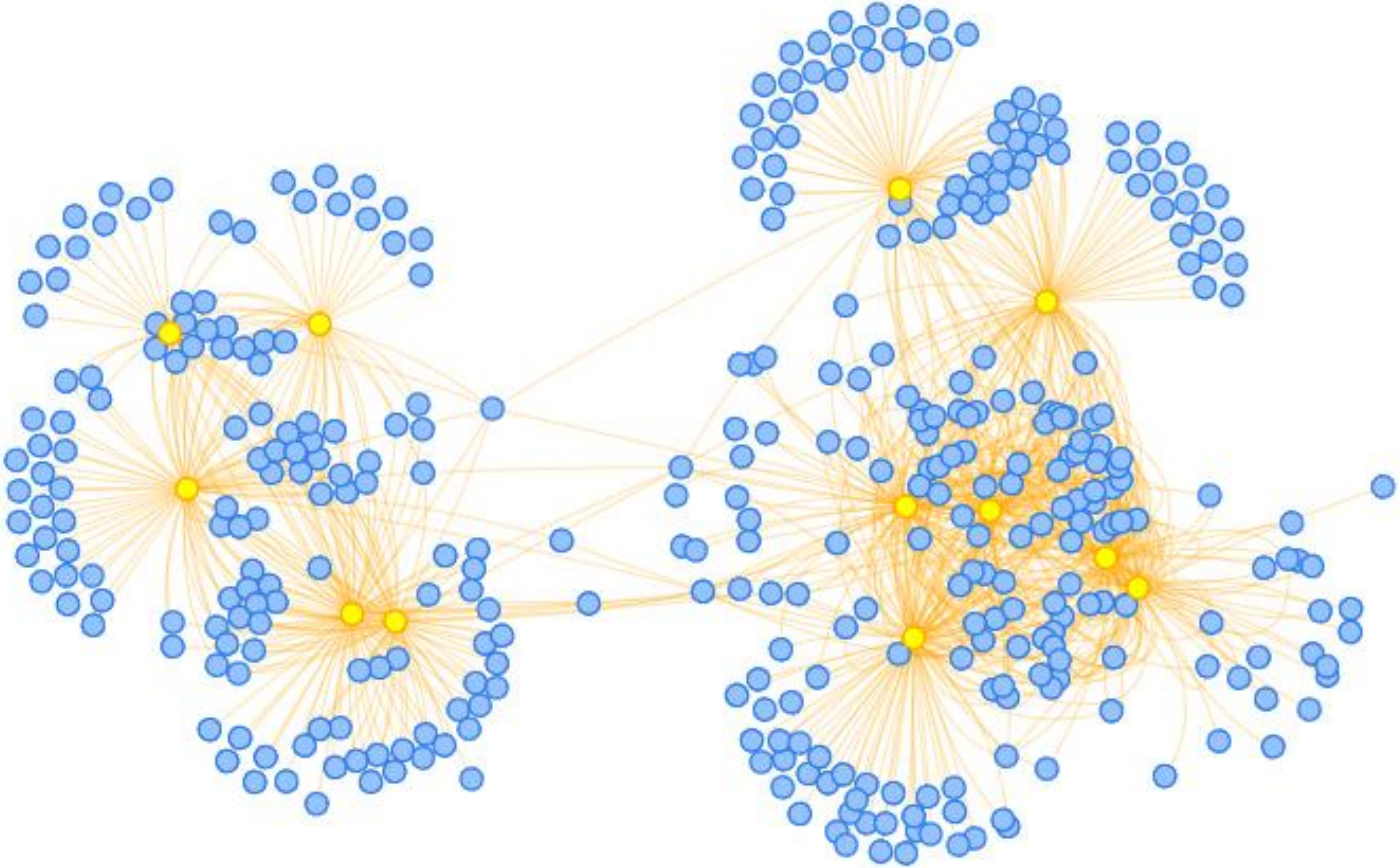
```
while (V1) {
   V1 = false;
   V2++;
   for (int V3 = 0; V3 < V4-V2; V3++) {
      if (V5[V3] > V5[V3 + 1]) {
         M1(&V5[V3], &V5[V3+1]);
         V1 = true;
      }
   }
}
```

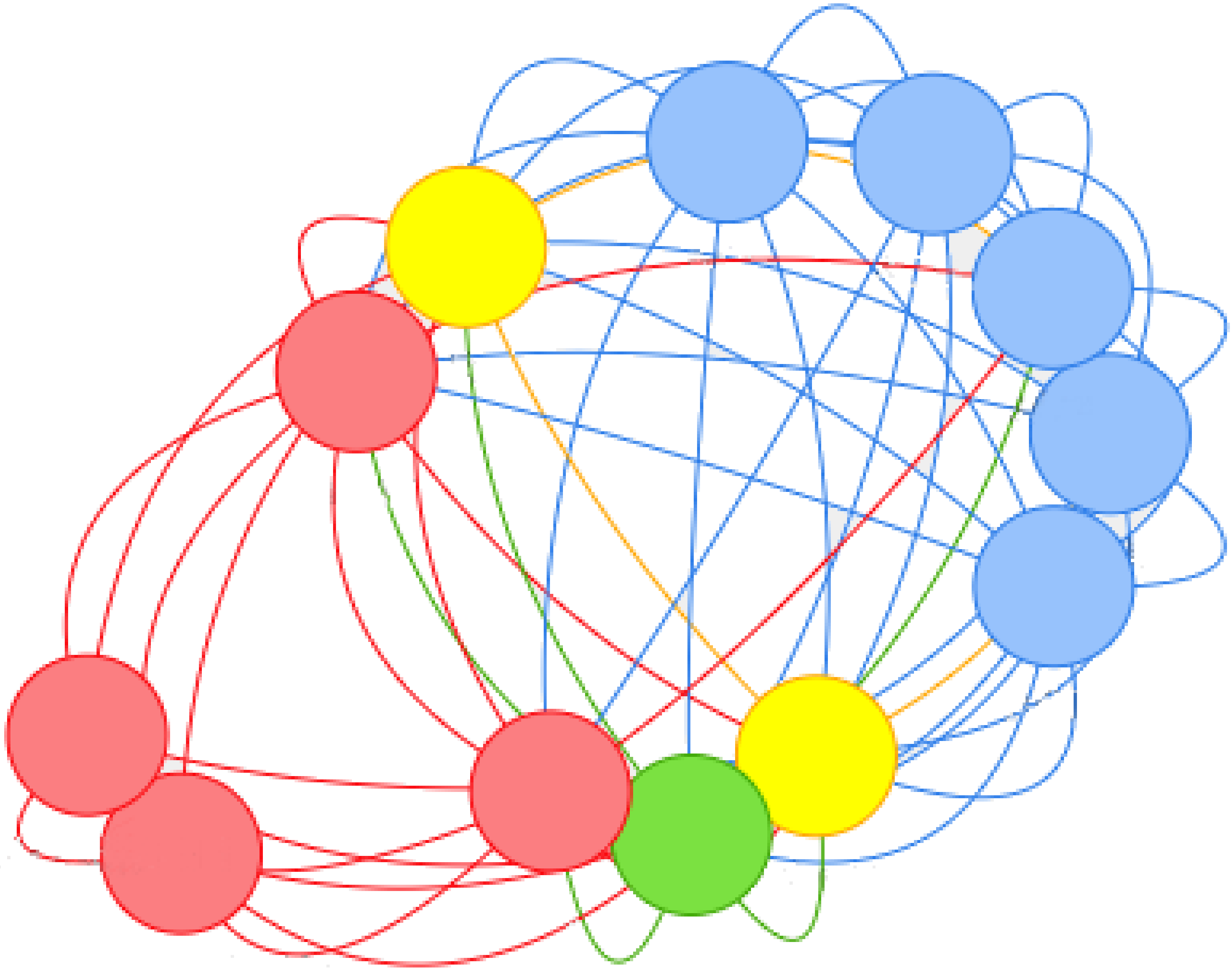
```
void M1(int V1[], int V2)
{
   int V3, V4;
   for (V3 = 0; V3 < V2-1; V3++)
      for (V4 = 0; V4 < V2-V3-1; V4++)
         if (V1[V4] > V1[V4+1])
            M2(&V1[V4], &V1[V4+1]);
}
```




seParenToken
)
ForStatement
SemicolonToken
;
LessThanExpression
sion
IdentifierName
LessThanToken
&lt;
SubtractExpression
Subtr
en
IdentifierToken
j
SubtractExpression
MinusToken
-
IdentifierName
Numeric
IdentifierName
MinusToken
-
IdentifierName
IdentifierToken
j
Nume
IdentifierToken
n
IdentifierToken
i


Code Contribution
Known Bug
Known Fix
Proposal
A
B
C
D
E
G
A'
B'
C'
D'
E'
G'
H'
I'
A''
C''
B''
G''
H''
E''
D''
J''
I''
A'''
C'''
B'''
G'''
D'''
J'''
match
partial
update
update
move
delete
insert
apply
apply





Machine Learning

Historical
Data IN

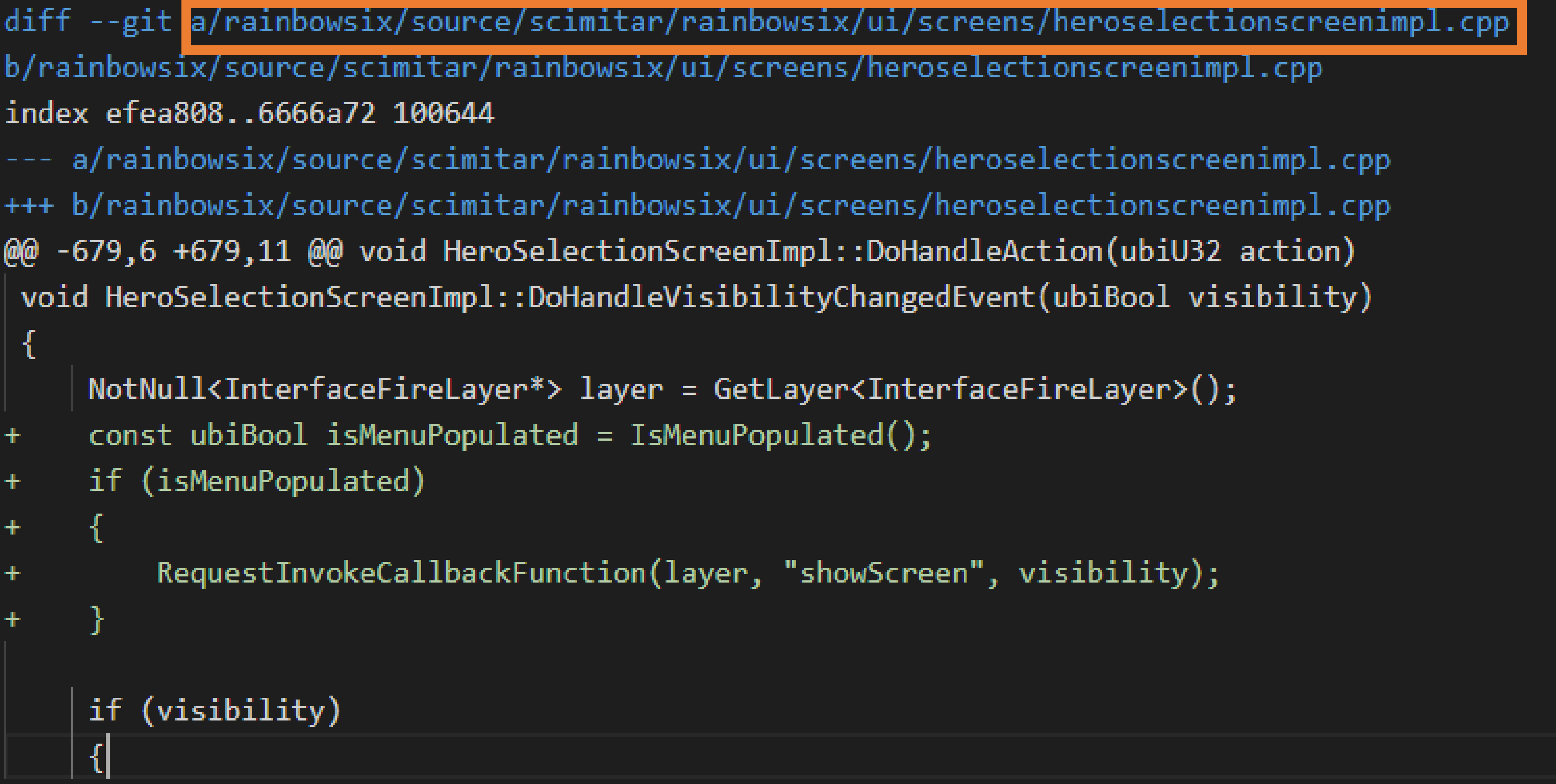
0.98	**0.10**	**0.13**	**0.14**
0.12	…	0.92	0.12
0.91	0.21	…	0.67
0.18	0.76	0.17	…

