Graduation Project Defense

Optimizing Continuous Integration Using Artificial Intelligence

University of Carthage
École Polytechnique de Tunisie

Host Organization UNIVERSITÉ LAVAL

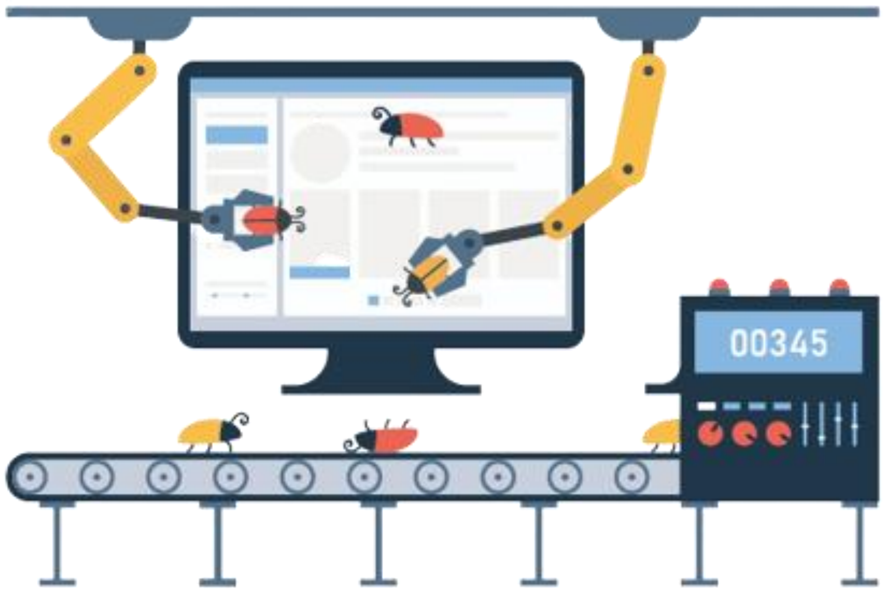
Elaborated by **Maher Dissem**

Supported on July 4th, 2022, before the examination board:

Dr.	Mohamed Hamdi	Associate Professor – SUP'COM	President
Dr.	Ramzi Guetari	Associate Professor – EPT	Rapporteur
Prof.	Naceur Benhadj Braiek	Professor - EPT	Academic Supervisor
Dr.	Mohamed Aymen Saied	Assistant Professor - ULaval	Supervisor
Mr.	Mohamed Mouine	Ph.D. Student - ULaval	Supervisor

Academic year: 2021-2022

CONTENTS



I

Project’s Description

1. General Context
2. Continuous Integration
3. Problem Statement
4. Project’s Objectives

Project’s Description
General Context

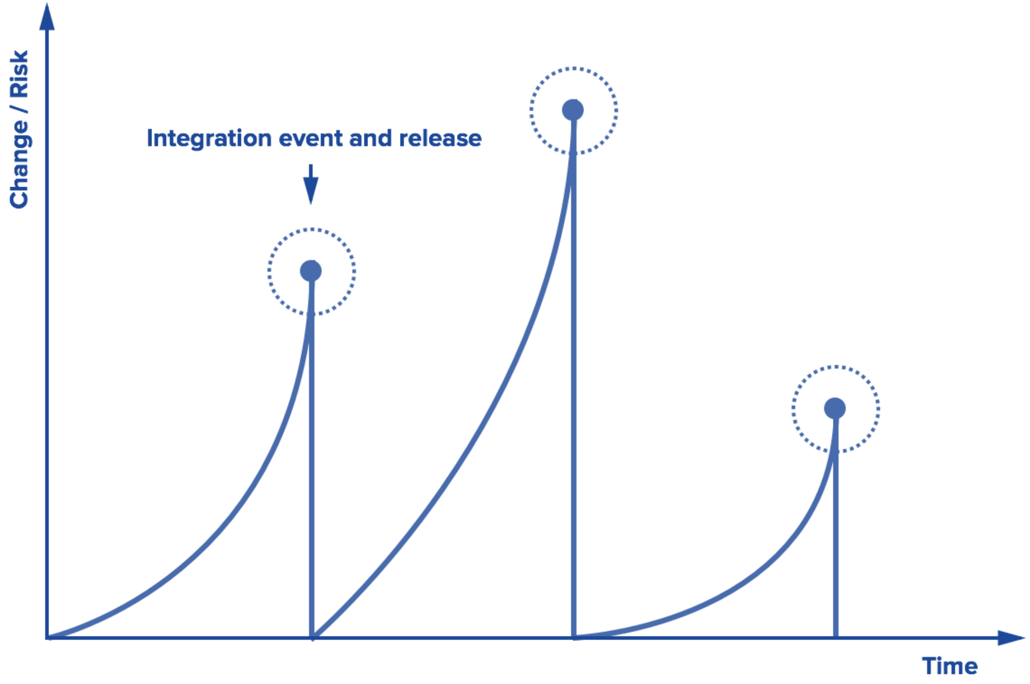
As projects become more complex, the risk of human error causing consequent defects also grows.

Software flaws and **delivery delays** can harm a company’s reputation resulting in lost consumers.

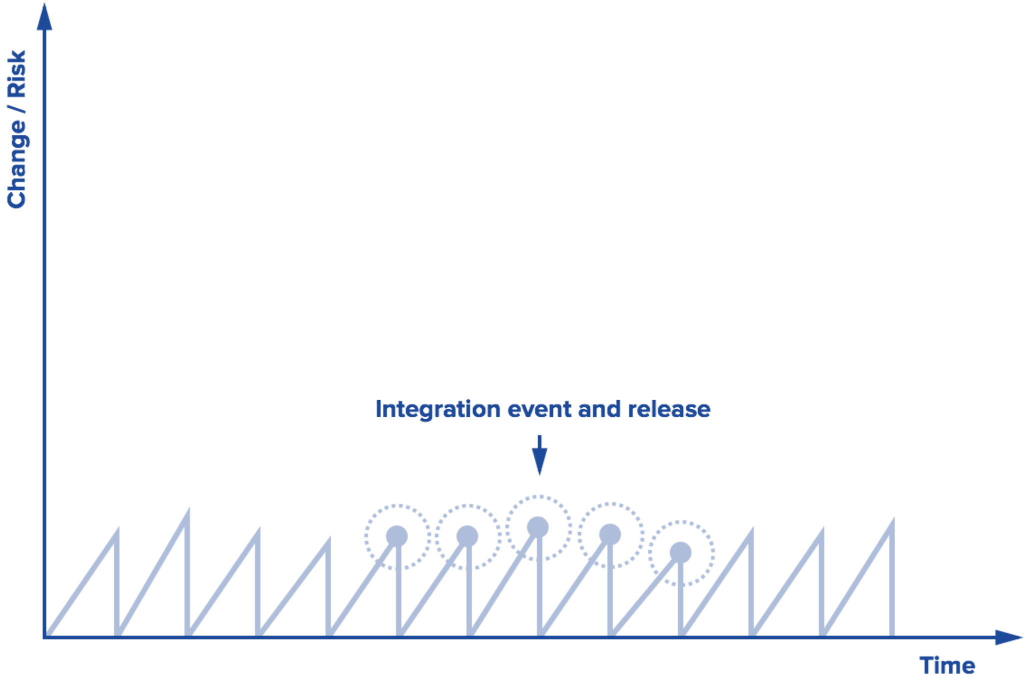
Project's Description
Continuous Integration

A modern practice where developers frequently push their modifications into the main codebase.

The software is built and tested after every commit.



