УНИВЕРЗИТЕТ У БЕОГРАДУ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ

Милош М. Котлар

ДЕТЕКЦИЈА АНОМАЛИЈА КОРИШЋЕЊЕМ МЕТА ПОДАТАКА У АУТОМАТИЗОВАНИМ СИСТЕМИМА ЗА МАШИНСКО УЧЕЊЕ

докторска дисертација

Београд, 2022

UNIVERSITY OF BELGRADE
SCHOOL OF ELECTRICAL ENGINEERING

Miloš M. Kotlar

ANOMALY DETECTION USING META FEATURES IN AUTOMATED MACHINE LEARNING SYSTEMS

doctoral dissertation

Belgrade, 2022

Комисија за преглед и оцену дисертације

Ментори:

Доц. др Марија Пунт
Доцент, Универзитет у Београду, Електротехнички факултет

Проф. др Захарије Радивојевић
Ванредни проф., Универзитет у Београду, Електротехнички факултет

Чланови Комисије:

Проф. др Милош Цветановић
Ванредни проф., Универзитет у Београду, Електротехнички факултет

Проф. др Драган Бојић
Редовни проф., Универзитет у Београду, Електротехнички факултет

Проф. др Синиша Влајић
Редовни проф., Универзитет у Београду, Факултет Организационих наука

Проф. др Горан Квашчев
Ванредни проф., Универзитет у Београду, Електротехнички факултет

Датум одбране: _____

Захвалница

Пре свега, желео бих да се захвалим свима који су током мог школовања учествовали у стицању знања и вештина потребних за истраживање, као и помогли приликом реализације идеја описаних у овом раду. Захваљујем се менторима, професорима др Марији Пунт и др Захарију Радивојевићу, као и др Милошу Цветановићу и др Вељку Милутиновићу јер су увек веровали у мене и својом сталном подршком ми помогли да истрајем. На крају највећу захвалност дугујем пријатељима и породици, који су ме инспирисали и допринели у креирању сопствених идеја, дали подршку, инспирацију, љубав, и били ту када је било потребно.

Наслов дисертације: Детекција аномалија коришћењем мета података у аутоматизованим системима за машинско учење

Резиме: Генерисање велике количине података условљено развојем крајњих уређаја (енг. *Edge Devices*) и интернет ствари (енг. *Internet of Things*) довело је до убрзаног развоја технологија и алгоритама за машинско учење који се користе у системима за анализу и обраду података. Са великом количином података у системима за њихову анализу и обраду, заснованим на алгоритмима за машинско учење, перформансе система искључиво зависе од квалитета података, одабраног модела и параметара модела. Аномалије у подацима представљају инстанце које се разликују од дистрибуције података, утичу на квалитет података и могу да се детектују коришћењем алгоритама за машинско учење. Предлог модела и параметара модела за детекцију аномалија искључиво зависи од експертизе креатора система или доменског експерта. У случајевима када не постоји узорак података са обележеним аномалијама, што је чест случај у подацима из реалног света, предлог модела за детекцију аномалија није тривијалан. Предлог модела за детекцију аномалија се може аутоматизовати, при чему такав систем за аутоматизовано машинско учење (енг. *AutoML*) предлаже модел за детекцију аномалија у подацима на основу података, мета података, одговарајуће оптимизационе метрике и претходно стеченог знања. Како би се омогућила имплементација аутоматизованог система за детекцију аномалија, потребно је дефинисати скуп функција за израчунавање мета података који ће се користити за предлагање модела за одговарајућу оптимизациону метрику.

Предмет истраживања представља развој проширивог система за израчунавање мета података. Идеја је да се систем за израчунавање мета података заснива на функцијама које користе доменско знање и испуњавају критичне захтеве за примену у системима за аутоматизовано машинско учење, а то су скалабилност и перформансе. Циљ истраживања је да се предложи скуп функција за израчунавање мета података који ће испуњавати критичне захтеве за наведену примену. Како би се предложио и евалуирао скуп функција за израчунавање мета података потребно је да се постојећа решења упореде кроз различите аспекте комплексности. Такође, потребно је да се дизајнирају експерименти и добију резултати који ће моћи да се користе у будућим истраживањима у области аутоматизованог машинског учења. На основу евалуације експерименталних резултата показано је да предложени мета подаци постижу тачност од 87% и да испуњавају критичне захтеве за примену у аутоматизованим системима за машинско учење, док постојећа решења постижу тачност од 73% над целим репозиторијумом. У ситуацијама када не постоји значајна количина скупова података предложено решење постиже и за 25% лошије перформансе. Значај истраживања представља могућност имплементације аутоматизованих система за детекцију аномалија заснованим на предложеном скупу функција за израчунавање мета података. У случајевима када не постоји узорак података са обележеним аномалијама, или подаци нису присутни, креатор података или доменски експерт ће моћи ефикасно да карактеризује аномалије у подацима на основу доменског знања.

Кључне речи: детекција аномалија, *automl*, карактеристике података, функције за мерење сличности, мета подаци, мета учење, пренос знања између модела

Научна област: Техничке науке - Електротехника и рачунарство

Ужа научна област: Рачунарска техника и информатика

УДК: 621.3:004.8

Disertation Title: Anomaly detection using meta features in automated machine learning systems

Abstract: Proliferation of data and devices led to the rapid development of technology and machine learning algorithms used in data analysis and processing systems. With a large amount of data in systems for their analysis and processing, system's performance depends solely on the quality of the data, the selected algorithm and the algorithm's parameters. Data anomalies are instances that differ from data distribution, affect data quality, and can be detected using machine learning algorithms. Selected algorithm and the parameters for anomaly detection depend exclusively on the expertise of the system creator or domain expert. In cases where there is no sample data with labeled anomalies, which is often the case in real-world, choosing right algorithm for anomaly detection is not trivial problem. Algorithm selection for anomaly detection tasks can be automated by using automated machine learning system (*AutoML*) that proposes an algortihm for detecting anomalies based on data and meta-features. A growing number of research papers shed light on AutoML frameworks, which are becoming a promising solution for building complex machine learning models without human expertise and assistance. The key challenge in enabling AutoML frameworks to build an efficient model for anomaly detection tasks is to determine the best underlying model for a given task and optimization metric. The meta-learning approaches based on a set of meta features that describes data properties can enable efficient model selection in AutoML frameworks. The existing meta-learning approaches based on statistical and information-theoretic meta features require large amounts of data and computational resources to extract data properties.

The subject of research within this doctoral dissertation is the development of an extensible system for extracting meta features based on domain-specific knowledge. In order to evaluate the proposed set of meta-features, the goal is to compare the existing solutions through different aspects of complexity against the proposed solution. Also, the goal is to design experiments and get results that can be used in future research in the field of automated machine learning in general. Based on the evaluation of experimental results, it is shown that the proposed meta features achieve accuracy of 87% and meet the critical requirements for application in AutoML systems, while the existing solutions achieve accuracy of 73%. In cases where there is no significant number of datasets available for evaluation, the proposed solution achieves 25% worse performance compared against the existing solutions. The significance of the research is the possibility of implementing AutoML systems based on the proposed set of meta features. In cases where there is no sample data with labeled anomalies, or data is not present, the data creator or domain expert will be able to effectively characterize the anomalies in the data, based on domain-specific knowledge.

Keywords: anomaly detection, automl, data properties, distance functions, meta features, meta-learning, transfer learning

Scientific field: Technical sciences - Electrical and computer engineering

Specific scientific field: Computing and informatics

UDC: 621.3:004.8

Садржај

1 Увод

Убрзан развој крајњих уређаја (енг. *Edge Devices*) и интернет ствари (енг. *Internet of Things*) довео је до генерисања велике количине података који су погодни за анализу и обраду [1], [2]. Системи који се користе за анализу и обраду тих података могу да буду засновани на алгоритмима машинског учења. Перформансе таквих система искључиво зависе од квалитета података, одабраног модела и параметара тог модела [3]. Аномалије у подацима представљају инстанце које се разликују од остатка дистрибуције података, утичу на квалитет података и могу да се детектују коришћењем алгоритама машинског учења. Детекција аномалија представља битан фактор приликом анализе података, директно утиче на квалитет података и има велику примену у различитим областима [4], [5], [6], [7]. Најчешћи случајеви детекције аномалија у подацима су за потребе праћења квалитета података, и заступљени су у доменима транспорта, медицине, софтверских записа, интернет безбедности и другим [8], [9]. Аномалије могу да се појаве у различитим типовима података и потребно их је детектовати као грешке у подацима или као нову категорију [10], [11]. У зависности од типа и домена података у ком се детектују, аномалије могу да имају различите карактеристике и могу да представљају грешке или нове категорије у подацима, што чини њихову детекцију могућом коришћењем одговарајућег модела.

Потреба за детекцијом аномалија захтева креирање модела који ће моћи ефикасно да детектује аномалије у подацима. Такав модел може да се разликује у зависности од домена примене, при чему одабир модела и параметара модела за детекцију аномалија искључиво зависи од експертизе креатора система или доменског експерта. Ако постоји узорак обележених аномалија у подацима, при чему је однос нормалних инстанци и аномалија балансиран, детекција аномалија може да се посматра као проблем бинарне класификације. Међутим, у случајевима када не постоји узорак података са обележеним аномалијама, што је чест случај у индустрији, одабир модела за детекцију аномалија представља захтеван задатак. Процес одабира модела и параметара модела може да се аутоматизује коришћењем аутоматизованих система за машинско учење (*AutoML*). Такви системи решавају проблем одабира модела и омогућавају кориснику који није експерт из одређеног домена, или нема довољно доменског знања, да креира модел за детекцију аномалија без претходног знања о карактеристикама, имплементацији и начину рада модела. Како би аутоматизовани систем за машинско учење могао да одабере модел који даје задовољавајуће резултате приликом детекције аномалија за дате податке и оптимизациону метрику, неопходно је применити мета учење за одабир одговарајућег модела.

Мета учење представља методу машинског учења где се применом одговарајућих функција над подацима добијају мета подаци који су предиктивни за одабир модела заснованог на алгоритмима машинског учења, [12]. Мета подаци се израчунавају применом функција над скуповима података, и на тај начин представљају њихове карактеристике. У аутоматизованим системима за машинско учење, мета подаци се користе приликом одабира модела на основу претходно стеченог знања и података који систем евалуира. Како би се креирале функције за израчунавање мета података, неопходно је одредити карактеристике аутоматизованих система за машинско учење и на основу тога дефинисати критичне захтеве које функције морају да испуњавају. На тај начин је могуће дефинисати скуп функција за израчунавање мета података које испуњавају критичне захтеве потребне за коришћење у аутоматизованим система за детекцију аномалија. Критични захтеви су условљени окружењем у ком се систем налази као и архитектуром на којој се систем извршава, и могу да се односе на стриктна временска ограничења као и на доступне ресурсе којима систем располаже [13].

Циљ истраживања овог рада је да се предложи скуп функција за израчунавање мета података који ће испуњавати критичне захтеве за примену у аутоматизованим системима за машинско учење. Како би се предложио и евалуирао скуп функција за израчунавање мета података, постојећа и предложено решење ће се упоредити кроз различите аспекте комплексности. Предложене функције за израчунавање мета података ће се евалуирати коришћењем функција за мерење сличности између мета података, где је потребно проверити да ли одређени тип или група функција даје боље резултате у односу на остале функције. Резултати експеримената ће показати да предложено решење постиже тачност од 87%, док постојећа решења постижу тачност од 73% над целим репозиторијумом, док у ситуацијама када не постоји значајна количина скупова података предложено решење постиже и за 25% лошије перформансе. Креирани експерименти и добијени резултати ће моћи да се користе у будућим истраживањима у области аутоматизованог машинског учења. Такође, биће анализирана ефикасност имплементираног решења на различитим архитектурама и местима извршавања како би се показало у којим окружењима предложено решење може да се имплементира и под којим условима. Значај овог истраживања представља могућност имплементације аутоматизованих система за детекцију аномалија заснованим на предложеном скупу функција за израчунавање мета података који ће испуњавати критичне захтеве за наведену примену. Рад садржи шест поглавља, преглед коришћене литературе, преглед скраћеница, преглед слика, преглед табела и биографију аутора.

Друго поглавље дефинише домен истраживања, све потребне појмове за разумевање проблема и врши упоредну анализу постојећих решења у домену мета учења и мета података за одабир алгоритма за детекцију аномалија у подацима. Поглавље даје преглед типова и домена података у којима се јављају аномалије, преглед различитих типова аномалија, алгоритама за детекцију аномалија и аутоматизованих система за машинско учење. Након тога је дат преглед и класификација постојећих решења у домену мета учења кроз различите аспекте комплексности. На крају су анализиране функције које се користе за мерење сличности између мета података коришћењем функција за мерење удаљености.

Треће поглавље дефинише скуп критичних захтева које функције за израчунавање мета података морају да задовоље како би могле да се користе у аутоматизованим системима за детекцију аномалија. Након тога, дефинисане су полазне хипотезе на које овај рад треба да одговори. Затим је дат предлог функција за израчунавање мета података заснованих на доменском знању које могу да се користите у аутоматизованим системима за детекцију аномалија.

Четврто поглавље дефинише логичку структуру компоненте за одабир алгоритма у системима за аутоматизовано машинско учење, која се заснива на предложеном скупу функција за израчунавање мета података. Након тога је описан начин функционисања такве компоненте и представљена је топологија компоненте у различитим окружењима.

Пето поглавље дефинише поставку експеримената, укључујући скупове података, евалуационе метрике, алгоритме за детекцију аномалија, функције за мерење сличности између мета података, као и преглед коришћених архитектура за извршавање експеримената у различитим окружењима. Након тога је дат опис експеримената који проверавају дефинисане полазне хипотезе. Добијени експериментални резултати су анализирани на начин да се одговори на постављене полазне хипотезе.

Шесто поглавље сумира резултате приказане у докторској дисертацији, излаже закључак и могуће правце даљег истраживања.

2 Дефиниција проблема и преглед постојећих решења

У овом поглављу се даје преглед области неопходних за дефиницију проблема и анализирају се постојећа решења у домену мета учења кроз различите аспекте комплексности. Прво је дат преглед података кроз различите типове и домене у којима се јављају аномалије, након чега су представљене аномалије у подацима, њихове карактеристике и дата је класификација на основу локалитета и димензионалног простора аномалија. Након тога је дат преглед алгоритама за детекцију аномалија, представљени су основни стохастички и детерминистички алгоритми, и извршена је кратка математичка анализа алгоритама. Затим су представљени аутоматизовани системи за машинско учење, главне карактеристике тих система, дискутоване су различите архитектуре и локације извршавања, и дат је преглед постојећих решења из отворене литературе и индустрије.

Да би се омогућило коришћење аутоматизованих система за машинско учење у детекцији аномалија, неопходно је коришћење мета података за одабир одговарајућег алгоритма. Због тога је дат преглед постојећих решења у домену мета података и извршена је компаративна анализа кроз различите аспекте комплексности. На крају, представљене су функције за мерење сличности између мета података, које су неопходне за функционисање аутоматизованих система за машинско учење.. На основу наведених области и њихове анализе, кроз следећа поглавља могуће је уочити карактеристике постојећих решења, као и дефинисати полазне хипотезе на које овај рад треба да одговори.

2.1 Подаци за анализу и обраду

У неком систему, подаци се генеришу на основу процеса који могу да осликавају активности тог система или да описују понашања неког актера система. Тако генерисани подаци нису у структуираном формату погодном за даљу анализу, због чега се јавља потреба за системима за њихову анализу и обраду. Системи за анализу и обраду података користе генерисане податке са једног или више извора и обрађују их како би произвели резултат у облику структуираног приказа или изведених података погодних за даљу анализу или обраду. Када активности или понашања у систему нису у складу са уобичајеним шаблонима, креирају се аномалије у подацима. Квалитет података утиче на перформансе модела за анализу и обраду података и због тога детекција аномалија представља важну компоненту приликом креирања одговарајућег модела. У зависности од случаја коришћења, типа података и домена података, карактеристике аномалија могу да се разликују. У наставку је дат преглед случајева коришћења алгоритама за детекцију аномалија за различите типове и домене података.

2.1.1 Тип података

Тип податка представља скуп вредности које тај податак може да има. Сваки тип података има различите карактеристике, што значи да скупови података са истим типовима деле заједничке карактеристике. У зависности од извора података, типови могу да се разликују, где један скуп података може да садржи један или више типова података. Како се у овом раду анализирају аномалије у подацима, дат је преглед типова података из отворене литературе и индустрије у којима се појављују аномалије. Одабир типова података анализираних у овом раду је извршен прегледом доступних скупова података са обележеним

аномалијама [14]. Одабрани типови података за анализу су временски подаци, вишедимензионални подаци, просторни подаци и номинални подаци.

Временски подаци су заступљени у системима који производе податке са одређеном фреквенцијом. Пример тих података може да буде очитавање резултата са сензора или праћење записа софтверског система. У овом случају се ради о системима који производе токове података. У временским подацима су заступљене аномалије које могу да буду у релацији са временском димензијом, где се такве инстанце разликују од остатка дистрибуције. Вишедимензионални подаци су заступљени у системима који прате податке о одређеним ентитетима или акцијама. Пример оваквих података може да буде праћење података о пацијентима у медицини или праћење банковних трансакција. У вишедимензионалним подацима су заступљене аномалије које могу да буду у релацији са више атрибута, при чему се те инстанце разликују од остатка дистрибуције. Просторни подаци су заступљени у системима који прате податке о географским ентитетима. Пример оваквих података може да буде праћење сензора који су распоређени на локацијама или праћење статуса групе рачунара који се налазе на различитим физичким локацијама. У просторним подацима су заступљене аномалије које могу да буду у релацији са географском локацијом или координатама, при чему се разликују од остатка дистрибуције. Номинални подаци су заступљени у системима који прате информације о ентитетима који имају номиналне атрибуте. Номинални тип података је често комбинован са осталим типовима података. Пример тих података може да буде праћење података о корисницима система или обрада текста. Номинални подаци не могу да описују количину података, не може да се одреди редослед и не може да се одреди квалитет података. У просторним подацима су заступљене аномалије које могу да буду у релацији вредностима номиналних података при чему се разликују од остатка дистрибуције.

Како би се боље представили различити типови података, у табели 1 су приказане карактеристике података по типовима различитих димензионалности, величина, и броја аномалија. Из сваке групе података узет је по један репрезентативни пример. Сваки ред у табели садржи назив скупа података, опис, тип, број атрибута, број редова и број аномалија.

Табела 1: Репрезентативни примери различитих типова података са описом, бројем атрибута, редова и аномалија. За сваки тип података узет је по један репрезентативни пример.

Скуп података	Опис	Тип података	Број атрибута	Број редова	Број аномалија
ELB request count	Метрике *AWS* сервера које представљају рад процесора, диска и мрежне комуникације	Временски подаци	2	4032	2
KDD-Cup99 HTTP	Шаблони понашања мрежне комуникације укључујући широк спектар упада симулираних у војном мрежном окружењу	Више-димензионални подаци	29	620097	1052
Australian centre for cyber security	Малициозни шаблони понашања мрежне комуникације	Просторни подаци	48	2540044	321283
Thyroid disease	Номинални подаци анамнеза пацијената	Номинални подаци	21	6915	249

2.1.2 Домен података

Домен података представља област из које подаци долазе. Домени података имају различите карактеристике, па тако скупови података из истих домена деле заједничке карактеристике. Извор података зависи од домена података, при чему један скуп података

може да буде креиран из више извора који су део једног или више домена. У наставку је дат преглед различитих домена података из отворене литературе и индустрије у којима се појављују аномалије. Одабир домена података анализираних у овом раду је извршен прегледом доступних скупова података са обележеним аномалијама. Одабрани домени података за анализу су производња, транспорт, економија, медицина, обрада текста, софтверски система, и социјалне мреже као графови података.

У зависности од домена података, аномалије могу да имају различите карактеристике. У процесу производње, системи су опремљени са великим бројем сензора који се користе за проверу квалитета производње и праћења процеса. Грешка у одређеном сензору може да доведе до успоравања или заустављања процеса производње. Један случај коришћења детекције аномалија може да буде предвиђање и детекција грешака у сензорима, при чему је циљ спречити успоравање или заустављање процеса производње [9]. У домену саобраћаја и транспорта, детекција аномалија може да се користи за откривање гужви и уских грла у транспортним мрежама, као другачије шаблоне понашања. Радови у отвореној литератури [15], [16] представљају приступе за детекцију аномалија анализирањем временских и просторних података који детектују другачије шаблоне понашања у транспорту. Такође, заступљен случај примене детекције аномалија јесте откривање упада у софтверске системе и преваре у банковним трансакцијама [17], [18]. Овакве нежељене активности могу да имају велики утицај на приватност и сигурност система и корисника. Такође, још један пример детекције аномалија представља медицина, где се у системима за надгледање пацијената користе методе за детекцију аномалија које могу да имају утицај на исход третирања пацијената. Такви системи су опремљени са сензорима и врше детекцију аномалија у временском подацима. Сензори прате стање пацијента и предвиђају контекстуално понашање које се не уклапа у нормалне шаблоне понашања [19]. Из наведених примера се показало да је детекција аномалија битан фактор у домену проналажења скривеног знања, где је циљ да се елиминишу грешке у подацима како би се побољшале перформансе система [20].

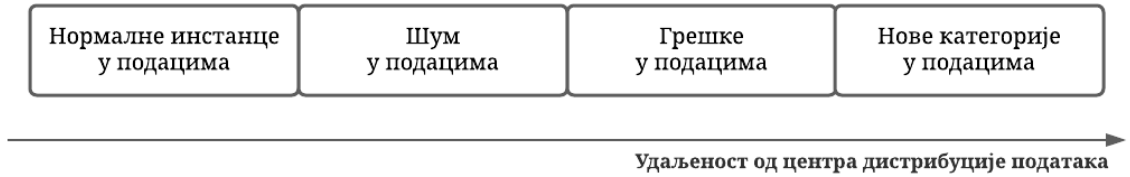
Како би се боље представили различити домени података, у табели 2 су приказане карактеристике података по доменима различитих димензионалности, величина, и броја аномалија. Из сваке групе података узет је по један репрезентативни пример. Сваки ред у табели садржи назив скупа података, опис, домен, број атрибута, број редова и број аномалија.

Табела 2: Репрезентативни примери различитих домена података описом, бројем атрибута, редова и аномалија. За сваки домен података узет је по један репрезентативни пример.

Скуп података	Опис	Домен података	Број атрибута	Број редова	Број аномалија
Machine temperature system failure	Подаци сензора температуре унутрашње компоненте велике индустријске машине	Производња	2	22695	4
NYC taxi	Агрегирани подаци укупног броја путника у сегментима од 30 минута	Транспорт	2	10320	5
Exchange CPC results	*Cost-per-click* резултати	Економија	2	1624	1
Breast cancer diagnostic	Дијагностика рака дојке коришћењем карактеристика слике	Медицина	30	367	10
Letter recognition	Карактеристике 26 слова Енглеског алфабета	Обрада текста	32	1599	100
Spacecraft anomaly data	Софтверски логови свемирских летелица *Solar-Terrestrial Physics Division of the National Geophysical Data Center*	Софтверски система	20	4654	3
Twitter volume UPS	Број спомињања кључне речи у сегментима од 5 минута на друштвеној мрежи	Графови	2	15866	5

2.2 Аномалије у подацима

Аномалије у подацима представљају инстанце које се по карактеристикама, односно вредностима разликују од осталих података. Аномалије могу да буду грешке у подацима, могу да представљају шум у подацима, као и да означавају нове категорије у подацима, које још нису откривене. Грешке у подацима углавном представљају вредности које не испуњавају структурна или вредносна ограничења и треба их уклонити из података. Инстанце које се посматрају као шум у подацима се сматрају да нису релевантне за одређени скуп података и због тога их треба уклонити из података. Нове категорије у подацима представљају групу података која има вредност и може да се користи за даљу анализу. На слици 1 је илустровано поређење нормалних инстанци и аномалија, при чему аномалије могу да буду грешке, шум или нове категорије у подацима. Тачна граница између ових класа се одређује на основу доменског знања и може да се разликује се за сваки скуп података. Када се детектује грешка или шум у подацима, циљ је да се таква инстанца уклони из података. У супротном, када се детектују нове категорије у подацима циљ је класификовати такве инстанце, које се или уклањају из података или означавају као нова класа. У овом раду се грешке, шум и нове категорије у подацима шум посматрају једнако и у зависности од скупа података се одређује граница између нормалних инстанци и аномалија.



Слика 1: Класификација инстанци података на основу удаљености од центра дистрибуције података. Нормалне инстанце у подацима су близу центра дистрибуције и имају очекиване шаблоне понашања. Шум у подацима се налази даље од центра дистрибуције у односу на нормалне инстанце. Грешке и нове категорије у подацима се по шаблонима понашања разликују од нормалних инстанци и шума у подацима.

Како се аномалије креирају услед грешке или појаве нових категорија у систему који има извор података, аномалије могу да буду присутне у различитим типовима података. Коришћењем структурних и вредносних провера на самом извору података, могуће је смањити количину аномалија у подацима. У зависност од домена где се подаци користе, тип података се разликује. У зависност од домена података зависи ког ће типа подаци бити. На пример, ако се подаци генеришу на извору са одређеном фреквенцијом, онда они имају временску димензију. Аномалије у временским подацима могу да буду транзијентне, што значи да се само у под одређеним условима креирају, и то углавном зависи од тренутног окружења и контекста у ком се систем налази. Сличан приступ може да се примени и на друге типове и домене података. Међутим, како би извођење било тачно, неопходно је да доменски експерт утврди карактеристике аномалија за одређени домен података. Доменски експерт би могао да на основу система који генерише податке, окружења и контекста у ком се систем налази одреди карактеристике аномалија које могу да се појаве. У том случају, доменски експерт предлаже карактеристике аномалија које могу да се појаве у подацима и како те аномалије утичу на квалитет података.. Присуство доменског експерта приликом решавања одређеног проблема је заступљено у домену машинског учења и користи се у различитим применама [21]. У овом раду, присуство доменског експерта је коришћено за одређивање карактеристика аномалија у подацима.

Аномалије у подацима у зависности од типа података и домена из којих подаци долазе могу да имају различите карактеристике. Један скуп података може да садржи аномалије са различитим карактеристикама. У зависности од карактеристика аномалија, оне могу да се поделе по типу, локалитету и димензионалном простору, чији преглед је дат у наставку.

2.2.1 Тип аномалија

Прост пример аномалије у подацима представља екстрема вредност. Та вредност се разликују од остатка дистрибуције и углавном представља грешку у подацима. Пример екстремне вредности може да буде висина људи на неком узорку података. Ако је за инстанцу у подацима, за висину дата негативна вредност или друга екстремна вредност, та инстанца се може сматрати аномалијом. У том случају, циљ је елиминисати такве вредности из скупа података. Следећи пример аномалије у подацима може да се представи кроз просечну температуру по месецима. Ако се посматра само вредност температуре без додатног контекста као што је месец, наведена инстанца не може да се означи као аномалија у подацима. На пример, ако је температура 25° целзијуса, та инстанца неће бити обележена као аномалија. Међутим, ако је наведена температура измерена у Фебруару, инстанца ће бити обележена као

аномалија због контекста и окружења у ком се налази. Наведени пример представља контекстуалну аномалију где је неопходно поред података укључити и контекст како би се аномалије разликовале од нормалних инстанци у подацима. Даље, аномалија може да се представи кроз пример система за анализу резултата пацијента. Ако се резултати пацијента посматрају као изоловане вредности, могуће је детектовати само просте типове аномалија, као што су екстремне вредности. На пример, ако се анализира комплетна крвна слика пацијента и вредности посматрају изоловано, не може да се утврди да ли је у питању аномалија, тј. да ли је потребна посебна терапија. Међутим, ако се вредности посматрају колективно, онда се таква инстанца разликује од остатка дистрибуције и као таква представља аномалију у подацима. Наведени пример представља колективну аномалију, где ако се атрибути инстанци у скупу података посматрају изоловано, није могуће детектовати аномалију, већ је неопходно посматрати податке колективно. Колективне аномалије могу да представљају и нову категорију у подацима чији облици понашања до тог тренутка нису уочени [22].

