

Інформаційна технологія виявлення дублікатів програмного коду

Виконав:

студент групи 1-КН-18М

Никуляк А.В.

Керівник:

к.т.н проф.

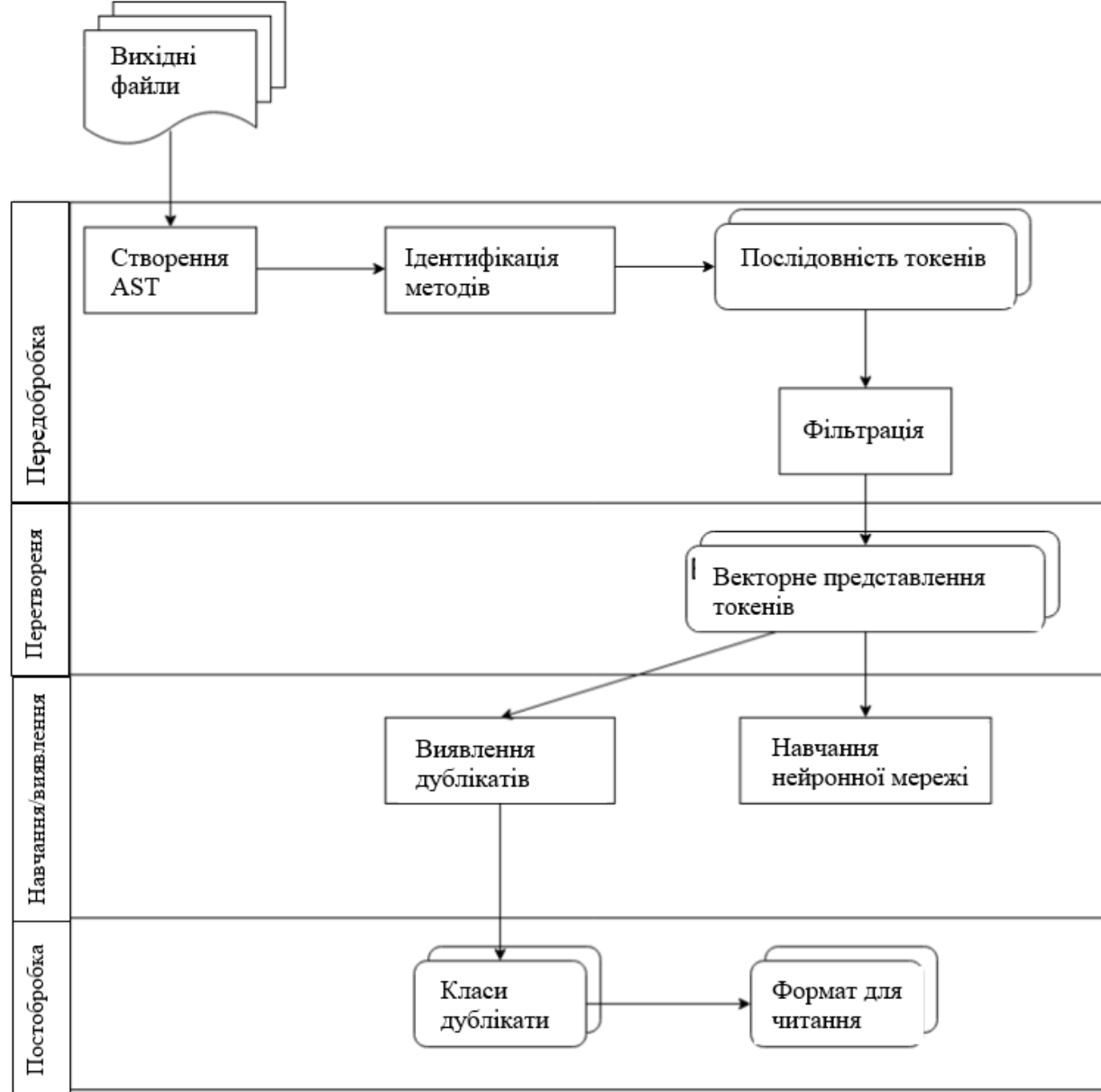
Месюра В.