Traditional Integration



Continuous Integration

Project's Description

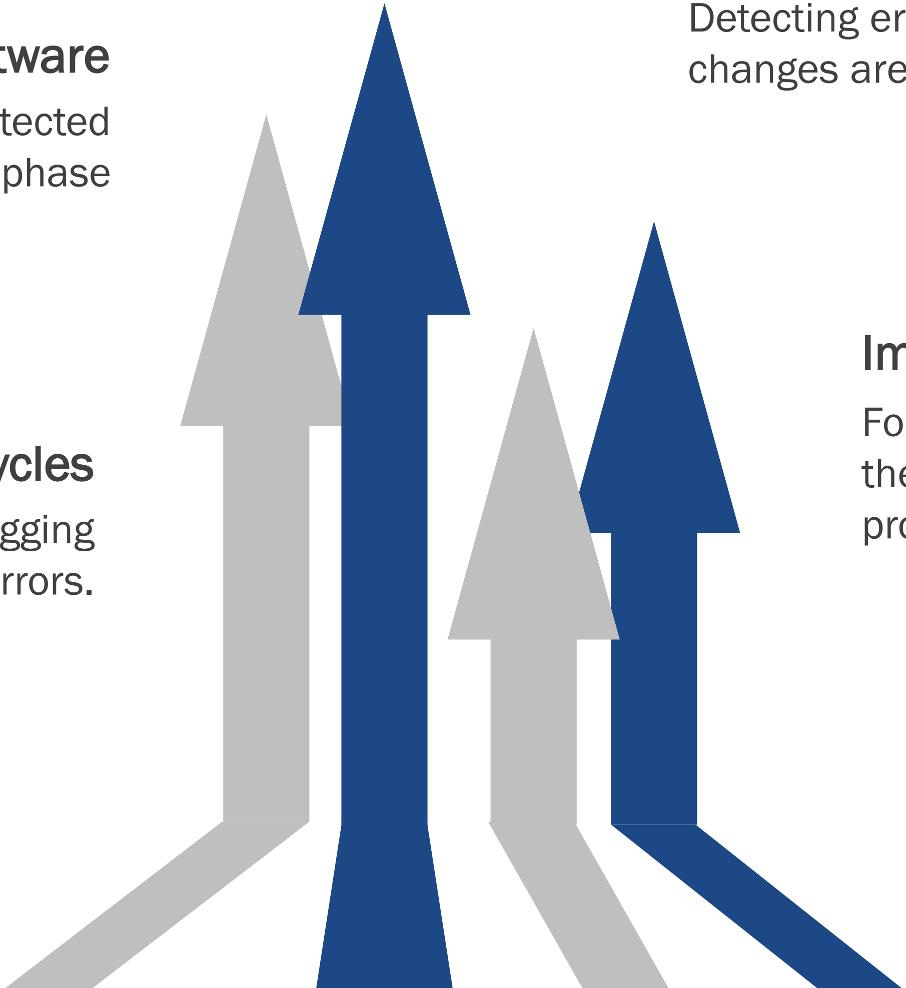
Benefits of Continuous Integration

Lower risk of delivering defective software

Most failures will be detected during the testing phase

Faster release cycles

Less time is spent debugging errors.



Faster errors debugging

Detecting errors is easier when less changes are integrated at once.

Improved code quality

Focusing on the functionality of the code rather than on avoiding problems.

Project’s Description

Continuous Integration Workflow

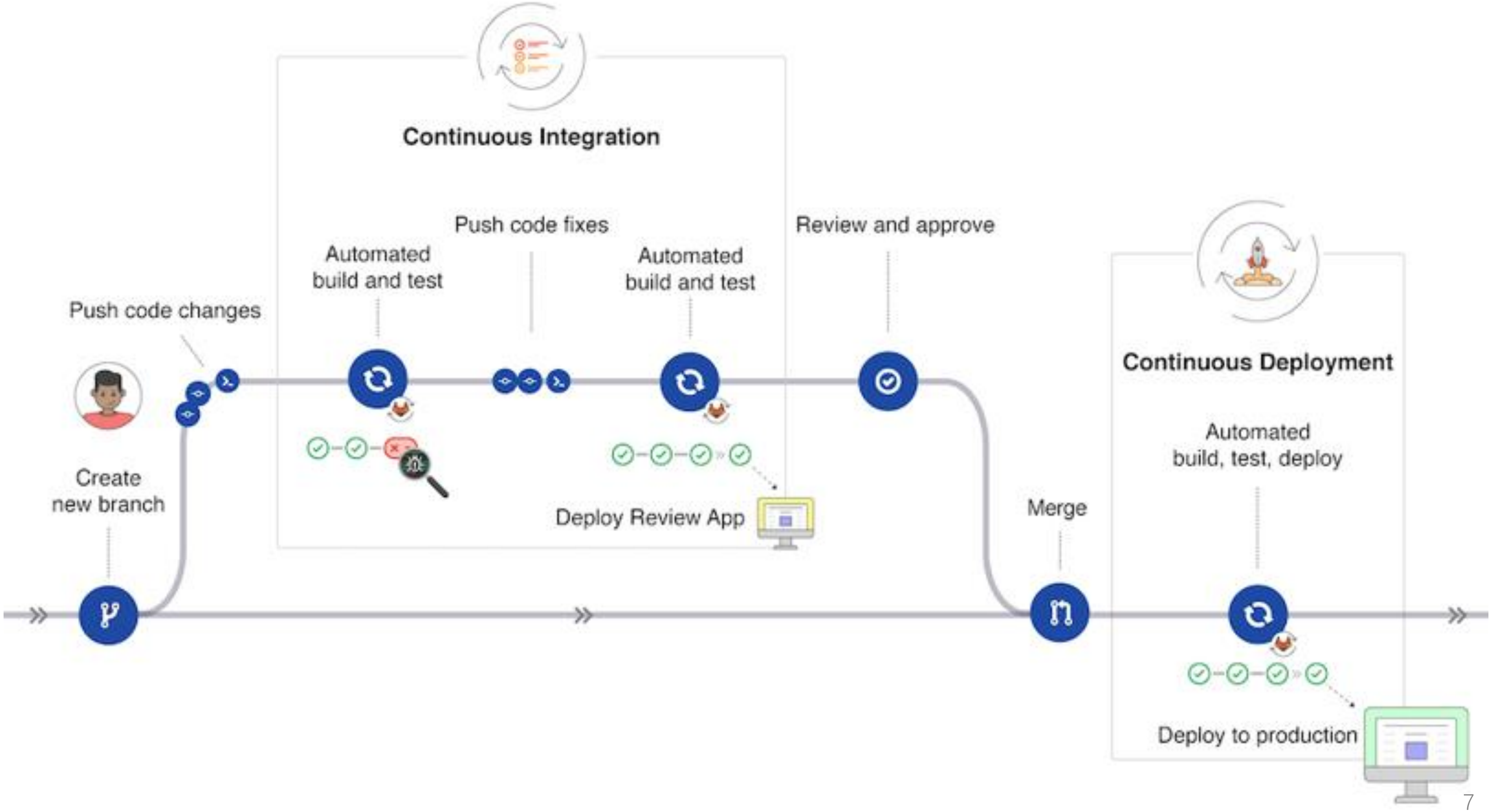


Image credits: Gitlab documentation

Project’s Description

Drawbacks of Continuous Integration

Power

Constant use of computational resources to run builds and tests.

Time

Long build durations.

Developers become less productive.