У табели 3 дат је преглед различитих типова аномалија кроз типове података и домене из којих подаци долазе. Прегледом отворене литературе и анализом различитих типова и домена података, дошло се до класификације типа аномалија у односу на тип и домен података. Такође, овакво поређење представља заступљене типове аномалија у тим категоријама, али не ограничава појаву других типова аномалија.

Табела 3: Типови аномалија који се јављају за различите типове и домене података. Наведено поређење представља заступљене типове аномалија у тим категоријама, али не ограничава појаву других типова аномалија.

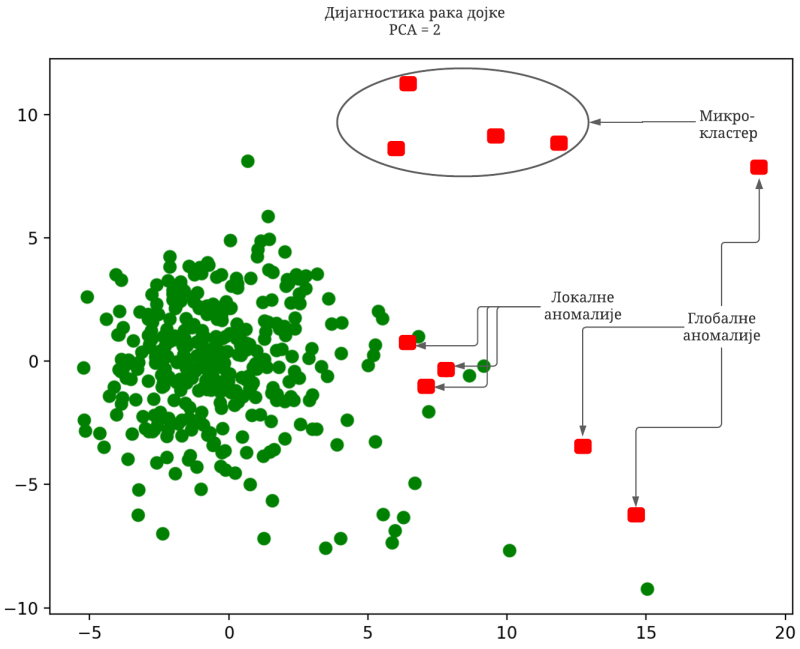
Домен података	Тип података			
	Временски подаци	Више-димензионални подаци	Просторни подаци	Номинални подаци
Производња	Екстремне вредности/ Контекстуалне аномалије	Екстремне вредности/ Колективне аномалије	Екстремне вредности	Екстремне вредности
Транспорт	Екстремне вредности/ Контекстуалне аномалије	Екстремне вредности/ Колективне аномалије	Екстремне вредности	Екстремне вредности
Економија	Екстремне вредности/ Контекстуалне аномалије	Екстремне вредности/ Колективне аномалије	Екстремне вредности	Екстремне вредности
Медицина	Екстремне вредности/ Контекстуалне аномалије	Екстремне вредности/ Колективне аномалије	Екстремне вредности	Екстремне вредности
Обрада текста	Контекстуалне аномалије	Екстремне вредности/ Колективне аномалије	Екстремне вредности	Екстремне вредности
Софтверски системи	Екстремне вредности/ Контекстуалне аномалије	Екстремне вредности/ Колективне аномалије	Екстремне вредности	Екстремне вредности
Графови	Контекстуалне аномалије	Екстремне вредности/ Колективне аномалије	Екстремне вредности	Екстремне вредности

Аномалије, поред њиховог типа, могу да се разликују по локалитету и димензионалном простору у ком се аномалије детектују. Локалитета аномалија и димензионални простор детекције аномалија представљају главне карактеристике аномалија по којима међусобно могу да се пореде.

2.2.2 Локалитет аномалија

Локалитет аномалија представља карактеристику аномалија која се односи на удаљеност од дистрибуције података, као и окружења у ком се аномалија налази. У зависности

од локалитета аномалија, оне могу да се класификују на глобалне аномалије, локалне аномалије и микро-кластере. На слици 2 су приказани различити локалитети аномалија на примеру скупа података из медицине. Глобалне аномалије представљају инстанце у подацима које се разликују од остатка дистрибуције и у својој околини немају инстанце са сличним вредностима. Пример глобалних аномалија је екстремна вредност у подацима и представља прост тим аномалија. Екстремне вредности углавном представљају грешке у подацима и треба их уклонити из података. Локалне аномалије представљају инстанце у подацима које се разликују од дистрибуције искључиво локалног подскупа података. Пример локалних аномалија су екстремне вредности искључиво у односу на локални подскуп података. Локалне аномалије могу да представљају понашање које се разликује од локалног подскупа података, али се не разликује од остатка дистрибуције података. Пример локалних аномалија може да се представи као повишени или снижени резултати комплетне крвне слике пацијента, при чему те вредности не представљају екстремне вредности у односу на цео скуп, али су изван референтних вредности. Локалне аномалије могу да представљају контекстуалне аномалије у подацима. Микро-кластери представљају инстанце у подацима које се разликују од дистрибуције података, док са суседним инстанцама креирају подскуп података са сличним шаблоном понашања који се разликује од остатка дистрибуције. Пример микро-кластера представља ново понашање у подацима које се разликује од дистрибуције података и које до тада није уочено. Микро-кластери могу да представљају колективне аномалије и да описује ново понашање у подацима.



Слика 2: Подаци о дијагностици рака дојке у 2-димензионалном простору са различитим локалитетима аномалија. Инстанце које се разликују од остатка дистрибуције су означене као глобалне аномалије. Инстанце које одступају од остатка дистрибуције само за непосредну околину су означене као локалне аномалије. Инстанце које одступају од остатка дистрибуције и имају сличне инстанце у својој околини су означене као микро-кластери.

Наведени локалитети аномалија утичу на методе детекције аномалија и дефинишу тип аномалија који се налази у подацима. Међутим, није дефинисано да тип аномалија може да се детерминистички представи помоћу само једног локалитета аномалија. На пример, одговарајућом трансформацијом података могуће је представити комплексне типове аномалија, као што су колективне и контекстуалне аномалије, помоћу простих типова аномалија, као што су екстремне вредност аномалија, и тако користити методе за детекцију простих аномалија. Како би се постигле трансформације над подацима и аномалија, неопходно је присуство доменског експерта и креатора података.

2.2.3 Димензионални простор аномалија

Димензионални простор аномалија представља карактеристику аномалија која се односи на број димензија или атрибута у скупу података у ком се детектују аномалије. У зависности од броја димензија у ком се аномалије детектују, оне могу да буду једнодимензионалне или вишедимензионалне. Ако се посматра једнодимензионални простор, онда се детекција аномалија врши по само једном атрибуту података. У том случају, могуће је детектовати аномалије коришћењем више димензија података независно, при чему се за сваку димензију креира одвојени модел. Наведени приступ се може користити приликом детекције простих аномалија, као што су екстремне вредности у подацима.

У вишедимензионалном простору, модел за детекцију аномалија обухвата више димензија при чему се креира један модел који учи шаблоне понашања на основу вредности више атрибута. Наведени приступ се може користити приликом детекције колективних и контекстуалних аномалија, као што су нове категорије у подацима. Приликом детекције аномалија у вишедимензионалном простору користе се комплексни модели који представљају шаблоне понашања више атрибута.

Редукцијом и интеграцијом података могуће је смањити комплексност проблема детекције аномалија у вишедимензионалном простору. Један од приступа редукције података представља креирање атрибута који представља семантичку интеграцију више атрибута. Како би подаци могли да се трансформишу, неопходно је добро познавање података, тј. присуство доменског експерта или креатора података.

2.3 Алгоритми за детекцију аномалија у подацима

Модели за детекцију аномалија се заснивају на алгоритмима који креирају шаблон понашања за нормалне инстанце у подацима. Након тога, за сваку инстанцу у подацима одређују колико је она удаљена од креираног шаблона понашања. Алгоритми за детекцију аномалија могу да се поделе у две категорије на основу резултата који производе: алгоритми који рачунају удаљеност од дистрибуције података и бинарни алгоритми. Ово поглавље прво даје преглед различитих типова алгоритама а затим за сваки тип алгоритма представља један пример алгоритма који ће се користити даље у раду.

Алгоритми који рачунају удаљеност од дистрибуције података као резултат дају вредност која се користи као индикатор колико је нека инстанца аномалија. Већина алгоритама који се користе за детекцију аномалија раде по наведеном принципу, што представља генерално добар приступ где се граница одређује на основу обележеног скупа података или уз присуство доменског експерта. Овај тип алгоритама даје више информација о аномалијама и због тога је практичан за анализу аномалија у подацима. Бинарни алгоритми за детекцију аномалија као резултат дају бинарну вредност која означава да ли је нека инстанца аномалија.

Бинарни тип алгоритама даје мање информација о аномалијама али је практичан из разлога што доноси одлуку да ли је инстанца аномалија, тако да има широку примену у системима за детекцију аномалија. У алгоритмима заснованим на рачунању удаљености се добија бинарни резултат применом одговарајуће границе која је одређена обележеним скупом података или присуством доменског експерта. Такође, одабир граничне вредности може да се израчуна претрагом простора коришћењем обележеног скупа података. Та вредност може да се разликује у зависности од опсега вредности у подацима, алгоритма који се користи за детекцију аномалија, као и од шума у подацима. Шум у подацима утиче на одређивање граничне вредности између нормалних инстанци и аномалија тако што повећава опсег нормалног шаблона понашања и тиме доводи да се аномалије теже детектују. Детекција аномалија у ситуацијама када не постоје обележени подаци се своди на одређивање граничне вредности која се налази у опсегу шума података. У тим ситуацијама, аномалије могу да се поделе на слабе и јаке аномалије. Слабе аномалије представљају шум у подацима које немају исте карактеристике као и јаке аномалије. Ако постоје обележени подаци, могуће је одредити граничну вредност између нормалних инстанци и аномалија коришћењем обележених података. Са обележеним подацима могуће је направити класификациони модел који ће ефикасно детектовати аномалије у подацима. Међутим, у већини случајева, обележени подаци не постоје или је подскуп обележених аномалија мали, што чини проблем класификације комплексним. Такође, ако постоје подаци за које се зна да не садрже аномалије и описују нормални шаблон понашања, могуће је креирати модел нормалног понашања и затим мерити удаљеност инстанци од креираног шаблона понашања.

На основу „*No Free Lunch*” теореме [23], није могуће креирати јединствени модел који ће давати добре резултате за све скупове података. Одабир модела за детекцију аномалија се своди на одабир алгоритма који ће за одговарајући тип података и домен из ког подаци долазе дати задовољавајуће резултате за тражену оптимизациону метрику. Тип и домен података утичу на перформансе алгоритама, па самим тим могу да утичу и на одабир алгоритма. Алгоритми за детекцију аномалија могу да се поделе на стохастичке и детерминистичке алгоритме, на основу резултата који производе. Стохастички алгоритми садрже одређени степен случајности приликом рачунања резултата. То значи да приликом довођења истог скупа података на улаз алгоритма, резултат не мора увек да буде идентичан. Алгоритми машинског учења који се користе за детекцију аномалија у ситуацијама када аномалије нису обележене могу да буду стохастички ако се заснивају на статистичким функцијама које користе карактеристике случајности. Са друге стране, детерминистички алгоритми не садрже компоненте које уносе случајност у израчунавање резултата. То значи да приликом довођења истог скупа података на улаз алгоритма, резултат је увек исти. Алгоритми машинског учења који се користе за детекцију аномалија у ситуацијама када аномалије нису обележене могу да буду детерминистички ако се не заснивају на статистичким функцијама које користе карактеристике случајности.

Ако постоји балансиран узорак података са обележеним аномалијама, модели засновани на алгоритмима за класификацију могу довољно добро да детектују аномалије у подацима. У супротном, модели засновани на алгоритмима за машинско учење који се користе у ненадгледаном учењу могу довољно добро да детектују аномалије у подацима. У оба случаја неопходно је креирати модел за детекцију аномалија у подацима. Један од главних фактора приликом одабира алгоритама је био степен трансформације који се врши над подацима. Алгоритми који имају нижи степен трансформација над подацима дају мање поуздане резултате, али резултати и детектоване аномалије у подацима могу лакше да се тумаче. Алгоритми који имају виши степен трансформација над подацима дају поузданије резултате, али резултати и детектоване аномалије у подацима су тежи за тумачење. Тумачење резултата

је неопходно како би се валидирали резултати и стекло знање о подацима и аномалијама у подацима.

Одабир алгоритма за детекцију аномалија представља битан задатак од који зависи од карактеристика података и аномалија и углавном се врши детаљном анализом података. Лош одабир алгоритма утиче на перформансе креираног модела. Како би се одабрали алгоритми који ће да се користи у експериментима и евалуацији резултата у овом раду, неопходно је одредити главне критеријуме за одабир алгоритама тако да се разликују по типовима података за које су они погодни, да су коришћени за различите типове аномалија, да буду из група стохастичких или детерминистичких алгоритама и да имају одређени степен трансформације над подацима. Степен трансформација над подацима представља могућност закључивања због чега је одређена инстанца означена као аномалија [24]. Подаци који се мапирају у више или ниже димензионалне просторе и не користе оригиналне атрибуте приликом одређивања аномалија се сматрају да имају већи степен трансформација. Одабир алгоритама је урађен тако што је из сваке групе алгоритама одабран по један алгоритам као репрезентативни пример који представља одређену групу алгоритама. На основу прегледа отворене литературе [14], може се закључити да су следеће групе алгоритама машинског учења заступљене приликом креирања модела за детекцију аномалија: алгоритми засновани на теорији вероватноће, статистички алгоритми, алгоритми засновани на декомпозицији, алгоритми засновани на удаљености, и алгоритми засновани на неуронским мрежама. У наставку је дат преглед алгоритама са освртом на основну математичку анализу и захтеве које алгоритам поставља како би креирао модел и детектовао аномалије у подацима.

2.3.1 Група алгоритама заснованих на теорији вероватноће

Група алгоритама заснованих на теорији вероватноће може да се користи за креирање модела за детекцију аномалија у подацима који су описани нормалном дистрибуцијом. Пример алгоритма из ове групе је Гаусова дистрибуција (енг. *Gauss distribution*). Гаусова дистрибуција описује податке коришћењем средње вредности и варијансе. У детекцији аномалија, мери се одступање инстанци од креиране нормалне дистрибуције у једнодимензионалном простору [25]. Применом функције (1) рачуна се густина вероватноће криве нормалне дистрибуције, где је σ стандардна девијација, μ средња вредност и x инстанца за коју се рачуна вредност.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \quad (1)$$

Како се аномалије у подацима у већини случајева детектују у вишедимензионалном простору, Гаусова дистрибуција се може проширити на вишедимензионални простор. У тој ситуацији се користи мултиваријантна Гаусова дистрибуција, где се рачуна густина вероватноће криве нормалне дистрибуције за више атрибута. Функције (2) представљају начин рачунања променљивих, где је Σ стандардна девијација, μ средња вредност, m број атрибута, x инстанца за коју се рачуна вредност и x^T транспонована вредност x.

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$
$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} (x^{(i)} - \mu)(x^{(i)} - \mu)^T \quad (2)$$

Услов за примену Гаусове дистрибуције за детекцију аномалија је да су подаци описани нормалном дистрибуцијом и да се аномалије разликују од те дистрибуције. Једном када се модел прилагоди дистрибуцији података, применом функције (3) се рачуна вероватноћа да инстанца припада дистрибуцији где је Σ стандардна девијација, μ средња вредност, n укупан број инстанци и x инстанца за коју се рачуна вредност.

$$p(x) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)} \quad (3)$$

Након тога се проверава да ли је вредност већа од постављене граничне вредности која раздваја нормалне инстанце од аномалија, применом функције (4), где је ε гранична вредност.

$$p(x) < \varepsilon \quad (4)$$

Ако су доступни означени подаци, гранична вредност аномалија може да се одреди помоћу означених података. Описани алгоритам представља детерминистички алгоритам јер се не заснива на функцијама које имају карактеристике случајности. Овај алгоритам даје могућност примене над било којим типом и доменом података под условом да су подаци описани нормалном дистрибуцијом. Подаци се представљају параметрима за опис нормалне дистрибуције и на тај начин се ради значајна трансформација над подацима, па није могуће једноставно представити резултате и повезати их са улазним подацима.

2.3.2 Група статистичких алгоритама

Група статистичких алгоритама за детекцију аномалија користи релацију између атрибута како би се креирао шаблон нормалног понашања из података. Инстанце које се не уклапају у креиран шаблон нормалног понашања се означавају као аномалије. Линеарна регресија представља статистички алгоритам који може да се користи за детекцију аномалија. Коришћењем линеарне регресије креира се регресиона права и рачуна удаљеност инстанци од регресионе праве. Ако је удаљеност између инстанце и регресионе праве већа од граничне вредности, та инстанца се означава као аномалија. Креирање регресионе праве се врши решавањем низа линеарних једначина, применом функције (5) где су X, β и ϵ компоненте линеарних једначина.

$$y = X\beta + \epsilon \quad (5)$$

На основу претходне функције се добијају коефицијенти регресионе праве. Након тога се за сваку инстанцу рачуна удаљеност од праве и проверава да ли је вредност већа од постављене граничне вредности која раздваја нормалне инстанце од аномалија, применом функције (6) где је ε гранична вредност.

$$\varepsilon < y \quad (6)$$

Ако су доступни означени подаци, гранична вредност се може одредити помоћу означених података. Описани алгоритам представља стохастички алгоритам јер компоненте линеарних једначина могу да буду промењиве. Овај тип алгоритама може да детектује аномалије у подацима који су линеарно зависни. На пример, у временским подацима са високом корелацијом између атрибута, алгоритам може ефикасно да детектује аномалије [26]. Коришћењем оригиналних података креира се регресиона права, и на тај начин се ради проста

трансформација над подацима, па је могуће једноставно представити резултате и повезати их са улазним подацима.

2.3.3 Група алгоритама заснованих на декомпозицији

Група алгоритама заснованих на декомпозицији може да се користи за детекцију аномалија у подацима применом трансформација над подацима. Алгоритми из ове групе примењују трансформације над подацима тако што трансформишу податке у суб-димензионални простор који повећава разлике у шаблонима понашања између нормалних инстанци и аномалија. Након трансформације, подаци се враћају у оригинални простор и рачуна се грешка приликом реконструкције података. Добијена грешка се узима као индикатор који показује колико је инстанца аномалија. Робусна анализа главних компонената (енг. *RPCA*) представља алгоритам заснован на декомпозицији који може да се користи за детекцију аномалија. Применом декомпозиције појединачних вредности (енг. *SVD*) [27], алгоритам мапира податке у суб-димензионални простор. Декомпозиција појединачних вредности се заснива на рачунању матрица са ортонормалним сопственим векторима, применом функције (7) где су U и V ортогоналне матрице са ортонормалним сопственим векторима, док је Σ дијагонална матрица са елементима једнаким корену позитивне сопствене вредности.

$$\mathrm{M} = U * \Sigma * V^{*} \qquad (7)$$

Након извршених трансформација над подацима рачуна се ϵ грешка у приликом трансформације. Затим се проверава да ли је вредност већа од постављене граничне вредности која раздваја нормалне инстанце од аномалија, применом функције (8), где је ε гранична вредност.

$$\varepsilon < \epsilon \qquad (8)$$

Описани алгоритам представља детерминистички алгоритам јер се не заснива на функцијама које имају карактеристике случајности. Ова група алгоритама може да има примену у вишедимензионалним подацима који садрже временске и просторне податке са релацијом између атрибута, која је уграђена у просторе нижих димензија. Када се подаци трансформишу у суб-димензионални простор аномалије у подацима имају већу грешку приликом трансформације. Над подацима се врши значајна трансформација, па није могуће једноставно представити резултате и повезати их са улазним подацима.

2.3.4 Група алгоритама заснованих на удаљености

Група алгоритама заснованих на удаљености за детекцију аномалија користи удаљеност од дистрибуције у локалном простору. У зависности од типа алгоритма, удаљеност инстанце података се дефинише као удаљеност од центра кластера, удаљеност од осталих података, или густина локалног простора. Кластеризација методом К-средњих вредности (енг. *Kmeans*) креира кластере у подацима и мери удаљеност између инстанци података и центра кластера. Та вредност означава колико је одређена инстанца аномалија. Инстанце које се не уклапају у креиран шаблон нормалног понашања се означавају као аномалије. Алгоритам итеративно рачуна координате кластера, применом функције (9) где је $S_i^{(t)}$ скуп инстанци података које припадају i кластеру у t јединици времена, док је x_j инстанца из скупа података.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad (9)$$

Коришћењем наведене функције се кластери распоређују у димензионалном простору након чега се рачуна удаљеност d између центра кластера и инстанци кластера. Затим се проверава да ли је удаљеност већа од постављене граничне вредности која раздваја нормалне инстанце од аномалија, применом функције (10), где је ε гранична вредност.

$$\varepsilon < d \quad (10)$$

Инстанце које се не уклапају у креиран шаблон нормалног понашања се означавају као аномалије. Ако су доступни означени подаци, гранична вредност се може одредити помоћу означених података. Описани алгоритам представља стохастички тип алгоритма јер се заснива на функцијама које имају карактеристике случајности. Остали типови алгоритама засновани на удаљености на исти начин одређују аномалије у подацима али користе другачије функције за мерење удаљености и груписање података [28]. Овај алгоритам даје могућност примене над било којим типом и доменом података. На основу података се одређују кластери у подацима, али се не мењају атрибути података и на тај начин се ради проста трансформација над подацима, па је могуће једноставно представити резултате и повезати их са улазним подацима.

2.3.5 Група алгоритама заснованих на неуронским мрежама

Група алгоритама заснованих на неуронским мрежама коришћењем обележених података може да креира модел за детекцију аномалија. У том случају се проблем детекције аномалија своди на класификацију при чему се различити облици понашања описују везама у неуронској мрежи. Међутим, потреба за детекцијом аномалија се углавном јавља када аномалије нису обележене у подацима. Како се у овом раду евалуирају скупови података без обележених аномалија, могуће је креирати модел класификације са једном класом која представља нормално понашање у подацима. Количина нормалних инстанци је већа у односу на количину аномалија у подацима, тако да неуронска мрежа описује шаблон понашања нормалних инстанци, што доводи да аномалије имају велику грешку приликом класификације. Аутоенкодер (енг. *Autoencoder*) је тип неуронске мреже који се користи за детекцију аномалија тако што трансформише податке у суб-димензионални простор [29]. Прегледом отворене литературе показано је да постоји више типова аутоенкодера који могу да се користе за детекцију аномалија [30]. *Sparse* тип аутоенкодера има више неурона у скривеном слоју у односу на улазни и излазни слој, и користи се за издвајање битних карактеристика података. *Denoising* тип аутоенкодера креира излаз уношењем шума у податке и користи се за реконструкцију података. *Contractive* тип аутоенкодера представља робусну репрезентацију података која је мање осетљива на мале варијације у подацима, *Convolutional* тип аутоенкодера се користи за представљање улазних података помоћу простих карактеристика и користе се у домену обраде слика, као што је мењање димензија слика. У овом раду се користи *Sparse* тип аутоенкодера јер се користи за издвајање битних карактеристика података где је могуће раздвојити аномалије од нормалних инстанци приликом креирања излазних података. Функција (11) представља пример рада аутоенкодера, где је E функција за енкодовање података у суб-димензионални простор, D функција за декодовање података и x инстанца из скупа података.

$$x' = D(E(x)) \quad (11)$$

Након извршених трансформација над подацима се рачуна ϵ грешка у трансформацијама. Затим се проверава да ли је вредност већа од постављене граничне вредности која раздваја нормалне инстанце од аномалија, применом функције (12), где је ε гранична вредност.

$$\varepsilon < \epsilon \quad (12)$$

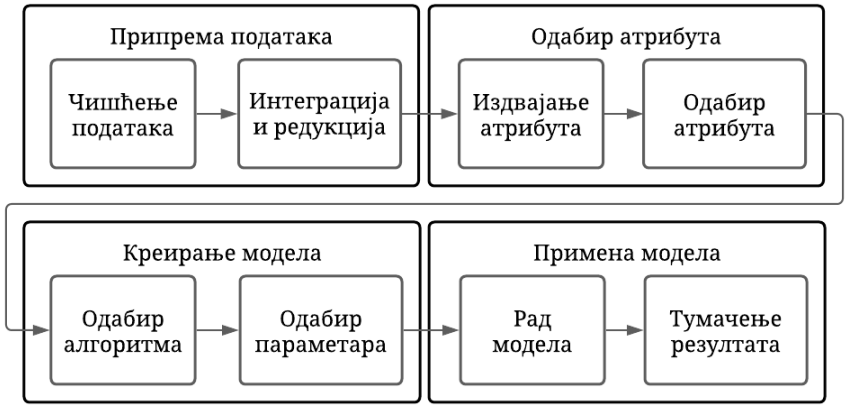
Описани алгоритам представља детерминистички алгоритам јер се не заснива на функцијама које имају карактеристике случајности [31]. Над подацима се врши значајна трансформација, па није могуће једноставно представити резултате и повезати их са улазним подацима. Овај алгоритам даје могућност примене над било којим типом и доменом података, тако да је погодан за детекцију аномалија за наведене типове и домене података. Поред аутоенкодера, постоје и остали типови неуронских мрежа који се користе за детекцију аномалија који врше просте трансформације над подацима где је могуће повезати улазне податке са моделом. На пример, у домену медицине неопходно је повезати улазне податке са моделом како би се закључило због чега се јавила аномалија у подацима и како могу да се касније детектују хеуристичким методама [32]. Дубоке конволуционе неуронске мреже се користе у различитим доменима за обраду слика где је потребно анализом слике детектовати аномалије. Оне прво трансформишу слику у карактеристике коришћењем конволуционих слојева. Након тога се у потпуно повезаној мрежи врши уклапање карактеристика у очекивани облик понашања. Дуготрајна краткорочна меморија (енг. *LSTM*) је посебна архитектура рекурентне неуронске мреже. Таква мрежа евалуира податке кроз време при чему ако се подаци довољно разликују од претходне секвенце, на пример за једну стандардну девијацију, онда може да се та инстанца посматра као аномалија у подацима.

2.4 Аутоматизовани системи за машинско учење

Аутоматизовано машинско учење представља процес аутоматизације примене алгоритама машинског учења. Аутоматизовани системи за машинско учење обухватају процесе припреме података, процесе интеграције и редукције података, процесе одабира алгоритма и параметара алгоритма погодних за одређени скуп података и оптимизациону метрику, и примену модела и добијања резултата. Овакви системи могу да буду корисни у случајевима када је потребно динамички одредити оптимални модел за дати проблем и оптимизациону метрику или креирати модел без развоја.

Припрема података за анализу и обраду, као и одабир и имплементација модела представља комплексан проблем, и као такав је погодан за аутоматизацију и примену алгоритама машинског учења. Аутоматизовани систем за машинско учење представљен је кроз следеће компоненте: компонента за припрему података, компонента за одабир атрибута, компонента за одабир модела и компонента за примену модела над подацима [33]. Свака од наведених компоненти се састоји од мањих модула и представља засебну област истраживања, док наведене компоненте повезане на приказан начин чине комплетан систем, као што је приказано на слици 3. Компоненте у аутоматизованим системима за машинско учење представљају комплексне целине које треба посебно анализирати. Компоненте аутоматизованог система су повезане на начин да излаз из претходне компоненте се користи као улаз следеће компоненте. У зависности од имплементације, компоненте везане за

припрему података и одабир атрибута могу да буду опционе, у ком случају је одговорност корисника да припреми податке на одговарајући начин.