Інформаційна технологія виявлення дублікатів програмного коду

- **Метою** магістерської кваліфікаційної роботи є підвищення точності виявлення дублікатів програмного коду за рахунок розробки інтелектуального методу
- **Об'єкт дослідження** – процес виявлення дублікатів програмного коду.
- **Предмет дослідження** – методи виявлення дублікатів програмного коду та їх властивості.

Типи дублікатів

Вихідний код	Тип-1	Тип-2
<pre>int main() { int x = 1; int y = x + 5; return y; }</pre>	<pre>int main() { int x = 1; int y = x + 5; return y;//output }</pre>	<pre>int func2() { int p = 1; int q = p + 5; return q; }</pre>
Тип-3	Тип-4	
<pre>int main() { int s = 1; int t = s + 5; t = t/++s; return t; }</pre>	<pre>int func4() { int n = 5; return ++n; }</pre>	

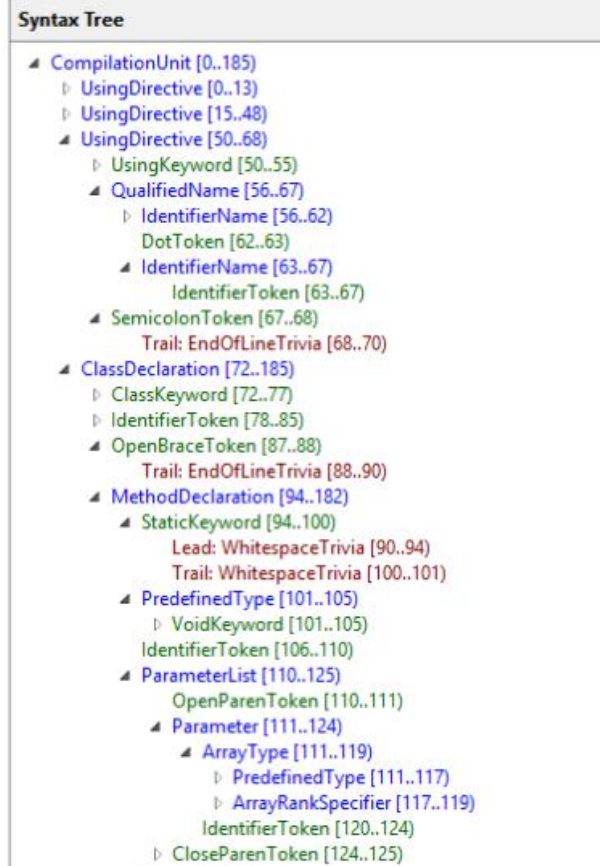


Абстрактне синтаксичне дерево

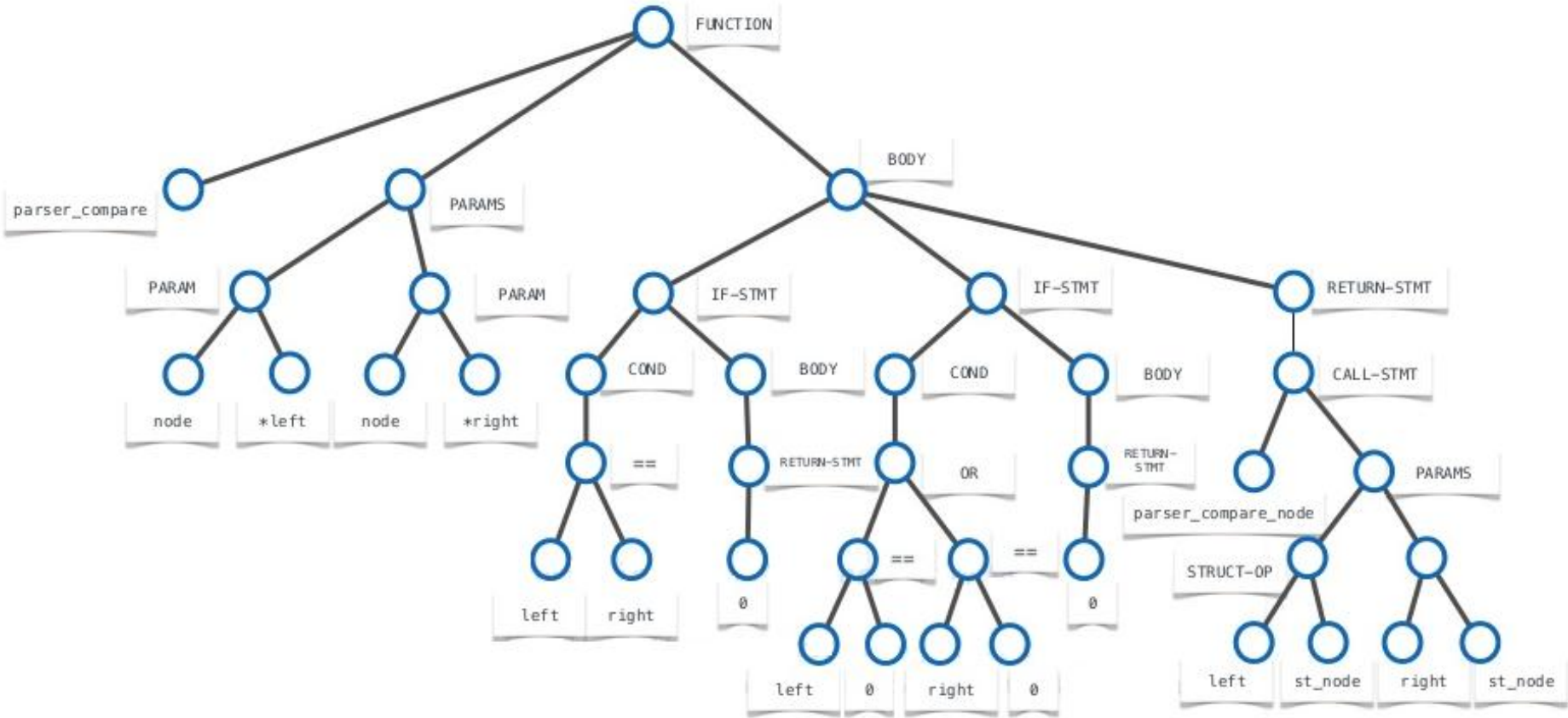
```
static int
parser_compare(PyST_Object *left, PyST_Object *right)
{
    if (left == right)
        return (0);

    if ((left == 0) || (right == 0))
        return (-1);

    return (parser_compare_nodes(left->st_node, right->st_node));
}
```



