Зборник радова Факултета техничких наука, НовиСад

УДК: 007:004.056
ДОИ: https://doi.org/10.24867/35OI02Ijacic

Емпиријска анализа ефикасности статичке анализе и АИ модела у детекцији плагијата програмског кода

An Empirical Study of the Effectiveness of Static Analysis and AI Models for Source Code Plagiarism Detection

Ирена Ијачић, *Факултет техничких наука, Нови Сад*

Студијски програм – ИНЖЕЊЕРСТВО ИНФОРМАЦИОНИХ СИСТЕМА

Кратак садржај – *Овај рад представља емпиријску анализу ефикасности статичке анализе и анализе засноване на вјештачкој интелигенцији у детекцији плагијата програмског кода. У раду је описана употреба сопственог рјешења за статичку анализу, као и избор великог језичког модела DeepSeek у облаку за потребе детекције плагијата. Рад наглашава важност развоја алата за провјеру плагијаризма. Поред тога, истичу се предности и ограничења сваке од поменутих анализа у зависности од типа измјена у коду. На основу добијених резултата предложени су правци даљег развоја, укључујући хибридни приступ који комбинује оба поменута приступа.*

Кључне речи: *детекција плагијата, анализа изворног кода, статичка анализа, АИ модели*

Abstract – *This paper presents an empirical analysis of the effectiveness of static analysis and artificial intelligence-based analysis in detecting plagiarism in source code. The study describes a custom solution for static analysis, as well as the selection of the large language model DeepSeek in the cloud for plagiarism detection purposes. The paper emphasizes the importance of developing tools for plagiarism verification. Furthermore, the advantages and limitations of each analysis method are highlighted depending on the type of modifications in the code. Based on the obtained results, directions for further development are proposed, including a hybrid approach that combines both methods.*

Keywords: *plagiarism detection, source code analysis, static analysis, artificial intelligence models*

НАПОМЕНА: Овај рад проистекао је из мастер рада чији ментор је био др Владимир Мандић, ванр. проф.

1. УВОД

Плагијат изворног кода постаје све већи проблем као резултат наглог раста онлајн репозиторијума и све веће доступности дигиталних докумената, што у великој мјери утиче на квалитет образовања у области рачунарства. Када се помене појам плагијата у коду не мисли се само на директно копирање туђег програма, већ често укључује и дјелимичне измјене тог кода, као што су промјене имена промјењивих, редослиједа функција или структуре програма. Овакве измјењене верзије кода представљају изазов за откривање. Претходне студије показују да плагијат обухвата читав низ стратегија за преузимања туђег кода, што укључује опсег од минималних површинских измјена до значајних прилагођавања, при чему основна идеја оригиналног програма остаје задржана. [1] Временом, истраживачи су се бавили темом провјере плагијата у коду, обрађујући различите врсте клонова који се могу јавити и развијајући одговарајуће алате који ће исте препознати. Значајан број постојећих рјешења се ослања на статичку анализу кода, гдје се комбинацијом познатих метрика врши процјена система или његових компоненти на основу њихове структуре, садржаја или документације. Поред статичке анализе, напредак у области вјештачке интелигенције омогућио је примјену АИ модела у детекцији плагијата програмског кода. Циљ овог рада је компаративна анализа метода статичке анализе и анализа заснованих на вјештачкој интелигенцији у откривању плагијата изворног кода.

2. ТЕОРИЈСКЕ ОСНОВЕ И СРОДНИ РАДОВИ

У овом поглављу представљене су теоријске основе појмова на којима се рад заснива.

2.1. Појам плагијата кода

Сам појам плагијата има више различитих дефиниција. Научници су описивали овај појам на различите начине, али увијек суштина дефиниције остаје иста, а то је да плагијат уствари представља копирање и измјену туђег кода без адекватног навођења његовог аутора.

Издвајају се четири основна типа плагијата која су посебно битни за разумијевање и детекцију плагијата кода. Типови клонова су [2]:

- Тип I: Дијелови кода су потпуно исти, осим евентуалних промјена у размацима и коментарима.
- Тип II: Структура дијелова кода је иста, али се разликују називи идентификатора, типови података, размаци и коментари.

- Тип III: Код овог типа, поред промјена у идентификаторима, називима промјењивих, типовима података и коментарима, поједини дијелови кода могу бити обрисани или измјењени, или се могу додати нови дијелови кода.
- Тип IV: Два дијела кода имају различит текст, али остварују исту функционалност.