Слика 3: Аутоматизовани системи за машинско учење се састоје од компоненте за припрему података, компоненте за одабир атрибута, компоненте за креирање модела и компоненте за примену модела. На слици је приказан начин повезивања компоненти и ток података кроз систем.

Први корак у оваквом систему представља припрема података која може да се подели на чишћење података и интеграцију и редукцију података. Оба процеса су заступљена у методама за проналажење скривеног знања и представљају битне факторе за постизање добрих перформанси модела. Чишћење података укључује избацивање инстанци које представљају невалидне вредности у подацима, и немају значај приликом креирања модела на основу шаблона понашања. Интеграција и редукција података представља процес мењања структуре података у циљу наглашавања битних карактеристика података. Интеграција података представља спајање више атрибута са циљем да се на бољи начин представи структура података и релације између атрибута. Редукција представља процес селекције и избацивања или редуковања атрибута који немају вредност за креирање шаблона понашања у моделу. Када се подаци очисте од грешака и трансформишу тако да на добар начин описују шаблоне понашања прелази се на следећи корак.

Други корак у систему представља одабир атрибута који је битан за креирање модела. Тај корак чине одабир релевантних атрибута и избацивање редундантних атрибута, углавном у вишедимензионалним подацима. Применом ових корака упрошћавају се шаблони понашања и смањује се време потребно за тренирања и закључивање система. Процес одабира атрибута се врши применом алгоритама који траже подскуп атрибута који даје малу грешку приликом извршавања модела. У отвореној литератури постоји доста радова који се баве овим процесом [34]. Избацивање редундантних атрибута може да се врши директно применом алгоритама претраге у ком случају атрибути погодни за избацивање дају велику грешку приликом извршавања модела.. Приликом креирања модела за вишедимензионалне податке, могућа је појава линеарне зависности грешке приликом извршавања модела у односу на број атрибута у подацима. У теорији, већи број атрибута представља више података из којих модел може да учи, док се у пракси показало да такав приступ углавном изазива појаву шума у подацима и повећава редундантности у подацима. У алгоритмима машинског учења то значи да са додавањем димензија захтева велика количина додатних података за евалуацију модела како би перформансе модела остале приближно исте.

Следећи корак представља одабир модела за одговарајући проблем и оптимизациону метрику. Овај корак се састоји из одабира алгоритма и параметара тог алгоритма. Одабир алгоритма представља процес који из скупа понуђених алгоритама одабира алгоритам који ће дати најбоље резултате за дату оптимизациону метрику. Одабир алгоритма може да се врши применом мета учења, које се користи за преношење стеченог знања са једног случаја коришћења на други. Један приступ мета учења представља евалуацију простих алгоритама на основу којих могу да се процене перформансе комплекснијих алгоритама. Други приступ је да се на основу претходно стеченог знања врши одабир алгоритма који је за сличне податке имао задовољавајуће перформансе. Након одабира алгоритма, неопходно је одабрати параметре алгоритма како би се максимизовале перформансе алгоритма. Параметри алгоритма се користе како би се контролисао процес учења алгоритма, при чему је овде циљ закључити који су оптимални параметри за одређени алгоритам, што захтева примену процеса учења. Оптимални одабир параметара алгоритама минимизује грешку у резултатима приликом њихове евалуације. Један од простих начина проналажења оптималних параметара алгоритма представља исцрпни метод тражења оптималних параметара који дају најбоље резултате. Овакав начин претраге мора бити праћен изврпавањем модела и евалуацијом добијених резултата како би се закључило која комбинација даје најбоље резултате.

Последњи корак представља примену модела и валидацију резултата читавог система. Примена модела се састоји из извршавања модела и анализе добијених резултата. Приликом анализе резултата, ако се закључи да добијени резултати не задовољавају одговарајуће критеријуме, неопходно је идентификовати компоненте система које могу да се оптимизују. Примена система може да се врши у доменима за које тренутно постоји имплементација, а то зависи од тренутне потребе у индустрији. Потреба за оваквим системима се пропорционално повећава са повећавањем количине података, које је условљено развојем крајњих уређаја. Већина оваквих система се користи за обраду слике и текста, док се у последње време јавља потреба за аутоматизованим системима и у другим областима у којима се генеришу велике количине података. Аутоматизовани систем за детекцију аномалија би могао да упрости проблем одабира и креирања модела за дате података и оптимизациону метрику. Такав систем би омогућио корисницима да без познавања алгоритама за детекцију аномалија и без експертског знања из области машинског учења могу ефикасно да детектују аномалије у подацима. Како би се илустровала комплексност одабира модела за детекцију аномалија, може се рећи да, на пример, линеарна регресија може постићи значајне резултате у детекцији аномалија у подацима са временском димензијом због линеарне повезаности података. Међутим, линеарна регресија може да произведе лоше резултате за вишедимензионалне податке који нису линеарно зависни [35].

Перформансе описаног аутоматизованог система зависе од оптимизације компоненти и улазних података. Ако су компоненте оптимизоване за одређени тип и домен улазних података, перформансе система могу да буду задовољавајуће. У супротном, потребно је анализом компоненти утврдити да ли је проблем у имплементацији компоненти или у типу и домену података за који систем није намењен. У првом случају, могуће је извршити оптимизацију појединачних компоненти тако да користе алгоритме прилагођене за дату примену. У другом случају, потребно је прилагодити систем тренирањем за одређени тип и домен података. Овај рад се бави искључиво одабиром алгоритма коришћењем техника мета учења и не анализира остале компоненте и модуле система за аутоматизовано машинско учење, већ предлаже да се користе постојећа решења из ове области која су представљена у следећим поглављима.

2.4.1 Карактеристике аутоматизованих система за машинско учење

Аутоматизовани системи за машинско учење могу да се користе у различитим доменима. На пример, у домену обраде слике се користе за класификацију објеката на слици, текстуалне репрезентације слике, детекцију објеката на слици и идентификацију и обележавање објеката на слици [36]. У домену обраде текста, овакви системи могу да се користе за предлагање речи у реченици које недостају, допуњавање реченица, као и мењање редоследа речи у реченици [37]. У већини случајева, овакви системи врше неки облик преноса знања из једног домена у други. Како би системи могли да се имплементирају и користе у различитим доменима, неопходно је дефинисати карактеристике тих система. У зависности од домена у ком се систем користи, постављају се карактеристике система које морају да буду задовољене како би систем могао да функционише.

Две главне карактеристике анализиране за различите домене су време извршавања система и доступни ресурси за рад система. У односу на те карактеристике одређује се архитектура за имплементацију система. Време извршавања система може да буде условљено окружењем у ком се систем користи. Ако окружење у ком се систем користи нема стриктне рокове за израчунавање резултата, одабир архитектуре зависи од доступних ресурса на локацији извршавања. Са друге стране, ако примена система поставља стриктне временске захтеве за извршавање система, неопходно је валидирати да ли постоји могућност извршавања система на локацији извршавања. Ако постоји, потребно је одабрати архитектуру погодну за ту примену. У супротном систем треба креирати тако да задовољи постављене карактеристике коришћењем одговарајуће архитектуре и окружења у ком се систем извршава.

У зависности од локације извршавања система које је одређено доменом примене, различите архитектуре за извршавање система могу да буду погодне. Следеће поглавље анализира различите архитектуре и локације извршавања система кроз аспекте од интереса, као што је време извршавања и доступни ресурси.

2.4.2 Архитектуре и локације извршавања аутоматизованих система за машинско учење

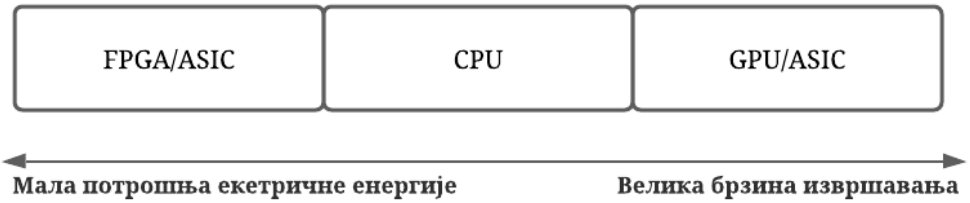
Преглед архитектура и локација извршавања представља битан фактор приликом имплементације аутоматизованог система за машинско учење. Како би се одабрала архитектура погодна за извршавање система и како би систем могао да се користи у датом окружењу неопходно је разумевање различитих архитектура и локација извршавања.

Аутоматизовани систем за машинско учење је неопходно тренирати како би могао да приликом извршавања врши пренос знања и одабир одговарајућег модела на основу претходно стеченог знања. Тренирање алгоритама машинског учења представља временски захтеван задатак у односу на закључивање, што даље може да захтева велике количине ресурса. На основу прегледа радова у отвореној литератури [38], закључује се да је академска заједница фокусирана на тренирање модела, док је индустрија више фокусирана на закључивање и примену модела креирањем прилагођених и хибридних решења, како на софтверском нивоу тако и на хардверском нивоу.

Локације извршавања анализиране у овом раду представљају решење у облаку, које има могућност скалирања ресурса, решење на крајњим уређајима, који имају ограничене ресурсе али су близу извора података и хибридно решење. У зависности од локације извршавања, неопходно је одабрати погодну архитектуру за тренирање и извршавање система. У

зависности од локације извршавања и количине података, анализиране су следеће архитектуре: процесори опште намене (*CPU*), графичке процесорске јединице (*GPU*), поље програмабилних матрица (*FPGA*) и интегрисана кола за специфичне намене (*ASIC*).

За апликације које користе велике количине података, у окружењима са ограниченим ресурсима као што су крајњи уређаји, није могуће тренирати модел због недостатка ресурса. Такође, у апликацијама које раде у реалном времену и које имају стриктне критичне захтеве за кашњење резултата, тренирање система у облаку представља комплексан проблем, пошто се обрада обавља изван локације извршавања. Тренутно, апликације за обраду великих података користе *CPU* и *GPU* архитектуре за тренирање модела у облаку, док су *FPGA* и *ASIC* решења заступљена приликом тренирања модела на крајњим уређајима [39]. На слици 4 су приказане карактеристике различитих архитектура кроз аспекте од интереса као што је потрошња електричне енергије и брзина извршавања. Тренирање модела је процес који захтева више ресурса у односу на закључивање, поготово када се користи велика количина података. Закључивање често има строге захтеве за кашњењем, посебно за апликације у реалном времену.



Слика 4: Однос између времена извршавања и потрошње електричне енергије за различите архитектуре. Потрошња електричне енергије је сразмерна брзини извршавања. Архитектуре су приказане у односу на брзину коју постижу, односни колико ресурса захтевају.

Компаније које пружају решења у облаку креирају системе где је циљ да се они користе као сервиси, као што је случај и са аутоматизованим системима за машинско учење. Како би аутоматизовани систем за машинско учење био у могућности да врши одабир оптималног алгоритма за детекцију аномалија, неопходно је да се такав систем тренира. Тренирање таквог система може да буде временски и просторно захтевно. У ситуацијама када се систем константно тренира са новим скуповима података, као и када је количина података за обраду велика, неопходно је одабрати архитектуру за извршавање таквог система како би се постигле добре перформансе у погледу времена извршавања и доступних ресурса.

Извршавање различитих система у облаку је данас присутно у доста примена јер даје могућност скалирања ресурса по потреби. Међутим, постоји још један тренд који премешта извршавање система из облака ка крајњим уређајима. Рад [40] даје предлог архитектуре система који се извршава на крајњим уређајима, при чему таква архитектура показује предност у односу на решења у облаку тако што се смањује комуникација док време извршавања и перформансе система остају исте. Овакав приступ би могао да буде од великог интереса у удаљеним окружењима где су стриктно ограничени ресурси.

Ако време извршавања система у облаку и време кашњења резултата задовољава постављене захтеве, могуће је имплементирати систем за аутоматизовано машинско учење у облаку. Међутим, у системима који имају критичне захтеве који су у реалном времену, углавном није могуће разматрати решења у облаку због кашњења резултата или могуће грешке у комуникацији. Ако има довољно ресурса да се систем извршава на одабраној локацији, нема

потребе за коришћење решења у облаку. У супротном, неопходно је креирати хибридно решење где се део система налази у облаку док се остатак налази на крајњем уређају.

Класификација различитих архитектура је приказана у табели 4. Свака категорија садржи примере из отворене литературе, ако такви примери постоје. Категорије без примера из отворене литературе представљају категорије коју вреди даље истраживати како би се дошло до корисних закључака. Међутим, то није циљ овог рада, па се такве категорије неће даље анализирати. Може се закључити да различити категорије у класификацији због различитих карактеристика архитектура не дозвољавају директно поређење перформанси система, потребних ресурса и кашњења приликом рачунања резултата.

Табела 4: Преглед различитих локација извршавања и архитектура које могу да се користе за аутоматизоване системе за машинско учење. За сваку категорију је дат по један или више примера из отворене литературе.

Архитектура	Локација извршавања		
	Решење у облаку	Хибридно решење	Решење на крајњем уређају
CPU	*Intel - AI DevCloud, Skylake*	*Intel FRD*	*Pentium M, Core Solo*
GPU	*Amazon EC2*	није доступно	*Nvidia-JetsonTX, Tegra X, Drive*
FPGA	*Baidu XPU*	*hpFog*	*Intel Stratix, Xilinx Virtex*
ASIC	*Google TPU*	није доступно	*IBM TruNorth, Intel Loihi*

На основу наведене анализе и радова из отворене литературе, може да се закључи да оптимално решење у смислу одабира архитектуре и локације извршавања зависи од примене система и окружења у ком се систем извршава. На крајњим уређајима, где су ресурси ограничени, *FPGA* и *ASIC* архитектуре се могу третирати као оптимална за закључивање, због њихове ефикасности и флексибилности у погледу захтеваних ресурса. У решењима која се извршавају у облаку, где могу да се обрађују велике количине података, *GPU* и *ASIC* архитектуре су оптималне за тренирање система, због њиховог капацитета обраде података и могућности да паралелизују извршавање. У табели 5 су поређене различите архитектуре кроз аспекте комплексности.

Табела 5: Поређење различитих архитектура кроз различите аспекте комплексности као што је број транзистора и фреквенција. Наведени аспекти највише утичу на постављање захтева за потребне ресурсе за њихов рад.

Архитектура	*Intel Xeon*	*Tesla V100*	*Xilinx Virtex Ultrascale*	*Google TPU*
Година производње	2016	2017	2014	2017
Тип	*CPU*	*GPU*	*FPGA*	*ASIC*
Технологија креирања	22nm	12nm	22nm	28nm
Фреквенција (*MHz*)	4300	1530	200	700
Тип меморије	*DRAM*	*DRAM*	*BRAM*	*SRAM*
Број транзистора	$7.2\ x\ 10^9$	$21.1\ x\ 10^9$	$20\ x\ 10^9$	$2.5\ x\ 10^9$

2.4.3 Преглед постојећих аутоматизованих система за машинско учење

Аутоматизовани системи за машинско учење се користе у различитим доменима за решавање разноврсних проблема као што је обрада текста и слика, класификација и регресија. Да би се смањили трошкови развоја софтвера за специфичну намену применом машинског

учења, аутоматизовање читавог процеса развоја софтвера смањује трошкове и време развоја. Прегледом отворене литературе је показано да постоје имплементације за специфичну намену, као и за општу намену [41]. У наставку је дат преглед постојећих аутоматизованих система за машинско учење који се користе у индустрији и који су имплементирани као софтвер отвореног кода. Представљена решења могу да се користе за различите типове и домене података. Такође, представљена решења могу да се извршавају на различитим архитектурама и садрже све наведене кораке описане у претходном поглављу.

AutoWEKA [42] представља аутоматизовани систем за машинско учење који је намењен за скупове података опште намене и врши избор алгоритама и параметара алгоритама. У комбинацији са *WEKA* пакетом даје добре предлоге различитих алгоритама за широк спектар типова и домена података. *AutoWEKA* користи Бајесову оптимизацију приликом одабира параметара алгоритма. Циљ је да се корисницима без доменског знања о машинском учењу омогући да ефикасно врше одабир алгоритама и оптимизацију параметара за дате податке и оптимизациону метрику. *Auto-sklearn* представља проширење *AutoWEKA* пакета за коришћење у *Python* програмском језику као замену за имплементацију алгоритама машинског учења. *Auto-PyTorch* представља проширење система који користи комбиновану примену алгоритама машинског учења дубоких неуронских мрежа.

AutoGluon [43] представља аутоматизовани систем за машинско учење који је лако проширив са применама у дубоком учењу и апликацијама из индустрије које обухватају текст, слике и табеларне податке. Систем омогућава једноставно креирање прототипа модела за дубоко учење. *X2O* [44] представља систем који омогућава корисницима без експертског знања у машинском учењу да креирају експерименталне моделе који могу да се користе за евалуацију и тестирање различитих модела над скуповима података. Овај систем је намењен за коришћење у академске сврхе јер не представља стабилан систем за коришћење у индустрији, већ даје могућност за добијање експерименталних резултата брзо. *TPOT* [45] је систем имплементиран у *Python* програмском језику који врши оптимизацију процеса машинског учења помоћу алгоритама генетског учења. *TransmogrifAI* [46] представља аутоматизовани систем за машинско учење имплементиран у *Scala* програмском језику који се користи у *Apache Spark* програмском оквиру. У табели 6 су представљене главне карактеристике наведених аутоматизованих система за машинско учење.

Табела 6: Поређење постојећих система за машинско учење по улазним подацима и компонентама система. Улазни подаци могу да буду табеле, текст или слике. Тип модела може да буде надгледани или ненадгледани.

Систем	Улазни подаци	Припрема података	Одабир атрибута	Тип модела	Евалуација резултата
AutoWEKA	табела	не	да	Надгледани	да
Auto-sklearn	табела	не	да	Надгледани	да
Auto-PyTorch	табела	да	да	Надгледани	да
AutoGluon	табела/текст/слика	да	да	Надгледани	да
X2O	табела	да	да	Надгледани/ Ненадгледани	да
TPOT	табела	не	да	Надгледани	да
TransmogrifAI	табела	да	да	Надгледани	да

2.5 Мета учење у аутоматизованим системима

Мета учење представља механизам учења о процесу учења одређеног система. Мета учење може да се примени приликом одабира атрибута, одабира алгоритама, као и одабира параметара алгоритма за одређени проблем. Циљ мета учења је да креира релације између карактеристика података и перформанси различитих алгоритама, тако да буде предиктивно за перформансе алгоритама. Мета учење се у већини случајева своди на израчунавање мета података који су предиктивни за перформансе алгоритама. То значи да се може направити релација између мета података и перформанси алгоритама који обрађују податке. У овом раду, мета учење се посматра као приступ за одабир оптималног алгоритма у аутоматизованим системима за машинско учење у домену детекције аномалија.

Мета подаци се израчунавају из података коришћењем одговарајућих мета функција. У одређеним случајевима, мета подаци се израчунавају из креираног модела и касније користе за креирање релација између карактеристика модела и перформанси алгоритама. Такође, постоје типови мета учења где се не израчунавају мета подаци, већ се користе прости алгоритми који су предиктивни за перформансе комплекснијих алгоритама. Мета учење је заступљено у надгледаним и ненадгледаним моделима за машинско учење. У оба случаја, мета подаци се користе како би описали главне карактеристике података и тако пренели знање на друге домене где је оно предиктивно за перформансе модела.

У овом раду, мета учење и мета подаци се користе за описивање карактеристика података који се разликују по типовима и доменима. Коришћењем израчунатих мета података одређује се сличност између скупова података коришћењем функција за мерење удаљености. Циљ је да мета подаци карактеризују аномалије у подацима и на тај начин скупове података који имају сличне карактеристике аномалија чине сличним. У неким случајевима, мета подаци се одређују на основу домена у ком треба да се примене. У супротном, користи се предефинисани скуп мета података који је погодан за општи случај коришћења.

Како би се јасно дефинисали мета подаци који су предиктивни за одабир алгоритма, функција (13) представља математичку репрезентацију мета података [47]. Ако је d скуп података за који се израчунавају мета подаци и g функција за израчунавање мета података са параметрима h_i, онда се мета подаци израчунава применом функције g која је предиктивна на перформансе алгоритма P са параметрима h_j.

$$\begin{aligned} d &\in R^{N*M} \\ f(d) &= g(d, h_i) \\ f(d) &\rightarrow P(d, h_j) \end{aligned} \qquad (13)$$

Скуп података d може да се посматра као матрица са N димензија и M атрибута, при чему сваки елемент матрице припада скупу реалних бројева. Функција за израчунавање мета података као резултат даје један или више мета података.

Постојећа решења у области мета података се заснивају на статистичким функцијама и функцијама теорије информација. На основу количине података која је потребна за израчунавање, потребних ресурса за израчунавање, и различитих типова функција које се користе за израчунавање, функције за израчунавање података могу да се поделе у следеће групе: просте функције, статистичке функције, функције засноване на теорији информација, доменске функција и функције које израчунавају мета податке на основу модела. Специјални тип мета учења представља метод обележавања где се уместо израчунавања мета података

примењују једноставне функције које дају резултате перформанси алгоритама који могу да се користе за процењивање перформанси комплексних алгоритама. У наставку овог поглавља је дат преглед постојећих решења на основу наведене класификације.

2.5.1 Мета подаци засновани на простим функцијама

Мета подаци засновани на простим функцијама представљају карактеристике података које могу да се израчунају из података на брз и ефикасан начин, коришћењем простих функција које имају константно време израчунавања. За њихово израчунавање се користе функције које имају малу комплексност и захтевају мало ресурса. Рад [47] даје преглед мета података и врши детаљну анализу различитих типова коришћењем аутоматизованог алата за њихово израчунавање. Пример простих мета података представља број атрибута у скупу података, број инстанци у скупу података, однос различитих типова атрибута, као и однос различитих класа у подацима. Овај тип мета података се добија применом мета функција над подацима при чему оне представљају основне карактеристике података. Како би се добили резултати мета података, није неопходно присуство свих података, већ је могуће на основу димензија података израчунати мета податке. Ови мета подаци се такође називају општим мета подацима јер описују основна својства скупа података.

Примена овог типа мета података је заступљена у аутоматизованим системима за машинско учење за одабир алгоритма, параметара алгоритма, као и за одабир атрибута из скупа података за одређени проблем. Једноставност функција које се користе за израчунавање мета података омогућава њихово коришћење у различитим окружењима док карактеристике које оне описују представљају основне карактеристике које налазе примену у сваком домену.

2.5.2 Мета подаци засновани на статистичким функцијама

Мета подаци засновани на статистичким функцијама су засноване на статистичким својствима података, при чему се израчунавају из података коришћењем статистичких функција које имају линеарно или квадратно време извршавања. За њихово израчунавање користе се функције које могу да имају значајну комплексност и захтевају одређену количину ресурса. Статистичке функције често захтевају параметре и израчунавају се само за нумеричке типове атрибута. Оне чине највећу и најразноврснију групу мета података које могу да се израчунавају по атрибутима одвојено или колективно за више атрибута, при чему се користи нека врста агрегације. Пример статистичких функција представља дистрибуција података, медијана, стандардна девијација, корелација и коваријанса, минимум, максимум и средња вредност [47]. Како би се добили резултати мета података, неопходно је присуство свих података како би се израчунале статистичке карактеристике податка.

Примена овог типа мета података је заступљена у доменима где су неопходне статистичке карактеристике података како би се извршио одабир алгоритма, параметара алгоритма или одабир атрибута из скупа података за одређени проблем. Статистичке функције које се користе за израчунавање мета података омогућавају њихово коришћење у различитим окружењима док карактеристике које оне описују представљају карактеристике које налазе примену у различитим доменима [48, 49].

2.5.3 Мета подаци засновани на функцијама теорије информација

Мета подаци засновани на функцијама теорије информација представљају карактеристике података о количини информација и комплексности података. За њихово израчунавање користе се функције које могу да имају значајну комплексност, које имају линеарно или квадратно време извршавања и захтевају одређену количину ресурса. Функције теорије информације не захтевају параметре и израчунавају се из података коришћењем дискретних вредности. Оне чине значајну групу мета података које могу да се израчунавају по атрибутима одвојено или колективно за више атрибута, при чему се користи нека врста агрегације. Пример функција теорије информација представљају својства као што су ентропија која обухвата количину информација и комплексност података, међусобне информације које углавном одређују однос атрибута и класе која се користи за проблеме класификације [50]. Неопходно је присуство свих података како би се израчунала количина информација у подацима.

Примена овог типа мета података је заступљена у аутоматизованим системима за машинско учење за одабир алгоритма, параметара алгоритма, као и за одабир атрибута из скупа података за одређени проблем. Углавном се користе за представљање различитих образаца понашања [51], за извођење препорука високог квалитета података и за представљање унутрашњих корелација између различитих класа [52].

2.5.4 Мета подаци засновани на доменском знању

Мета подаци засновани на доменском знању представљају карактеристике података које могу да се израчунају из података на брз и ефикасан начин, коришћењем простих функција и доменског знања. За њихово израчунавање се користе функције које имају малу комплексност, захтевају мало ресурса и које имају константно или линеарно време извршавања. Функције су засноване на логичким изразима и садрже доменско знање израчунато или одређено из података. Пример доменских мета података се разликује од домене и представља карактеристике везане искључиво за тај домен. Овај тип мета података се добија применом мета функција над подацима. Како би се добили резултати мета података, није неопходно присуство података, већ је могуће на основу доменског знања одредити карактеристике. Ако подаци нису доступни, доменски експерт или креатор података их може проценити.

Примена овог типа мета података је заступљена само у одређеним доменима за које постоји имплементација аутоматизованих системима за машинско учење. До сада се у домену детекције аномалија нису користили мета подаци засновани на доменском знању. Пример доменских мета података у класификација текста, где је знање засновано на домену представља дужину речника, преклапање речи, број категорија текста, тврдоћу корпуса, ширину домена и слично [53].

2.5.5 Мета подаци засновани на моделима

Мета подаци засновани на моделима садрже својства која описују модел; они се израчунавају применом алгоритама машинског учења, наиме, стабла одлучивања и алгоритама за груписање. За њихово израчунавање се креира модел из ког се добијају карактеристике модела. Овај приступ за креирање модела користи алгоритме који имају значајну комплексност и захтевају одређену количину ресурса. Алгоритми који се користе

захтевају параметре и израчунавају се само за нумеричке типове атрибута. Оне чине посебну групу мета података који се користе за преношење знања са једног модела на други. Пример оваквих мета података представља број листова, број чворова, дубина и ширина дрвета коришћењем стабла одлучивања [47]. Како би се добили резултати мета података, неопходно је присуство свих података да би се креирао модел. Слично томе, радови [54], [55] анализирају мета податке за одређивање броја кластера у подацима тако што предлажу функције које описују структуре кластера у подацима.

Још једна група која израчунава мета податке из модела представља метод обележавања (енг. *Landmarking*). Метод обележавања је посебан случај мета учења који описује карактеристике података користећи просте алгоритме. Резултат мета учења није издвајање мета података из података или модела, већ је предвиђање који ће алгоритам пружити добре перформансе за дати скуп података применом простих алгоритама над подацима. Коришћени алгоритми се заснивају на алгоритмима за класификацију и груписање. Пример алгоритма је *EliteNN* који је заснован на *1NN* алгоритму за класификацију где се користи над редукованим бројем атрибута из података [56].

2.5.6 Класификација постојећих функција за израчунавање мета података

Наведена класификација представља анализу различитих приступа мета учења коришћењем мета података који се добијају применом одговарајућих функција над подацима или моделима, као и коришћењем једноставних функција које су предиктивне за перформансе комплексних алгоритама. Поред наведених приступа мета учења постоје и нови приступи засновани на техникама трансформације података, при чему се посматрају промене у понашању алгоритама машинског учења применом трансформација [57]. Мета учење се такође користи и за одабир атрибута који су предиктивни за перформансе модела [58] [59]. У табели 7 је дат преглед решења за израчунавање мета података по начина добијања мета података, комплексности функција, као и количини информација потребних за израчунавање.