Абстрактне синтаксичне дерево



Рекурентні нейронні мережі

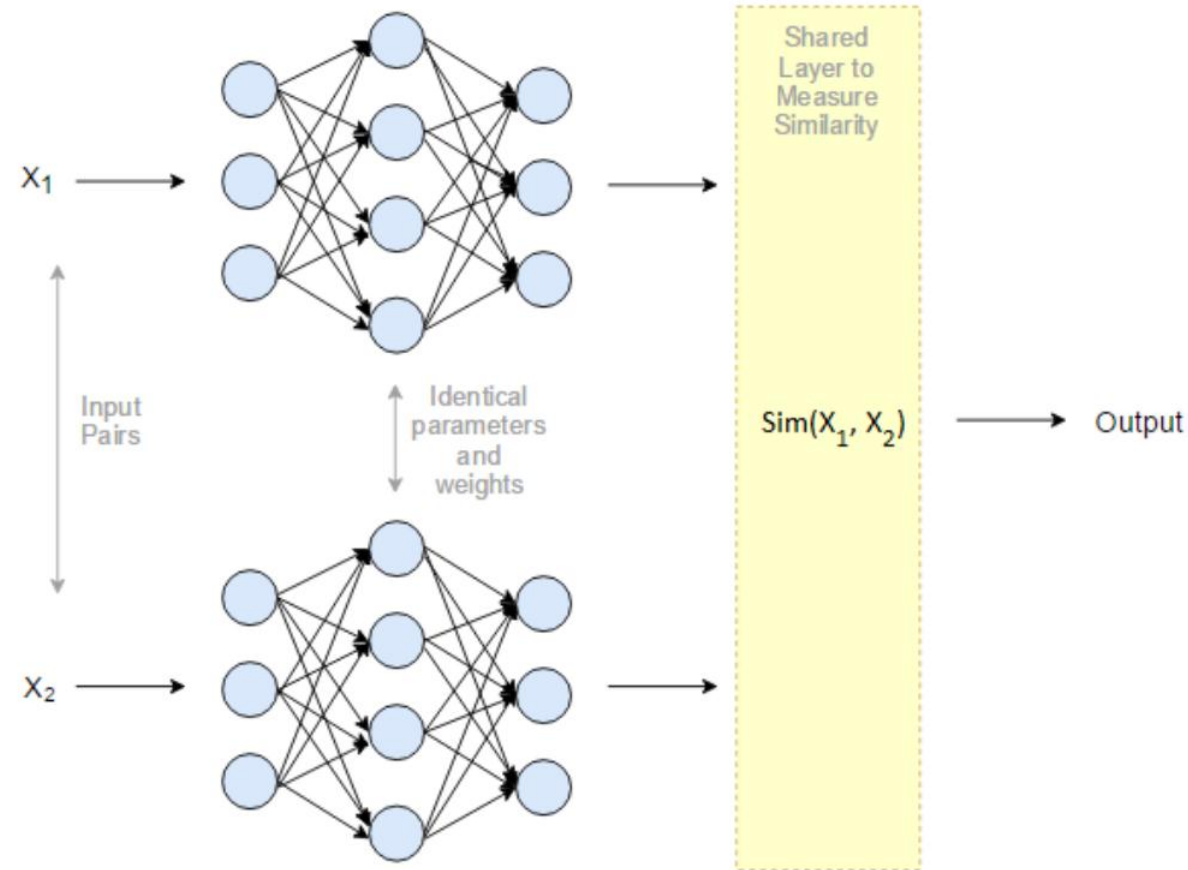
- **Рекурентні нейронні мережі** — це тип нейронних мереж, в яких є зворотний зв'язок. При цьому під зворотним зв'язком мається на увазі зв'язок від логічно найвіддаленішого елемента до менш віддаленого. Наявність зворотних зв'язків дозволяє запам'ятовувати і відтворювати цілі послідовності реакцій на один стимул.

Цільова функція нейронної мережі

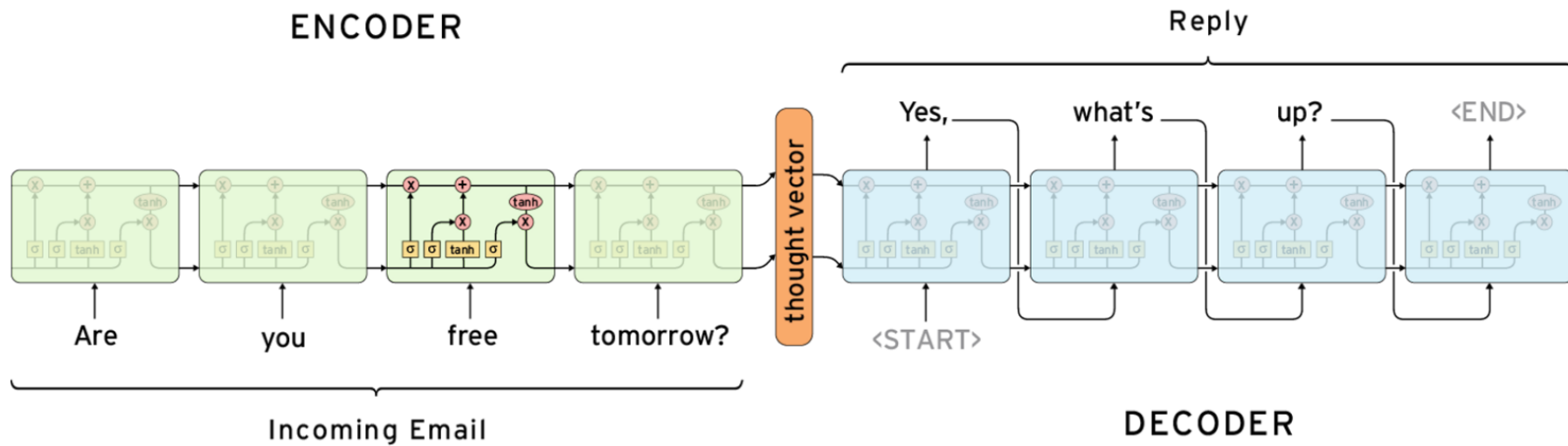
$$L(w) = \frac{\lambda}{2} \|w\|_2^2 + \frac{1}{2N} \sum_{(i,j) \in D} y_{ij} d_{ij}^2 + (1 - y_{ij}) \max(1 - d_{ij}^2, 0) \quad (2.9)$$