Data OUT
Model

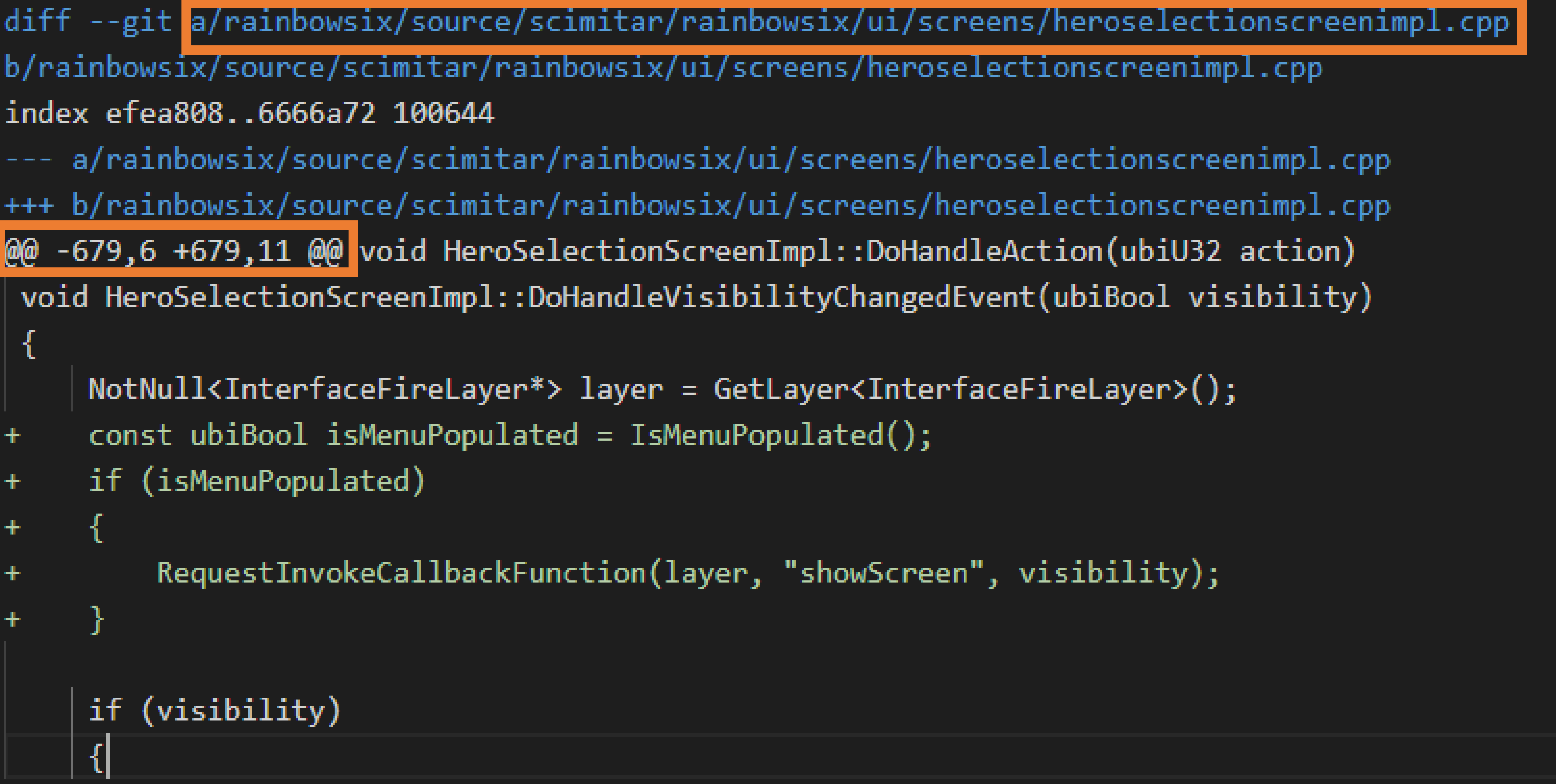
New Data

0.14	**0.153**	**0.84**	**0.14**
0.12	…	0.92	0.12
0.45	0.41	…	0.67
0.69	0.76	0.17	…

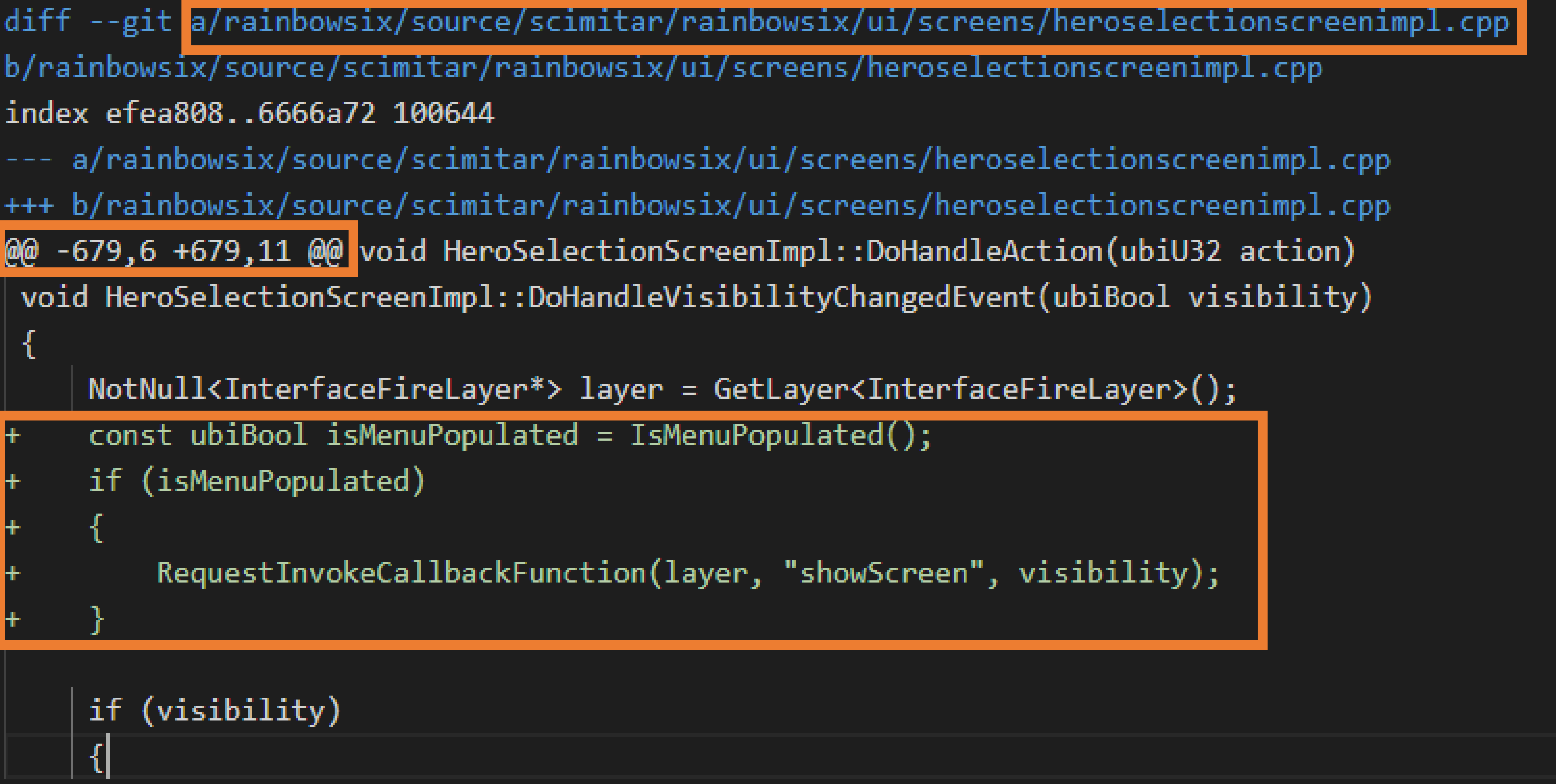
Prediction



```
diff --git a/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
b/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
index efea808..6666a72 100644
--- a/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
+++ b/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
@@ -679,6 +679,11 @@ void HeroSelectionScreenImpl::DoHandleAction(ubiU32 action)
 void HeroSelectionScreenImpl::DoHandleVisibilityChangedEvent(ubiBool visibility)
 {
     NotNull<InterfaceFireLayer*> layer = GetLayer<InterfaceFireLayer>();
+    const ubiBool isMenuPopulated = IsMenuPopulated();
+    if (isMenuPopulated)
+    {
+        RequestInvokeCallbackFunction(layer, "showScreen", visibility);
+    }
 
     if (visibility)
     {
```



```
diff --git a/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
b/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
index efea808..6666a72 100644
--- a/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
+++ b/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
@@ -679,6 +679,11 @@ void HeroSelectionScreenImpl::DoHandleAction(ubiU32 action)
 void HeroSelectionScreenImpl::DoHandleVisibilityChangedEvent(ubiBool visibility)
 {
     NotNull<InterfaceFireLayer*> layer = GetLayer<InterfaceFireLayer>();
+    const ubiBool isMenuPopulated = IsMenuPopulated();
+    if (isMenuPopulated)
+    {
+        RequestInvokeCallbackFunction(layer, "showScreen", visibility);
+    }
 
     if (visibility)
     {
```



```
diff --git a/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
b/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
index efea808..6666a72 100644
--- a/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
+++ b/rainbowsix/source/scimitar/rainbowsix/ui/screens/heroselectionscreenimpl.cpp
@@ -679,6 +679,11 @@ void HeroSelectionScreenImpl::DoHandleAction(ubiU32 action)
 void HeroSelectionScreenImpl::DoHandleVisibilityChangedEvent(ubiBool visibility)
 {
     NotNull<InterfaceFireLayer*> layer = GetLayer<InterfaceFireLayer>();
+    const ubiBool isMenuPopulated = IsMenuPopulated();
+    if (isMenuPopulated)
+    {
+        RequestInvokeCallbackFunction(layer, "showScreen", visibility);
+    }

     if (visibility)
     {
```