Табела 7: Преглед мета података по типу, да ли су до сада коришћени у домену детекције аномалија, нивоу потребних информација и количини података потребних за израчунавање. Важно је напоменути да се мета подаци засновани на доменском знању нису користили у домену детекције аномалија до сада. Ниво информација потребних за израчунавање мета података подељен је у следеће категорије од најмање до највеће, при чему свака категорија имплицитно укључује претходне категорије: (I) потребно је само познавање домена, (II) потребна је информација о количини података, (III) потребни су типови атрибута, (VI) потребна је дистрибуција података, (V) потребан је подскуп података и (VI) потребан је цео скуп података. Мета подаци израчунати из података описују карактеристике података и директно се израчунавају из података. Мета подаци израчунати из модела описују карактеристике модела који је креиран коришћењем података, што значи да се индиректно израчунавају из података.

Мета подаци	Извор мета података	Коришћени у домену детекције аномалија	Подаци потребни за израчунавање	Велика количина података потребна	Значајна комплексност израчунавања	Референце
Прости мета подаци	подаци	да	III - типови атрибута	не	не	[47] [48]
Статистички мета подаци	подаци	да	IV - дистрибуција података	да	да	[47] [48] [50] [49]
Мета подаци засновани на теорији информација	подаци	да	V - подскуп података	да	да	[51] [52]
Доменски мета подаци	подаци	не	II - количина података	не	не	[53]
Мета подаци засновани на моделу	модел	не	VI - скуп података	да	да	[55] [54]
Метод обележавање	модел	да	V - подскуп података	да	не	[56]

2.6 Мерење сличности између мета података

У алгоритмима машинског учења, проблем мерења сличности односно удаљености представља битан аспект који је предиктиван за перформансе алгоритма и захтева детаљну анализу. Одабиром адекватне функције за мерење удаљености постижу се добре перформансе алгоритма тако што се користи оптимална метрика за мерење сличности између података. Удаљеност између два скупа података означава метрику која одређује сличност, односно разлику између тих скупова података, кроз карактеристике података у одређеном домену. У домену детекције аномалија те карактеристике могу да буду тип аномалија, локалитет аномалија или димензионални простор аномалија. За ту сврху се углавном користе функције за мерење удаљености. Примери алгоритама који користи функције за мерење удаљености су алгоритми засновани на густини простора. У тим алгоритмима се удаљеност мери тако што се рачуна удаљеност по свим атрибутима инстанце. Приликом рачунања удаљености, атрибути инстанци могу да буду различитих типова и интервала вредности. Одређене функције за мерење удаљености захтевају рачунање удаљености по атрибутима који немају исте типове података. То даље доводи да одређени атрибути имају већи утицај на резултат од других атрибута. Пример је рачунање удаљености између инстанци које имају номиналне и

нумеричке атрибуте, при чему нумерички атрибути имају већи утицај на резултат. Како би се решио тај проблем добра пракса је да се вредности нормализују и стандардизују пре примене функције за рачунања удаљености. Стандардизација значи да је опсег вредности атрибута стандардизован да би се измерило колико је стандардних девијација атрибут инстанце удаљен од средње вредности.

Један пример коришћења удаљености у алгоритмима машинској учења представља рачунање грешке модела. Приликом евалуације резултата модела неопходно је одредити начин за мерење одступања добијеног резултата у односу на жељену вредност. То може да се врши рачунањем средње вредности свих резултата који се затим упоређује са средњом вредношћу жељених резултата.

Мерење удаљености представља битан модул у аутоматизованим системима за машинско учење приликом одабира алгоритма. За одабир алгоритма се користе мета подаци где је циљ преношење претходно стеченог знања на основу мерења сличности између скупова података. Мерење удаљености или мерење сличности представља одређивање скупова података који су близу у одређеном димензионалном простору на основу карактеристика тих података, што су мета подаци.

Након прегледа отворене литературе закључено је да се функције за мерење сличности између података заснивају на простим функцијама за мерење удаљености које се користе у различитим доменима. У следећим поглављима је дат преглед постојећих функција за мерење удаљености где је представљена математичка анализа и случајеви коришћења. Наведене функције за мерење удаљености се могу користити над подацима, али и над скалираним подацима тако што се додаје фактор на вектор у подацима. На тај начин даје предности појединим векторима који утичу на удаљеност података. У супротном, ако је потребно да се сви атрибути података посматрају једнако, неопходно је урадити стандардизацију над подацима.

2.6.1 Еуклидска функција за мерење удаљености

Еуклидска (енг. *Euclidean*) функција за мерење удаљености представља растојање између два вектора. Растојање између два вектора се рачуна као удаљеност координата у Декартовском координатном систему. Број димензија је условљен величином вектора. Применом функције (14) се рачуна удаљеност између две координате где p и q представљају координате у n-димензионалном простору, q_i и p_i представљају елементе вектора, док n представља димензионални простор.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \qquad (14)$$

На основу функције, удаљеност се рачуна као корен квадрата разлике вредности елемената вектора. Елементи вектора могу да буду представљени коришћењем различитих јединица и могу да имају различите опсеге. Ако би се функција применила на такве податке, елементи који имају већи интервал вредности би имали и већи утицај на резултат. У том случају је неопходно нормализовати податке пре примене наведене функције како би сви елементи вектора имали једнак утицај на резултат. У случајевима са доста итерација, неопходно је смањити комплексност функције избацивањем корена. Димензионални простор

у случају мерења сличности између скупова података представља број мета података који се користи за поређење или број атрибута у скупу података ако се сличност мери између самих података.

Еуклидска функција за рачунања удаљености се такође зове и Л2 удаљеност која је доста је заступљен у алгоритмима машинског учења. У домену аутоматизованих система за машинско учење, функција се користи за мерење сличности односно удаљености између скупова података. Такође, у алгоритмима за машинско учење, ова врста мерења удаљености може да се користи за рачунање грешке резултата тако што се рачуна удаљеност између добијеног резултата и очекиваног резултата.

2.6.2 *Manhattan* функција за мерење удаљености

Manhattan функција за мерење удаљености представља растојање између два вектора. Растојање између два вектора се рачуна као збир апсолутних разлика елемената вектора. Број димензија је условљен величином вектора. Применом функције (15) се рачуна удаљеност између две координате где p и q представљају координате у n-димензионалном простору, q_i и p_i представљају елементе вектора, док n представља димензионални простор.

$$d(p,q) = \sum_{i=1}^{n} |q_i - p_i| \qquad (15)$$

Елементи вектора могу да буду представљени коришћењем различитих јединица и могу да имају различите интервале вредности. Ако би се функција применила на такве податке, елементи који имају веће интервале би имали и већи утицај на резултат, па је због тога неопходно нормализовати податке пре примене наведене функције како би сви елементи вектора имали једнак утицај на резултат. *Manhattan* функција за рачунања удаљености је погодна када вектори представљају униформну матрицу целобројних вредности. *Manhattan* дистанца представља рачунање удаљености између два вектора, што у овом случају могу да представљају атрибути скупова података или мета подаци израчунати из података.

Овај тип рачунања дистанце се такође зове и Л1 удаљеност и доста је заступљен у алгоритмима машинског учења. У односу на Еуклидску удаљеност, ова функција је погодна ако се број елемената у вектору повећава. Ако су вектори за поређење вишедимензионални при чему је број димензија велики, овај тип функције даје боље резултате јер има мању комплексност израчунавања у поређењу са претходном функцијом.

2.6.3 *Hamming* функција за мерење удаљености

Hamming функција за мерење удаљености представља растојање између два вектора који имају бинарне вредности. Растојање између два вектора се рачуна као збир елемената који се разликује у векторима. Број димензија је условљен величином вектора. Применом функције (16) се рачуна удаљеност између два бинарна вектора где p и q представљају векторе у n-димензионалном простору, q_i и p_i представљају елементе вектора, док n представља димензионални простор.

$$d(p,q) = \sum_{i=1}^{n} q_i <> p_i \qquad (16)$$

Hamming функција за рачунање удаљености је погодна када вектори представљају номиналне или текстуалне вредности, при чему сваки елемент представља један карактер. Ако су текстуалне вредности идентичне, удаљеност ће бити *0*. Овакве врсте вектора су заступљене приликом енкодовања података. Представљена функција може да се посматра као посебан случај мерења удаљености где се користе бинарне вредности. Функција за мерење удаљености има широку примену у машинском учењу, приликом обраде текста и приликом тренирања бинарних неуронских мрежа [60]. Због карактеристика функције, закључено је да није погодна за коришћење у овом раду, тако да се она даље неће анализирати.

2.6.4 *Minkowski* функција за мерење удаљености

Minkowski функција за мерење удаљености представља апстракцију претходних функција за мерење удаљености тако што уводи додатни параметар који представља степен функције. Коришћењем функције могу да се изведу претходне Л функције за мерење удаљености. Број димензија је условљен величином вектора док је степен одређен параметром k. Применом функције (17) се рачуна удаљеност између две координате где p и q представљају координате у n-димензионалном простору, q_i и p_i представљају елементе вектора, док n представља димензионални простор.

$$d(p,q) = \sqrt[\frac{1}{k}]{\sum_{i=1}^{n} (q_i - p_i)^k} \qquad (17)$$

Minkowski функција за мерење удаљености има широку примену у машинском учењу и због начина параметризације, и може да се користи у различитим доменима. Због карактеристика функције, закључено је да није погодна за коришћење у овом раду, тако да се она даље неће анализирати. Функција је погодна за оптимизацију у генетским алгоритмима [61].

2.6.5 Класификација постојећих функција за мерење удаљености

Наведене функције за мерење удаљености представљају заступљене функције из отворене литературе које се користе у алгоритмима машинског учења за мерење сличности између података, као што су алгоритми класификације и груписања засновани на густини. Такође, наведене функције се користе за евалуирање резултата тако што мере удаљеност између добијене и очекиване вредности.

Због мале комплексности наведене функције су погодне за коришћење у алгоритмима машинског учења. Поред ових алгоритама, алгоритми машинског учења могу да се користе за мерење удаљености и сличности. На пример, неуронска мрежа може да се користи за мерење удаљености између скупова података тако што се тренирањем модела креира очекивани шаблон понашања који касније може да се користи приликом одлучивања. Прегледом отворене литературе је закључено да је добра пракса коришћење простих функција за мерење

удаљености, како не би утицале на перформансе модела. Коришћењем комплексних функција за мерење удаљености ствара се опасност од мењања релација у подацима и самим тим шаблона понашања, што може да утиче на перформансе модела. Мерења сличности између података представља битан модул у алгоритмима машинског учења и овај рад у наставку дефинише нове функције за мерење удаљености коришћењем постојећих алгоритама машинског учења. У табели 8 је дат приказ функција за мерење удаљености кроз аспекте комплексности, домена коришћења и типа улаза.

Табела 8: Функције за мерење сличности кроз аспекте типа улаза, комплексности и домена коришћења. Функције имају малу комплексност што их чини погодним за коришћење у алгоритмима машинског учења, јер не утичу на шаблоне понашања трансформацијама над подацима.

Функција за мерење удаљености	Тип улаза	Комплексност	Домен коришћења	Референце
Еуклидска	Вектори	*O(n)*	Геометрија, машинско учење, мерење сличности	[62]
Manhattan	Матрица целобројних вредности	*O(n)*	Геометрија, матрице, машинско учење	[62]
Hamming	Текст, бинарни вектори	*O(n)*	Бинарна класификација, енкодовање текста	[60]
Minkowski	Вектор или матрица целобројних вредности	*O(n)*	Геометрија, матрице, машинско учење	[61]

3 Предлог решења

У овом поглављу се даје предлог функција за израчунавање мета података на основу доменског знања за одабир алгоритама за детекцију аномалија. Прво је представљена потреба за дефинисањем новог скупа мета података за детекцију аномалија. Постојећа решења се заснивају на функцијама за израчунавање мета података које имају значајну комплексност и нису прилагођене за детекцију аномалија у аутоматизованим системима за машинско учење. Аутоматизовани системи за детекцију аномалија постављају захтеве који морају да буду испуњени како би могли да се користе у различитим окружењима, а односе се на перформансе и скалабилност компонената. Након тога је дат преглед критичних захтева које мета подаци морају да испуне како би могли да се користе у аутоматизованим системима за машинско учење. Затим је представљено предложено решење и дефинисане су главне карактеристике решења.

У аутоматизованим системима за машинско учење, одабир алгоритма се врши на основу мета података и претходно стеченог знања. Циљ је предложити алгоритам који ће за дати проблем и оптимизациону метрику дати најбоље могуће перформансе. У одређеним случајевима, скупови података са аномалијама нису обележени па није могуће применити алгоритме класификације за тренирање модела и детекцију аномалија, већ је неопходно користити овакве системе за одабир оптималног алгоритма.

3.1 Потреба за дефинисањем новог скупа мета података

Како би се решио проблем одабира алгоритма за детекцију аномалија, радови из отворене литературе се баве различитим приступима [63]. Један од приступа представља примену методе исцрпног тестирања, где се тестирањем свих постојећих комбинација алгоритама долази до оптималног алгоритма. Овакав приступ је показао да методе исцрпног тестирања захтевају велике количине података и ресурса како би могли да тестирају све постојеће алгоритме. Аутоматизовани системи за машинско учење у зависности од локације извршавања и случаја коришћења могу да имају стриктне захтеве за време извршавања и доступне ресурсе, па овакав приступ није погодан за коришћење у аутоматизованим системима.

Други приступ представља примену мета података за одабир оптималног алгоритма, и овај приступ је детаљно представљен и анализиран у наставку. Постојећи мета подаци који се користе за одабир алгоритама представљају статистичке карактеристике података и количину информација у подацима, и не дефинишу карактеристике аномалија у подацима. Са друге стране, мета подаци који се израчунавају коришћењем простих функција могу брзо да израчунају мета податке али углавном не дају добре резултат јер не могу да карактеризују комплексне шаблоне понашања. Приступи засновани на мета подацима захтевају мање обраде и мање ресурса у односу на приступ исцрпног тестирања. Међутим, постојећа решења не могу једноставно да се примене на проблем детекције аномалија у подацима, јер су прилагођени за обраду текста и слика.

Детекција аномалија у подацима побољшава квалитет података и представља неопходан корак у ситуацијама када се анализирају хетерогени подаци који са одређеном фреквенцијом долазе са неког извора или када се детектују правила понашања у временским сегментима. Ако овакви системи раде у реалном времену, захтеви су стриктни и није могуће применити методе које захтевају велике количине ресурса или података.

3.1.1 Дефинисање критичних захтева за израчунавање мета података

Постојећи мета подаци за предлог алгоритма се заснивају на статистичким функцијама и функцијама теорије информација које захтевају велике количине података. Аутоматизовани системи за машинско учење у зависности од случаја коришћења постављају захтеве везане за перформансе и скалабилност који морају да буду испуњени како би могли да се ефикасно користе. Мета подаци коришћени за одабир алгоритма треба да дају добре перформансе за различите типове података и оптимизационе метрике. Такође, у ситуацијама када обележени подаци нису доступни, што је чест случај у домену детекције аномалија, доменски експерт треба да буде у могућности да представи карактеристике аномалија са одређеном грешком. На основу свега наведеног, неопходно је дефинисати критичне захтеве који ће се касније користити приликом валидације предложеног решења, како би се омогућило коришћење мета података у аутоматизованим системима за машинско учење.

Први критични захтев се односи на разноврсност улазних података при чему је неопходно да мета подаци карактеризују аномалије у различитим типовима података и за различите домене података како би могли да се користе у аутоматизованим системима. У супротном, систем би давао добре перформансе искључиво у одређеним случајевима, што се своди на креирање модела за тај одређени случај и не задовољава карактеристике аутоматизованих система за машинско учење. Следећи критични захтев се односи на разноврсност оптимизационих метрика за које се тражи оптимални алгоритам. Неопходно је да аутоматизовани систем даје добре перформансе за различите оптимизационе метрике како би могао да се користи за различите проблеме. Затим, у зависности од локације извршавања, неопходно је користити мета податке који не захтевају велику количину ресурса како би систем могао да се извршава и задовољи захтеве дефинисане од окружења. На крају, ако подаци нису обележени, потребно је омогућити да се на основу знања доменског експерта израчунају мета подаци за предлог алгоритма који ће ефикасно вршити детекцију аномалија.

Предложени критични захтеви које функције за израчунавање мета података треба да испуне, а односе се на скалабилност и перформансе су:

- Неутралност – мета подаци треба да описују карактеристике аномалија за различите области и типове података; алгоритми одабрани на основу мета података за различите области и типове података треба да имају могућност да постижу задовољавајуће резултате. Ако се покаже да предложени мета подаци дају задовољавајуће резултате само за одређену област или тип података, сматра се да мета подаци не испуњавају овај критични захтев.
- Скалабилност – функције за израчунавање мета података треба да буду једноставне за израчунавање и без сложених операција; комплексност мета података треба да буде линеарна.
- Повезаност – мета подаци треба да ефикасно описују карактеристике аномалија за различите оптимизационе метрике; алгоритми одабрани на основу мета података за различите оптимизационе метрике треба да имају могућност да постижу задовољавајуће резултате. Ако се покаже да предложени мета подаци дају задовољавајуће резултате само за одређену оптимизациону метрику, сматра се да мета подаци не испуњавају овај критични захтев.
- Једноставност – креатор података или доменски експерт треба да има могућност да израчуна мета податке у ситуацијама када подаци нису обележени или нису присутни; Случај коришћења укључује извор података који генерише податке са одређеном

фреквенцијом или креирање система за податке који тек треба да буду генерисани.

Чињеница је да још није предложен проширив скуп функција за израчунавање мета података за одабир алгоритма у домену детекције аномалија на основу доменског знања, који ће смањити комплексност и потребне ресурсе за израчунавање. Коришћењем мета података заснованих на доменском знању омогућило би се ефикасно обезбеђивање резултата за дати проблем и оптимизациону метрику, што би убрзало процес учења и утицало на перформансе аутоматизованог система. Како би се дефинисали мета подаци на основу доменског знања, неопходно је поставити полазне хипотезе које ће се касније валидирати а односе се на мета податке који би се користили у аутоматизованим системима за машинско учење.

3.1.2 Полазне хипотезе

Након прегледа отворене литературе, дефинисања проблема и креирања критичних захтева који морају да буду испуњени за мета податке како би се користили у аутоматизованим системима за машинско учење, неопходно је дефинисати полазне хипотезе како би се јасно дефинисало шта све предложено решење треба да испуни. Полазне хипотезе ће се валидирати у експериментима који ће дати одговоре на питања, која могу да буду корисна за даља истраживања у овој области. Полазне хипотезе у овом раду су дефинисане на следећи начин:

- Могуће је предложити скуп мета података заснованом на доменском знању за карактеризацију аномалија у подацима
- Предложени скуп мета података може да задовољи наведене критичне захтеве
- Предложени скуп мета података постиже исте или боље резултате у односу на постојеће скупове мета података за различите типове података, различите типове аномалија, као и за различите области из којих подаци долазе
- Предложени скуп мета података је могуће проценити искључиво на основу доменског знања и без присуства података
- За предложени скуп мета података је могуће одредити тип функција за мерење удаљености између скупова података који у већини случајева даје боље резултате од осталих типова функција које се најчешће користе за мерење удаљености између скупова података

Прегледом отворене литературе закључено је да постојећа решења испуњавају критичне захтеве везане за неутралност и повезаност. Постојећа решења заснована на простим мета подацима испуњавају критичне захтеве везане за скалабилност. Постојећа решења заснована на статистичким функцијама, и функцијама теорије информација не испуњавају критичне захтеве везане за скалабилност јер је комплексност функција квадратна и могу да захтевају значајне количине ресурса за израчунавање . Критични захтев везан за једноставност до сада није анализиран у отвореној литератури, па се сматра да постојећа решења не испуњавају наведени критични захтев. Међутим, прости мета подаци не захтевају захтевну обраду над подацима па се сматрају да на основу знања доменског експерта могу да се процене са одређеном грешком, и тако испуњавају наведени критични захтев. Важност горе наведених захтева ће током времена расти заједно са количином података који се генеришу, јер квалитет анализе и обраде података зависи од одабраног модела.

3.2 Предлог функција за израчунавање мета података на основу доменског знања

Потреба за систематским приступом за одабир алгоритма за детекцију аномалија који ће да се користи у системима за аутоматизовано машинско учење отвара могућност за креирање мета података на основу доменског знања у домену детекције аномалија. Анализом постојећих решења показано је да постоји могућност да постојећа решења не задовољавају један или више критичних захтева. Како би се задовољили наведени критични захтеви, овај рад предлаже скуп мета података заснованих на доменском знању. Доменско знање аномалија обухвата карактеристике аномалија које могу да се користе за обележавање аномалија у подацима [64]. Карактеристике података су представљене у претходном поглављу и обухватају локалитет аномалија и димензионални простор аномалија. На основу тих података могу да се разликују типови аномалије у подацима. Такође, поред карактеристика аномалија, могу да се користе прости мета подаци као што је број аномалија у подацима, тип података и домен података.

Ако је d скуп података са n инстанци, свака инстанца садржи $x = [v1, v2, v3, \dots]$ вектор са m атрибута и опционом класом која обележава да ли је та инстанца аномалија. Мета податак c може да се дефинише као резултат функције $f(d) = c$ која када се примени на скуп података d израчунава вектор вредности које представљају мета податке скупа података d. Израчунати мета подаци утичу на одабир алгоритма за детекцију аномалија када је у питању d скуп података. Предложени мета подаци који описују карактеристике аномалија су представљени у следећим поглављима.

3.2.1 Предлог функције за израчунавање локалитета аномалија

У зависности од средине у којој се аномалије налазе, оне могу да буду глобалне аномалије, локалне аномалије и микро-кластери. Инстанце које имају екстремне вредности у поређењу са осталим инстанцама су означене као глобалне аномалије. Инстанце са вредностима које се разликују од своје околине у n-димензионалном простору, али су у просеку вредности свих података су означене као локалне аномалије. Инстанце које имају одступања у вредностима у односу на дистрибуцију података и имају суседе са сличним карактеристикама представљају микро-кластере. Ова карактеристика аномалија је предложена да се посматра као мета податак јер може да буде предиктивна за перформансе алгоритма. Локалитет аномалија не мора да буде јединствен за скуп података, што значи да се у скупу података може појавити више од једног типа локалитета аномалије. У скупу података d локалитет аномалије се израчунава применом функције (18), где је i инстанца података за коју се рачуна локалитет аномалија, d скуп података, ε околина око инсанце из скупа података, λ гранична вредност између нормалних инстанци и аномалија, $score$ функција која одређује колико је нека инстанца аномалија. Гранична вредност се одређује на основу узорка обележених аномалија у скупу података или на основу доменског знања [65].

$$loc(i,d) = \begin{cases} global, & \text{if } score(i,d) \geq \lambda \\ local, & \text{if } score(i,\varepsilon(i,d)) \geq \lambda \text{ and } score(i,d) < \lambda \\ cluster, & \text{if } score(i,d) \geq \lambda \text{ and } score(\varepsilon(i,d),d) > \lambda \end{cases} \quad (18)$$

Мета податак заснован на локалитету аномалија захтева малу количину означених података како би могао да се израчуна. У случајевима када не постоји узорак података са обележеним аномалијама, или подаци нису присутни, креатор података или доменски експерт ће моћи ефикасно да карактеризује аномалије у подацима искључиво на основу доменског знања. На пример, глобалне аномалије се разликују од остатка дистрибуције као појединачне инстанце и оне представљају екстремне вредности у односу на све инстанце у скупу података. Локалне аномалије су инстанце са већим одступањем у поређењу само са суседним инстанцама у дистрибуцији. Микро-кластери имају одступање веће од нормалних инстанци и заједно са суседима који имају сличне карактеристике креирају микро-кластер. Микро-кластери често указују на новину у подацима и представљају нову групу која до тог тренутка није била откривена.

Како би дефинисани мета податак могао да се користе у аутоматизованим системима за машинско учење, неопходно је валидирати да испуњава дефинисане критичне захтеве. Експерименти су дизајнирани тако да валидирају критичне захтеве, док ће се у наставку дати кратка анализа за сваки од критичних захтева. Први критични захтев се односи на неутралност, где за различите типове података и домене из којих подаци долазе могу да се јаве наведени локалитети аномалија. Други критични захтев представља скалабилност, где предложена функција за израчунавање не садржи комплексне итерације и није неопходно итерирати кроз све инстанце како би се закључило који све типови аномалија постоје, већ је могуће узети мали узорак обележених аномалија. Трећи критични захтев се односи на повезаност, и он ће се валидирати у експериментима. На крају, једноставност је постигнута могућношћу да мета подаци буду процењени на основу доменског знања експерта из наведене области, који може да процени који локалитет аномалија може да се очекује у подацима.

3.2.2 Предлог функције за израчунавање димензионалног простора аномалија

У зависности од броја атрибута у скупу података у ком се детектују, аномалије могу да буду представљене у једнодимензионалном и вишедимензионалном простору. Детектовање аномалија у једнодимензионалном простору захтева креирање модела за сваки атрибут у скупу података. У супротном, креира се један модел за све атрибуте у подацима који се користи за детекцију аномалија. Детекција аномалија у једнодимензионалном простору је заснована на дистрибуцији података једног атрибута где аномалије често представљају екстремне вредности или грешке у временским подацима. Такве аномалије се могу ефикасно детектовати коришћењем пробабилистичких и статистичких метода. Детекција аномалија у вишедимензионалном простору се заснива на дистрибуцији података n-тог простора атрибута. Већина постојећих примера детекције аномалија у индустрији заснована је на вишедимензионалном простору, где се аномалије описују као понашање бар два атрибута у скупу података. Наведена карактеристика аномалија је предложена да се посматра као мета податак јер може да буде предиктивна за перформансе алгоритма. Димензионални простор аномалија је јединствен за скуп података. Одређеним трансформацијама над подацима димензионални простор може да се трансформише из вишедимензионалног у једнодимензионални при чему су карактеристике аномалија трансформисане у суб-димензионални простор. У скупу података d локалитет аномалија се израчунава применом функције (19), где је $attrNum$ функција за одређивање броја атрибута у скупу података.

$$space(d) = \begin{cases} univariate, & \text{if } attrNum(d) = 1 \\ multivariate, & \text{if } attrNum(d) > 1 \end{cases} \quad (19)$$

Мета подаци засновани на димензионалном простору аномалија захтевају искључиво број атрибута у скупу података. У случајевима када не постоје подаци за које ће да се врши детекција аномалија, креатор података или доменски експерт ће моћи ефикасно да одреди број атрибута искључиво на основу доменског знања.

Како би предложени мета податак могао да се користе у аутоматизованим системима за машинско учење, неопходно је валидирати да испуњава дефинисане критичне захтеве. Експерименти су дизајнирани тако да валидирају критичне захтеве, док је у наставку дата кратка анализа за сваки од критичних захтева. Први критични захтев се односи на неутралност, где за различите типове података и домене из којих подаци долазе може да се јави различити димензионални простор аномалија. Други критични захтев представља скалабилност, где предложена функција за израчунавање не садржи комплексне итерације и није неопходно итерирати кроз скуп података, већ је могуће узети само димензије скупа података. Трећи критични захтев се односи на повезаност, и он ће се валидирати у експериментима. На крају, једноставност је постигнута могућношћу да мета податак буду одређен на основу доменског знања експерта из наведене области.

3.2.3 Предлог функције за израчунавање броја аномалија

Однос аномалија у подацима се добија коришћењем просте функције која израчунава број инстанци које су аномалије у односу на укупан број инстанци у скупу података. Ова карактеристика аномалија је предложена да се посматра као мета податак јер може да буде предиктивна за перформансе алгоритма. Идеја за предлог овог мета податка долази из простих мета података где се израчунава однос између две класе у бинарној класификацији. У скупу података d број аномалија се израчунава применом функције (20), где је $anomNum$ функција која даје број инстанци које су означене као аномалије, док је $instNum$ функција за одређивање броја инстанци у скупу података.

$$ratio(d) = \frac{anomNum(d)}{instNum(d)} \quad (20)$$

Мета податак заснован на броју аномалија захтева искључиво број аномалија у скупу података. Ако скуп података са означеним аномалијама није доступан у тренутку одабира алгоритма или се број аномалија у подацима мења кроз време, доменски експерт или креатор података може да процени овај мета податак.