2.2. Статичка анализа кода

Статичка анализа кода представља процес процјене система или његових компоненти на основу њихове структуре, садржаја или документације. Ова анализа пружа могућност доношења закључка о томе како се понаша сам програм, а да се тај исти програм не мора претходно извршити. Када се говори о области детекције плагијата у коду, статичка анализа пружа могућност поређења два програма у погледу њихове структуре, синтаксе и семантичких елемената, како би се утврдило да ли је код копија неког другог рада. У том случају, статичка анализа може бити изражена као комбинација више познатих метрика које ће покрити различите области провјере.

2.3. Појам вјештачке интелигенције

Постоји више различитих дефиниција за појам вјештачке интелигенције. Већина практичара се слаже око два главна циља АИ, а то су изградња интелигентне машине и сазнавање о природи интелигенције. Једна од кључних методологија вјештачке интелигенције је дубоко учење (енгл. *Deep Learning*), а дефинише се као техника која тренира вјештачке неуронске мреже на огромним скуповима података у облику обиљежених података (енгл. *labelled data*). Дубоко учење је основ за задатке великих језичких модела [3]. Неки од најпознатијих језичких модела су: GPT-3, GPT-3.5, GPT-4, BARD, DeepSeek.

2.4. Преглед постојећих истраживања

На основу извршеног прегледа релевантне литературе може се закључити да и статичке методе анализе и методе засноване на вјештачкој интелигенцији представљају ефикасна рјешења у случајевима када се плагијат налази у површинским измјенама кода. Међутим, уколико су извршене значајније измјене на нивоу логике или структуре кода, предност се даје приступима заснованим на АИ. Ови приступи у таквим условима показују већу прецизност, бољи одзив и укупно боље резултате. С друге стране, статичка анализа има предност када је ријеч о потребним рачунарским ресурсима за анализу и брзину њеног извршавања, нарочито при обради великих скупова података. Ове карактеристике чине је бољом опцијом за специфичне случајеве у којима су перформансе и скалабилност најважнији.

3. МЕТОДОЛОГИЈА И ЕКСПЕРИМЕНТАЛНИ ОКВИР

У оквиру ове секције описани су кораци спроведене методологије.

3.1. Циљ истраживања и истраживачка питања

Циљ овог рада је евалуација два технолошка приступа за потребе детекције постојања плагијата у контексту провјере студентских радова. Допринос рада се огледа у компаративној анализи два различита приступа детекције на истом скупу података, што може пружити корисне податке за образовне институције и бројне развојне тимове. На основу постављеног истраживачког циља могу се формулисати следећа истраживачка питања:

1. Која метода (статичка анализа или анализа заснована на вјештачкој интелигенцији) даје прецизније резултате и већи одзив у откривању плагијата изворног кода у поређењу са познатом референтном вриједношћу?
2. Како се резултати статичке и АИ анализе разликују у случајевима када су измјене у коду површинске (на примјер промјена имена промјењивих или формата) у односу на случајеве који укључују дубље семантичке измјене (на примјер измјене у алгоритамској логици)?
3. Која метода је временски ефикаснија и једноставнија за имплементацију и примјену на великим колекцијама задатака?
4. У којој мјери су резултати сваке методе транспарентни и интерпретабилни крајњем кориснику (на примјер наставницима или истраживачима)?

Полазна претпоставка истраживања је да између статичке анализе и АИ заснованог приступа не постоје значајне разлике у ефикасности детекције плагијата, које се могу квантификовати кроз F1-score метрику.

3.2. Скуп података

За потребе истраживања коришћени су радови студената прикупљени у оквиру курса „Алгоритми и структуре података“, на смјеру Инжењерство информационих система Техничког факултета у Новом Саду. Радови се организују у сетове, тако да сваки сет садржи референтни рад, као и друге радове који служе за поређење и провјеру.
Међу поменутим радовима, издвајају се три типа примјера над којима је вршена анализа, а то су:

1. Студентски радови на исту тему - радови који се односе на исти задати алгоритам.
2. Сет контролних примјера – ови радови нису написани од стране студената, већ се ради о ауторским радовима који служе као референца (енгл. *ground truth*) јер је познато шта је оригинал, које модификације су извршене, односно о каквој врсти измјене се ради.
3. Студентски радови на различиту тему коришћени за тестирање *false positive* детекције.