- де w - ваги нейронної мережі,
- D - набір навчальних пар,
- d_{ij}^2 - квадратична відстань
- l_2 відстань між i і j послідовностями (розраховане між двома останніми шарами сіамської ШНМ),
- $y_{ij} \in 0, 1$ – як і було розглянуто раніше, значення відповідає за схожість або відмінність послідовностей.

Архітектура нейронної мережі

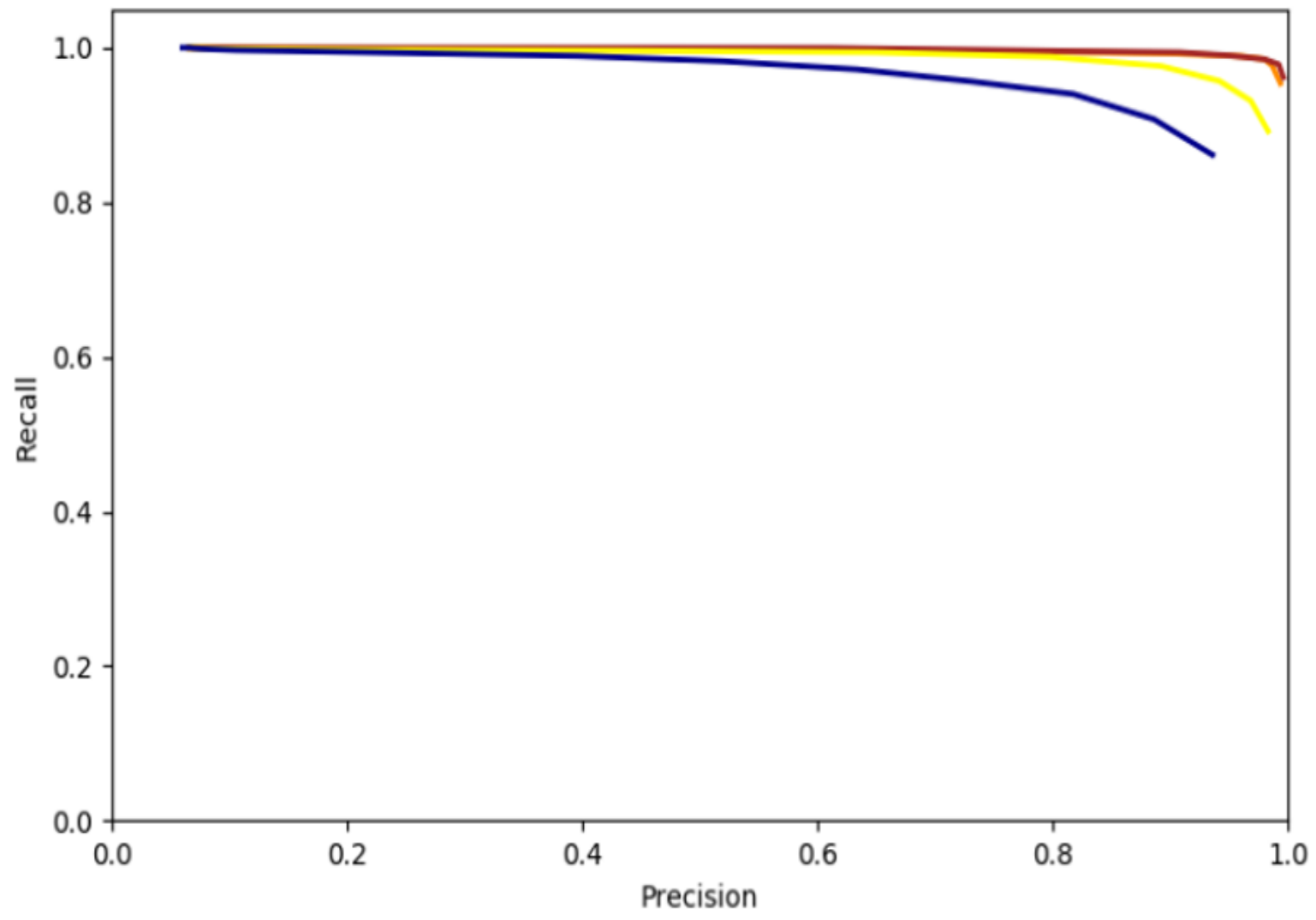


Модель Sequence-to-Sequence



Результати тестування методів виявлення дублікатів

Метод	BigCloneBench			OJClone		
	P	R	F ₁	P	R	F ₁
Deckard	0.83	0.91	0.86	0.79	0.86	0.82
CDLH	0.85	0.80	0,82	0.71	0.67	0.68
SourcererCC	0.76	0.82	0.78	0.63	0.74	0.68
AST+RNN	0.94	0.96	0.95	0.94	0.95	0.94



Програмна ралізація

Code Duplicates Detector

File Edit Run

GeneticMutator.cs

```
160     case TokenType.Multiplicative:
161         var additive = _random.Next( 2 ) == 0 ? Tok
162         return new BinaryNode( additive, leftMutate
163
164     default:
165         throw new NotSupportedException( nameof( nc
166     }
167 }
168
169 for (var j = 0; j < generationPopulation; ++j)
170 {
171     var (first, second) = TournamentSelection(bestKeeper
172
173     var mutatedNode = _geneticMutator.MutateNodeWithCrc
174     (
175         first.Expression,
176         first.Size,
177         second.Expression,
178         second.Size
179     );
180     var (optimized, count) = _optimizer.MutateNodeWithC
181
182     var difference = _computer.ComputeGeneticDifference
183     _bestKeeper.Submit(new ComputedNodeResult(optimized
184 }
185