Како би предложени мета податак могао да се користи у аутоматизованим системима за машинско учење, неопходно је валидирати да испуњава дефинисане критичне захтеве. Експерименти су дизајнирани тако да валидирају критичне захтеве, док је у наставку дата кратка анализа за сваки од критичних захтева. Први критични захтев се односи на неутралност, где за различите типове података и домене из којих подаци долазе могу да се јаве различите количине аномалија. Други критични захтев представља скалабилност, где предложена функција за израчунавање не садржи комплексне итерације и није неопходно итерирати кроз скуп података, већ је могуће узети само димензије скупа података. Трећи критични захтев се односи на повезаност, и он ће се валидирати у експериментима. На крају, једноставност је

постигнута могућношћу да мета податак буду процењен на основу доменског знања експерта из наведене области.

3.2.4 Предлог функције за израчунавање типа података

Коришћењем мета података за одабир алгоритма на основу података се врши преношењем знања између скупова података. Тако се дошло на идеју да се тип података користи као мета податак који ће омогућити дељење знања између истих типова података тако што ће они имати исту вредност мета податка. Тип података је мета податак која омогућава повезивање скупова података са истим типовима података. Након прегледа отворене литературе, закључено је да се скупови података који се користе за детекцију аномалија могу класификовати у следеће категорије према типу података: номинални, временски, просторни и вишедимензионални подаци. Ако скуп података садржи атрибуте са одређеним типом података, а ти атрибути су укључени у димензионални простор детекције аномалија, може се сматрати да скуп података припада том типу података. Тип података не мора да буде јединствен за скуп података, што значи да се у скупу података може појавити више од једног типа. У скупу података d тип података се израчунава применом функције (21), где је δ гранична вредност за број атрибута у вишедимензионалним подацима, док наведене функције одређују присуство различитих типова података.

$$data\_type(d) = \begin{cases} nominal, & \text{if } hasText(d) \\ temporal, & \text{if } hasTime(d) \\ spatial, & \text{if } hasCoord(d) \\ high\_dim, & \text{if } attrNum(d) \geq \delta \end{cases} \quad (21)$$

Мета податак заснован на типу захтева малу количину података како би могао да се израчуна. Како би предложени мета податак могао да се користи у аутоматизованим системима за машинско учење, неопходно је валидирати да испуњава дефинисане критичне захтеве. Експерименти су дизајнирани тако да валидирају критичне захтеве, док је у наставку дата кратка анализа за сваки од критичних захтева. Први критични захтев се односи на неутралност, где за различите домене података може јавити различити тип података. Други критични захтев представља скалабилност, где предложена функција за израчунавање не садржи комплексне итерације и није неопходно итерирати кроз све инстанце како би се закључило који све типови података постоје, већ је могуће узети мали узорак података. Трећи критични захтев се односи на повезаност, и он ће се валидирати у експериментима, што се односи на чињеницу да алгоритми треба да буду одабрани тако да дају добре резултате за различите типове и домене података. На крају, једноставност је постигнута могућношћу да мета податак буде одређен на основу доменског знања експерта из наведене области који може да одреди који се тип података очекује. Како би се представили различити типови података који се користе у детекцији аномалија, постоји велики број радова који сумирају главне разлике између типова и представљају алгоритме за детекцију аномалија [65] [26] [66] [67] [68] [69] [70] [71] [72] [65] [73].

3.2.5 Предлог функције за израчунавање мета података на основу домена података

Домен података у коме се детектују аномалије је предложен да се посматра као мета податак јер може да буде предиктиван за перформансе алгоритма. Оваквим мета податком се постиже дељење знања између података из истог домена тако што ће они имати исту вредност мета податка. Домен података представља мета податак која омогућава повезивање скупова података који долазе из истог домена. Након прегледа отворене литературе, закључено је да се домени података у којима се појављују аномалије и постоје јавно доступни обележени подаци могу класификовати у следеће категорије према домену података: производња, транспорт, финансије, медицина, обрада текста, софтверски логови и друштвене мреже у облику графова. У скупу података d домен података се израчунава применом функције (22), где наведене функције одређују домене података.

$$data\_domain(d) = \begin{cases} manuf, & \text{if } isManuf(d) \\ transp, & \text{if } isTransp(d) \\ finance, & \text{if } isFinance(d) \\ medicine, & \text{if } isMedicine(d) \\ text, & \text{if } isText(d) \\ software, & \text{if } isSoftware(d) \\ graph, & \text{if } isGraph(d) \end{cases} \quad (22)$$

Ако скуп података описује податке из одређеног домена, може се сматрати да скуп података припада том домену података. Домен података не мора да буде јединствен за скуп података, што значи да скуп података може да долази из неколико домена.

3.2.6 Карактеристике предложених функција за израчунавање мета података

Мета подаци су одабрани на начин да имају малу комплексност приликом израчунавања и да се заснивају на доменском знању. Доменско знање мета података омогућава доменском експерту да израчуна или процени мета податке. Линеарна комплексност мета података омогућава ефикасно израчунавање и извршавање у окружењима која имају ограничене ресурсе. Предложени мета подаци треба да испуњавају постављене критичне захтеве везане за скалабилност и једноставност коришћењем простих функција за израчунавање мета података заснованих на доменском знању. Експерименти у овом раду ће анализирати испуњеност критичних захтева и дати одговор на дефинисане полазне хипотезе. У табели 9 су представљене карактеристике предложених функција за израчунавање мета података на основу доменског знања.

Табела 9: Карактеристике предложених функција за израчунавање мета података на основу доменског знања. Тип мета податка се означава којој групи мета податак припада, док подаци потребни за израчунавање означавају са којом количином података могу да се одреде мета подаци.

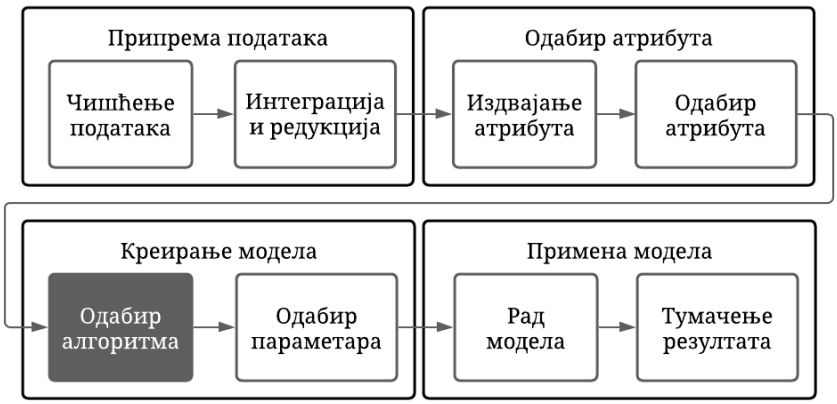
Предложени мета подаци	Тип мета података	Подаци потребни за израчунавање
Локалитета аномалија	Доменски мета податак	скуп података
Димензионални простор аномалија	Доменски мета податак	број атрибута
Број аномалија	Доменски мета податак/прост мета податак	скуп података
Тип података	Доменски мета податак/прост мета податак	тип атрибута
Домен података	Доменски мета податак/прост мета податак	тип атрибута

4 Пројектовање компоненте за одабир алгоритма у аутоматизованим системима за машинско учење

У овом поглављу је представљена имплементација предложене компоненте за одабир алгоритама у аутоматизованим системима за машинско учење. Прво је дат преглед компоненти које чине систем. Затим је дата логичка структура компоненте за одабир алгоритма при чему се анализирају модули компоненте са имплементационим детаљима. Након тога је дат начин коришћење компоненте за одабир алгоритма који ради у две фазе, фази тренирања и одлучивања. На крају су представљене различите топологије компоненте за различите локације извршавања.

4.1 Преглед компоненти у аутоматизованим системима за машинско учење

Пројектовање компоненте аутоматизованог система за машинско учење која ће се користити за одабир алгоритма у детекцији аномалија представља неопходан корак како би се креирали експерименти и дали одговори на полазне хипотезе. Аутоматизовани систем за машинско учење се састоји од компоненте за припрему података, компоненте за одабир атрибута, компоненте за креирање модела и компоненте за примену модела. Свака компонента представља област која захтева посебну пажњу, тако да је један део система везан за предложено решење изолован и извршена је његова имплементација. За креирање компоненте система за аутоматизовано машинско учење, неопходно је дефинисати модуле компоненте и зависност између тих модула. Таквим приступом је могуће развијати компоненте система независно, при чему се свака компонента система посебно имплементира и тестира. Једном када се компоненте система интегришу, могуће је тестирати цео систем. У склопу овог рада је имплементирана компонента система за одабир алгоритама, као што је приказано на слици 5. Оваквим приступом се омогућава интеграција имплементиране компоненте у постојеће аутоматизоване системе за машинско учење који би се прилагодили и за проблем детекције аномалија.



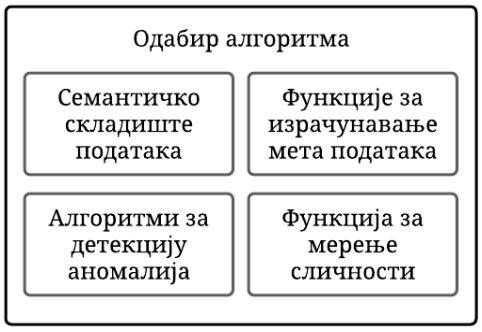
Слика 5: Аутоматизовани системи за машинско учење се састоји од компоненте за припрему података, компоненте за одабир атрибута, компоненте за креирање модела и компоненте за примену модела. На слици је означена компонента која ће бити имплементирана у овом раду.

Одабир модела за детекцију аномалија у аутоматизованим системима за машинско учење представља специјални случај преношења знања са претходно евалуираних и карактеризованих података које садрже аномалије у подацима. Оптимални алгоритам за одређени скуп података и оптимизациону метрику се одабира на основу већ евалуираних скупова података који имају сличне карактеристике података и аномалија. Алгоритми који су за сличне скупове података дали добре резултате се узимају као оптимални алгоритми за дати скуп података и оптимизациону метрику. Перформансе модела зависе од одабира алгоритма, оптимизације параметара алгоритма, и припреме података која укључује интеграцију и редукцију података. Оптимизација параметара алгоритама се врши углавном хеуристичким методама или применом алгоритама машинског учења. Овај рад се бави искључиво одабиром алгоритама.

Овакав приступ је инспирисан надгледаним учењем при чему се пре рада система захтевају скупови података који имају обележене аномалије. Пре него што систем буде у могућности да закључује, неопходно је тренирати систем тако што се евалуирају постојећи скупови података са обележеним аномалијама коришћењем различитих алгоритмима. Такође, ти скупови података су карактеризовани мета подацима који се израчунавају из података. Касније, добијени подаци се користе за одређивање најближих скупова података за дати скуп података и на тај начин се бира оптимални алгоритам за одговарајућу оптимизациону метрику. Изворни код имплементиране компоненте, репозиторјум скупова података и креирани експерименти са резултатима су јавно доступни [74].

4.2 Логичка структура компоненте за одабир алгоритма

У овом поглављу је представљена имплементација компонента за одабир алгоритма на основу мета података. Резултат компоненте представља одабрани алгоритам за који се касније извршава оптимизација параметара како би се постигле оптималне перформансе. Логичка структура компоненте за одабир алгоритма коришћеног у овом раду се састоји од модула за израчунавање мета података, модула за детекцију аномалија, модула за мерење сличности и модула који представља семантичко складиште података, као што је приказано на слици 6. У наставку је дат приказ појединачних модула који заједно формирају наведену компоненту.



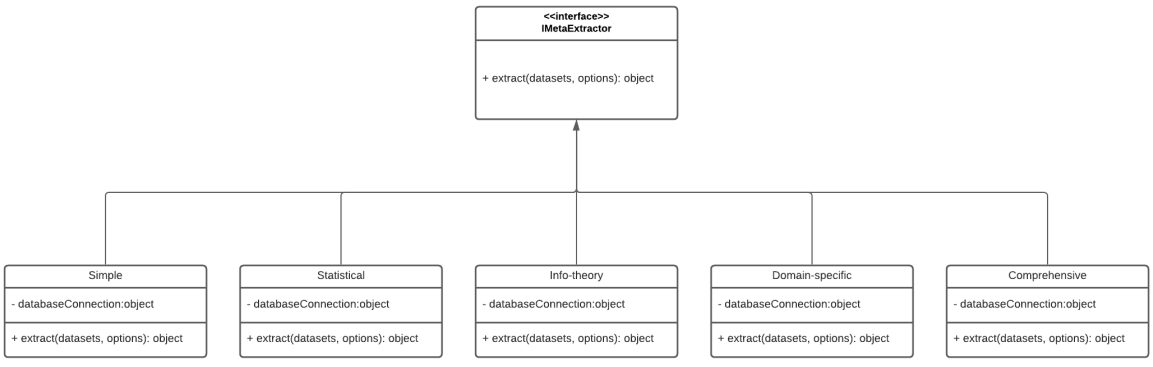
Слика 6: Компонента за одабир алгоритма која се састоји од модула за израчунавање мета података који се користе за карактеризацију података и аномалија у подацима, модула за детекцију аномалија у подацима и модула за мерење сличности између скупова података које се користе за приликом одабира алгоритма, као и модула који се користи за чување карактеристика аномалија у подацима и резултата евалуације алгоритама.

Сви модули су имплементирани коришћењем *Python* програмског језика и *TensorFlow* библиотеке. Овај програмски језик се користи у многим случајевима када је потребно имплементирати алгоритме машинског учења јер нуди погодне механизме за манипулацију над подацима. Такође, предност коришћења овог програмског језика је могућност репрезентације кода преко графа извршавања. Генерисани граф извршавања се креира приликом компилације кода и представља универзалну репрезентацију која може да се мапира на произвољну архитектуру. Неопходно је валидирати предложено решења на различитим архитектурама како би се дошло до закључака везано за одабир архитектуре у односу на локацију извршавања и случај коришћења компоненте. Како оваква имплементација не захтеве измене у коду приликом мапирања на алтернативне архитектуре, *TensorFlow* библиотека је погодна за експерименте који ће евалуирати извршавање на различитим локацијама коришћењем различитих архитектура. Постојање *ASIC* архитектуре за коришћени програмски језик и библиотеку чини га још погоднијим за имплементацију компоненте.

4.2.1 Модул за израчунавање мета података

Модул за израчунавање мета података представља главни модул компоненте за одабир модела. Резултат овог модула су израчунати мета подаци који представљају карактеристике података и аномалија. Улаз овог модула представља скуп података за који треба да се израчунају мета подаци. Наведени модул комуницира са семантичким складиштем података где се смештају израчунати мета подаци. Приликом тренирања система, израчунати мета подаци се шаљу у семантичко складиште података. Приликом одлучивања система, израчунати мета подаци се враћају главном процесу који коришћењем осталих модула одређује оптимални алгоритам за дати скуп података.

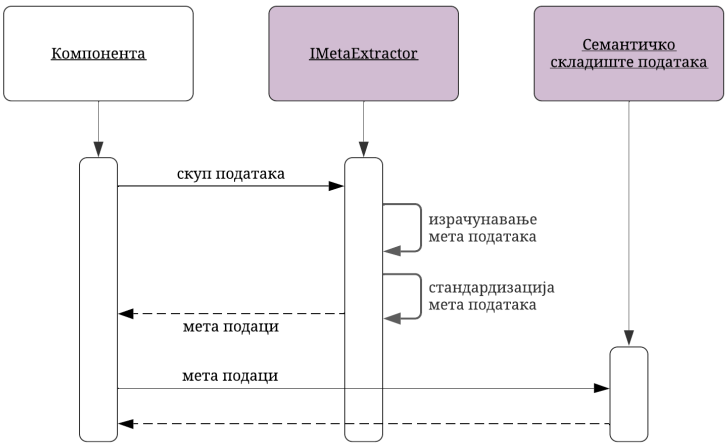
На слици 7 је приказан класни дијаграм модула за израчунавање мета података који имплементира облик понашања који се референцира као фасада у развоју софтвера. Често се карактеризација података врши помоћи више од једне групе функција за израчунавање мета података, па због тога је неопходно оставити могућност да компонента буде проширива и модуларна. Модуларност компоненте омогућава имплементацију нових или додавање постојећих функција за израчунавање мета података.



Слика 7: Класни дијаграм модула за израчунавање мета података на основу скупа података.

Приликом карактеризације података, мета подаци који нису представљени у истим јединицама могу да више или мање утичу резултат одабира алгоритма јер имају одређени утицај на мерење сличности између скупова података, па је због тога неопходно оставити

могућност за нормализацију и стандардизацију мета података пре враћања резултата компоненти. На слици 8 је приказан дијаграм секвенци који дефинише начин рада овог модула у фази тренирања. У фази закључивања, последњи корак се не извршава.

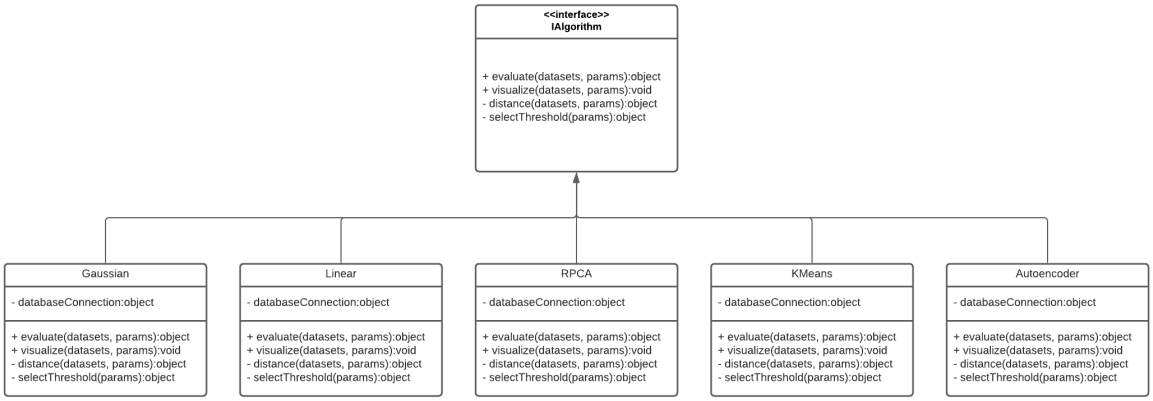


Слика 8: Дијаграм секвенци модула за израчунавање мета коришћењем различитих функција за израчунавање мета података који дефинише начин рада у фази тренирања. У фази закључивања, последњи корак се не извршава.

4.2.2 Модул за детекцију аномалија

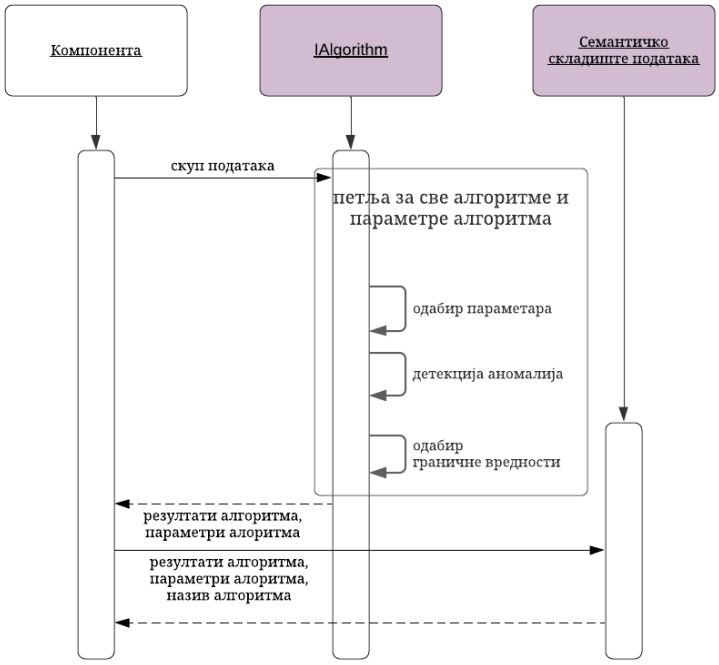
Извршавање алгоритама детекције аномалија представља се као модул за евалуацију скупова података коришћењем различитих алгоритама. Циљ је евалуација скупова података коришћењем више алгоритама који представљају основу за касније одлучивање компоненте. Улаз овог модула представља скуп података и одабрани алгоритам са параметрима који треба да изврши детекцију аномалија. Резултат модула представља резултат извршеног алгоритма у облику оптимизационих метрика које се користе у систему. Одабир алгоритама и оптимизационих метрика је дефинисан у следећем поглављу приликом поставке експеримената. У фази тренирања, овај модул комуницира са семантичким складиштем података при чему шаље резултате алгоритма за одређени скуп података. У фази закључивања, модул шаље резултате процесу који упоређује добијене резултате са очекиваним резултатима. Перформансе алгоритама за скуп података могу да зависе од типа алгоритма и параметара тог алгоритма. Ако се евалуира стохастички тип алгоритма неопходно је извршити евалуацију истог скупа податак више пута како би се добили валидни резултати. Такође, ако алгоритам има параметре који утичу на резултат, неопходно је оставити могућност за извршавање алгоритма са различитим вредностима и комбинацијама параметара како би се добили валидни резултати.

На слици 9 је приказан класни дијаграм модула за извршавање алгоритама за детекцију аномалија који имплементира облик понашања који се референцира као фасада у развоју софтвера. Овакав приступ је неопходан због потребе за извршавањем различитих алгоритама са различитим параметрима.



Слика 9: Класни дијаграм модула за детекцију аномалија коришћењем различитих алгоритама.

Алгоритми могу да се разликују по типу података који примају па је неопходно обезбедити могућност трансформације података пре примене алгоритама. У експериментима који евалуирају и пореде предложено решење неопходно је поредити више алгоритама. На слици 10 је приказан дијаграм секвенци који представља начин рада модула у фази тренирања. У фази закључивања, последњи корак се не извршава.



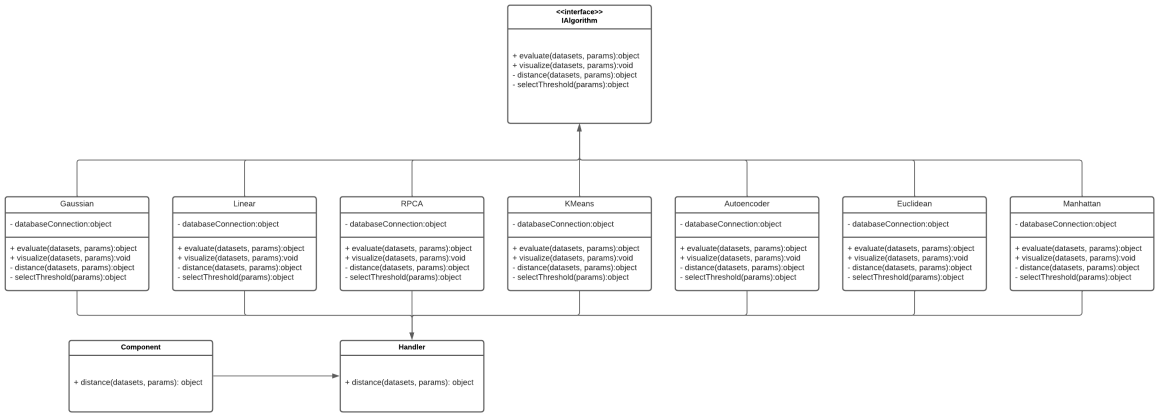
Слика 10: Дијаграм секвенци модула за детекцију аномалија у фази тренирања. У фази закључивања, последњи корак се не извршава.

4.2.3 Модул за мерење сличности

Модул за мерење сличности између скупова података се заснива на функцијама за мерење удаљености. Компонента на основу карактеристика података, што су мета подаци, сортира евалуиране скупове података по сличности или удаљености. Улаз за овај модул представља скуп података за који треба да се рангирају постојећи скупови података. Резултат

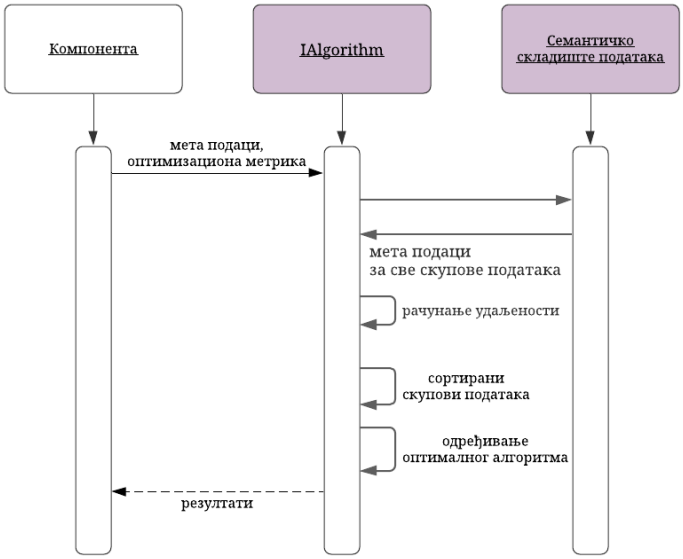
модула представља низ рангираних скупова података. На основу тога је могуће у фази одлучивања одабрати алгоритме који су дали добре резултате за одговарајућу оптимизациону метрику. Модул из семантичког складишта података дохвата карактеристике евалуираних скупова података. Затим се врши одабир функције за мерење удаљености. Одабир се врши на основу коришћених мета података. Модул ради искључиво у фази закључивања, док у фази тестирања није активан.

На слици 11 је приказан класни дијаграм модула за израчунавање случности који имплементира облик понашања који се референцира као фасада и ланац одговорности у развоју софтвера. Овакав приступ је неопходан због потребе за извршавањем различитих функција за мерење удаљености при чему се резултати агрегирају и прослеђују даље.



Слика 11: Класни дијаграм модула за мерење сличности између скупова података коришћењем различитих функција и израчунатих мета података.

Када функција за мерење удаљености креира низ сортираних скупова података, неопходно је одредити који је број скупова података који утиче на одабир алгоритма. У зависности од области примене аутоматизованог система броја алгоритама између којих се врши одабир може да се разликује, и одабир је дефинисан у поставци експеримената. На слици 12 је приказан дијаграм секвенци рада модула за мерење удаљености.

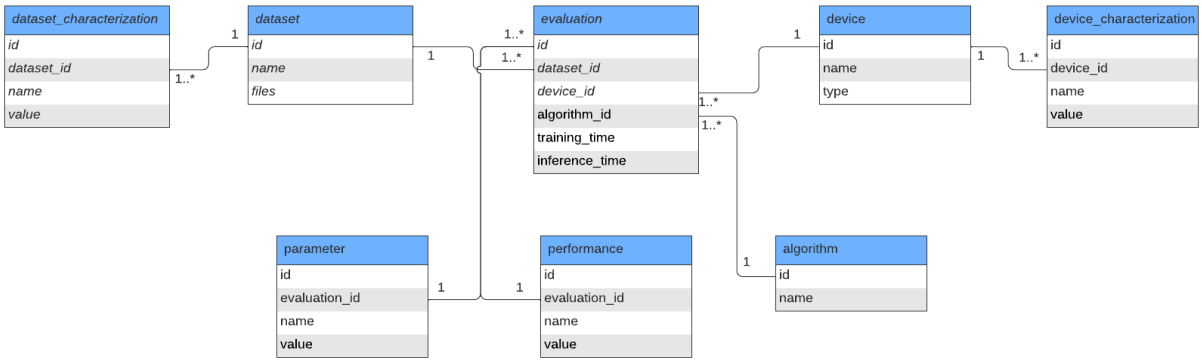


Слика 12: Дијаграм секвенци модула за мерење сличности између скупова података коришћењем функција за мерење удаљености.