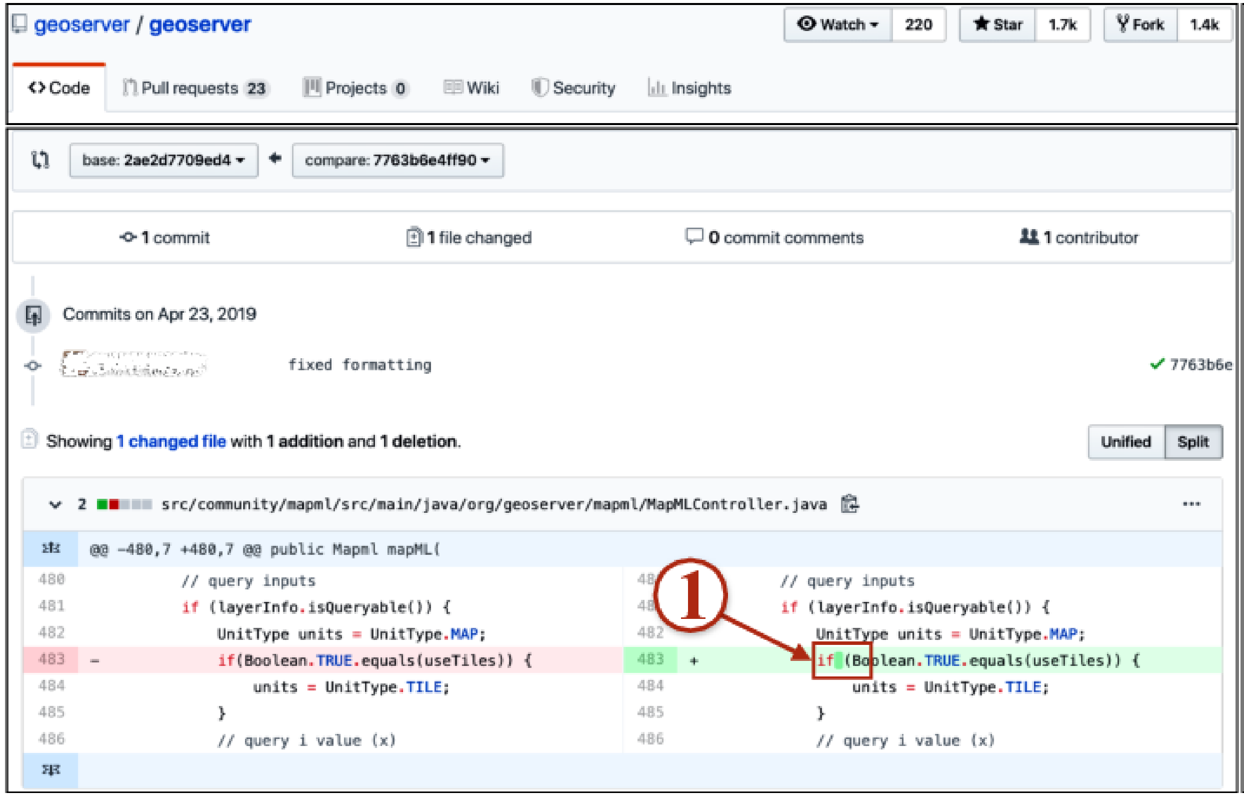
High cost

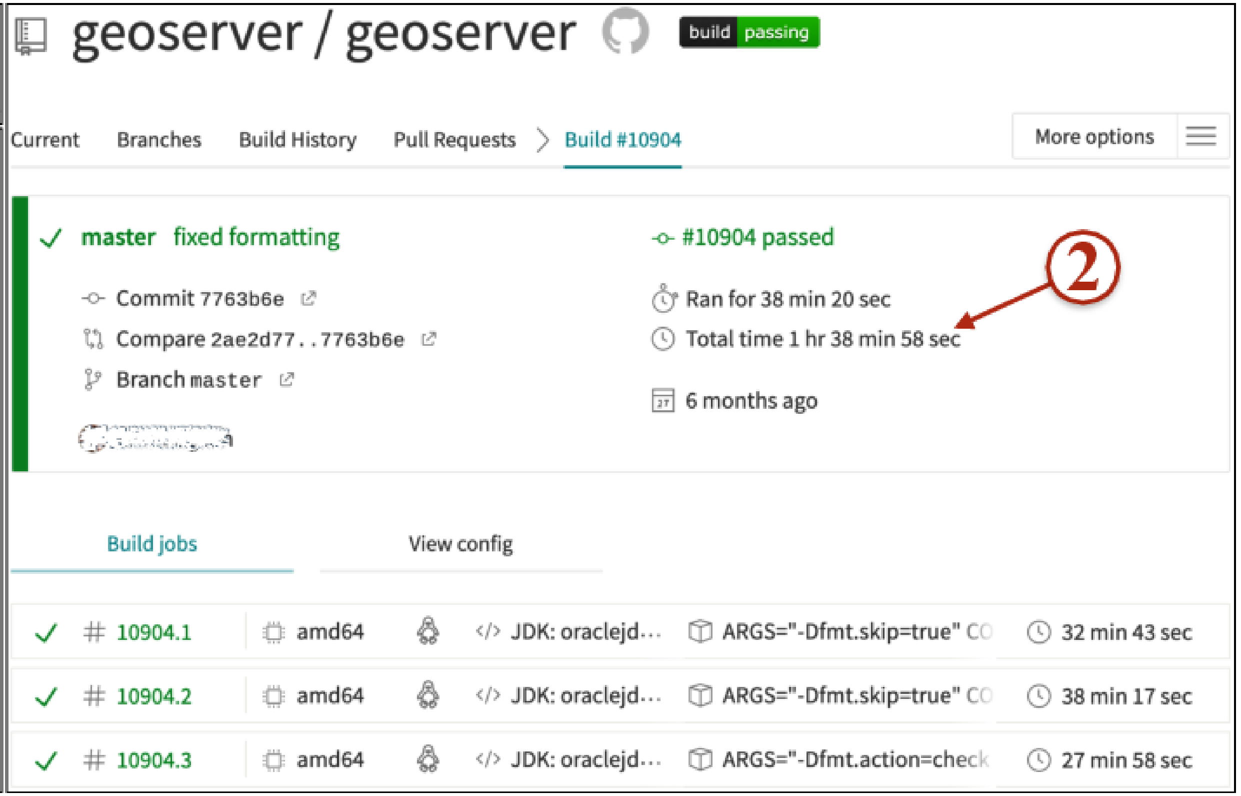
Google estimates its CI systems in millions of dollar and Mozilla estimate theirs as 201 k$/month

This high cost hinders the adoption of CI by small companies.

Project’s Description
Optimization Strategy

Motivating Example
The build process took **1 hr 38 min** for a simple code reformatting commit





Project’s Description
Project’s Objectives

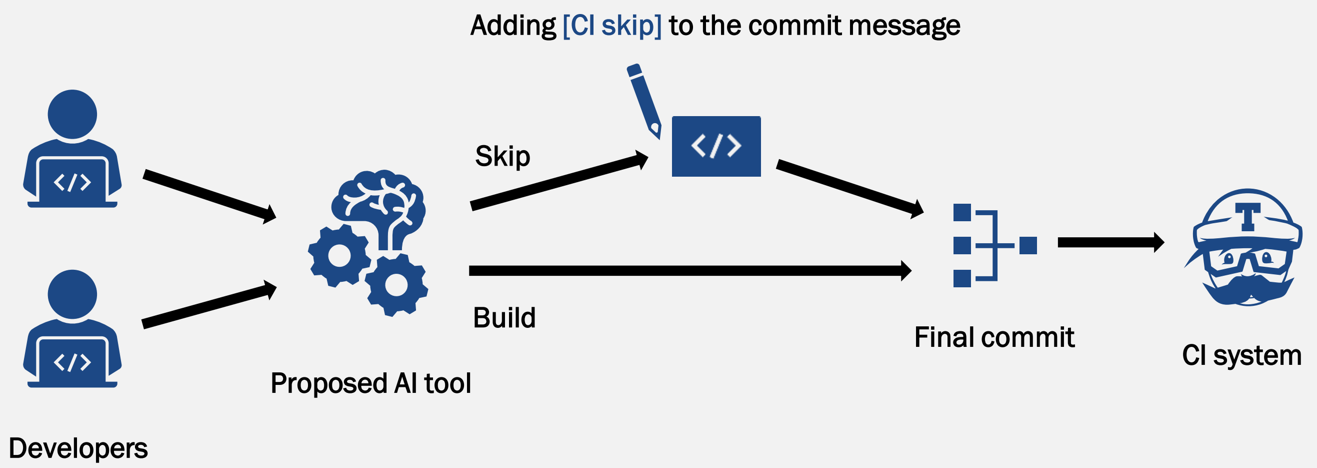
Goal

We aim to reduce the number of executed builds by **skipping** unnecessary ones such as:

- code reformatting
- documentation edits
- code comments modification

We model the problem as a **binary classification** task and aim to solve it using Machine Leaning.

Framework of the solution



II

Proposed Solution

1. Dataset
2. Decision Tree Model
3. Deep Reinforcement Learning
4. Solution Details

Proposed Solution
Dataset

Dataset consisting of the commit history of several open-source software projects using CI that have a significant number of skipped commits.