Machine Learning Features

File Age

Files

Directories

Machine Learning Features

File Age

Files

Directories

Dev R-XP

Dev S-XP

Contributors

Machine Learning Features

Subsystems

Fan Out

Test Coverage

Entropy

File Age

Files

Fan In

Subsystem Stability

Directories

Dev S-XP

Dev R-XP

Unique Change

File Stability

Log Density

Complexity

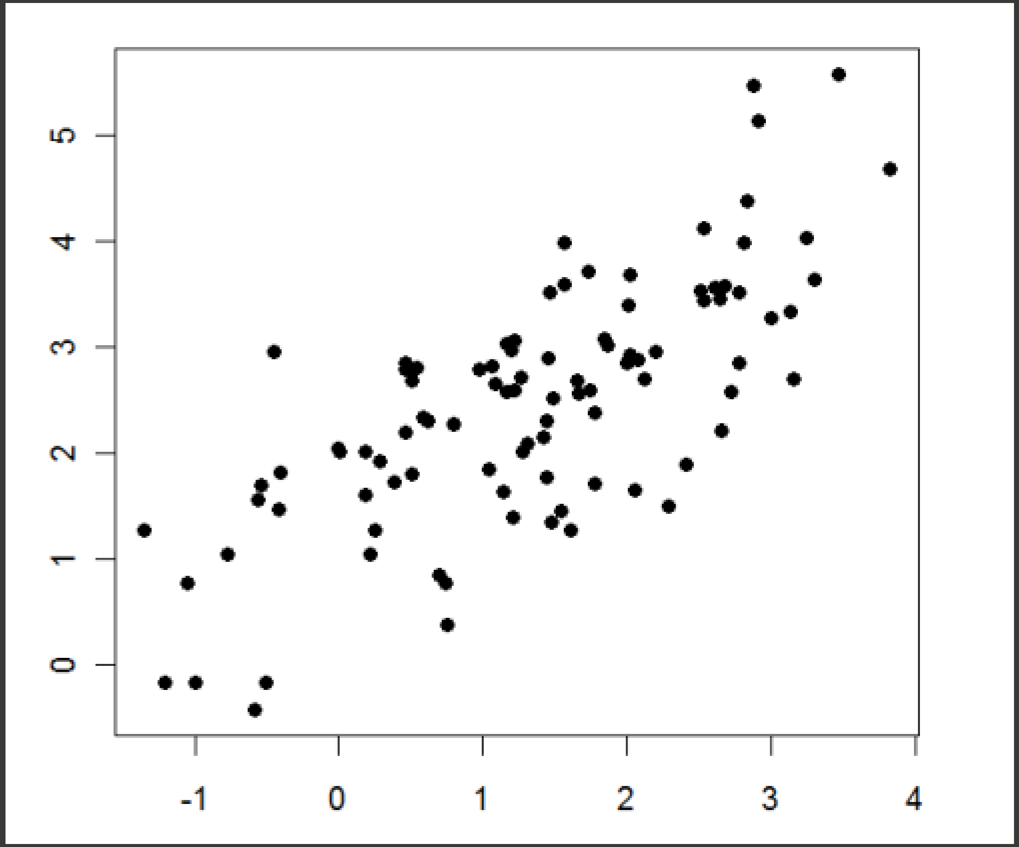
Contributors

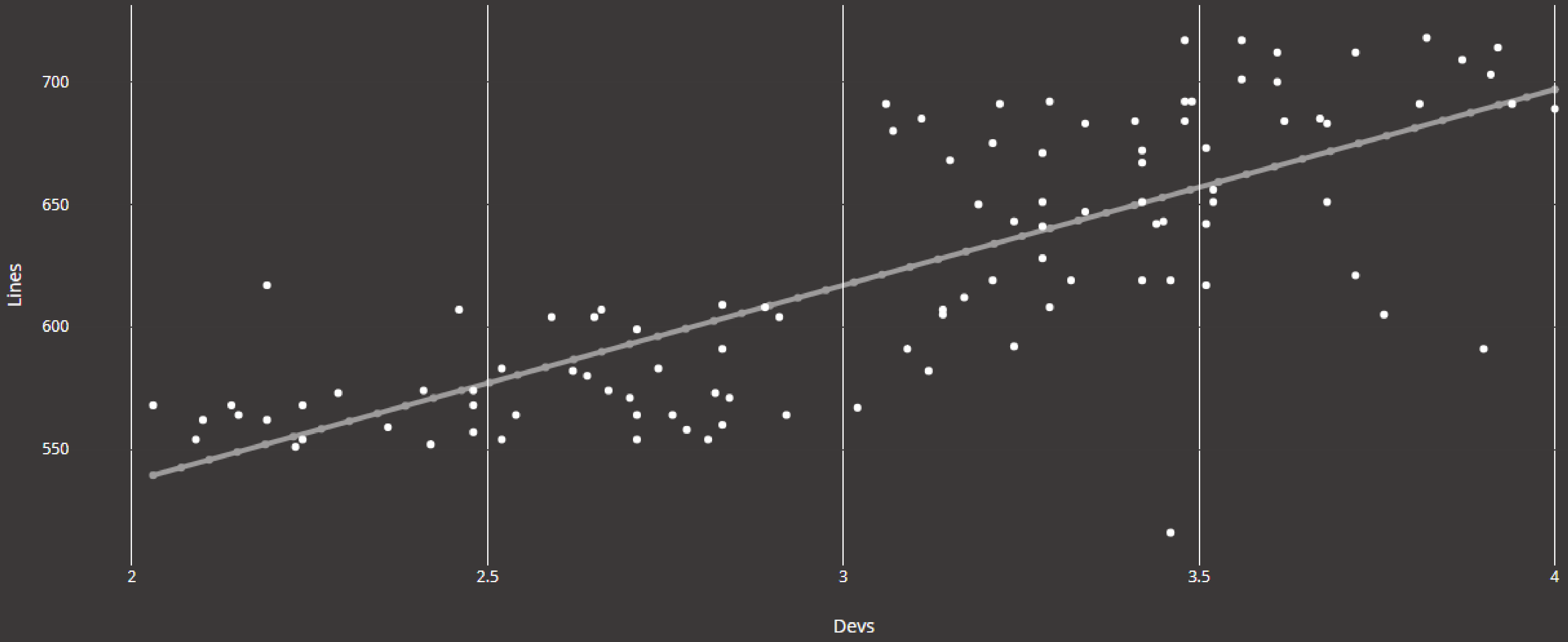
Comment Density

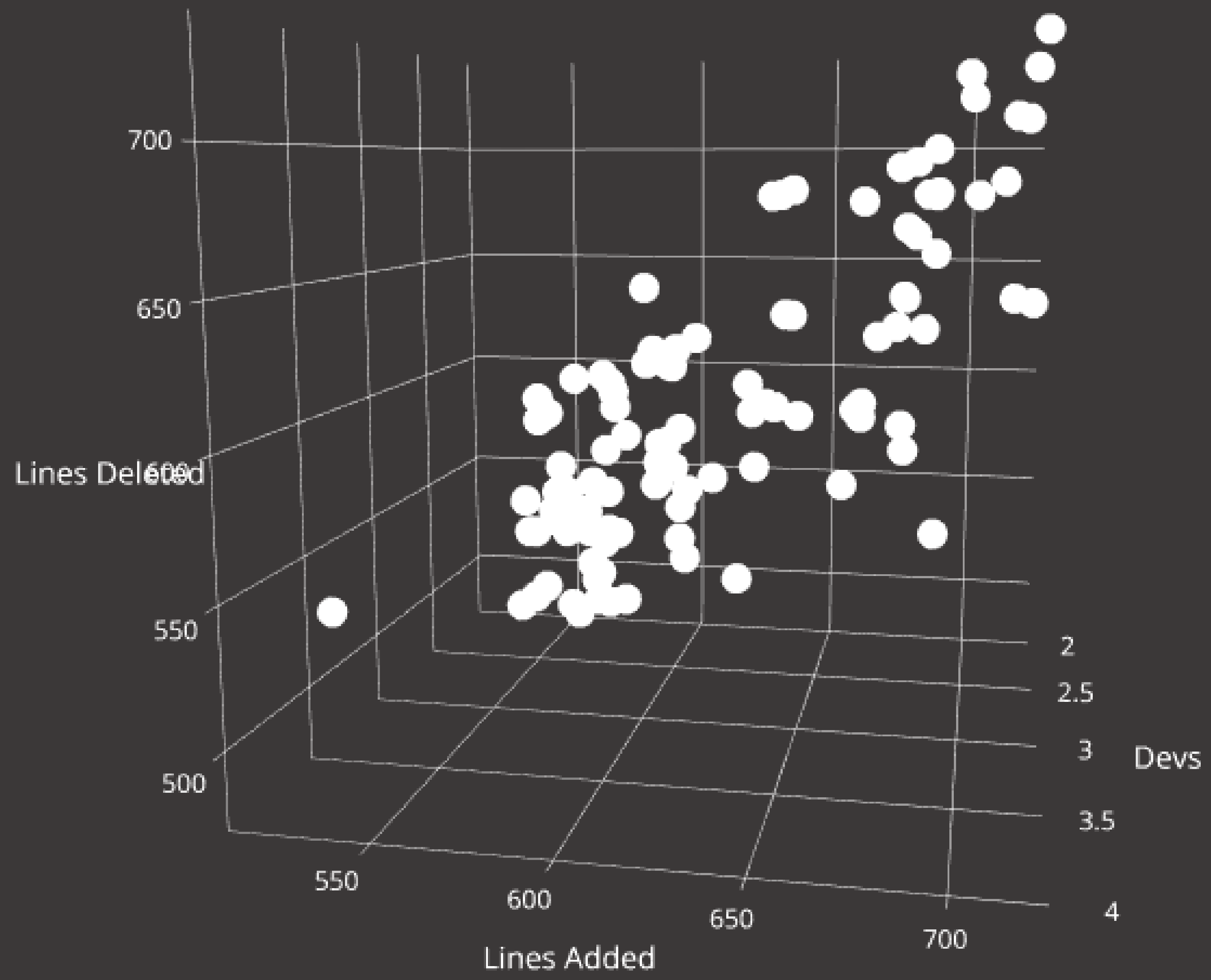
Machine Learning

$$x = \begin{bmatrix} f_{A1}\ f_{B1}\ f_{C1}\ f_{D1}\ \cdots\ f_{M1}\ C_1 \\ . \\ . \\ . \\ f_{AN}\ f_{BN}\ f_{CN}\ f_{DN}\ \cdots\ f_{MN}\ C_2 \end{bmatrix}$$

Feature Matrix




Lines
700
650
600
550
2
2.5
3
3.5
4
Devs


700
650
Lines Deleted
600
550
500
550
600
650
700
Lines Added
2
2.5
3
3.5
4
Devs

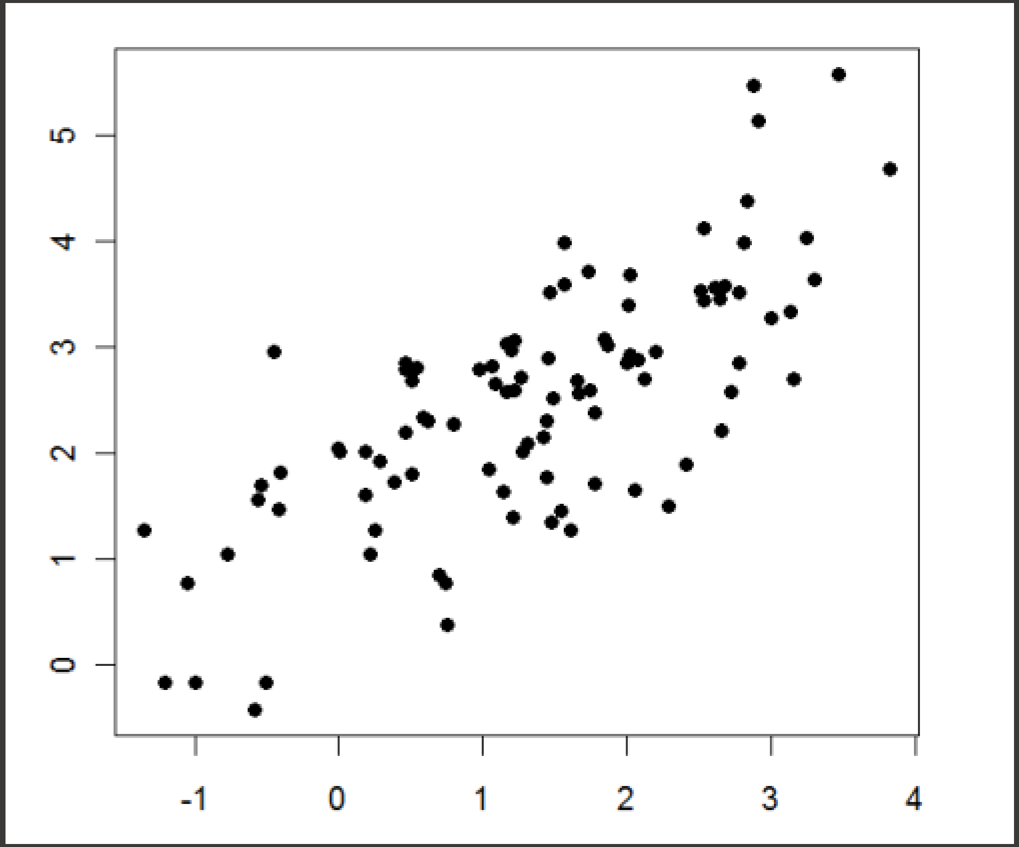
Machine Learning

$$x = \begin{bmatrix} f_{A1}\ f_{B1}\ f_{C1}\ f_{D1}\ \cdots\ f_{M1}\ C_1 \\ . \\ . \\ . \\ f_{AN}\ f_{BN}\ f_{CN}\ f_{DN}\ \cdots\ f_{MN}\ C_2 \end{bmatrix}$$