4.2.4 Семантичко складиште података

Модул који представља семантичко складиште података се користи за чување карактеристика података у облику мета података и резултата евалуације алгоритама за детекцију аномалија. Овај модул једини има стање, што значи да остали модули могу да се скалирају и прошире у случају потребе за више ресурса. Како су ентитети компоненте добро структурирани, одлучено је да се приступи имплементацији релационе базе података. Улаз овог модула представљају подаци различитих ентитета које је потребно чувати.

На слици 13 је приказан релациони модел базе података. Модел је креиран са намером да буде лако проширив, што значи да додавање новог приступа за рачунање мета података или новог алгоритма за евалуацију не захтева промене у моделу базе података. Приликом моделовања базе података све норме за моделовање базе података су испоштоване. Табеле које се користе за карактеризацију података и перформансе о алгоритмима садрже колоне које представљају назив и вредност метрике. Модел базе података је креиран са намером да се представља семантичко складиште података које може даље да се анализира.



Слика 13: Релациони модел семантичког складишта података за чување информација о мета подацима и резултатима различитих алгоритама за детекцију аномалија.

Модул се у фази тренирања користи за складиштење података, док се у фази закључивања користи за читање података. Са растом базе података потребно је разматрати укључивање индекса како би брзина рада модула била задовољавајућа за различите случајеве коришћења и архитектуре на којима се систем извршава. Велика количина података у бази представља могућност виртуелизације података са циљем да се донесу закључци на вишем нивоу апстракције. За анализирање података могу да се користе алгоритми за проналажење скривеног знања (енг. *Data Mining*).

4.3 Коришћење компоненте за одабир алгоритма

Имплементирана компонента је намењена да се користи у две фазе - фаза тренирања и фаза закључивања. Фаза тренирања је неопходна како би компонента могла да креира податке у семантичком складишту података који би се касније користили приликом процеса одлучивања. Пре него што компонента може да врши закључивање неопходно је да се тренира са различитим скуповима података.

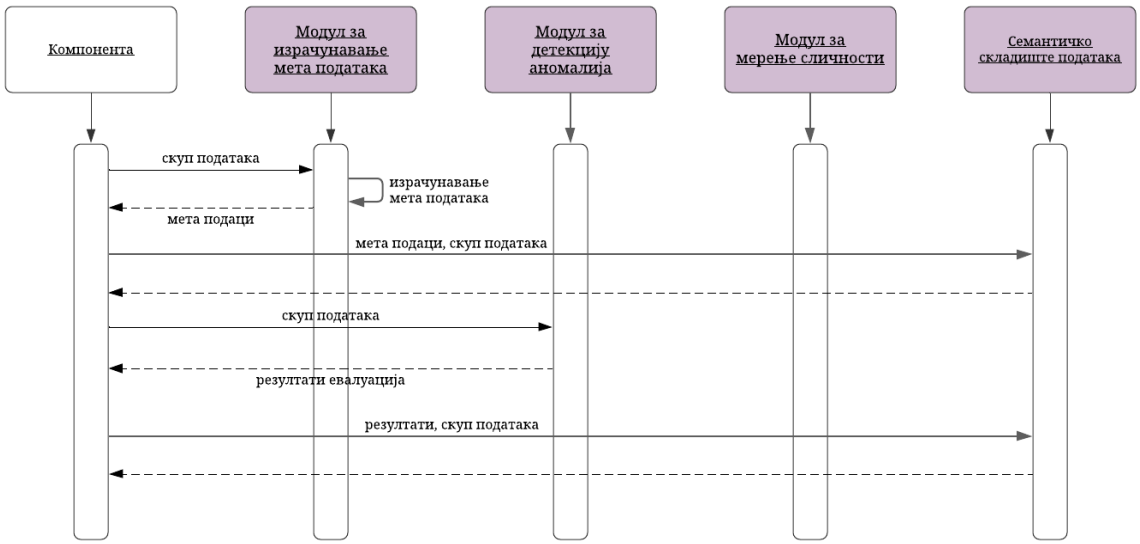
Модули у компоненти су међусобно зависни тако да излаз једног модула представља улаз другог модула. Компонента је пројектована са циљем да буде проширива тако да је потребно користити облик понашања за извршавање команди који омогућава креирање модула као команди при чему се објекат са тренутним стањем прослеђује између модула. Оваквом имплементацијом се омогућава проширивост компоненте и интеграција са другим компонентама аутоматизованог система за машинско учење. Компоненте у систему су повезане дефинисаним форматима улазних података потребних за извршавање. Компоненте могу независно да се имплементирају и оптимизују, што омогућава модуларну имплементацију система где се комбинују различите компоненте. Када се компонента имплементира или оптимизује, потребно је извршити интеграцију коришћењем дефинисаних улазних формата. Улаз компоненте треба да одговара излазном формату претходне компоненте, док излаз компоненте треба да одговара улазном формату следеће компоненте.

4.3.1 Тренирање компоненте

Тренирање компоненте представља прву фазу рада система. Пре тренирања компоненте неопходно је креирати репозиторијум података за тестирање са обележеним аномалијама у подацима. Затим, пре довођења скупова података на улаз система за евалуацију, неопходно је форматирати и обрадити податке тако да буду погодни за карактеризацију

података и аномалија. Обрада података се врши у компоненти пре имплементиране компоненте. Када је репозиторијум креиран и подаци спремни за карактеризацију почиње процес тренирања.

Тренирање система може да се подели у неколико целина. Први корак представља карактеризација података применом функција за израчунавање мета података. У сврху комбинованих решења као и евалуације предложеног решења, неопходно је омогућити извршавање више типова функција за израчунавање мета података које представљају различите групе. Израчунати мета подаци се шаљу у семантичко складиште података. Након карактеризације података применом функција за израчунавање мета података, прелази се на корак евалуације алгоритама. Алгоритми се евалуирају коришћењем различитих параметара и оптимизационих метрика помоћу креираног репозиторијума података. Након евалуације скупова података, подаци о алгоритму, параметрима алгоритма и резултати алгоритма се шаљу у семантичко складиште података. У семантичком складишту података се креира веза између добијених оптимизационих метрика алгоритама и мета података, при чему у овом случају мета подаци представљају карактеристике аномалија у подацима. На слици 14 је приказан дијаграм секвенци рада компоненте у фази тренирања.



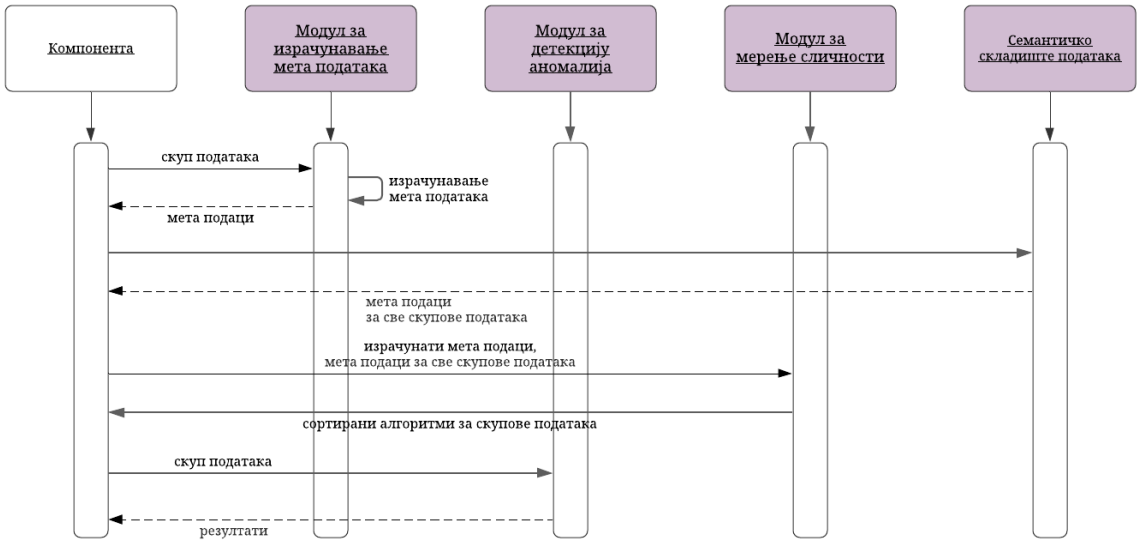
Слика 14: Дијаграм секвенци рада компоненте у фази тренирања. Процес се састоји од израчунавања мета података, евалуирања скупова података коришћењем алгоритама за детекцију аномалија, као и чувања резултата у семантичком складишту података.

4.3.2 Закључивање компоненте

Закључивање компоненте представља другу фазу рада и врши се након тренирања система. Пре пуштања компоненте у фазу закључивања неопходно је извршити тренирање као што је описано у претходном поглављу. Када је семантичко складиште података попуњено са тест подацима, компонента је у могућности да врши одабир алгоритама за дати скуп података и оптимизациону метрику.

Закључивање компоненте може да се подели у неколико целина. Први корак представља карактеризацију података применом функција за израчунавање мета података. У сврху комбинованих решења као и евалуације предложеног решења и креирања

експеримената, неопходно је омогућити извршавање више типова функција за израчунавање мета података које представљају различите групе. Израчунати мета подаци се не шаљу у семантичко складиште података, већ се прослеђују даље у модул за мерење удаљености где се траже слични скупови података. На основу карактеризације скупова података, модул за мерење сличности применом функција за рачунање удаљености сортира скупове података који су коришћени за тренирање. Након тога се врши одабир алгоритма на основу добијених резултата. Када се одабир алгоритма заврши, прелази се на модул за извршавање алгоритма, који није део имплементиране компоненте. Ради валидације предложеног решења, модул за извршавање алгоритама за детекцију аномалија је коришћен за валидацију имплементиране компоненте и предложеног решења. За дату оптимизациону метрику компонента предвиђа алгоритам узимајући алгоритам од сличних скупова података који је дао оптималне перформансе. На слици 15 је приказан дијаграм секвенци рада компоненте у фази закључивања.



Слика 15: Дијаграм секвенци рада компоненте у фази закључивања. Процес се састоји од израчунавања мета података, рачунања удаљености између скупова података коришћењем мета података и затим одлучивања. Након тога се скуп података евалуира одабраним алгоритмом и врши се валидација добијених резултата.

4.4 Топологија компоненте за различите локације извршавања

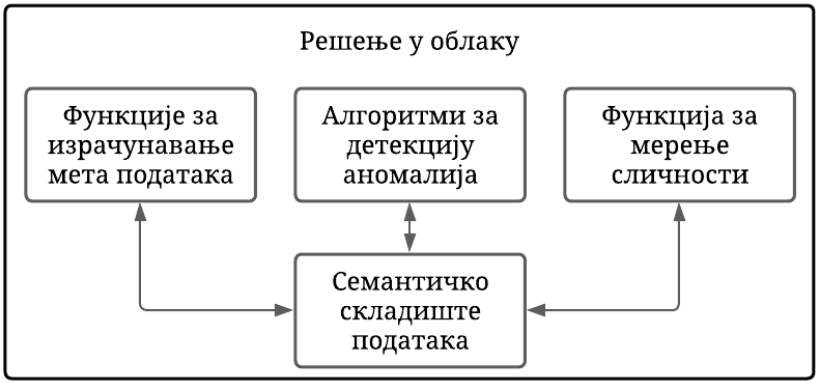
Системи за аутоматизовано машинско учење у зависности од примене могу да се извршавају на различитим локацијама. Од локације извршавања зависи одабир архитектуре на којој ће систем да се извршава. Ово поглавље даје преглед различитих топологија имплементираног система за различите локације извршавања. Локација извршавања система зависи од случаја коришћења. На основу прегледа отворене литературе је направљена подела у ради на решење у облаку, решење на крајњем уређају и хибридно решење. Локације извршавања дефинишу захтеве везане за доступне ресурсе на основу којих се дефинишу погодне архитектуре за коришћење на тој локацији извршавања. У наставку је дат предлог топологије имплементиране компоненте у односу на локацију извршавања. Топологије описане у следећим поглављима ће се евалуирати у експериментима ради дискусије и закључивања које архитектуре су погодне за које локације извршавања.

Главни задатак приликом одабира архитектуре за одређену локацију извршавања представљају ресурси доступни на тој локацији извршавања. Ресурси од интереса могу бити време закључивања система на одређеној локацији као и време тренирања система. Поред времена извршавања као главног ресурса који се узима у обзир приликом имплементације система, у одређеним случајевима, ресурси као што су потрошња електричне енергије или заузеће меморије могу да представљају битне факторе приликом одабира архитектуре. У ситуацијама када су ресурси ограничени, као што су крајњи уређају, потребно је одабрати архитектуру која прави компромис између времена извршавања и потрошње електричне енергије. Такве ситуације су типичне за системе који не раде у реалном времену и имају стриктно ограничене ресурсе. Меморија представља још један битан ресурс који може да ограничи рад система и углавном је у релацији са временом извршавања. Ако меморија представља проблем, могуће је применом одговарајућих оптимизационих механизама направити компромис између времена извршавања и меморије. Пример је чување података на оптимизован начин који је мање погодан за обраду тако да треба више времена да се подаци доведу у стање погодно за коришћење. Различите топологије система постављају карактеристике које могу да се покажу као погодне за одређене локације извршавања. На одабир топологије погодне за коришћење на различитим локацијама извршавања могу да утичу и сами подаци као и тип извора података. Ове карактеристике ће се дискутовати за представљене локације извршавања.

4.4.1 Решење у облаку

Већина аутоматизованих система за машинско учење који се користе у индустрији и академији је заснована на решењу у облаку. Решења у облаку омогућавају скалабилност и као таква представљају приступ погодан за процесирање велике количине података. Скалабилност ових решења представља главну предност у односу на остала решења где је могуће скалирати архитектуру на којој се систем извршава тако да ресурси не представљају проблем. Ако случај коришћења нема стриктне захтеве што се тиче времена извршавања и извор се налази на локацији извршавања, ова локација извршавања је погодна за извршавање имплементиране компоненте. Међутим, ако случај коришћења има стриктне захтеве, као што је време извршавања, неопходно је анализом компоненте закључити да ли је могуће извршавање на наведеној локацији. Ако се закључи да није могуће, потребно је евалуирати остале локације извршавања за рад компоненте. Такође, ако случај коришћења има стриктне захтеве везане за кашњење резултата или поузданости компоненте у смислу повезаности и конзистентног рада, неопходно је евалуирати остале локације извршавања за извршавање компоненте.

Решење у облаку представља топологију где се сви модули налазе у облаку и при чему се цела компонента извршава у облаку. Архитектуре коришћене у овом окружењу укључују оне које имају добре перформансе у смислу времена извршавања, док остали аспекти нису од значаја с обзиром да ресурси нису ограничени. Један од битних фактора је финансијска исплативост имплементације система која неће бити даље дискутована у овом раду. На слици 16 је представљена архитектура компоненте у облаку где се сви модулу компоненте налазе на истој локацији извршавања.



Слика 16: Архитектура компоненте у облаку где се сви модули компоненте налазе на истој локацији извршавања.

Тренирање са великом количином података може да се скалира на архитектуре као што су *GPU* и *ASIC* који имају добре перформансе за велике количине података. Како потрошња електричне енергије није битна у овом случају, скалирање и паралелизација је могућа. Сви модули компоненте могу ефикасно да производе резултате и тако омогућавају добре перформансе система у погледу времена извршавања.

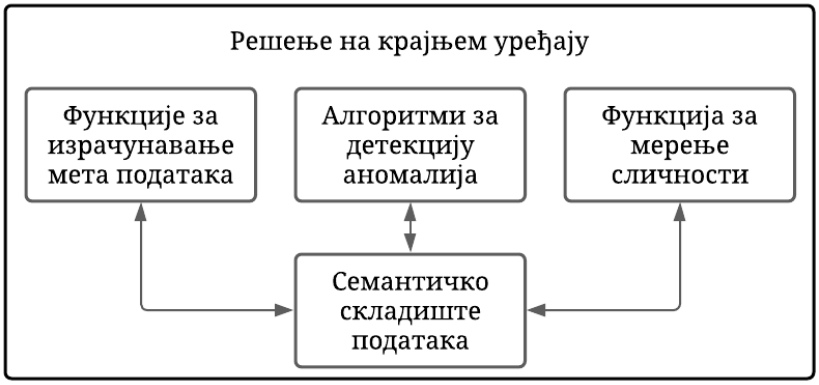
Проблем код ове имплементације представља комуникација са извором података и типом података који стиже до система. Ако су подаци временски и захтевају константан процес тренирања и обраде, комуникација и кашњење резултата може да представља проблем који доводи до разматрања осталих локација извршавања за рад компоненте.

4.4.2 Решење на крајњем уређају

Крајњи уређај представља место где се налази извор података или локација коришћења система. Пример оваквог окружења може да представља сензор неког система или мобилни телефон корисника. У оба случаја, ресурси су ограничени што даље захтева одабир архитектуре извршавања која задовољава постављене захтеве. Ако се компонента налази у фази имплементације могуће је утицати на одабир архитектуре где се врши оптимизација ресурса кроз одабир погодне архитектуре за ту примену. У супротном, неопходно је користити постојећу архитектуру, и самим тим се евалуира изводљивост имплементације компоненте. Ако компонента не користи временске податке који константно долазе са неког извора, могуће је имплементирати компоненту на крајњем уређају ако то ресурси омогућавају. Међутим, ако компоненте ради у реалном времену и окружење поставља стриктне временске захтеве, неопходно је валидирати изводљивост имплементације компоненте с обзиром на ограничене ресурсе. Такође, ако се систем налази у окружењу које захтева комуникацију са извором података, проблеми као што су комуникација са извором података и доступни ресурси који морају да буду довољни за извршавање компоненте утичу на одабир архитектуре.

Решење на крајњем уређају представља топологију где се сви модули налазе на уређају где се систем користи или на уређају где се налази извор података. Архитектуре коришћене у овом окружењу укључују оне које имају карактеристике погодне за дату локацију извршавања. Ресурси од интереса могу да буду време извршавања, доступна меморија и потрошња електричне енергије јер се ради о уређајима који немају константан извор електричне енергије.

На слици 17 је представљена архитектура компоненте на крајњем уређају где се сви модулу компоненте налазе на истој локацији извршавања.



Слика 17: Архитектура компоненте на крајњем уређају где се сви модули компоненте налазе на истој локацији извршавања.

Тренирање система представља захтевну операцију коју некад није могуће извршити на крајњем уређају, у ком случају се приступа алтернативним решењима. Ако ресурси уређаја могу да задовоље критичне захтеве локације извршавања, у ситуацијама када се може вршити одабир архитектуре, *FPGA* и *ASIC* представљају погодне архитектуре. Такође, *CPU* и *GPU* са мањим капацитетом који захтевају мало ресурса у односу на решења у облаку могу да се користе. Како је потрошња електричне енергије битна у овом окружењу, скалирање и паралелна обрада је ограничена.

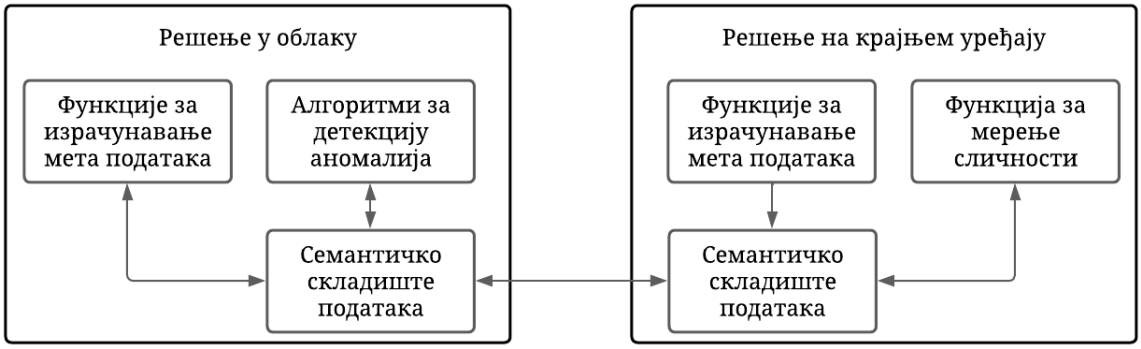
Проблем код ове имплементације представља комуникација са извором података и типом података који стиже до компоненте. Такође, ограничени ресурси могу да утичу на рад и изводљивост имплементације система. Ако су подаци временски и захтевају константан процес тренирања и обраде или ако комуникација и кашњење имају стриктне захтеве, неопходно је разматрање алтернативних топологија.

4.4.3 Хибридно решење

Претходно наведене локације извршавања имају предности у ситуацијама када ресурси доступни на одређеној локацији дозвољавају такву топологију и имплементацију система. Међутим, није увек могуће обезбедити да се систем имплементира у облаку или на уређају где се користи. Због тога се хибридно решење у овом раду представља као решење где се одређени модули компоненте извршавају на различитим локацијама. Ако се систем налази у фази пројектовања, могуће је утицати на одабир архитектуре где може да се врши оптимизација ресурса кроз одабир погодне архитектуре за ту примену. У супротном, неопходно је користити постојећу архитектуру, и на основу тога одредити који делови компоненте се извршавају на којим локацијама. Овакво решење подразумева да се део компоненте извршава у облаку док се остатак извршава на уређају корисника или на извору података. Архитектура која се користи на таквим локацијама извршавања је углавном одређена да се у облаку користе *CPU* и *GPU*, док се на крајњим уређајима користе *FPGA* и *ASIC*. Како решење у облаку омогућава велике количине ресурса које могу по потреби да се скалирају, погодно је тренирање компоненте имплементирати у облаку. Исти приступ се примењује на крајњим уређајима, где је битно да нема кашњења и да имплементација буде стабилна у смислу комуникације. Због

тога се намеће захтев да се на крајњим уређајима користи део компоненте за закључивање где се семантичко складиште података шаље из облака на крајњи уређај.

На слици 18 је приказана архитектура компоненте за одабир алгоритма где се део компоненте налази у облаку док се остатак налази на крајњем уређају. Тренирање са великом количином података може да се скалира на *GPU* и *ASIC* који имају велику пропусну моћ, док се закључивање врши на крајњим уређајима који задовољавају захтеве локације извршавања. Како потрошња електричне енергије није ограничена у фази тренирања, скалирање и паралелна обрада може да се постигне на дефинисаној локацији извршавања.



Слика 18: Архитектура компоненте за одабир алгоритма где се део компоненте налази у облаку док се остатак налази на крајњем уређају.

Ако се систем тренира само први пут пре закључивања, оваква топологија је погодна за имплементацију компоненте. Међутим, ако систем ради са временским подацима који константно долазе са неког извора, треба обратити пажњу на комуникацију између решења у облаку и решења на крајњем уређају. Ако систем ради у реалном времену и окружење поставља стриктне временске захтеве, неопходно је валидирати изводљивост имплементације компоненте с обзиром на ограничења у комуникацији. У поређењу са претходним решењима, ово решење се издваја по томе што омогућава рад система чак и када се деси прекид у комуникацији, при чему компонента користи податке који су доступни на крајњем уређају како би могла да врши закључивање.

Главни недостатак овог решења је потреба за комуникацијом између окружења која отвара питања приватности података, сигурности и стабилности компоненте. Стабилност се односи на константну комуникацију између модула за тренирање и закључивање. Проблем сигурности захтева интеграцију заштићене комуникације од неауторизованог приступа. Приватност представља још један фактор који утиче на изводљивост имплементације компоненте где је потребно заштити приватност података редукцијом или трансформацијом сензитивних података пре слања у друго окружење.

5 Евалуација предложеног решења и пројектоване компоненте

У овом поглављу је извршена евалуација предложеног решења и пројектоване компоненте поставком и описом експеримената који врше евалуацију и дају одговоре на дефинисане полазне хипотезе. Након тога су представљени резултати експеримената и дати су одговори на полазне хипотезе. У првом делу поглавља је представљена поставка експеримената где је дат преглед коришћених скупова података, алгоритама за детекцију аномалија, евалуационих метрика, функција за мерење удаљености и архитектура коришћених за имплементацију компоненте. Након тога је дат опис експеримената који представља начин извршавања експеримената, при чему сваки експеримент даје одговор за једну или више постављених полазних хипотеза. Експерименти се састоје од евалуације предложеног решења у односу на постојећа решења и евалуације имплементиране компоненте на различитим архитектурама. Након тога су представљени резултати експеримената, на основу чега су дати одговори на дефинисане полазне хипотезе. Експерименти су креирани тако да тестирају перформансе различитих решења за мета податке, као и да валидирају да је могуће коришћење предложених мета података у аутоматизованим системима за машинско учење. Методологија поставке и описа експеримената је рађена по узору на методологију представљену у раду [75], и представља начин поставке експеримената која се анализира одвојено од описа експеримената. Оваква методологија омогућава проширивање поставке експеримената без потребе за мењањем описа експеримената.

5.1 Поставка експеримената

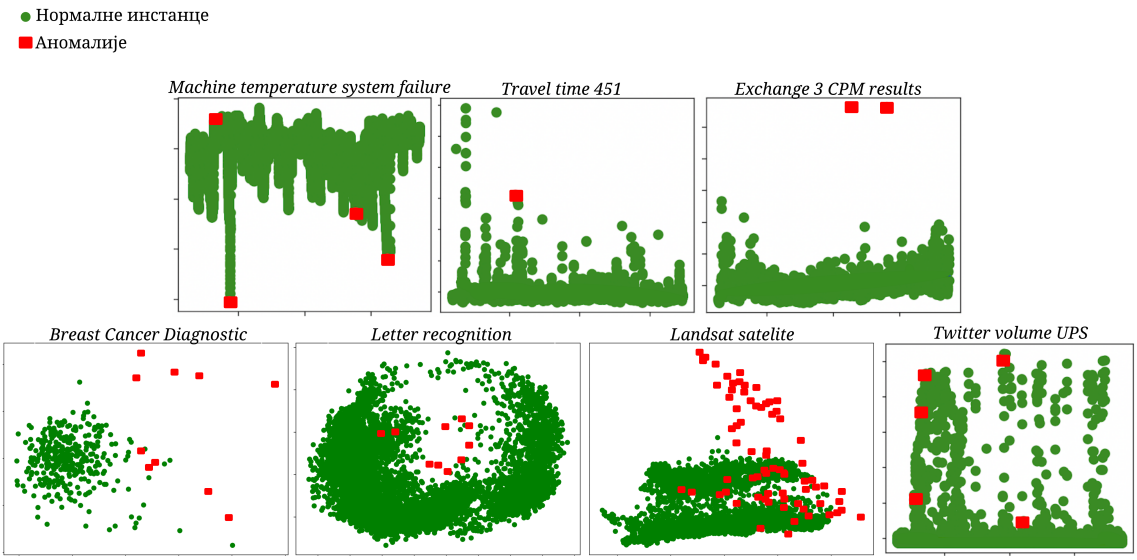
Ово поглавље представља поставку експеримената где је описан креирани репозиторијум података који се користи у експериментима. Репозиторијум садржи различите скупове података и за циљ има да својом разноврсношћу утиче на резултате експеримената тако да они буду валидни за различите типове података и домене у којима предложено решење може да се користи. Такође, дискутована је појава различитих врста парадокса и како они могу да утичу на експерименте. Након тога је дат преглед алгоритама за детекцију аномалија и евалуационих метрика коришћених у експериментима. Евалуационе метрике су од значаја како би се представило по којим критеријумима ће се предложено решење поредити са постојећим решењима. Затим, дат је преглед коришћених мета података и функција за мерење удаљености. На крају, дат је преглед коришћених архитектура на којима се имплементира пројектована компонента. Раздвајањем поставке експеримената и описа експеримената омогућава се проширивање поставке експеримената са додатим материјалима у итерацијама истраживања. Ниво детаља у поставци експеримената је изабран тако да буде једнак или више грануларан у поређењу са добро познатим решењима, као што је [47].