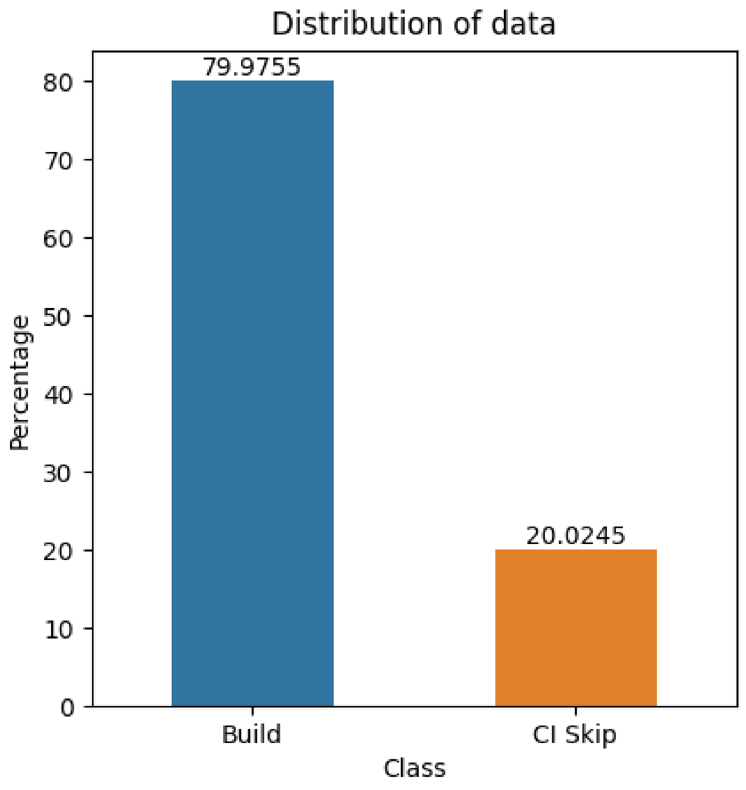
Statistics about the commit	Commit purpose	Link to previous commit
Number of subsystems Number of directories Number of files Number of lines added Number of lines removed Number of comments added Types of modified files Commit message	Fixing commit Documentation commit Building commit Meta-files commit Merging commit Media commit Source commit Formatting commit Maintenance commit	Project’s number of recent skips Committer's number of recent skip Previous commit result Age Number of developers Developer’s experience Sub-system experience Committer's recent experience

Proposed Solution
Dataset

Data Imbalance

The dataset is heavily imbalanced.
- A majority of commits are executed.
- A minority of commits are CI skipped.

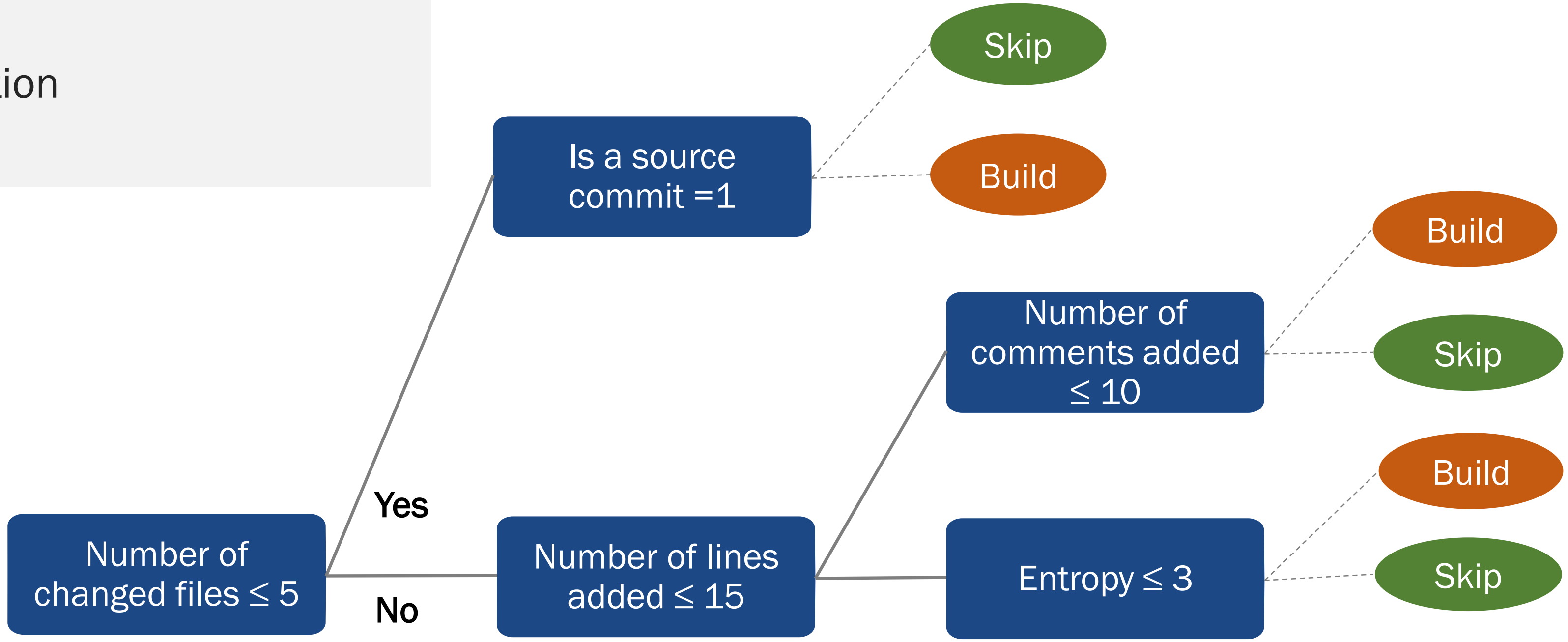
Developers are not familiar with the option to CI skip commits.



Proposed Solution
Decision Tree Model

We employ the **Decision Tree** model to predict which commits to CI skip.

- Quick results
- Interpretable classification



Proposed Solution

Decision Tree Building Algorithm

Building a Decision Tree using the Gini Index

```
for each tree node do
    for each feature do
        calculate the Gini index for all thresholds
        calculate the average Gini index
        calculate the Gini gain
    end for
    select the feature with the highest Gini gain
    set node
    split dataset
end for
```

Gini Index: $G(s) = 1 - \sum_{c \in C} -p(c)^2$

Gini Gain: $GG(s) = G(s) - \sum_{t \in T} p(t)G(t)$

Low Gini index due to data imbalance.

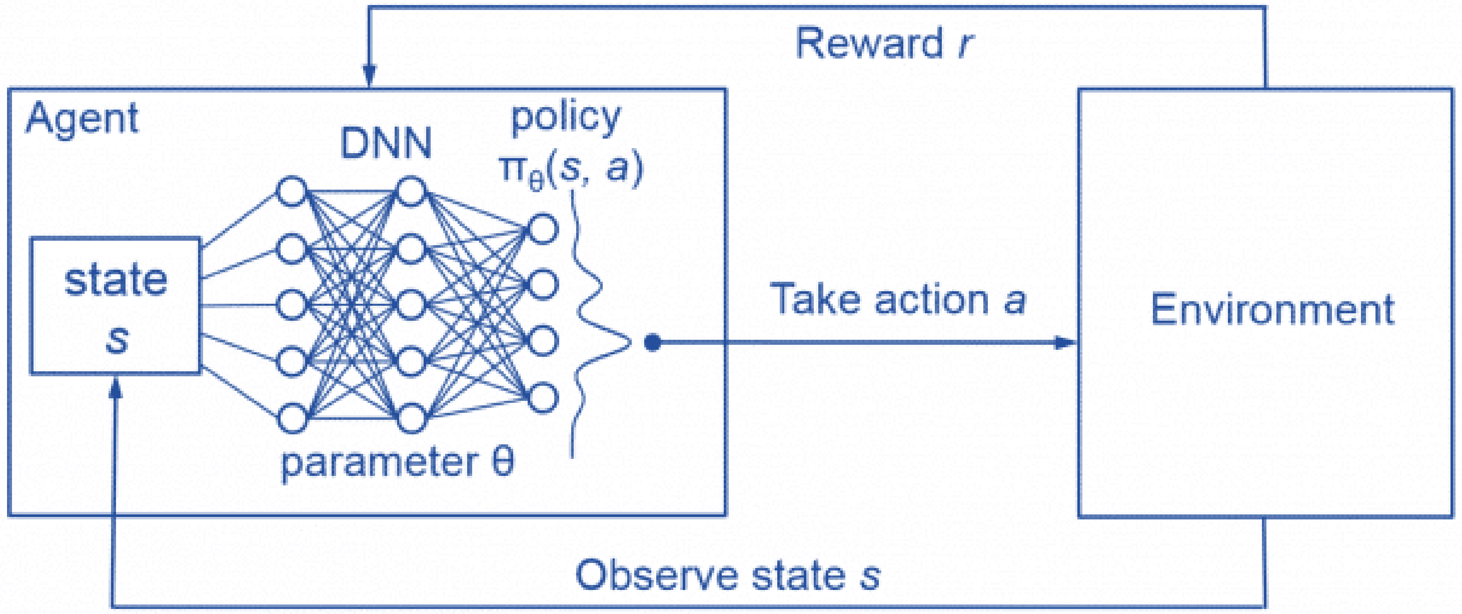
Proposed Solution

Our Approach

Proposed Solution

To solve this problem, we implement a novel deep reinforcement learning based algorithm to build decision trees that take into consideration the imbalanced nature of data.