Feature Matrix

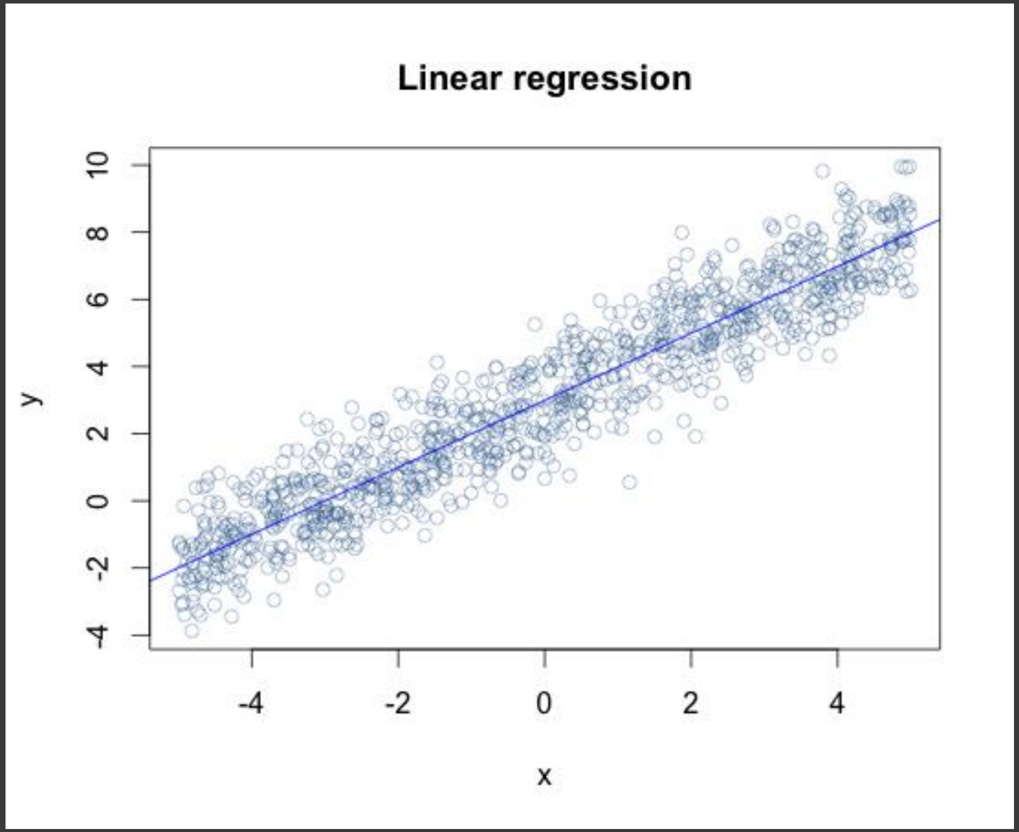
$$\sigma_{jk} = \frac{1}{n-1} \sum_{i=1}^{N} (x_{ij} - \bar{x}_{ij})(x_{ik} - \bar{x}_k)$$

Covariance Matrix

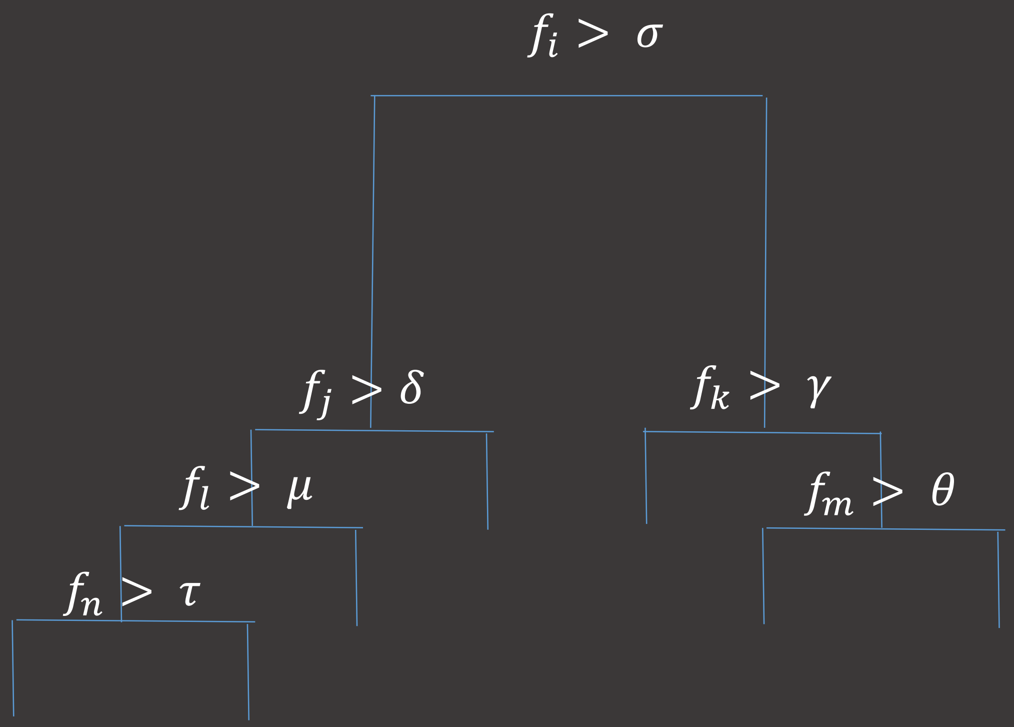


Machine Learning

*Tries to determine the best regression formulae: Subsystems*x + Directories*y + Files*z + LineAdded+l + LineDeleted*-0.00 + LineTotal*-0.00 + Devs*-0.05 + Age*0.20 + UniqueChange*0.30 + Exp*-0.00 + RExp*-0.00 + Sexp*0.50*