5.1.1 Прикупљање података за експерименте

Проблем детекције аномалија може да се посматра као задатак бинарне класификације где једна класа представља аномалије док друга класа представља нормалне инстанце у подацима. Постоји велики број јавно доступних скупова података са бинарним класама, као што је *UCI* репозиторијум за машинско учење [76]. Ови скупови података могу да се трансформишу за проблем детекције аномалија насумичним узорковањем малог броја

инстанци из једне класе и креирањем новог скупа података. Међутим, такав приступ не гарантује креирање скупа података који садржи карактеристике аномалија и постојање шаблона понашања аномалија из реалног света. Чак и да је могуће креирати такав скуп података, није гарантовано да се аномалије не уклапају у нормалне шаблоне понашања. Због свега наведеног, овај приступ се не сматра валидним за прикупљање података са обележеним аномалија погодним за експерименте у овом раду.

У овом раду је креиран репозиторијум података који се састоји од скупова података који припадају различитим изворима. Креирани репозиторијум садржи 63 скупа података са обележеним аномалијама, при чему ти подаци имају различите типове и припадају различитим доменима. Вишедимензионални подаци припадају домену медицине, обраде текста, и обраде записа (енг. *Logs*) у софтверским системима. На пример, један од скупова података представља вишедимензионалне карактеристике слика дојке, где класа означава да ли је присутан канцер, што ако јесте случај, та инстанца је означена као аномалија. Следећи пример вишедимензионалних података је детекција аномалија у записима софтверског система где је циљ детекција веза у мрежној комуникацији које нису ауторизоване. Временски подаци припадају домену графова, транспорта, софтвера и економије. Пример скупа података је анализа веза друштвене мреже у облику графа где су дате објаве са временском димензијом и циљ је детекција инстанци које се разликују од осталих података. Следећи пример представљају софтверски логови машина у облаку где је циљ детекција аномалија у раду процесора и оперативне меморије кроз време. Скупови података су прикупљени из следећих извора [77] [78]. На слици 19 су представљене дистрибуције скупова података, где је за сваки домен одабран по један скуп података. Дистрибуције су представљене у 2-димензионалном простору где је извршена трансформација података из оригиналног простора како би се дистрибуција визуелно приказала.



Слика 19: Дистрибуција скупова података у 2-димензионалном простору за различите домене. Инстанце означене црвеном бојом представљају аномалије у подацима. Циљ је да се покаже разноврсност скупова података који су одабрани за тестирање у експериментима.

Скупови података прикупљени на наведени начин задовољавају захтеве који се односе на главне карактеристике аномалија као што је одступање од остатка дистрибуције и количину аномалија. Експерименти су засновани на индустријским скуповима података где скупови

података различитих типова и покривају 7 различитих домена који по отвореној литератури представљају већину апликација релевантних за домен детекције аномалија. У табели 10 је дат приказ различитих карактеристика аномалија у креираном репозиторијуму за типове података и домене којима подаци припадају. На основу локалитета аномалија, скупови података обухватају локалне и глобалне аномалије, са малим процентом микро-кластера. На основу димензионалног простора аномалија, креирани репозиторијум садржи значајан број скупова података са једнодимензионалним аномалијама које су повезане са темпоралном димензијом и одређеним бројем вишедимензионалних скупова података.

Табела 10: Креирани репозиторијум података се састоји од 63 скупа података са означеним аномалијама, при чему подаци имају различите типове и припадају различитим доменима. Један скуп података може садржати више од једног типа података и локалитета аномалија. Колона укупно представља укупан број скупова података за одређену класификацију, док колона просек представља просечан број аномалија у скуповима података за одређену класификацију. Креирани репозиторијум скупова података представља свеобухватну колекцију за евалуацију перформанси различитих алгоритама за детекцију аномалија.

		Локалитет аномалија					
		Локалне аномалије		Глобалне аномалије		Микро-кластери	
		укупно	просек	укупно	просек	укупно	просек
Тип података	Вишедимензионални подаци	5	$12\ x\ 10^{-4}$	5	$47\ x\ 10^{-4}$	2	$25\ x\ 10^{-4}$
	Временски подаци	41	$64\ x\ 10^{-6}$	14	$61\ x\ 10^{-6}$	1	$18\ x\ 10^{-4}$
	Номинални подаци	0	0	1	$36\ x\ 10^{-4}$	1	$36\ x\ 10^{-4}$
	Просторни подаци	1	$30\ x\ 10^{-4}$	0	0	0	0
Домен података	Производња	4	$15\ x\ 10^{-4}$	0	0	0	0
	Транспорт	5	$67\ x\ 10^{-6}$	3	$15\ x\ 10^{-4}$	0	0
	Економија	6	$14\ x\ 10^{-4}$	0	0	0	0
	Медицина	0	0	3	$33\ x\ 10^{-4}$	2	$36\ x\ 10^{-4}$
	Обрада текста	2	$88\ x\ 10^{-5}$	2	$86\ x\ 10^{-4}$	0	0
	Софтвер	30	$97\ x\ 10^{-6}$	2	$17\ x\ 10^{-5}$	2	$16\ x\ 10^{-4}$
	Графови	0	0	10	$22\ x\ 10^{-6}$	0	0

Описани начин прикупљања података је неопходан из разлога да се постигне добра основа за извршавање експеримената где скупови података не утичу на предложено решење тако се привидно добију добри резултати само за одређени тип података, одређени локалитет аномалија или одређени домен података. Оваквим одабиром скупова података се смањује могућност појаве различитих врста парадокса. Симпсонов парадокс представља феномен који се јавља у статистици као појава која је везана за одређену групу података која није присутна приликом узимања ширег узорка података. Пример парадокса генерално може да се представи кроз позитиван тренд који се јавља за две одвојене групе, док се негативан тренд јавља када се групе комбинују. Пример Симпсоновог парадокса у овом случају би могао да се дефинише као одређени тип података за које предложено решење даје добре резултате у поређењу са постојећим решењима. Међутим, ако се евалуира шири скуп података, што укључује остале типове података, показује се да предложено решење не даје боље резултате. Због тога су узети скупови података који елиминишу појаву Симпсоновог парадокса тако што подаци долазе са више извора и имају различите карактеристике. Одабрани скупови података су коришћени у индустрији и академији за различите врсте тестирања. Такође, одабрани скупови података припадају различитим доменима и садрже различите типове података.

Још један парадокс који је битан представља Берксонов парадокс који се испољава у облику резултата експеримената који нису логични. Овај парадокс се обично јавља када су

експерименти наклоњени једној страни при чему они не узимају све чињенице у обзир, и тако дају привидно добре резултате. Како би се елиминисао овај парадокс, предложено решење је поређено са постојећим решењима коришћењем истих скупова података. Евалуирање различитих решења коришћењем истих скупова података смањује вероватноћу да се наведени парадокс појави. Поред анализираних парадокса, постоје и други парадокси који нису примарно везани за статистичку анализу па због тога нису релевантни у овом раду.

Како би скупови података могли да се користе за евалуацију алгоритама за детекцију аномалија, неопходно је извршити стандардизацију података. Стандардизација података је урађена уклањањем средње вредности и скалирањем на јединичну варијансу. На тај начин се изједначава утицај различитих карактеристика података приликом креирања шаблона понашања.

5.1.2 Алгоритми за детекцију аномалија

Алгоритми за детекцију аномалија се користе приликом тренирања и закључивања компоненте. У фази тренирања, перформансе алгоритама се евалуирају коришћењем скупова података из креираног репозиторијума. У фази закључивања, добијене перформансе алгоритама се користе приликом одабира оптималног алгоритма за нови скуп података. Приликом одабира алгоритама за детекцију аномалија, степен трансформација над подацима и тип алгоритама су главне карактеристике које су вршиле одлучивање, као што је дискутовано у поглављу 2. На основу прегледа отворене литературе и заступљености коришћених алгоритама у индустрији, алгоритми су одабрани као репрезентативни примери из својих група. Алгоритми коришћени у експериментима за детекцију аномалија су Гаусова дистрибуција, линеарна регресија, робусна анализа главних компонената, кластеризација методом К-средњих вредности и аутоенкодер.

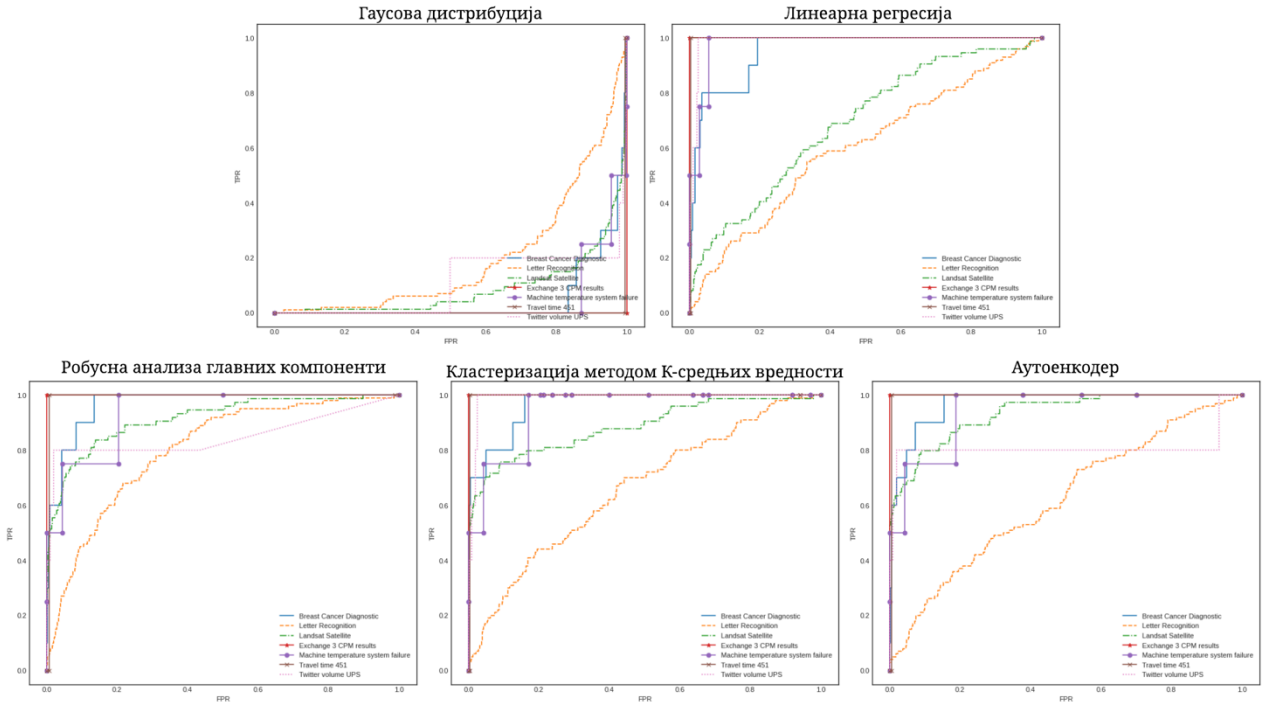
Одабир параметара за алгоритама је одређен на основу добрих пракси и познатих приступа из отворене литературе, као и на основу хеуристичког знања. Одређени параметри су оптимизовани насумичном претрагом простора. Гаусова дистрибуција је детерминистички алгоритам који нема параметре. Линеарна регресија је детерминистички алгоритам који има фреквенцију учења (α) и функцију грешке као параметере. Приликом евалуације резултата користи се средња квадратна грешка. На основу хеуристичког знања, фреквенција учења (α) је постављена на *0,3* и број епоха за тренирање је *5000*. Робусна анализа главних компонената је стохастички алгоритам који захтева број епоха, који је постављен на *1000*. Кластеризација методом К-средњих вредности припада групи стохастичких алгоритама, и као параметар има број кластера. На основу хеуристичког знања опсег вредности за параметар k је одабран да буде од *2* до *50*, док је број епоха *10*. Приликом евалуације резултата алгоритма, претрагом тог простора се долази до оптималног параметра за дати скуп података и оптимизациону метрику. Аутоенкодер је детерминистички алгоритам који садржи улазни слој који се одређује бројем атрибута у скупу података, *3* скривена слоја који имају *128*, *32*, и *128* неурона респективно, и излазни слој који је одређен бројем атрибута у скупу података. Приликом евалуације резултата користи се средња квадратна грешка. На основу хеуристичког знања, оптимизација мреже се врши *Adam* методом, док је величина серије *32* и број епоха за тренирање *100*.

Поред наведених алгоритама, разматрала се и примена алгоритама као што су алгоритми засновани на вероватноћи, као и алгоритми засновани на густини простора. Како циљ овог рада није предлог и евалуација предложених алгоритама, већ се они користе за евалуацију предложеног решења и имплементиране компоненте, наведени алгоритми и параметри су одабрани на основу прегледа отворене литературе и хеуристичког знања, док

додатни алгоритми нису даље разматрани. Оваквим одабиром алгоритама се добијају резултати експеримената где се са додавањем нових алгоритама или мењањем параметара не очекују велике промене у перформансама предложеног решења, јер одабир алгоритама обухвата различите степене трансформација над подацима.

Сви наведени алгоритми као резултат не дају бинарну вредност која означава да ли је инстанца аномалија, већ дају вредност која представља степен одступања од остатка дистрибуције. Приликом евалуације резултата алгоритма, врши се претрага простора за граничну вредност између нормалних инстанци и аномалија, и на тај начин се долази до оптималне вредности где је број итерација постављен на 5*000*. Како би се представиле перформансе алгоритама за различите граничне вредности, могуће је урадити анализу радне карактеристике пријемника (енг. *ROC*).

Ова анализа представља перформансе бинарног класификатора за различите граничне вредности између класа, што је случај и у детекцији аномалија. Анализа радне карактеристике пријемника се врши креирањем криве која представља перформансе алгоритма који врши бинарну класификацију. Крива се креира исцртавањем праве позитивне карактеристике (енг. *TPR*) у односу на лажно позитивну карактеристику (енг. *FPR*). Права позитивна карактеристика представља однос инстанци које су тачно класификоване као позитивне од свих позитивних инстанци. Слично томе, лажно позитивна карактеристика је однос инстанци које су погрешно класификоване да ће бити позитивне од свих негативних инстанци. На пример, у детекцији аномалија, права позитивна карактеристика представља инстанце које јесу аномалије и које су идентификоване као аномалије. Бинарни класификатори који дају вероватноћу која представља степен припадности некој класи, може са различитим граничним вредностима да осликава перформансе алгоритма. Анализа радне карактеристике пријемника представља однос између сензитивности и специфичности бинарног класификатора. Како би се валидирало да су одабрани алгоритми и скупови података погодни за коришћење у експериментима, урађена је анализа радне карактеристике пријемника. Циљ овакве анализе је да покаже да су алгоритми и скупови података одабрани тако да се добије униформна расподела перформанси, при чему постоје комбинације алгоритама и скупова података који дају лоше перформансе и са друге стране постоје комбинације које дају добре перформансе. На слици 20 је приказана анализа радне карактеристике пријемника за одабране алгоритме, где се за сваки алгоритам евалуира по један скуп података из различитих домена.

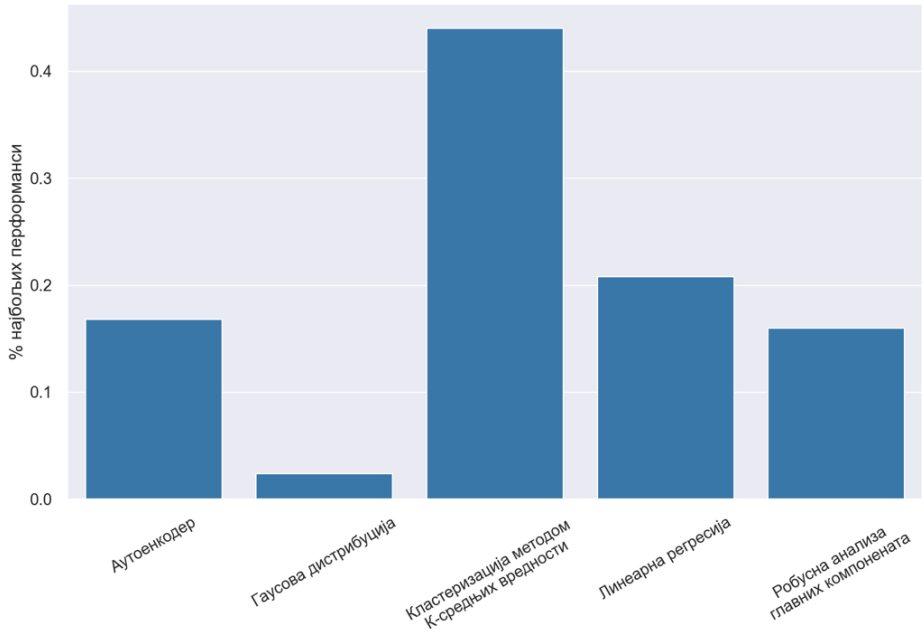


Слика 20: Анализа радне карактеристике пријемника за различите алгоритме и скупове података из различитих домена. Показано је да за различите домене података један алгоритам може да постигне другачије резултате за одређену оптимизациону метрику.

Главна особина радне карактеристике пријемника је што однос класа у скупу података не утиче на анализу, што је случај у детекцији аномалија. Због тога се ова врста анализе користи као метода за евалуирање алгоритама за детекцију аномалија. Пример аномалија које се ређе појављују у подацима и које се теже детектују су микро кластери. Ова група аномалија представља нову категорију у подацима и обично се детектује као колективна аномалија, што није тривијалан задатак.

Анализом радне карактеристике пријемника се даје преглед перформанси алгоритама, али се не користи за њихово поређење. Простор испод криве радне карактеристике пријемника може да се користи за поређење перформансе различитих алгоритама. Израчунавањем површине испод криве радне карактеристике пријемника могу да се пореде перформансе алгоритма. Вредност површине је пропорционална вероватноћи да насумично изабрана аномалија има веће одступање од остатка дистрибуције у односу на насумично изабрану нормалну инстанцу.

У зависности од алгоритама и скупова података, неки алгоритми могу да дају боље резултате од других. То може да зависи од типа алгоритма и параметара тог алгоритма. Ако алгоритам има параметре који могу да имају опсег вредности, коришћењем тог алгоритма већа је вероватноћа да ће одређена комбинација параметара дати боље резултате од алгоритама који немају параметре. На слици 21 је приказан проценат у коме је алгоритам за детекцију аномалија дао најбоље перформансе над скуповима података из креираног репозиторијума за *f1* оптимизациону метрику. На основу слике показано је да одређени алгоритми засновани на густини и алгоритми који имају параметре дају боље резултате за детекцију аномалија. Алгоритми засновани на теорији вероватноће дају добре резултате само за неколико скупова података из репозиторијума. Свакако то не може да се генерализује за све алгоритме и све скупове података. Одабир *f1* оптимизационе метрике је дискутован у следећем поглављу.



Слика 21: Поређење алгоритама за детекцију аномалија коришћењем креираног репозиторијума рачунањем процента у коме је одређени алгоритам дао најбоље перформансе за *f1* оптимизациону метрику. Показано је да алгоритми који имају параметре дају боље резултате, док алгоритми без параметара дају добре резултате само за неколико скупова података из репозиторијума.

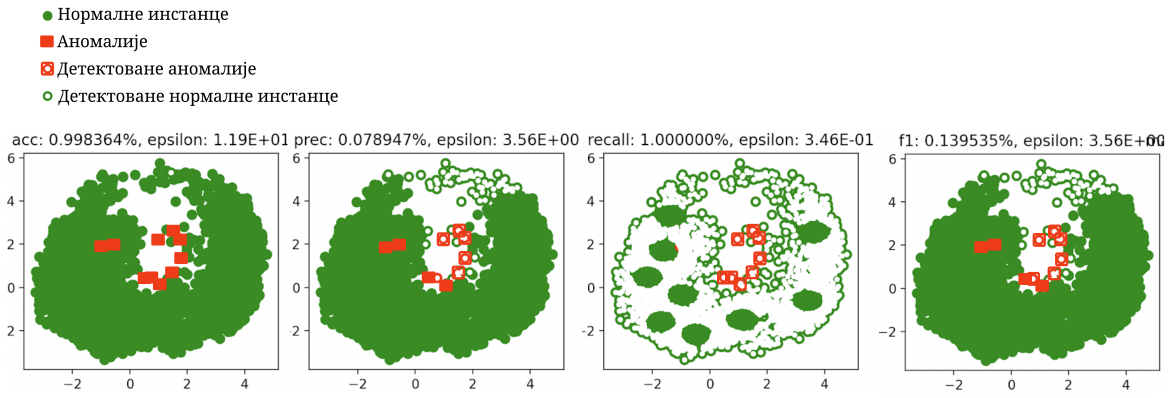
5.1.3 Евалуационе метрике

Евалуационе метрике представљају начин мерења перформанси алгоритама и користе се у експериментима за поређење различитих решења. Оптимизационе метрике анализиране у овом раду представљају заступљене метрике које се користе у алгоритмима машинског учења, и оне су: *accuracy*, *precision*, *recall* и *f1*. У детекцији аномалија, где је однос између нормалних инстанци и аномалија велики, ако се посматра оптимизациона метрика као што је *recall*, гранична вредност између нормалних инстанци и аномалија се одређује тако да све аномалије буду детектоване, а при томе и већина нормалних инстанци буду означене као аномалије. Резултати *recall* оптимизационе метрике могу привидно да буду задовољавајући, што доводи до случаја да креирани модел губи смисао јер се свака инстанца детектује као аномалија. У табели 11 су приказане средње вредности оптимизационих метрика алгоритама над скуповима података који су подељени по типу података, локалитету аномалија, и домену података. Све метрике осим *f1* имају привидно добре перформансе и не представљају валидне метрике за поређење. Због тога је као главна оптимизациона метрика у експериментима коришћена *f1* која даје добар однос између *precision* и *recall* оптимизационих метрика.

Табела 11: Средње вредности оптимизационих метрика алгоритама над скуповима података који су подељени по типу података, локалитету аномалија, и домену података. Показано је да су само одређене метрике погодне за представљање перформанси алгоритама за детекцију аномалија.

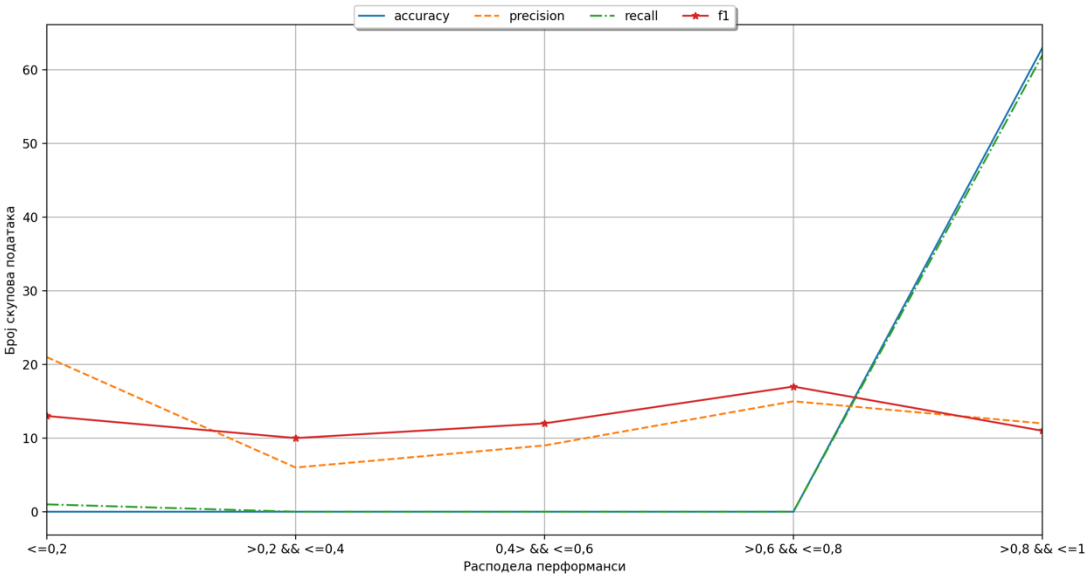
		Оптимизационе метрике			
		accuracy	*precision*	*recall*	*f1*
	Сви подаци	0,94	0,51	0,99	0,25
Тип података	Вишедимензионални подаци	0,92	0,51	1,00	0,22
	Временски подаци	1,00	0,50	0,98	0,33
Локалитет аномалија	Локалне аномалије	0,92	0,38	0,99	0,17
	Глобалне аномалије	0,95	0,65	1,00	0,36
	Микро-кластери	0,96	0,61	1,00	0,18
Домен података	Производња	0,97	0,81	0,99	0,05
	Транспорт	1,00	0,56	1,00	0,47
	Економија	1,00	0,60	0,99	0,50
	Медицина	0,97	0,66	1,00	0,32
	Обрада текста	0,85	0,25	1,00	0,14
	Софтвер	0,98	0,52	0,97	0,29
	Графови	1,00	0,73	1,00	0,48

У детекцију аномалија, претерано или недовољно тренирање се дешава одабиром оптимизационе метрике која привидно даје добре резултате. На слици 22 је дат пример перформанси кластеризације методом К-средњих вредности за различите оптимизационе метрике над једним скупом података из репозиторијума. Показано је да иако одређене оптимизационе метрике имају добре перформансе, алгоритам није детектовао аномалије у подацима или је детектовао нормалне инстанце као аномалије.



Слика 22: Кластеризација методом К-средњих вредности за различите оптимизационе метрике над једним скупом података. Показано је да одређене оптимизационе метрике иако имају добре перформансе не детектују аномалије у подацима или детектују нормалне инстанце као аномалије.

Дистрибуција перформанси алгоритама над репозиторијумом података треба да буде нормална за одређену оптимизациону метрику како би предложено и постојећа решења могла да се пореде и добију валидни резултати поређења. Како би се потврдио исправан одабир оптимизационих метрика, на слици 23 су приказане просечне вредности перформансе алгоритама над репозиторијум података где су оне груписане у категорије по перформансама. Показано је да за већину скупове података оптимизационе метрике *accuracy* и *recall* приказују привидно добре перформансе. Међутим, за *f1* оптимизациону метрику је показано да је униформна дистрибуција перформанси.



Слика 23: Дистрибуција просечне вредности перформанси алгоритама за детекцију аномалија над креираним репозиторијумом података коришћењем различитих оптимизационих метрика. Показано је да коришћењем *f1* оптимизациоине метрике дистрибуција перформанси је униформна.

5.1.4 Мета подаци за карактеризацију аномалија

Како би се дали одговори на полазне хипотезе, предложено решење је потребно поредити са постојећим решењима из отворене литературе и индустрије која се заснивају на мета подацима који се израчунавају из података. То обухвата мета податке који припадају групама простих мета података, статистичких мета података и мета података заснованих на теорији информација. У табели 12 је дат преглед различитих типова мета података коришћених у експериментима са бројем атрибута и кратким описом за сваку групу. Мета подаци засновани на доменском знању представљају предложено решење у овом раду.

Табела 12: Преглед различитих типова мета података коришћених у експериментима са бројем атрибута и кратким описом за сваку групу. Мета подаци засновани на доменском знању представљају предложено решење у овом раду. Предложено решење садржи укупно 5 мета података, али пошто мета подаци нису јединствени за један скуп података, већ је могуће имати више вредности за један мета податак, број мета података који се користи у имплементацији је 18.

Мета подаци	Број атрибута	Опис
Прости мета подаци	11	Прости мета подаци садрже карактеристике које су директно израчунате из података, захтевају мало ресурса и имају малу комплексност израчунавања
Статистички мета подаци	26	Статистички мета подаци садрже статистичке карактеристике које су израчунате из података, захтевају значајну количину ресурса и имају значајну комплексност израчунавања
Мета подаци засновани на теорији информација	8	Мета подаци засновани на теорији информација садрже карактеристике које су израчунате из података, захтевају значајну количину ресурса и имају значајну комплексност израчунавања
Мета подаци засновани на доменском знању	5 (18)	Предложено решење садржи карактеристике засноване на доменском знању које су директно израчунате из података, захтевају мало ресурса и имају малу комплексност израчунавања
Комплетан скуп мета података	93	Представља комплетан скуп анализираних мета података који су израчунате из података, захтевају значајну количину ресурса и имају значајну комплексност израчунавања. Ова група садржи просте и статистичке мета податке, мета податке засноване на теорији информација и предложено решење.