Proposed Solution
Deep Reinforcement Learning

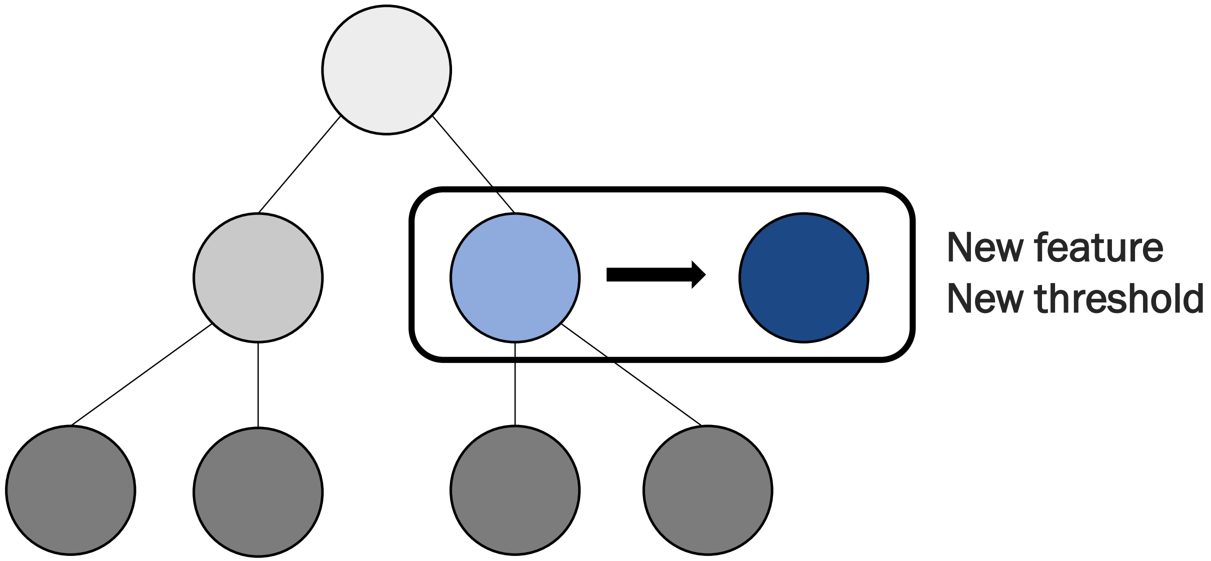


Proposed Solution

Solution Overview

Solution's Principle

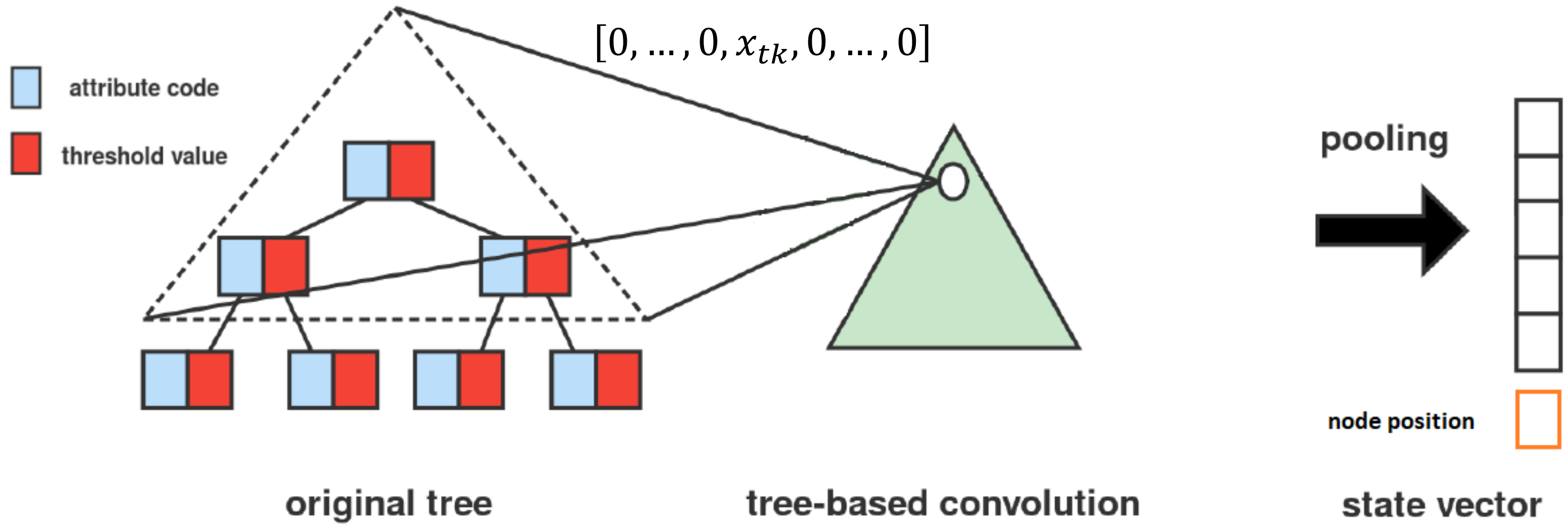
We iterate through the tree's nodes and adjust them so that the classification metric is improved.



Proposed Solution
State Observation

State Observation

The state is extracted using tree-based convolution.



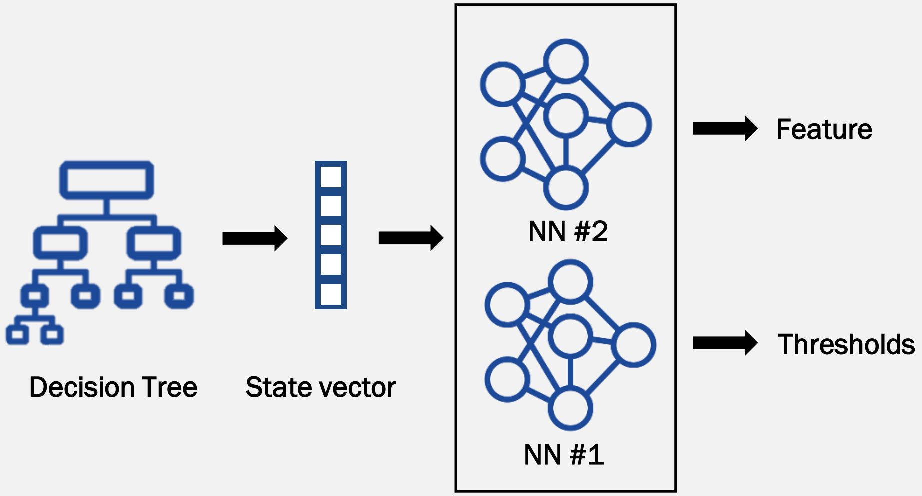
Proposed Solution
Agent's Actions

Agent's Actions

Each episode starts by generating an initial decision tree.

At each step, we choose an action $a_k = (k, x_k)$ for a single node.