Machine Learning



$f_i > \sigma$

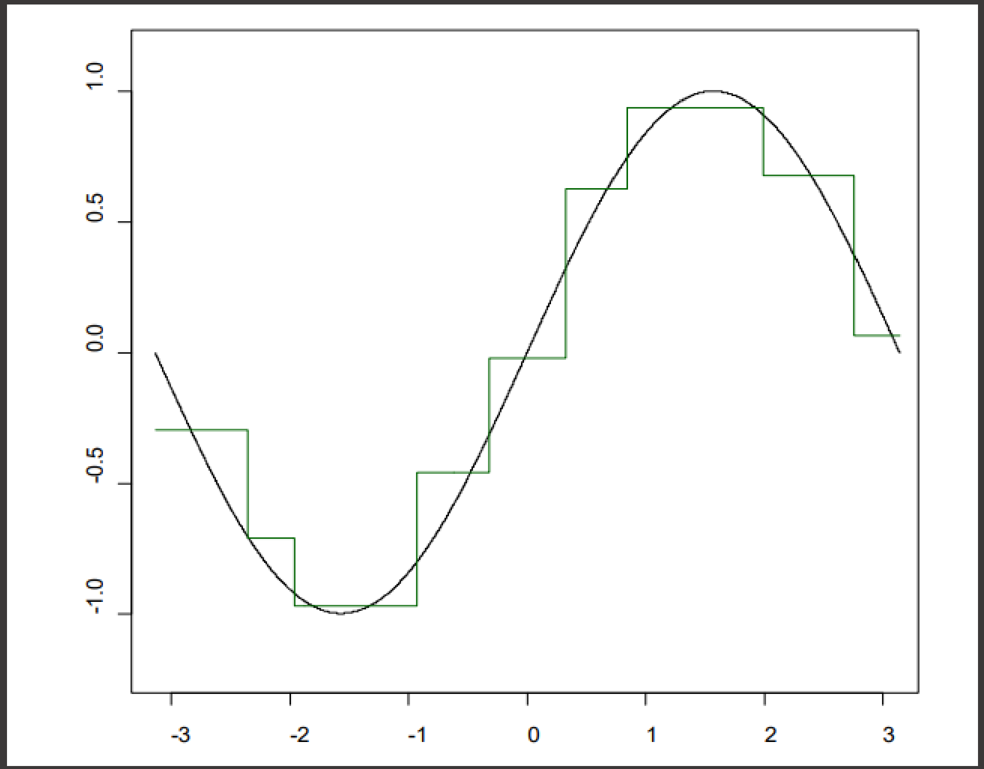
$f_j > \delta$

$f_k > \gamma$

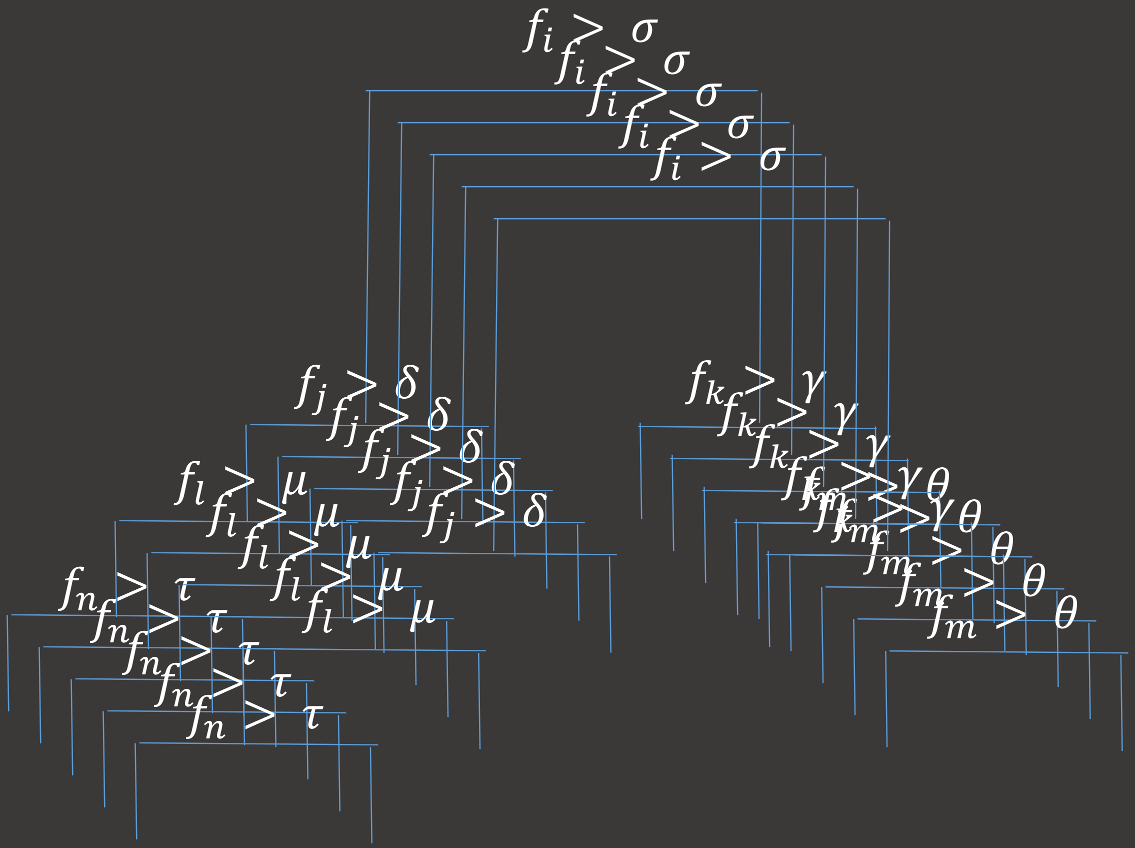
$f_l > \mu$

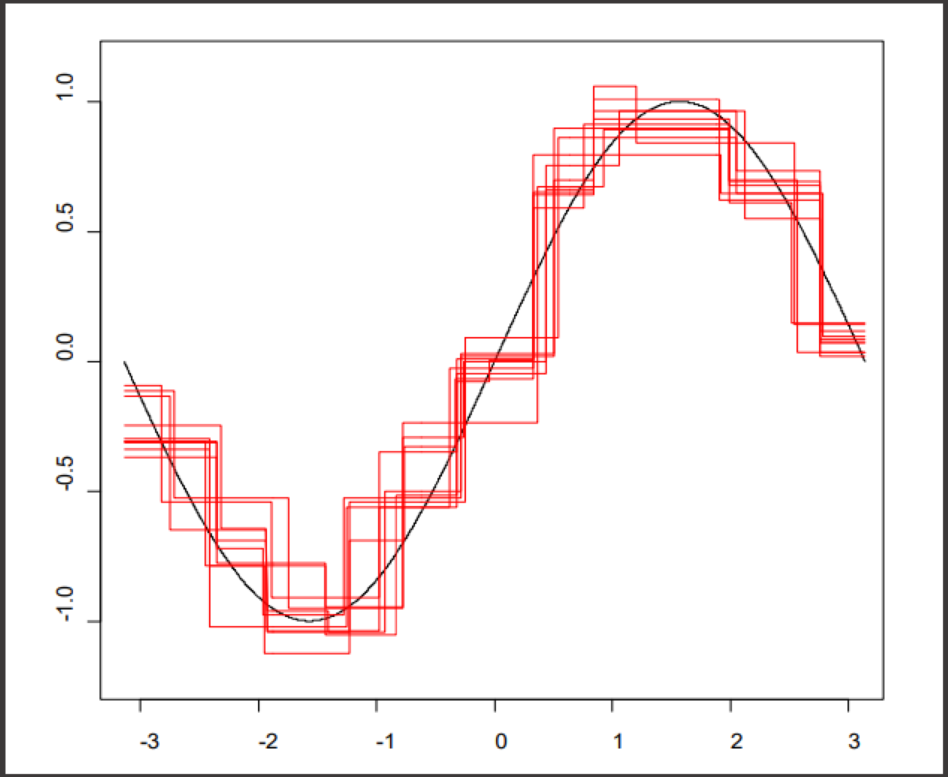
$f_m > \theta$

$f_n > \tau$

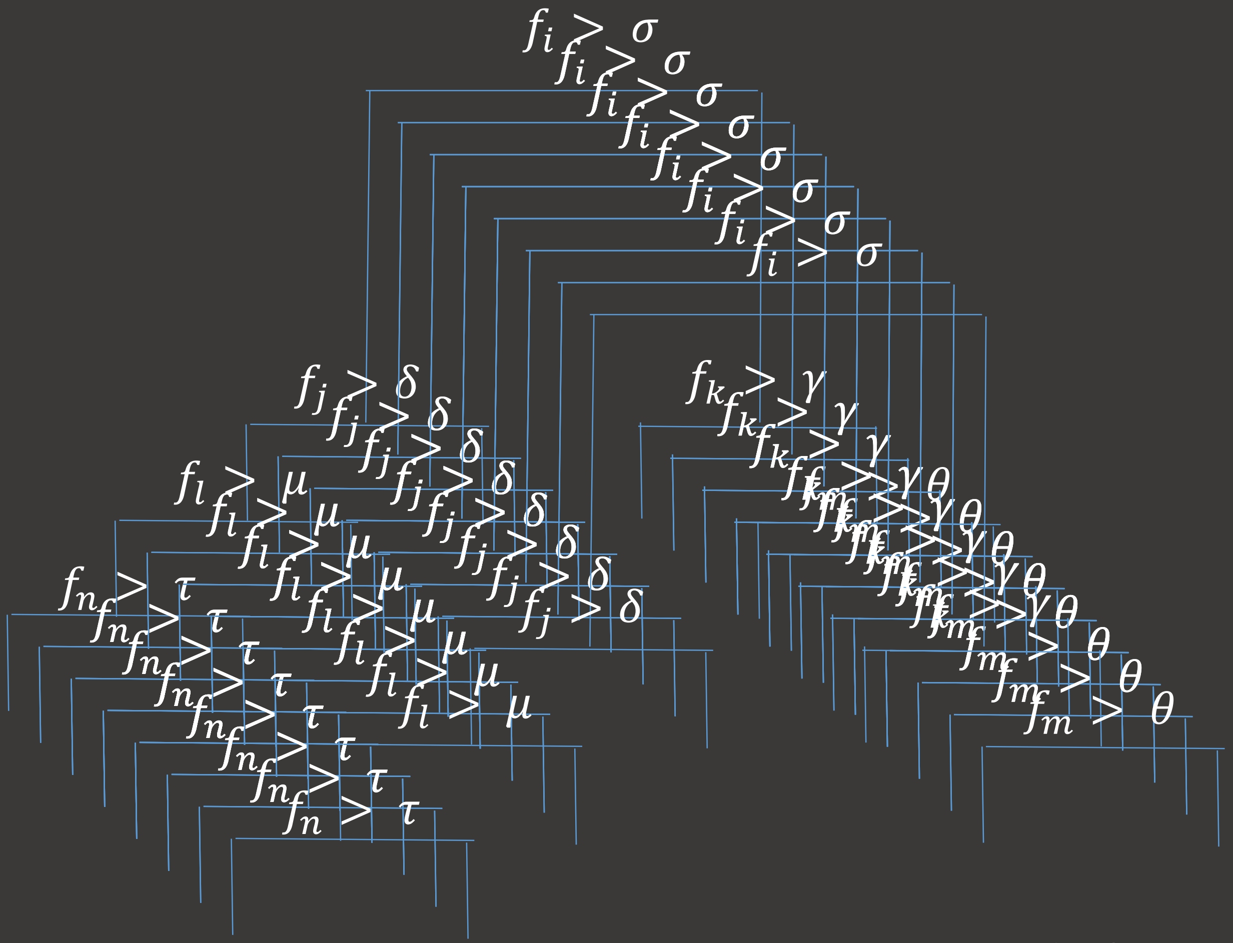


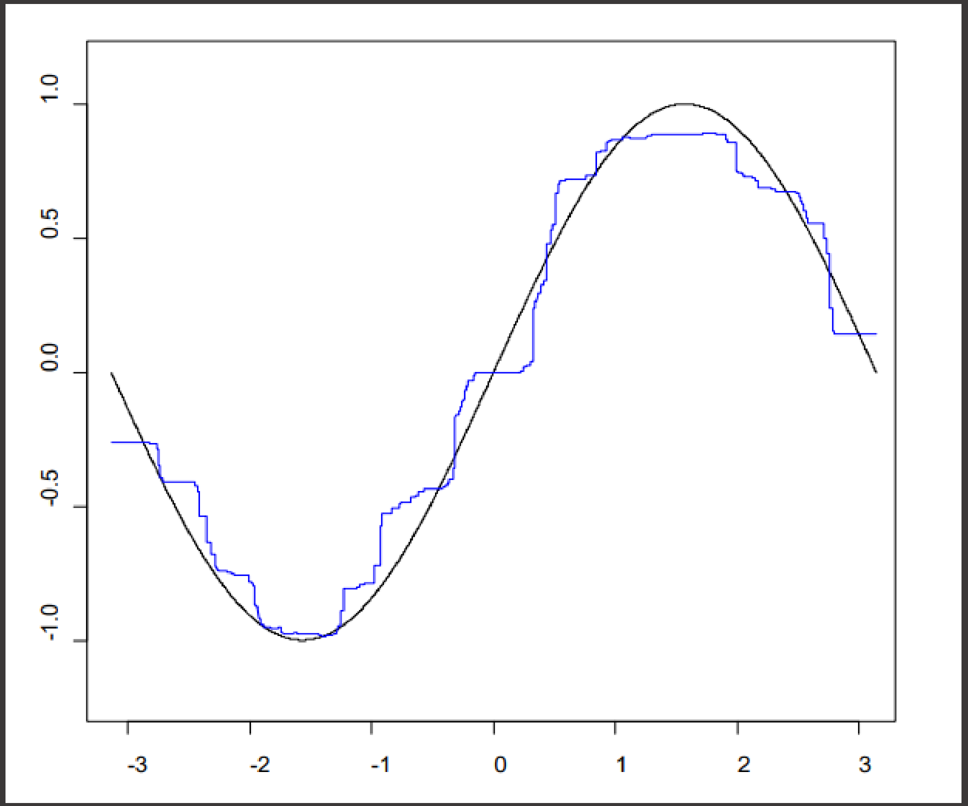
Machine Learning



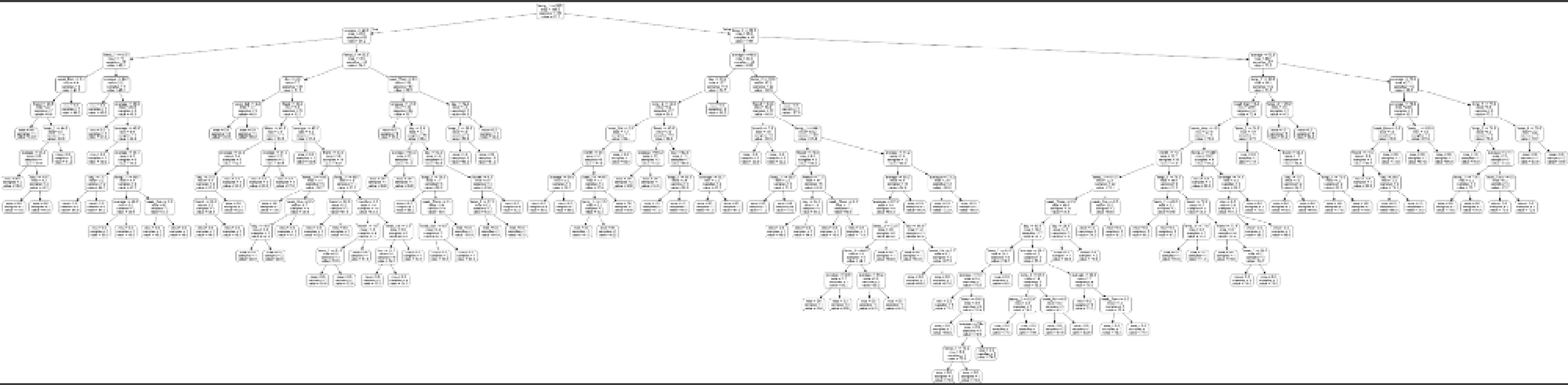


Machine Learning





Machine Learning



Machine Learning Performances

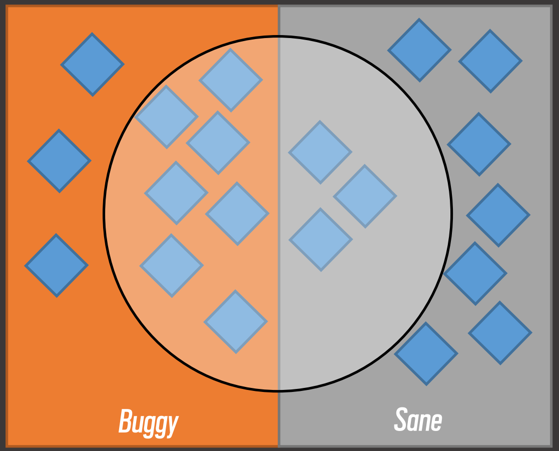
We can predict if a commit is introducing a defect with