3.3. Примјењене методе

Статичка анализа кода се користи као један од приступа за детекцију плагијата у коду. Циљ ове анализе јесте идентификација степена сличности између програмских кодова на основу њихове структуре, синтаксе и редослиједа операција. Иако је статичка анализа примарно усмјерена ка откривању површинских измјена у коду, комбинацијом различитих метрика могуће је креирати анализу која ће препознати и дубље измјене. Креиран је сопствени код за статистичку анализу који комбинује и рачуна више различитих метрика, а то су *Exact Match*, *Token Similarity*, Анализа структуре апстрактног синтаксног стабла (ACT), *Sequence similarity* и *Winnowing Similarity*.
За сваку од ових метрика дефинисана је тежина, као што приказане Табела 1, а коначан степен сличности добија се као тежински просјек.

Табела 1. Тежине одабраних метрика статичке анализе

Мјера	Тежина
Exact Match	0.2
Token Similarity	0.1
AST Similarity	0.2
Sequence Similarity	0.2
Winnowing Similarity	0.3

Поред статичке анализе за детекцију плагијата између програмских кодова користи се и анализа заснована на вјештачкој интелигенцији, која омогућава аутоматизовану анализу са фокусом на логику и структуру кода. Дакле, оваквом анализом се ставља фокус на препознавање сличности и у случајевима дубоких семантичких измјена, за које је потребан већи степен разумијевања програмског кода. Анализа заснована на вјештачкој интелигенцији реализована је употребом великог језичког модела. Ови модели су истренирани и научени да препознају одговарајуће обрасце у коду и на основу тога доносе закључке. За потребе овог рада одабран је DeepSeek модел у облаку и то његова верзија *deepseek-v3.1:671b-cloud*.

3.4. Метрике за процјену

Precision представља удио исправно идентификованих релевантних случајева у односу на укупан број случајева које је систем означио као релевантне. Precision мјери колико је резултат чист од лажних позитива:

$$Precision = TP/(TP + FP) \quad (1)$$

Recall представља удио исправно идентификованих релевантних случајева у односу на укупан број стварно релевантних случајева у референтном скупу. Recall показује колико система пропушта релевантних измјена:

$$Recall = TP/(TP + FN) \quad (2)$$

F1-score је хармонијска средина прецизности и обухвата (Precision и Recall), и користи се као уравнотежена метрика када је важно узети у обзир и лажне позитивне и лажне негативне резултате. Дефинисан је као:

$$F1 - score = 2 \cdot (Precision \cdot Recall)/(Precision + Recall) \quad (3)$$

Постављена је нулта хипотеза (H_0) којом се наводи да је просјечна F1-score вриједност добијена примјеном статичке анализе једнака просјечној F1-score вриједности добијеној примјеном АИ анализе, односно:

$$H_0: \mu_{F1stat} = \mu_{F1AI} \quad (4)$$

Алтернативна хипотеза (H_1) полази од супротне претпоставке, да просјечне F1-score вриједности нису једнаке, те да једна од метода показује већу ефикасност у откривању плагијата.

4. РЕЗУЛТАТИ ЕКСПЕРИМЕНТАЛНОГ ИСТРАЖИВАЊА

У оквиру секције 4 приказани су резултати извршених анализа.

4.1. Резултати статичке анализе

Резултати метрика статичке анализе за свих десет сетова података приказује Табела 2. Уочава се да статичка анализа постиже максималну прецизност у свим сетовима, што указује на одсуство лажних позитивних детекција. Са друге стране, вриједности одзива се више мијењају током анализе, нарочито у случајевима дубљих семантичких измјена, што се одражава и на ниже F1-score вриједности.
Просјечно вријеме извршавања статичке анализе указује на високу временску ефикасност и стабилност извршавања.

Табела 2. Резултати статичке анализе

Статичка анализа			
Сет	Precision	Recall	F1-score
1	1	0.67	0.8
2	1	1	1
3	1	0.6	0.75
4	1	0.75	0.857
5	1	1	1
6	1	0.75	0.875
7	1	0.5	0.67
8	1	0.5	0.67
9	1	0.5	0.67
10	1	0.5	0.67

Табела 3 садржи резултате анализе засноване на АИ моделу. АИ анализа постиже високе вриједности и прецизности и одзива у готово свим сетовима података, што доводи до високих F1-score вриједности. За разлику од статичке анализе, АИ приступ задржава висок одзив и у случајевима дубљих семантичких

измјена кода. Просјечно вријеме извршавања АИ анализе значајно је веће у односу на статичку анализу.