- k is the feature
- x_k is the threshold value



Proposed Solution

Reward Function

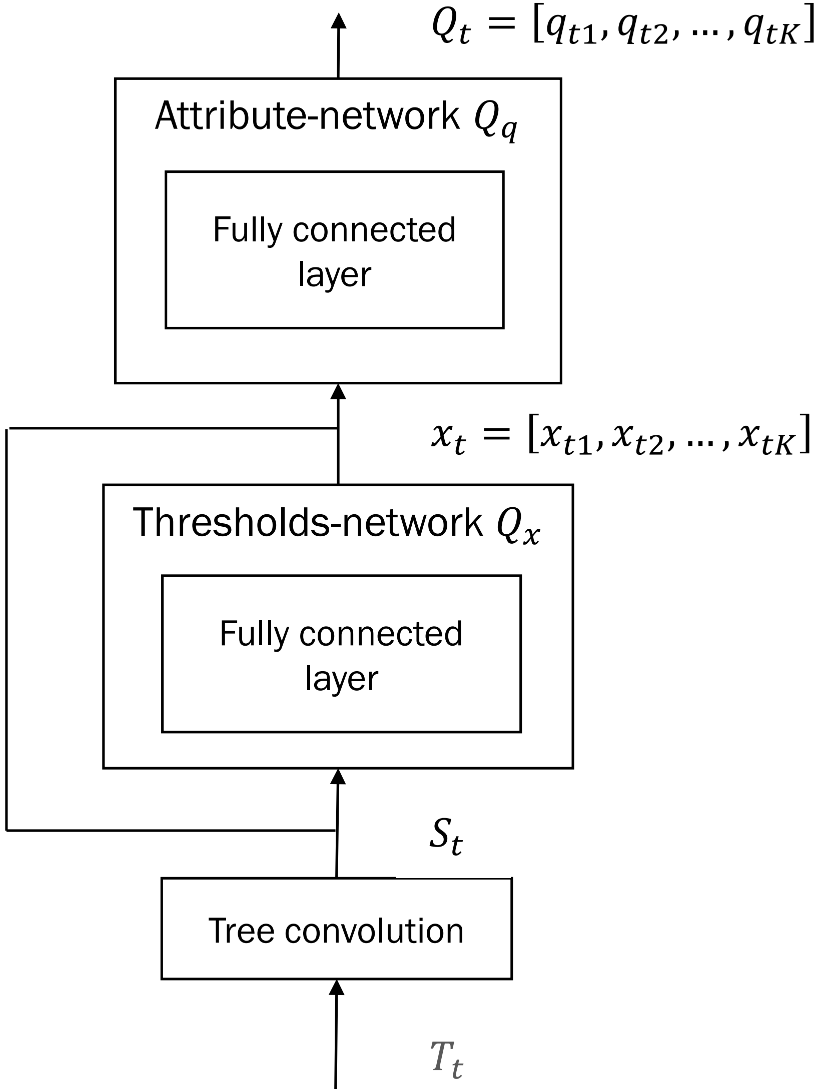
Reward Function

After each node modification t, we classify the dataset using the new tree.

The predicted results $\hat{Y}_t$ and the ground truth Y_t are used to calculate a classification metric m_t.

$$r_t = m_t - m_{t-1}$$

Proposed Solution
Training Process



Choose attribute K_t

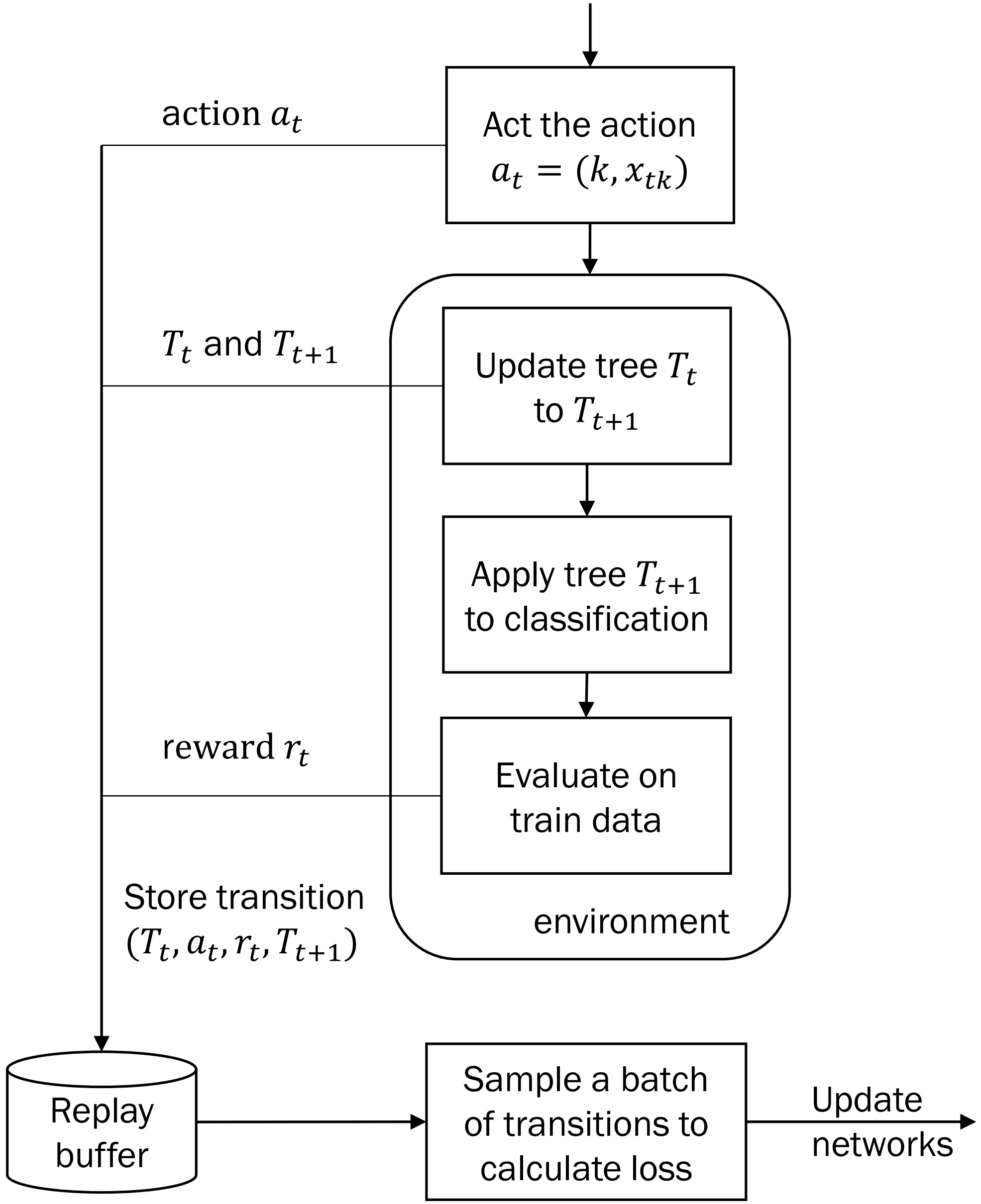
$$k_t = arg\max Q_t$$

$$= \underset{k \in [K]}{arg\max}\, Q_q\,(S_t, X_t; \theta_q)$$

Choose action

$$a_t = \begin{cases} (k_t, x_{tk}) \text{ with probability } 1 - \varepsilon \\ \text{random action with probability } \varepsilon \end{cases}$$

Proposed Solution
Training Process



III

Evaluation

1. Evaluation Methodology
2. Compared Techniques
3. Results

Evaluation

Evaluation Methodology

Classification Metrics

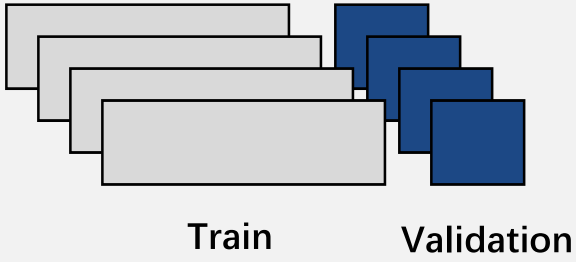
		Predicted Class	
		Positive	Negative
Actual Class	Positive	**TP**	**FN**
	Negative	**FP**	**TN**

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN}$$

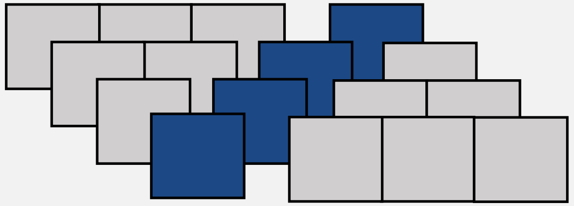
$$F1 = 2\,\frac{precision \cdot recall}{precision + recall}$$

$$AUC = \frac{1 + \frac{TP}{TP + FN} - \frac{FP}{FP + TN}}{2}$$

Validation Data



Within-project validation



Cross-project validation

Evaluation
Compared Techniques

Machine Learning

We also compare our solution to:

- Decision Tree built using the Gini index.
- Random Forest

Genetic Algorithm

This approach evolves a population of IF-THEN rules to find one with optimal performance.