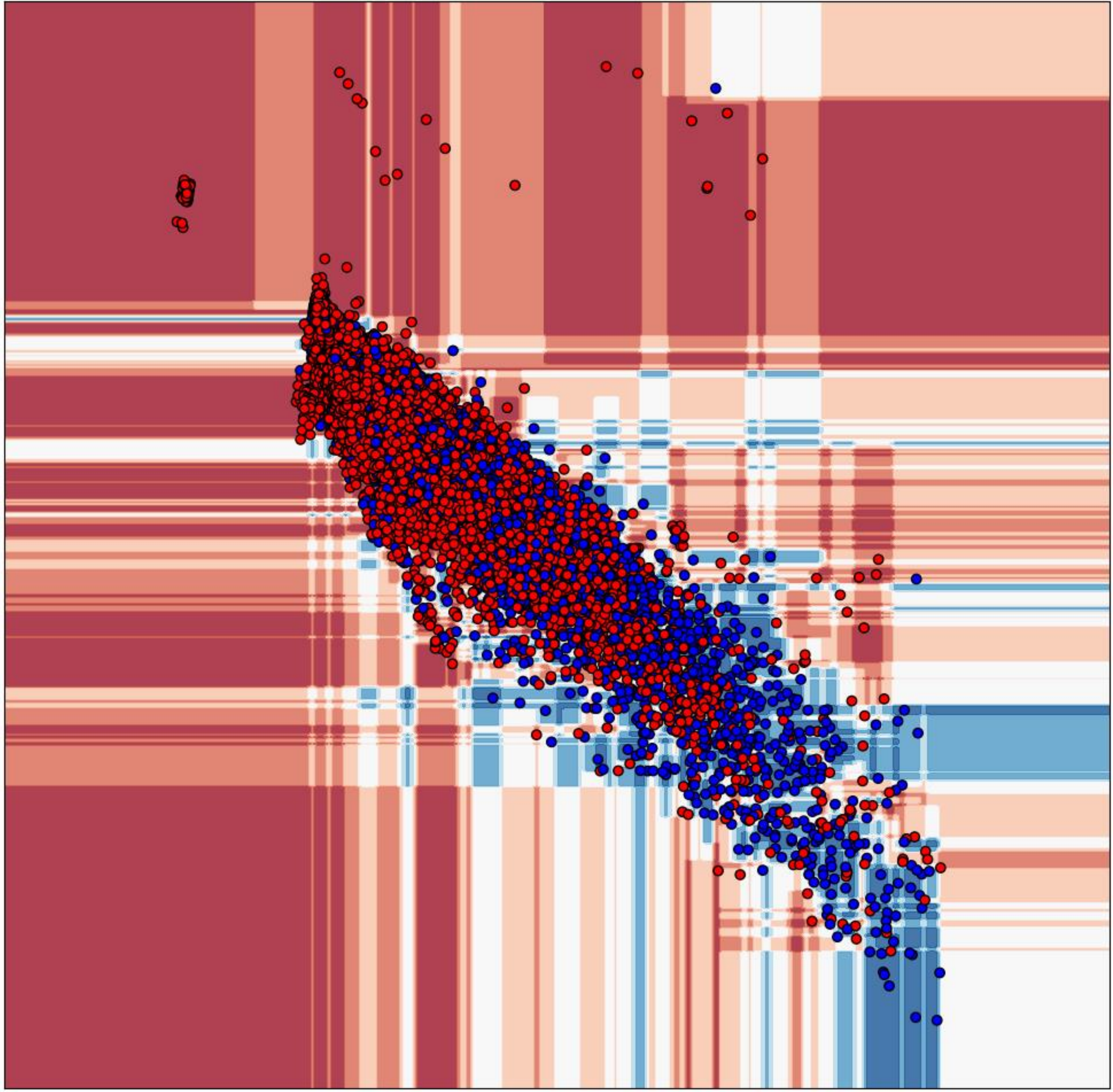
- *70.05% precision*
- *71.40% recall*

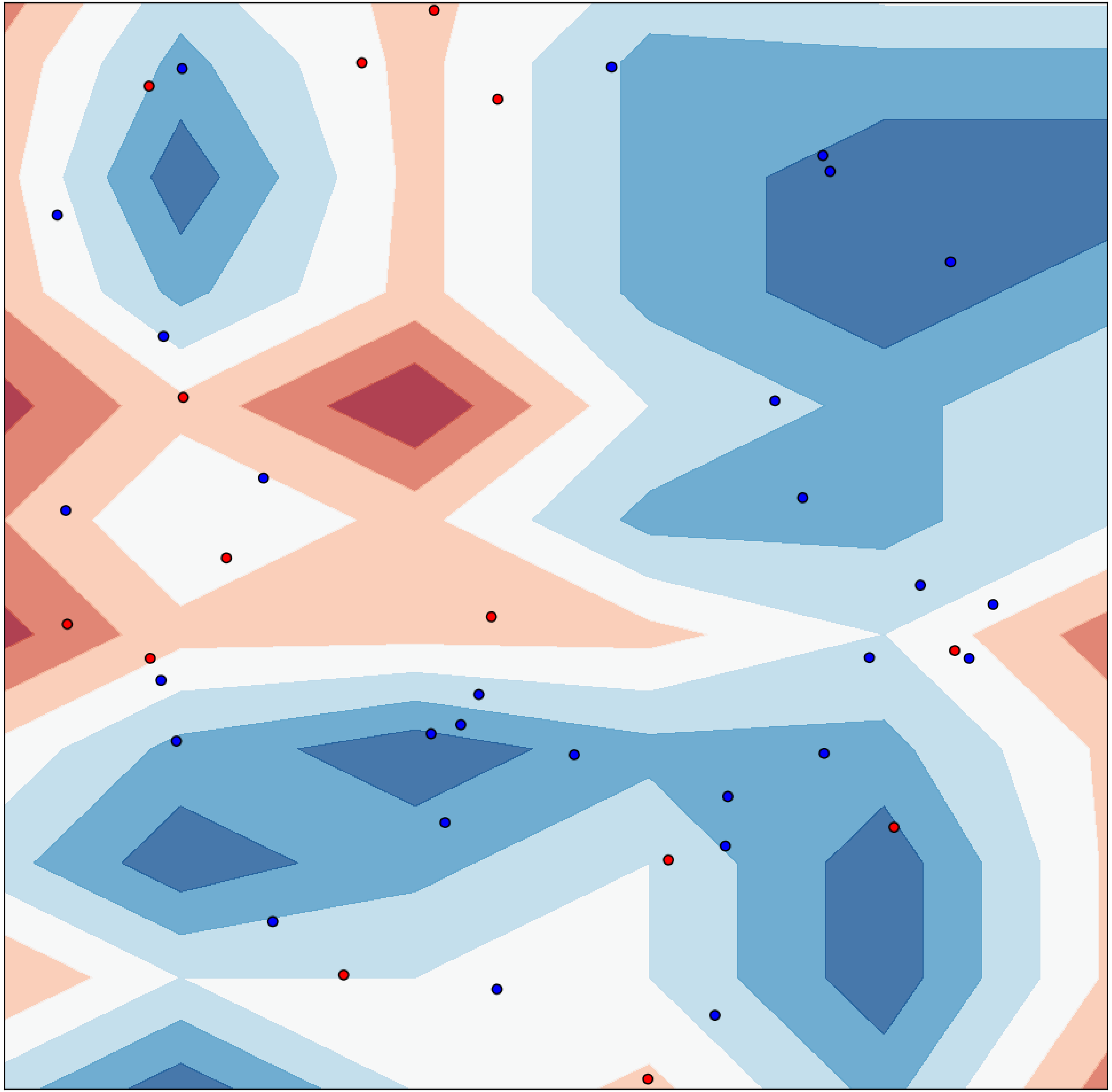
Machine Learning Performances

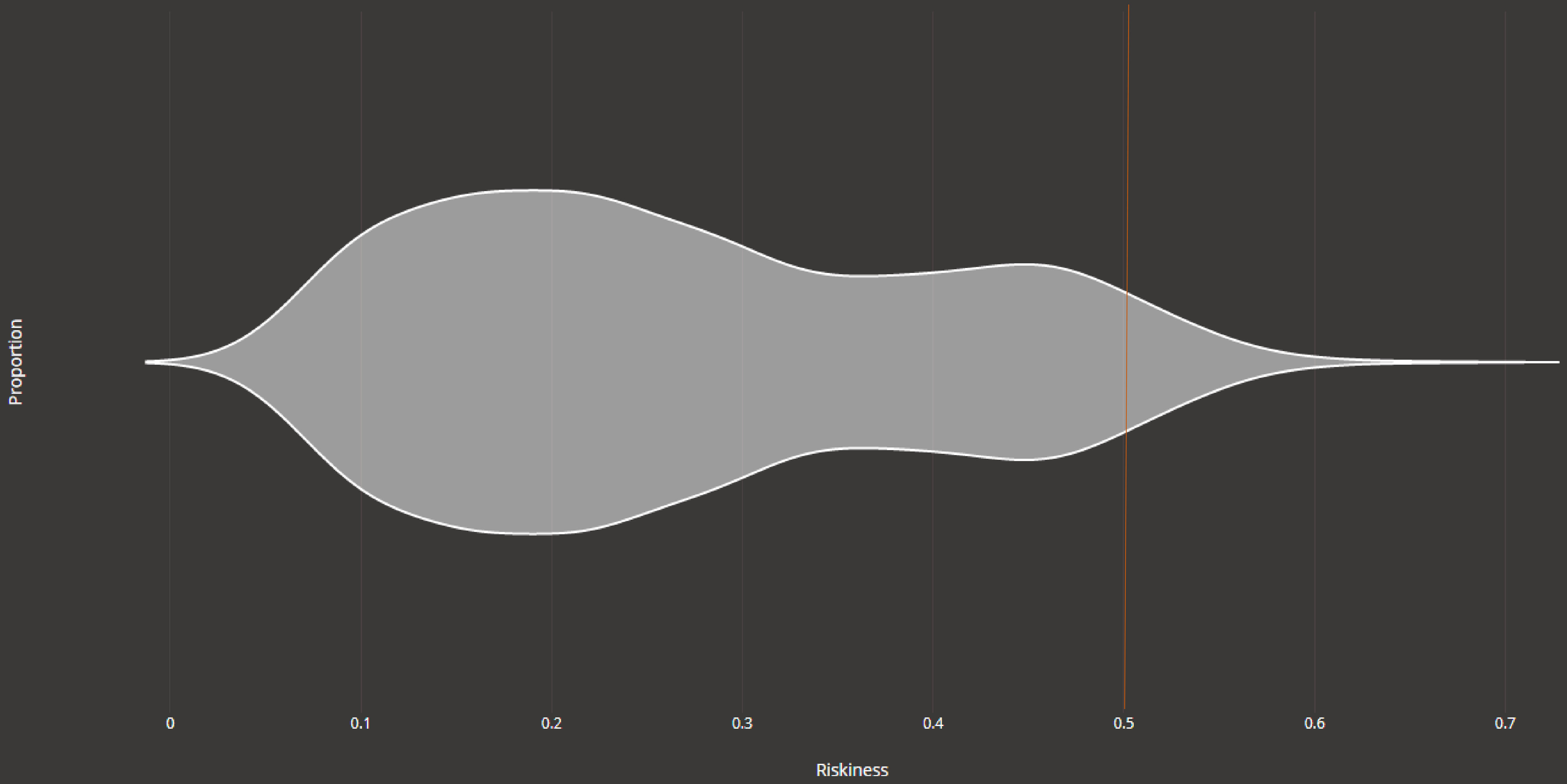
We can predict if a commit is introducing a defect with