Табела 3. Резултати анализе засноване на вјештачкој интелигенцији

АИ анализа			
Сет	Precision	Recall	F1-score
1	0.9	1	0.95
2	1	1	1
3	1	0.8	0.89
4	1	1	1
5	1	1	1
6	1	1	1
7	1	1	1
8	1	1	1
9	0.8	1	0.89
10	1	1	1

5. ДИСКУСИЈА

Кроз резултате спроведеног истраживања јасно се наводи да статичка анализа показује већу прецизност у поређењу са методом заснованом на вјештачкој интелигенцији. Ово указује да ова метода генерише мањи број лажно позитивних резултата, односно да је конзервативнија у означавању кода као плагираног. Са друге стране, анализа заснована на вјештачкој интелигенцији показује значајно већи одзив у односу на статичку анализу. Ови резултати указују да АИ методе успјешније идентификују већи број стварних случајева плагијата, али уз повећан ризик од лажно позитивних резултата. Статичка анализа показује већу осјетљивост на површинске измјене кода, али предност има у погледу времена извршавања, транспарентности и детерминистичности резултата. Резултати добијени овим истраживањем су у складу са постојећим научним истраживањима, уз изузетак прецизности, гдје је у овом истраживању утврђена предност статичке анализе. Примјена Wilcoxon-овог теста потврдила је постојање статистички значајне разлике у F1-score вриједностима између двије методе ($p < 0.05$), чиме се одбацује постављена нулта хипотеза.

6. ЗАКЉУЧАК

У оквиру овог рада обрађена је тема детекције плагијата између програмских кодова написаних у програмском језику C#. Резултати спроведеног истраживања показују да је статичка анализа показала предност у погледу тачности резултата и времена које је потребно за извршавање детекције, али се истовремено показала као недовољно ефикасна за детектовање дубљих измјена које не може поуздано да препозна. Супротно томе, анализа заснована на вјештачкој интелигенцији показује изузетне резултате управо у случају дубљих семантичких измјена кода. Изазови идентификовани током израде овог рада у највећој мјери односили су се на проналазак одговарајућег модела вјештачке интелигенције. Иако је област вјештачке интелигенције у процесу континуираног развоја, још увијек су присутна одређена ограничења у оквиру појединих примјена. Управо једна од таквих области јесте детекција плагијата у програмском коду, гдје постојећи модели не показују увијек задовољавајући ниво поузданости и прецизности. Додатан изазов при изради рада био је преглед постојеће научне литературе. Иако је ова тема тренутно изузетно актуелна, посебно у академским круговима гдје све већи број факултета и образовних институција укључује овакву врсту алата за провјеру студентских и ученичких радова, још увијек није написан довољан број релевантних научних радова који се баве овом проблематиком. Што се тиче доприноса спроведеног истраживања, он се прије свега огледа у развоју сопственог алгоритма за статичку анализу кода у сврху детекције плагијата, што је омогућило поређење статичке анализе и анализе засноване на вјештачкој интелигенцији кроз провјеру тачности, одзива, времена извршавања, степена транспарентности резултата, као и кроз провјеру детерминистичности спроведених анализа. Кроз поступак извођења овог истраживања, уочена су и одређена ограничења везана за постојеће АИ моделе у овој области, што оставља простор за будући рад. Правци даљег истраживања могу бити усмјерени ка развоју хибридног приступа који би комбиновао статичку анализу програмског кода и методе засноване на вјештачкој интелигенцији, како би се искористило најбоље од оба приступа.

7. ЛИТЕРАТУРА

[1] G. Cosma and M. Joy, “Towards a Definition of Source-Code Plagiarism,” *IEEE Transactions on Education*, vol. 51, no. 2, pp. 195–200, May 2008, doi: 10.1109/TE.2007.906776.

[2] D. Rattan, R. Bhatia, and M. Singh, “Software clone detection: A systematic review,” Jul. 2013. doi: 10.1016/j.infsof.2013.01.008.

[3] B. · January, “Learning Outcomes of Classroom Research.” [Online]. Available: https://www.researchgate.net/publication/358050961

Кратка биографија:



Ирена Ијачић, рођена 7. априла 2000. године у Требињу. Након завршене гимназије 2019. године уписала Факултет техничких наука, смјер Инжењерство информационих система. Основне академске студије завршила 2023. године са звањем „дипломирани инжењер информационих технологија“. **Контакт: irenaijacic17@gmail.com**