Evaluation
Results

Within-project validation								
Metric	F1 score				AUC			
Project \ Method	**Ours**	GAR	DT	RF	**Ours**	GAR	DT	RF
Candybar-library	**0.79**	1	0.75	0.72	**0.71**	1	0.49	0.47
Pghero	**0.6**	0.85	0.58	0.77	**0.71**	0.92	0.72	0.85
Mtsar	**0.71**	0.88	0.51	0.55	**0.9**	0.91	0.63	0.66
Steve	**0.36**	0.62	0.28	0.21	**0.6**	0.82	0.61	0.56
SemanticMediaWiki	**0.49**	0.45	0.24	0.04	**0.65**	0.69	0.54	0.5

Evaluation
Results

Cross-project validation								
Metric	F1 score				AUC			
Project \ Method	**Ours**	GAR	DT	RF	**Ours**	GAR	DT	RF
Candybar-library	**0.62**	0.92	0.48	0.53	**0.7**	0.86	0.47	0.61
Pghero	**0.6**	0.8	0.44	0.47	**0.76**	0.68	0.64	0.65
Mtsar	**0.54**	0.68	0.41	0.37	**0.59**	0.72	0.58	0.59
Steve	**0.44**	0.52	0.2	0.16	**0.68**	0.75	0.57	0.54
SemanticMediaWiki	**0.33**	0.41	0.27	0.19	**0.59**	0.64	0.54	0.53

IV Conclusion

Conclusion

Good Performance

Average F1 score of 60% for within-project validation and 50% for cross-project validation.

Interpretable Results

Developers will be presented with an explanation for the CI skip decision.

Adaptable Solution

The model will retain its knowledge and adapt to more information and to a shift in development focus.

Future Work

Automatically selecting optimal hyper-parameters. (tree depth)

THANK YOU FOR YOUR ATTENTION

Any questions?

Graduation Project Defense

Optimizing Continuous Integration Using Artificial Intelligence

University of Carthage
École Polytechnique de Tunisie

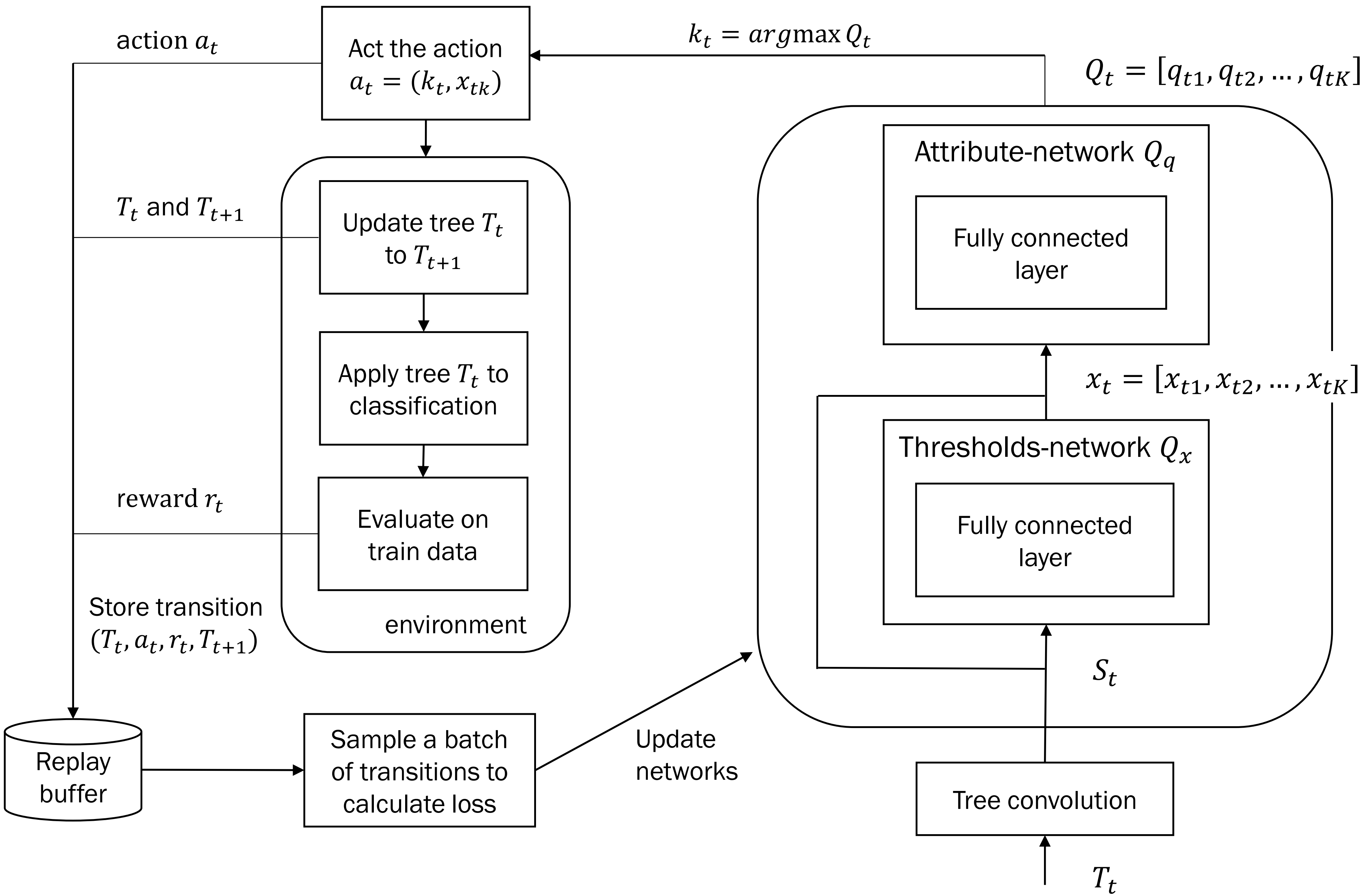
Host Organization UNIVERSITÉ LAVAL

Elaborated by **Maher Dissem**

Supported on July 4th, 2022, before the examination board:

Dr.	Mohamed Hamdi	Associate Professor – SUP'COM	President
Dr.	Ramzi Guetari	Associate Professor – EPT	Rapporteur
Prof.	Naceur Benhadj Braiek	Professor - EPT	Academic Supervisor
Dr.	Mohamed Aymen Saied	Assistant Professor - ULaval	Supervisor
Mr.	Mohamed Mouine	Ph.D. Student - ULaval	Supervisor