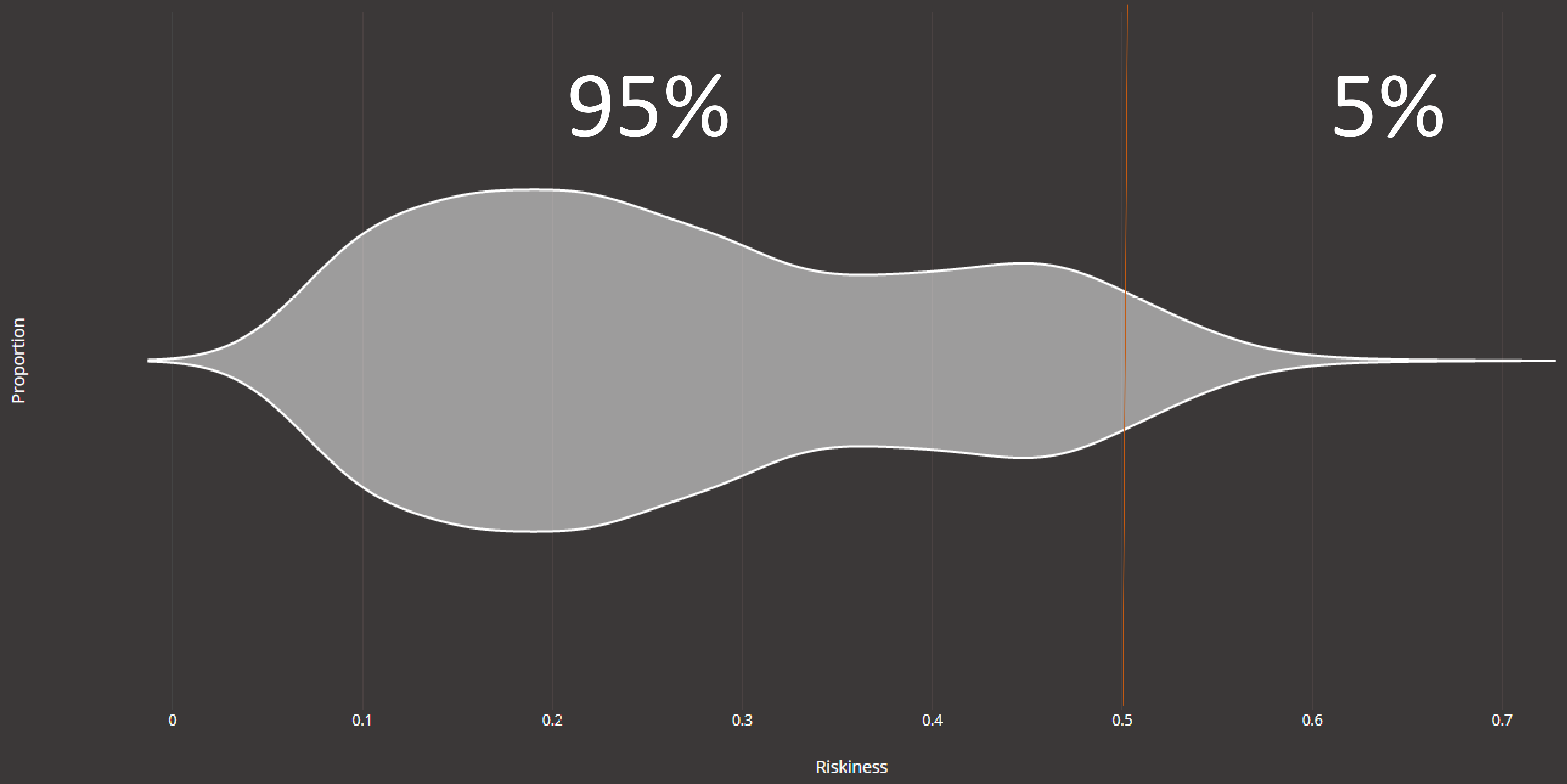
- *70.05% precision*
- *71.40% recall*



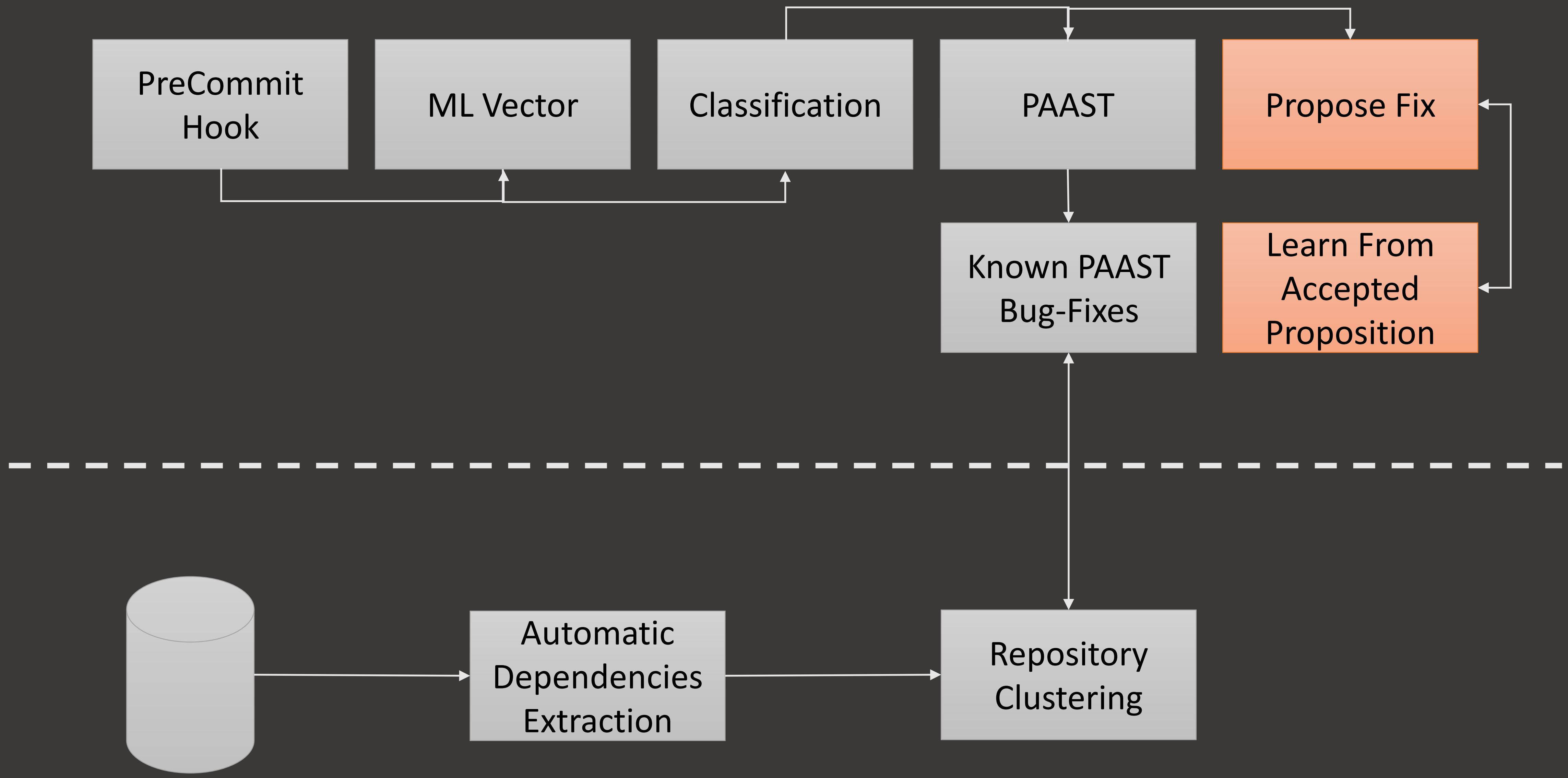





Proportion
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Riskiness


95%
5%
Proportion
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Riskiness

Prediction	Were we right?
Risk Factor %	True Positive %
0 - 10	1.45%
10 - 20	4.67%
20 - 30	11.10%
30 - 40	21.07%
40 - 50	43.95%
50 - 60	68.18%
60 – 70	79.91%
70 – 80	92.80%
80 - 90	96.11%
90 - 100	98.99%

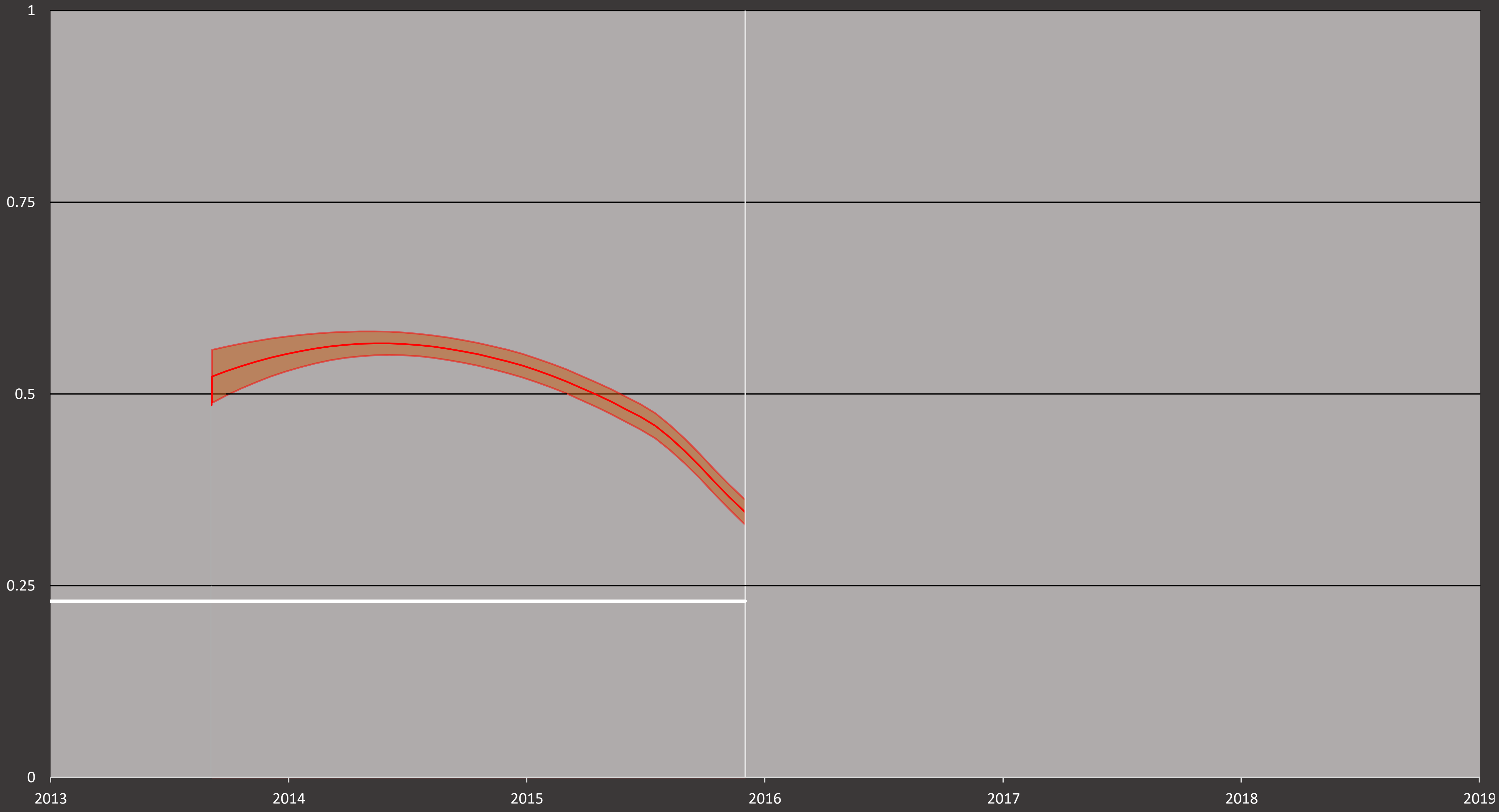

PreCommit Hook
ML Vector
Classification
PAAST
Propose Fix
Known PAAST Bug-Fixes
Learn From Accepted Proposition
Automatic Dependencies Extraction
Repository Clustering

laforge.ubisoft.com

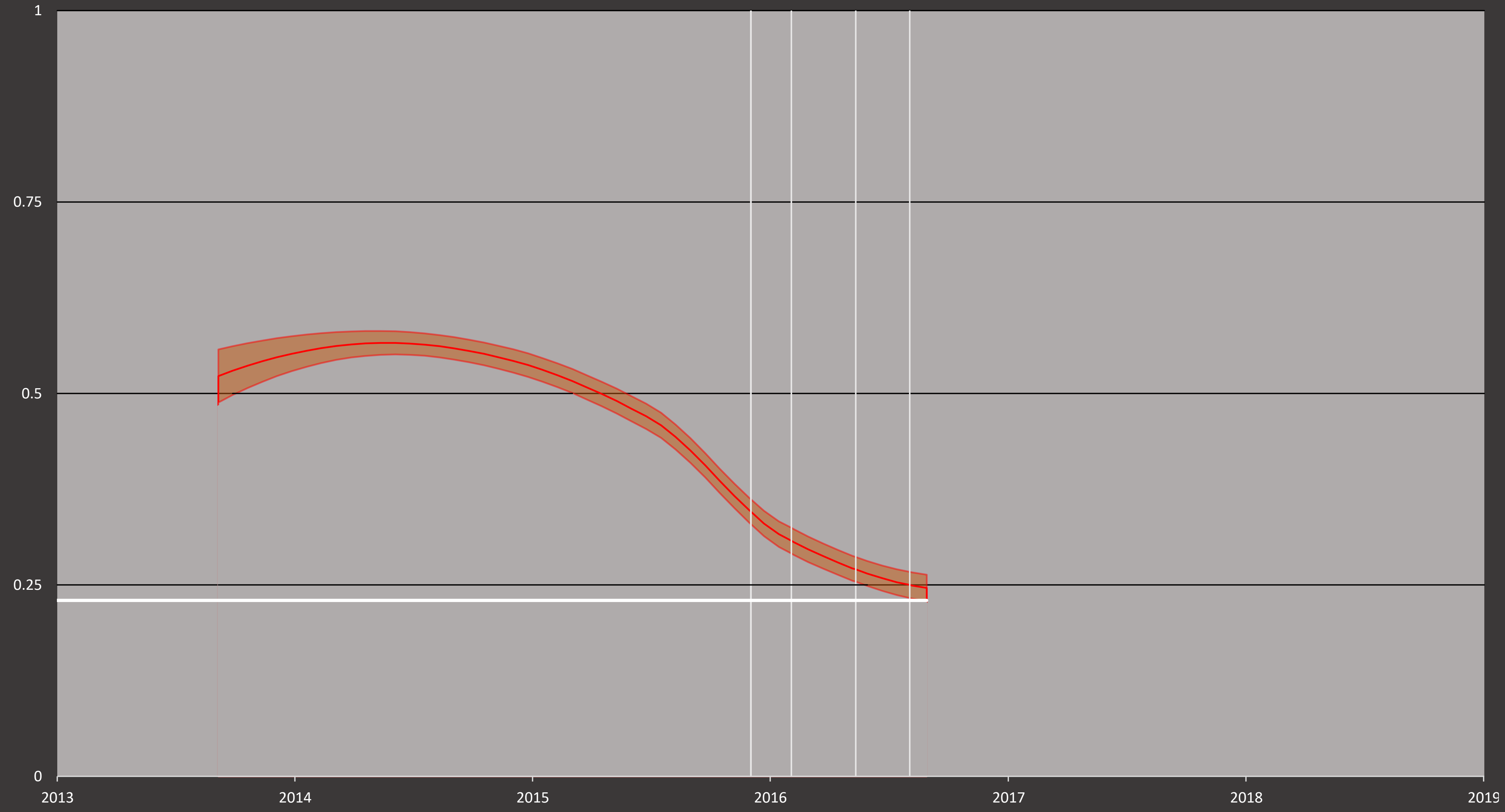
Rainbow Six Siege

- Released in December 2015
- 35+ million players
- 3 million players each day
- “Most Improved Game of 2017” according to multiple sites
- 600+ full-time people worldwide
- Around 200 testers
- Around 150 programmers
    - Around 100 programmers in game code itself