186 if( _random.NextDouble() <= FullMutationProbability )
187     return FullMutation();
188
189 return base.Mutate( node );
```

StringTokenizer.cs

```
52 var blockComment = false;
53
54 while( lineComment || blockComment || char.IsWhiteSpace(
55     || c == '/' && !IsEnd
56         && ((lineComment = Peek() == '/') || (blc
57 {
58     if( IsEnd ) return CurrentToken = blockComment ? Tok
59     if( !lineComment && !blockComment ) continue;
60
61     for (var j = 0; j < generationPopulation; ++j)
62     {
63         var (first, second) = TournamentSelection(bestKes
64
65         var mutatedNode = _geneticMutator.MutateNodeWithC
66         (
67             first.Expression,
68             first.Size,
69             second.Expression,
70             second.Size
71         );
72         var (optimized, count) = _optimizer.MutateNodeWit
73
74         var difference = _computer.ComputeGeneticDifferen
75         _bestKeeper.Submit(new ComputedNodeResult(optimi:
76     }
77
78     c = Read();
79     if( c == '\n' ) lineComment = false;
80     if( c != '*' || Peek() != '/' ) continue;
81     Forward();
82     blockComment = false;
```

Code

- ▲ C:\Users\Admin\source\repos\DuplDec\AbstractSyntaxTree.Genetic (3)
 - ▲ AbstractSyntaxTree.Analyzer (1)
 - StringTokenizer.cs (1)
 - ▲ AbstractSyntaxTree.Genetic (1)
 - GeneticMutator.cs (1)

Висновки

- Підвищення точності виявлення на 12%
- Підвищення повноти виявлення на 10%

в порівнянні з існуючими програмними реалізаціями методів

Дякую за увагу