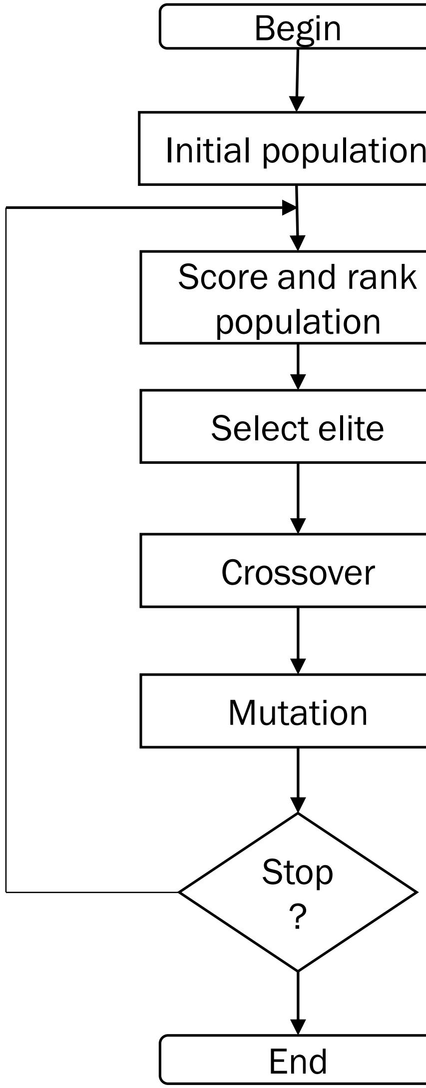
Academic year: 2021-2022



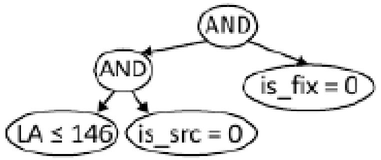
action a_t
Act the action $a_t = (k_t, x_{tk})$
$k_t = arg\max Q_t$
$Q_t = [q_{t1}, q_{t2}, \dots, q_{tK}]$
T_t and T_{t+1}
Update tree T_t to T_{t+1}
Apply tree T_t to classification
reward r_t
Evaluate on train data
environment
Store transition (T_t, a_t, r_t, T_{t+1})
Attribute-network Q_q
Fully connected layer
$x_t = [x_{t1}, x_{t2}, \dots, x_{tK}]$
Thresholds-network Q_x
Fully connected layer
S_t
Replay buffer
Sample a batch of transitions to calculate loss
Update networks
Tree convolution
T_t

Evaluation

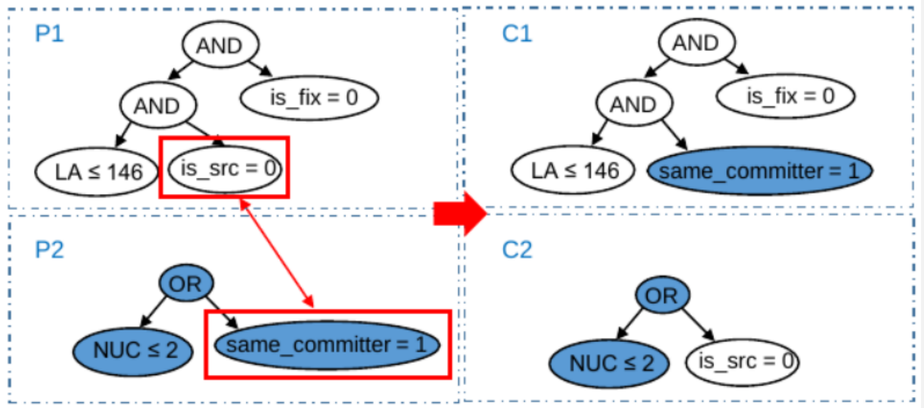
Evolutionary Search Approach

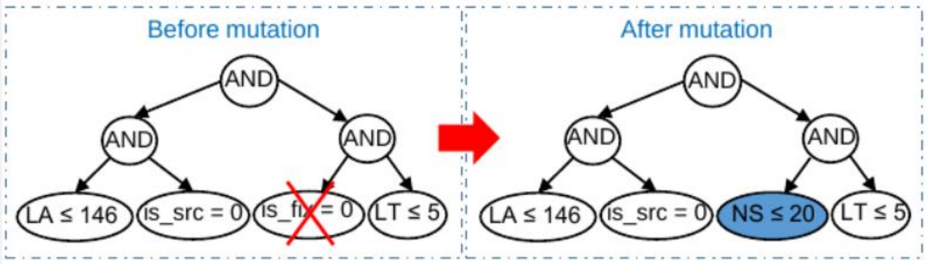


IF LA $\le$146 **AND** IS_SRC =0 **AND** IS_FIX =0
THEN Skip commit.

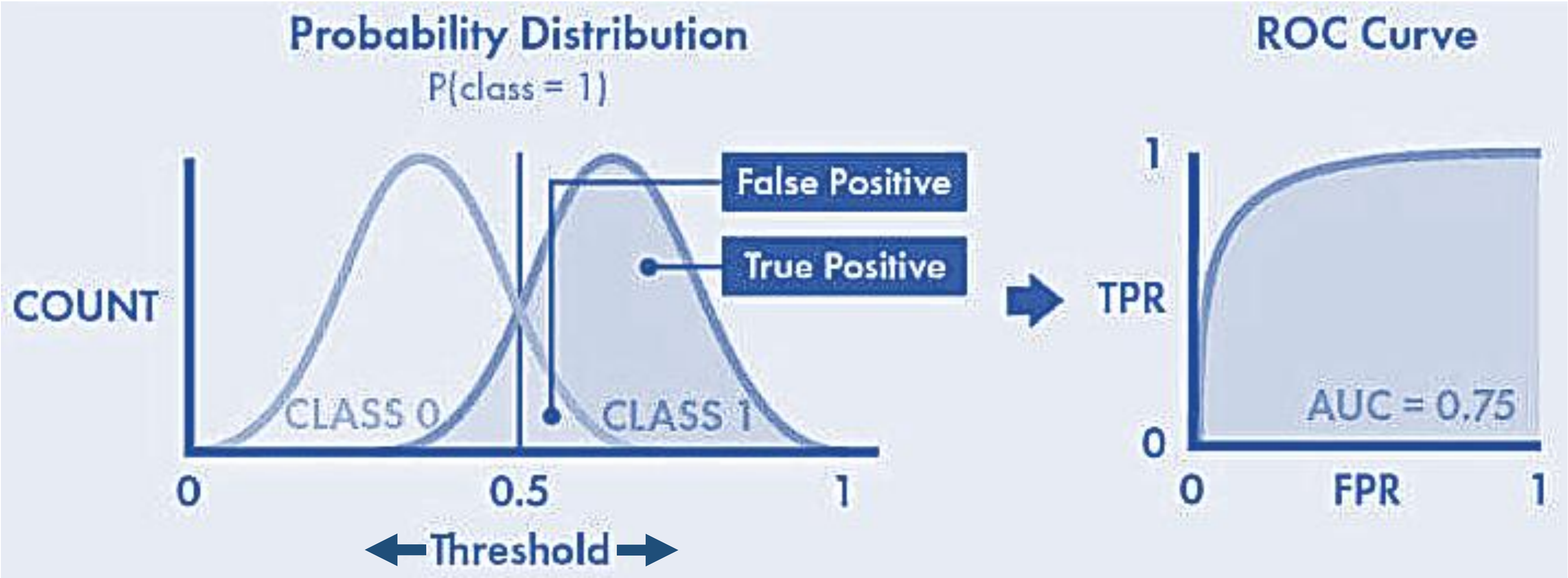


Fitness: $min_{i=1}^{n}\sqrt{FPR_i^{\,2} + (1 - TPR_i)^2}$

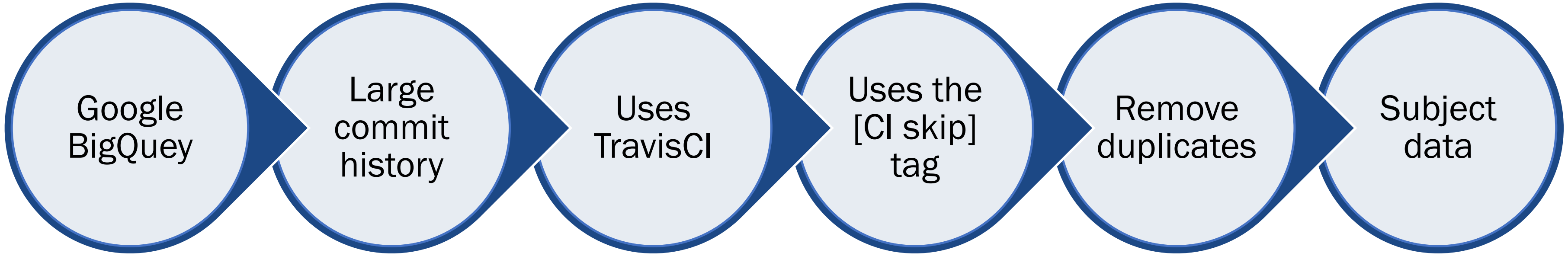




Evaluation
Classification Metrics



Proposed Solution
Dataset



Data filtering steps