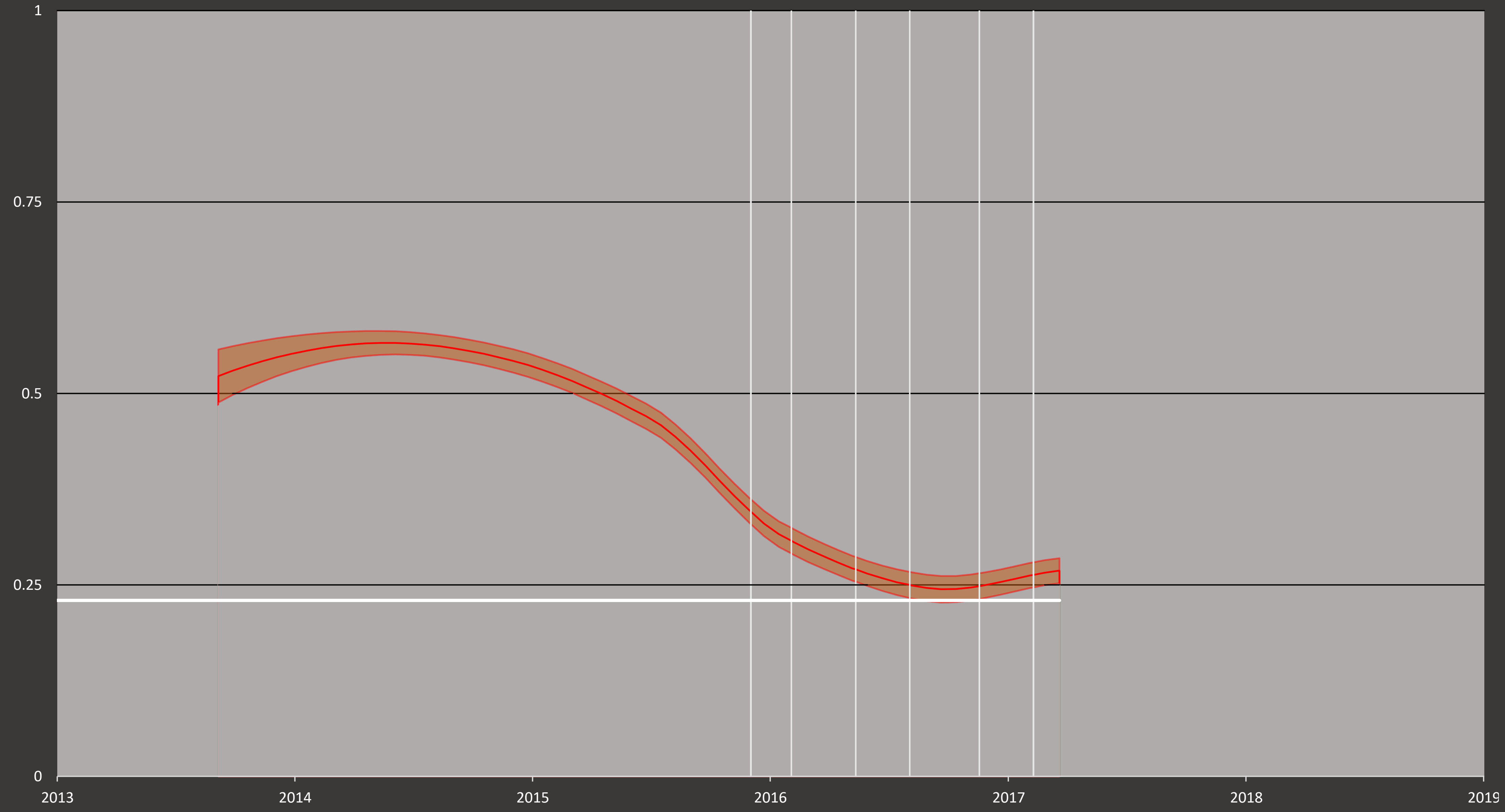
Bug Introduction Rate - Rainbow Six Siege



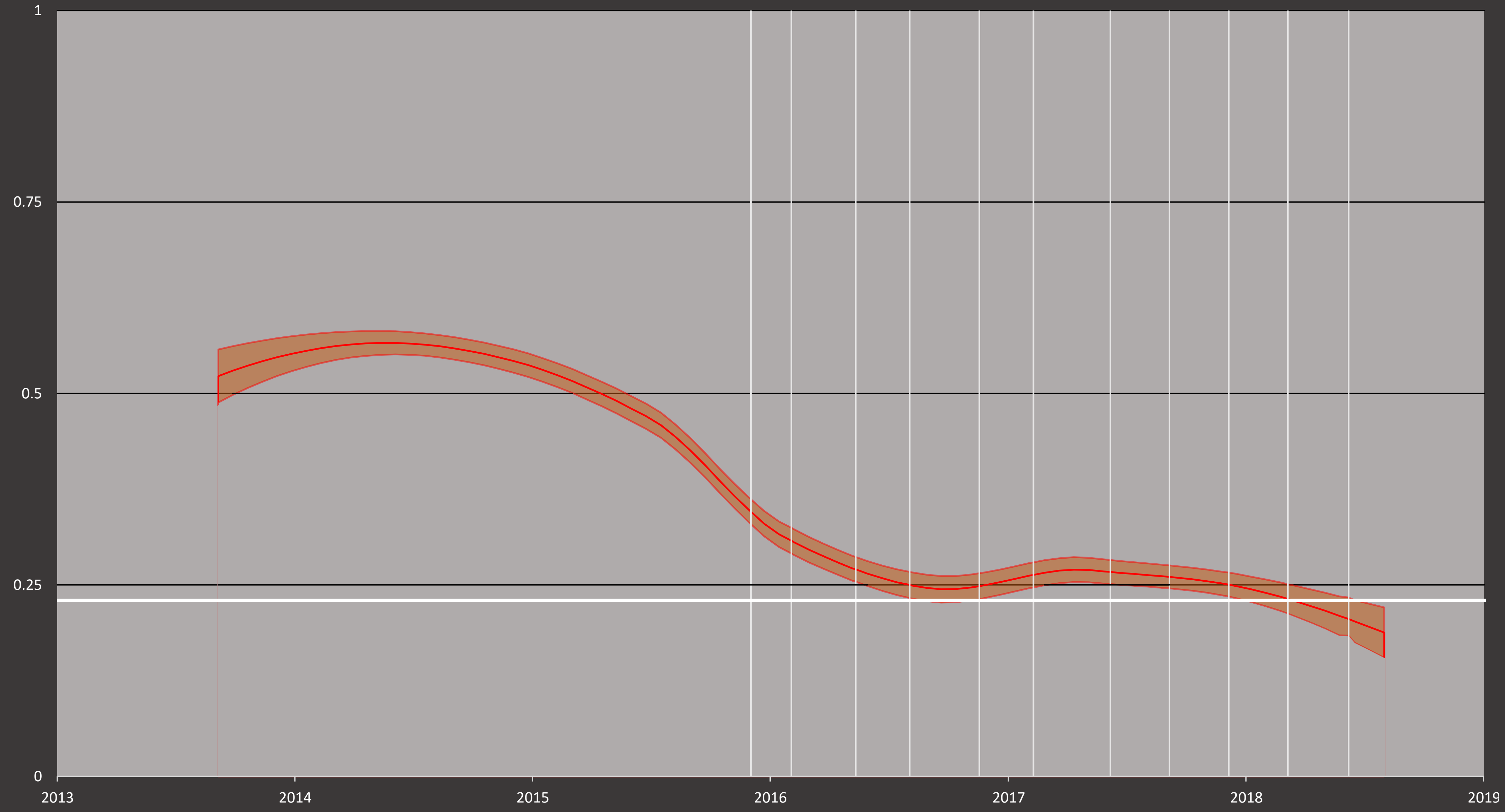
Bug Introduction Rate - Rainbow Six Siege

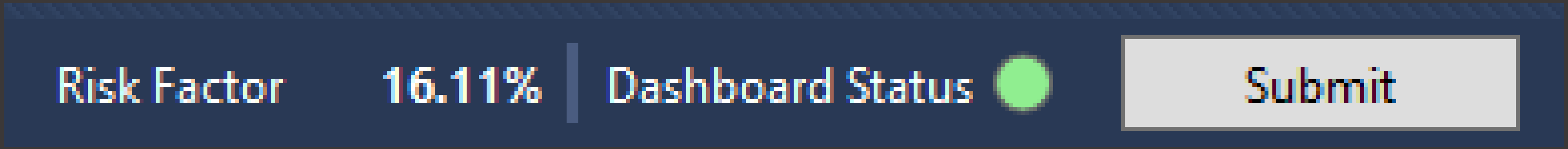


Bug Introduction Rate - Rainbow Six Siege



Bug Introduction Rate - Rainbow Six Siege





Risk Factor 16.11% | Dashboard Status 🟢

…/portal/sections/fullscreen/portalfullscreenviewcontentlistcontainer.h

⬇

- RS-504962 - [ShopShowcase] Tagging System - Elite Uniform is missing operator tag
- RS-501517 - [Customs][Local] Custom team names for local lobbies do not automatically update for non-host players
- RS-504185 - [ShopShowcase] Operator Icon not cleared when moving to a seasonal or universal weapon skin in the album fullscreen view
- …
- …

- RS-504962 - [ShopShowcase] Tagging System - Elite Uniform is missing operator tag
- …
- …

⬇

1. In the fullscreen view of owned weapon skins, within the Operators menu, scrolling from a normal weapon skin to a seasonal/universal one will not clear the Operator icon, causing it to overlap with the seasonal icon and text
2. When the host of a local customs lobby changes the team names using the "ALT" key (default) all joined players will not see the newly chosen names until they leave the game and rejoin it

```
EnablePlayer(position, false);
EnablePlayer(position, false, false);
```

```
manager.SetEntity(obj, true);
manager.SetEntity(obj, false);
```

Potential

- A lot of bugs can be seen in a single function.
- A lot of things to improve as well.
  - Missing .Reserve(…) calls

Potential

- A lot of bugs can be seen in a single function.
- A lot of things to improve as well.
  - Missing .Reserve(…) calls
  - Wrong big O

Potential

- A lot of bugs can be seen in a single function.
- A lot of things to improve as well.
  - Missing .Reserve(…) call
  - Wrong big O

```
for (ubiU32 i = 0; i < m_SomeArray.Size();)
{
    Element* element = m_SomeArray[i];
    if (SomeCondition(element))
        m_SomeArray.RemoveAt(i)
    else
        ++i;
}
```

```
m_SomeArray.RemoveIf([](Element* e){
    return SomeCondition(e); });
```

Potential

- A lot of bugs can be seen in a single function.
- A lot of things to improve as well.
    - Missing .Reserve(…) call
    - Wrong big O
    - Bad usage of heap
        - Array -> InplaceArray (aka std::vector vs. SmallVector)

Potential

- A lot of bugs can be seen in a single function.
- A lot of things to improve as well.
  - Missing .Reserve(…) call
  - Wrong big O
  - Bad usage of heap
    - Array -> InplaceArray (aka std::vector vs. SmallVector)
    - std::function -> inplace_function from SG14 (if bigger than SSO buffer)

Potential

- A lot of bugs can be seen in a single function.
- A lot of things to improve as well.
    - Missing .Reserve(…) call
    - Wrong big O
    - Bad usage of heap
        - Array -> InplaceArray (aka std::vector vs. SmallVector)
        - std::function -> inplace_function from SG14 (if bigger than SSO buffer)
        - String -> InplaceString (if bigger than SSO buffer)

Potential

- A lot of bugs can be seen in a single function.
- A lot of things to improve as well.
  - Missing .Reserve(…) call
  - Wrong big O
  - Bad usage of heap
    - Array -> InplaceArray (aka std::vector vs. SmallVector)
    - std::function -> inplace_function from SG14 (if bigger than SSO buffer)
    - String -> InplaceString (if bigger than SSO buffer)
- Will false positives ruin it?

Questions?