Наведене групе мета података представљају карактеристике података на начин да могу да се користе за одабир алгоритама у аутоматизованим системима за машинско учење, углавном за обраду текста и слика. За имплементацију постојећих решења се користи библиотека *pyMFE* [47] која представља савремено решење за мета учење имплементирано у *Python* програмском језику. Ово решење садржи комплетан скуп мета података који могу да се израчунавају из података. Како би у експериментима могли да се пореде мета подаци кроз дефинисане критичне захтеве, у табелама 13-17 је дат приказ мета података кроз дефинисане критичне захтеве. Приликом постављања полазних хипотеза је дат начин одређивања испуњености критичних захтева за различите групе мета података. Прегледом отворене литературе и дефинисањем критичних захтева могуће је одредити да ли група испуњава наведене критичне захтеве. Показано је да мета подаци засновани на простим функцијама и на доменском знању испуњавају критичне захтеве за примену у аутоматизованим системима за машинско учење.

Табела 13: Скуп простих мета података који се састоји од 11 атрибута. За сваки мета податак је дат кратак опис и преглед критичних захтева које испуњава.

Назив	Опис	Неутралност	Скалабилност	Повезаност	Једноставност
attr_to_inst	Однос између броја атрибута и инстанци	Да	Да	Да	Да
cat_to_num	Однос између броја категоричких и нумеричких атрибута	Да	Да	Да	Да
freq_class	Фреквенцију појављивања сваке различите класе	Да	Да	Да	Да
inst_to_attr	Однос између броја инстанци и атрибута	Да	Да	Да	Да
nr_attr	Укупан број атрибута	Да	Да	Да	Да
nr_bin	Укупан број бинарних атрибута	Да	Да	Да	Да
nr_cat	Укупан број категоричких атрибута	Да	Да	Да	Да
nr_class	Укупан број класа	Да	Да	Да	Да
nr_inst	Укупан број инстанци	Да	Да	Да	Да
nr_num	Укупан број нумеричких атрибута	Да	Да	Да	Да
num_to_cat	Однос између броја нумеричких и категоричких атрибута	Да	Да	Да	Да

Табела 14: Скуп статистичких мета података који се састоји од 26 атрибута. За сваки мета податак је дат кратак опис и преглед критичних захтева које испуњава.

Назив	Опис	Неутралност	Скалабилност	Повезаност	Једноставност
can_cor	Канонска повезаност између података	Да	Не	Да	Не
cor	Апсолутна вредност корелације атрибута	Да	Не	Да	Не
cov	Апсолутна вредност коваријансе атрибута	Да	Не	Да	Не
eigenvalues	Сопствена вредност матрице коваријансе	Да	Не	Да	Не
g_mean	Геометријска средина атрибута	Да	Не	Да	Не
h_mean	Хармонска средина атрибута	Да	Не	Да	Не
kurtosis	Расподела вероватноће случајне променљиве реалне вредности	Да	Не	Да	Не
mad	Средња апсолутна девијација прилагођена фактором	Да	Не	Да	Не
max	Максимална вредност атрибута	Да	Да	Да	Да
mean	Средња вредност атрибута	Да	Да	Да	Да
median	Медијана атрибута	Да	Да	Да	Да
min	Минимална вредност атрибута	Да	Да	Да	Да
nr_cor_attr	Број различитих високо повезаних парова атрибута	Да	Не	Да	Не
nr_disc	Број канонске корелације између сваког атрибута и класе	Да	Не	Да	Не
nr_norm	Број атрибута који имају нормалну расподелу	Да	Не	Да	Не
nr_outliers	Број атрибута са бар једном вредношћу одступања	Да	Не	Да	Не
p_trace	*Pillai's trace* - доказ да варијабла има статистички значајан утицај	Да	Не	Да	Не
range	Оквир атрибута или разлика максималне и минималне вредности	Да	Не	Да	Не
roy_root	*Roy's* највећи корен	Да	Не	Да	Не
sd	Стандардна девијација	Да	Не	Да	Не
sd_ratio	Статистички тест за хомогеност коваријанси	Да	Не	Да	Не
skewness	Искривљеност дистрибуције	Да	Не	Да	Не
sparsity	Метрика реткости атрибута	Да	Не	Да	Не
t_mean	Скраћена средња вредност атрибута	Да	Не	Да	Не
var	Вредност варијансе	Да	Не	Да	Не
w_lambda	*Wilks' lambda* вредност	Да	Не	Да	Не

Табела 15: Скуп мета података заснован на теорији информација који се састоји од 8 атрибута. За сваки мета податак је дат кратак опис и преглед критичних захтева које испуњава.

Назив	Опис	Неутралност	Скалабилност	Повезаност	Једноставност
attr_conc	Коефицијент концентрације сваког пара различитих атрибута	Да	Не	Да	Не
attr_ent	*Shannon* ентропија предиктивних атрибута	Да	Не	Да	Не
class_conc	Коефицијент концентрације између атрибута и класа	Да	Не	Да	Не
class_ent	*Shannon* ентропија класе	Да	Не	Да	Не
eq_num_attr	Број еквивалентних атрибута за предиктивни задатак	Да	Не	Да	Не
joint_ent	Заједничка ентропија између атрибута и класе	Да	Не	Да	Не
mut_inf	Међусобне информације између атрибута и класе	Да	Не	Да	Не
ns_ratio	Вредност шума у подацима	Да	Не	Да	Не

Табела 16: Скуп мета података заснован на доменском знању који се састоји од 5 атрибута. За сваки мета податак је дат кратак опис и преглед критичних захтеве које испуњава.

Назив	Опис	Неутралност	Скалабилност	Повезаност	Једноставност
data_type	Тип података	Да	Да	Да	Да
data_dom	Домен података	Да	Да	Да	Да
anom_loc	Локалитет аномалија	Да	Да	Да	Да
anom_dim	Димензионални простор аномалија	Да	Да	Да	Да
anom_ratio	Однос између броја аномалија и инстанци	Да	Да	Да	Да

Табела 17: Комбиновани скуп мета података заснован на статистичким функцијама и теорији информација који се састоји од 30 атрибута. За сваки мета податак је дат кратак опис и преглед критичних захтева које испуњава.

Назив	Опис	Неутралност	Скалабилност	Повезаност	Једноставност
gravity	Удаљеност између мањинских и већинских центара масе	Да	Не	Да	Не
iq_range	Вредност интерквартилног опсега	Да	Не	Да	Не
lh_trace	*Lawley-Hotelling trace*	Да	Не	Да	Не
attr_conc.sd	Стандардна девијација коефицијента концентрације сваког пара различитих атрибута	Да	Не	Да	Не
attr_count	Број атрибута	Да	Не	Да	Не
attr_ent.sd	Стандардна девијација *Shannon* ентропије	Да	Не	Да	Не
can_cor.sd	Стандардна девијација канонске повезаности	Да	Не	Да	Не
class_conc.sd	Стандардна девијација коефицијента концентрације између атрибута и класа	Да	Не	Да	Не
cor.sd	Стандардна девијација апсолутних вредности корелације атрибута	Да	Не	Да	Не
cov.sd	Стандардна девијација апсолутних вредности коваријансе атрибута	Да	Не	Да	Не
eigenvalues.sd	Стандардна девијација сопствене вредности матрице коваријансе	Да	Не	Да	Не
freq_class.sd	Стандардна девијација фреквенције појављивања сваке различите класе	Да	Не	Да	Не
g_mean.sd	Стандардна девијација геометријске средина атрибута	Да	Не	Да	Не
h_mean.sd	Стандардна девијација хармонске средина атрибута	Да	Не	Да	Не
iq_range.sd	Стандардна девијација вредности интерквартилног опсега	Да	Не	Да	Не
joint_ent.sd	Стандардна девијација заједничке ентропије између атрибута и класе	Да	Не	Да	Не
kurtosis.sd	Стандардна девијација расподела вероватноће случајне променљиве реалне вредности	Да	Не	Да	Не
mad.sd	Стандардна девијација средње апсолутне девијација прилагођене фактором	Да	Не	Да	Не
max.sd	Стандардна девијација максималне вредности атрибута	Да	Не	Да	Не
mean.sd	Стандардна девијација средње вредности атрибута	Да	Не	Да	Не
median.sd	Стандардна девијација медијане атрибута	Да	Не	Да	Не
min.sd	Стандардна девијација минималне вредности атрибута	Да	Не	Да	Не
mut_inf.sd	Стандардна девијација међусобне информације између атрибута и класе	Да	Не	Да	Не
range.sd	Стандардна девијација опсега атрибута	Да	Не	Да	Не
row_count	Број редова	Да	Не	Да	Не
sd.sd	Стандардна девијација стандардне девијације	Да	Не	Да	Не
skewness.sd	Стандардна девијација искривљености дистрибуција	Да	Не	Да	Не
sparsity.sd	Стандардна девијација метрике реткости атрибута	Да	Не	Да	Не
t_mean.sd	Стандардна девијација скраћене средње вредности	Да	Не	Да	Не
var.sd	Стандардна девијација варијансе	Да	Не	Да	Не

У експериментима су поређени искључиво мета подаци који се израчунавају из података, док остала решења у домену мета учења нису евалуирана, као што су мета подаци који се израчунавају из модела или приступ обележавања, јер имају другачије карактеристике па није могуће директним поређењем да се дође до одговора на постављене хипотезе.

5.1.5 Функције за мерење удаљености

Функције за мерење удаљености се користе у аутоматизованим системима за машинско учење као функције за мерење сличности између скупова података и представљају битан модул компоненте. Сличност између скупова података мери се као удаљеност између мета података. У другом поглављу, прегледом отворене литературе је дат преглед функција за мерење удаљености које се користе у алгоритмима машинског учења. Еуклидска и *Manhattan* функције представљају заступљене функције које се користе у алгоритмима за машинско учење. Остале функције за мерење удаљености, као што су *Hamming* и *Minkowski* су изведене из претходних функција или представљају њихову генерализацију, па немају примену у домену овог рада, због чега и нису даље анализиране и евалуиране у експериментима. Ако се одабере функција за мерење удаљености која не представља сличност на добар начин, компонента за одабир алгоритма може постићи лоше перформансе упркос мета подацима који испуњавају све дефинисане критичне захтеве.

Комплексност наведених функција омогућава да се оне користе у различитим алгоритмима за машинско учење на начин да не трансформишу податке приликом мерења удаљености. Са друге стране, њихова комплексност утиче на перформансе алгоритма, јер ако карактеристике података не могу да се пореде наведеним функцијама, резултати експеримената ће бити лоши. Слично као и са одабиром алгоритама за детекцију аномалија, циљ је узети алгоритме који имају различити степен трансформација над подацима. Тако се дошло до идеје да се поред наведених функција, алгоритми који су одабрани за детекцију аномалија користе и у сврху мерења удаљености. Оваквим приступом се омогућава поређење алгоритама из различитих група алгоритама за машинско учење који ће се користе за мерење удаљености између скупова података коришћењем мета података. Дакле, поред стандардних функција за мерење удаљености користе се и алгоритми који су се користили за детекцију аномалија, а то су Гаусова дистрибуција, линеарна регресија, робусна анализа главних компонената, кластеризација методом К-средњих вредности и аутоенкодер. Сви параметри алгоритама су коришћени на исти начин као и приликом њихове примене за детекцију аномалија, што је описано раније у овом поглављу. У наставку је дат преглед тих алгоритама при чему су представљене главне карактеристике као и начин мерења удаљености.

Гаусова дистрибуција представља алгоритам који за циљ има да представи дистрибуцију података кроз параметре као што су средња вредност и стандардне девијације. Овај алгоритам даје добре перформансе када су подаци описани нормалном дистрибуцијом. Због тога се не очекује да овај алгоритам да добре резултате за мерење удаљености јер мета подаци углавном немају нормалну дистрибуцију. Алгоритам мери удаљеност сваке инстанце од центра дистрибуције при чему та вредност представља удаљеност од осталих скупова података. За сваки скуп података се израчунава колико је удаљен од центра дистрибуције коришћењем мета података, тако да скупови података који имају сличну удаљеност се сматрају да су слични. Први корак представља одређивање параметара за описивање дистрибуције мета података. Следећи корак је рачунање удаљености од центра дистрибуције за сваки скуп података.

Линеарна регресија представља статистички алгоритам који за циљ има креирање регресионе праве која предвиђа континуалне вредности за податке који су линеарно зависни. Чест случај када су подаци линеарно зависни укључује постојање временске димензије у подацима. Идеја примене овог алгоритма за проблем мерења сличности је представљен као мерење удаљености за сваки скуп података од линеарне праве. Први корак за мерење удаљености је креирање регресионе праве из израчунатих мета података. Након тога се за сваки скуп података рачуна растојање коришћењем неке од простих функција за мерење удаљености. Израчунате удаљености се сортирају по близини при чему мета подаци који имају сличне удаљености се сматрају да су слични. Као и у претходном случају, овај метод захтева да подаци буду линеарно зависни, па се не очекује да резултати алгоритма у овој примени буду добри.

Робусна анализа главних компонената је алгоритам који се заснива на декомпозицији која трансформише податке у суб-димензионални простор. Након тога се подаци враћају у оригинални простор и прилико креирања података у том простору се мери грешка. Идеја је да израчуната грешка приликом реконструкције података представља сличност између података, где ће се скупови података са сличном грешком сматрати као слични. Први корак представља трансформацију израчунатих мета података у мањи суб-димензионални простор. Након тога се подаци реконструишу враћањем у оригинални простор. Приликом реконструкције рачуна се грешка која представља удаљеност од оригиналних података, и на основу те вредности се проналазе скупови података који су слични. Очекује се да овај алгоритам да добре резултате за различите класификације јер није условљен типом података или дистрибуцијом података.

Кластеризација методом К-средњих вредности је алгоритам који се користи за груписање података при чему се рачуна удаљеност од центра кластера и инстанце података припадају кластерима којима су близу. За потребу мерења удаљености, подаци могу да се сматрају да су слични ако њихови мета подаци имају сличну удаљености од центра кластера. Први корак је одређивање центра кластера након чега се мери удаљеност за сваки скуп података коришћењем мета података. Затим се за податке који имају сличне удаљености од центра кластера сматра да су близу. Овај приступ се заснива на просторној удаљености, а не семантичкој сличности. Како се удаљеност мери у димензионалном простору, користе се просте функције за мерење удаљености претходно описане у овом раду.

Аутоенкодер представља специфичан тип неуронских мрежа која се заснива на представљању података у суб-димензионалном простору. Слично као код робусне анализе главних компонената, приликом враћања у оригинални простор, реконструисани подаци могу да садрже грешку, поготово ако ти подаци представљају аномалије. За мерење удаљености, идеја је да вредност грешке коју подаци имају приликом њихове реконструкције представља удаљеност. Први корак је представљање израчунатих мета података у суб-димензионалном простору. Након тога се подаци реконструишу враћањем у оригинални димензионални простор. Приликом тог враћања се рачуна грешка у реконструкцији. Та грешка представља удаљеност па подаци који имају сличну вредност могу да се сматрају да су близу. Очекује се да овај алгоритам да добре резултате за различите класификације јер није условљен типом и дистрибуцијом података.

Наведене функције и алгоритми за мерење удаљености између података као резултат дају сортирану листу скупова података по близини коришћењем мета података. Како би се одредило колико се најближих скупова података узима приликом евалуације за пренос знања о алгоритму и перформансама, користи се параметар k који одређује број најближих скупова података који се узима приликом одабира алгоритма. Ако је k велики број, алгоритми који постижу добре резултате у општем случају или за велики број скупова података из креираног

репозиторијума, као што је кластеризација методом К-средњих вредности, имају већи утицај на одабир оптималног алгоритма. Међутим, ако је вредност *k* мала, карактеристике података и мета подаци имају већи утицај на одабир алгоритма. У опису експеримената и добијеним резултатима је објашњен начин одређивања параметра *k*.

5.1.6 Архитектуре за различите локације извршавања

Архитектуре одабране за евалуирање имплементиране компоненте у експериментима треба да покрију све битне карактеристике за различите локације извршавања и предложене топологије компоненте. Како би симулирали имплементацију решења у различитим окружењима, у наставку је дат преглед главних карактеристика сваког типа архитектуре, као и локације извршавања за коју је та архитектура погодна.

Прва архитектура која ће бити представљена је централна јединица процесирања која је за сада најзаступљенија архитектура која се користи. Ова архитектура се заснива на великом број транзистора и садржи неколико језгара са аритметичко логичким јединицама. Карактеристике ове архитектуре су релативно брзо време извршавања, као и потрошња електричне енергије. Због великог броја транзистора на чипу, као и високе фреквенције рада, потрошња електричне енергије може да буде значајна. У поређењу са другим архитектурама, представља добар однос између брзине извршавања и потребних ресурса за извршавање. Овај тип архитектуре је заступљен у облаку и на крајњем уређају, где се у зависности од потреба скалира број језгара са аритметичко логичким јединицама. У овом раду је ова архитектура евалуирана за потребе извршавања у облаку као и на крајњим уређајима. Уређај коришћен за тестирање је *Intel Xeon CPU 2,20 GHz*. Одабир уређаја је условљен имплементацијом и јавно доступним архитектурама.

Друга архитектура која ће бити представљена је графичка јединица процесирања која је за сада присутна у решењима у облаку где је могуће урадити велику паралелизацију приликом обраде података. Ова архитектура се заснива на великом број транзистора у поређењу са претходним решењем и садржи неколико хиљада језгара са аритметичко логичким јединицама. За разлику од претходног решења где језгра имају велику моћ процесирања, ово решење садржи велики број јединица које имају малу моћ процесирања. Карактеристике ове архитектуре су сличне као у претходном решењу, при чему је овде још већи интензитет карактеристика. То значи да је време извршавања још мање, при чему је и потреба за ресурсима већа. Због великог броја транзистора на чипу потрошња електричне енергије за имплементацију може да захтева велику количину ресурса за извршавање. У поређењу са другим архитектурама, представља добар приступ када је могуће паралелно процесирање података. У овом раду је ова архитектура евалуирана за потребе извршавања у облаку. Уређај коришћен за тестирање је *Nvidia Tesla K80 GPU*. Одабир уређаја је условљен имплементацијом и јавно доступним архитектурама.

Наведене архитектуре представљене до сада припадају *controlflow* парадигми где је циљ креирати програм са циљем да контролише ток података кроз хардвер. Архитектуре које ће бити представљене у наставку се реферишу као *dataflow* парадигма, где је циљ да програм конфигурише хардвер тако што се као резултат добије граф извршавања који се мапира на хардвер. Примери *dataflow* парадигме су *FPGA* и *ASIC* архитектуре. Парадигма се заснива на малом број транзистора у поређењу са *controlflow* парадигмом. За разлику од претходних решења где архитектуре имају велику моћ процесирања, представљено решење се заснива на потпуно другој парадигми програмирања. *Dataflow* парадигма захтева присуство *controlflow* архитектуре јер представља акцелератор, што значи да не извршава целу компоненту, већ се

модули погодни за извршавање на *dataflow* парадигми мигрирају. Карактеристике ових архитектура су такве да због малог броја транзистора у поређењу са претходним решењима захтевају мање ресурса. У поређењу са другим архитектурама, представљају добар приступ када је неопходно водити рачуна о потрошњи електричне енергије, при чему могу да се постигну добре перформансе из аспекта времена извршавања под одређеним условима. У овом раду је ова архитектура евалуирана за потребе извршавања у облаку и на крајњим уређајима. Уређај коришћен за тестирање је *Google TPU* са *8* језгара. Одабир уређаја је условљен имплементацијом и јавно доступним архитектурама. Како би се имплементирала компонента на *dataflow* архитектури, и при томе постигле добре перформансе неопходно је задовољити критичне захтеве везане за ту парадигму, као што је велики број итерација компоненте где се у свакој итерацији врши комплексна обрада над подацима. Поред тога, погодно је да подаци долазе са одређеном фреквенцијом са неког извора.

5.2 Опис експеримената

Експерименти креирани у овом поглављу имају за циљ да одговоре на дефинисане полазне хипотезе. Полазне хипотезе су дефинисане тако да валидирају да ли предложено решење постиже исте или боље перформансе у односу на постојећа решења, и под којим условима. Полазне хипотезе валидирају перформансе различитих решења и проверавају испуњеност дефинисаних критичних захтева. Критични захтеви су дефинисани тако да постављају захтеве од стране аутоматизованих система за машинско учење где се мета подаци користе за одабир алгоритма. Евалуација предложеног решења и имплементиране компоненте је подељена у пет различитих експеримената. Прва четири експеримента евалуирају предложено решење, испуњеност критичних захтева и функције за мерење сличности између мета података. Пети експеримент евалуира перформансе имплементиране компоненте кроз аспекте као што су време извршавања и потребни ресурси за извршавање на одабраним архитектура.

У претходном поглављу је дата поставка експеримената, где су дефинисани алгоритми и мета подаци који се користи у експериментима у наставку. Експерименти су дефинисани на начин да провере да ли су критични захтеви испуњени за предложено решење. Критични захтеви везани за повезаност и неутралност се проверавају коришћењем скупова података из различитих извора, различитих домена и са различитим типовима података. Критични захтев везан за скалабилност се потврђује коришћењем мета података који се израчунавају на основу доменског знања и простих функција. Критични захтев везан за једноставност се потврђује експериментима где се проверава могућност процене мета података у случајевима када подаци нису присутни, при чему доменски експерт или креатор података може да процени мета податке.

5.2.1 Експеримент 1 - Евалуација предложених мета података

Евалуација предложених мета података представља први експеримент и односи се на евалуацију предложеног решења. Како би се предложено решење евалуирало, неопходно је поредити га са постојећим решењима по аспектима од интереса. Постојећа решења се односе на мета податке из отворене литературе који се израчунавају из података и могу да се користе у евалуацији. Списак постојећих мета података који се евалуирају су представљени у претходном поглављу.

Полазне хипотезе које експеримент треба да валидира су:

- Предложени скуп мета података може да задовољи наведене критичне захтеве
- Предложени скуп мета података постиже исте или боље резултате у односу на постојеће скупове мета података за различите типове података, различите типове аномалија, као и за различите области којима подаци припадају

Експеримент је дизајниран тако да валидира постављене критичне захтеве везане за неутралност и повезаност. Неутралност дефинише да мета подаци треба да ефикасно описују карактеристике аномалија за различите области и типове података. Повезаност дефинише да мета подаци треба да ефикасно описују карактеристике аномалија за различите оптимизационе метрике. Експеримент пореди предложено и постојећа решења наведена у поставци експеримената. Оптимизационе метрике које се користе за евалуацију и поређење су наведене у поставци експеримената. Резултати експеримента треба да покажу да ли предложени скуп мета података постиже исте или боље резултате у односу на постојеће скупове мета података за различите типове података, различите локалитете аномалија, као и за различите области којима подаци припадају.

Експеримент се састоји од четири фазе. У првој фази се пореде различите групе постојећих решења. За сваки скуп података из репозиторијума се одређује оптимални алгоритам узимањем алгоритма који је дао најбоље перформансе за дату оптимизациону метрику сличног скупа података. Затим, коришћењем тог алгоритма се евалуира дати скуп података и добијени резултати се агрегирају и пореде за различите групе мета података. У једној итерацији се евалуира један скуп података.

Директним поређењем постојећих решења се долази до одговора на питање под којим условима предложено решење даје боље резултате. Међутим, није могуће потврдити да је одабран минимални скуп мета података који даје најбоље могуће резултате за постојећа решења или предложено решење. У другој фази се случајном претрагом простора тражи минимални скуп комбинацијом свих постојећих мета података, где величина минималног скупа који се тражи итерира од један до пет, јер је толико мета података предложено. Због времена трајања једне итерације, број итерација за сваку величину скупа је ограничен на *1000*, при чему се за сваку величину скупа узима најбоља комбинација. У итерацијама се искључиво евалуира *f1* оптимизациона метрика, како би се избегли привидно добри резултати који се појављују за остале оптимизационе метрике, и смањило време потребно за извршавање експеримената.

У трећој фази се случајном претрагом простора тражи минимални скуп мета података комбинацијом свих постојећих мета података који испуњавају критичне захтеве, где величина минималног скупа који се тражи итерира од један до пет, јер је толико мета података предложено. Због времена трајања једне итерације, број итерација за сваку величину скупа је ограничен на *1000*, при чему се за сваку величину скупа узима најбоља комбинација. У итерацијама се искључиво евалуира *f1* оптимизациона метрика, како би се избегли привидно добри резултати који се појављују за остале оптимизационе метрике, и смањило време потребно за извршавање експеримената.

У четвртој фази се случајном претрагом простора тражи минимални скуп комбинацијом предложених мета података који испуњавају критичне захтеве, где величина минималног скупа који се тражи итерира од један до пет, јер је толико мета података предложено. Због времена трајања једне итерације, број итерација за сваку величину скупа је ограничен на *1000*, при чему се за сваку величину скупа узима најбоља комбинација. Оваквим приступом је извршена парцијална претрага простора како би се дошло до оптималног скупа

мета података и показало који мета подаци дају добре резултате за одабир алгоритама за детекцију аномалија и величина скупа мета података.

За евалуацију резултата неопходно је мерити сличност између скупова података. Битно је да се обезбеди да се сва решења евалуирају у истом окружењу, односно под истим условима. То подразумева да се користи иста функција за мерење удаљености. Како није лако одредити функцију која за све случајеве даје добре резултате, експеримент евалуира различита решења истим скупом функција при чему је за представљање резултата узета функција која даје најбоље резултате.

Циљ експеримента је да евалуира различите групе мета података као и да провери да ли постоји минимални скуп који задовољава дефинисане критичне захтеве. Резултати експеримента су представљени по различитим типовима података, локалитету аномалија и различитим доменима. Оваквим приступом се показује у којим условима предложено решење даје боље резултате. Такав ниво грануларности омогућава лако тумачење резултата које је предиктивно за даље коришћење предложеног решења.

5.2.2 Експеримент 2 - Евалуација функција за мерење удаљености

Мерење удаљености између скупова података је предиктивно за резултате првог експеримента где се валидира предложено решење и због тога представља битну целину за коју је неопходно креирати експеримент. Циљ експеримента је да се евалуирају различите функција за мерење удаљености и да се закључи да ли постоји веза између типа мета података и типа функција за мерење удаљености. Ако постоји веза између типова мета података и функција за мерење удаљености, то омогућава креатору система за аутоматизовано машинско учење да користи оптимални тип функције за одговарајуће мета податке. У супротном, неопходно је користити скуп функција за мерење удаљености које би се користиле у таквом систему.

Полазна хипотеза коју експеримент треба да валидира је:

- За предложени скуп мета података је могуће одредити тип функција за мерење удаљености између скупова података који у већини случајева даје боље резултате од осталих типова функција које се најчешће користе за мерење удаљености између скупова података

Експеримент пореди различите функције за сваку групу мета података тако што се извршава евалуација перформанси одабраних алгоритама и приликом тога се користе различите функције за мерење удаљености. Ако се за одређени тип мета података једна група за мерење удаљености покаже као оптимална, може да се закључи да је та група погодна за евалуирани тип мета података. У супротном, закључује се да одабрани тип мета података нема одређену групу функција која је погодна за мерење удаљености. Оптимизационе метрике које се користе за евалуацију и поређење су наведене у поставци експеримената. Функције за мерење удаљености су одабране по групама, тако да свака група има различите карактеристике. Одабир функција за мерење удаљености је дат у поставци експеримената. Експеримент се извршава над целим репозиторијумом података за *f1* оптимизациону метрику.

Одабир алгоритама се врши рангирањем скупова података по близини и узимањем алгоритма који је најзаступљенији за првих *k* скупова података и дату оптимизациону метрику. Експеримент ће евалуирати резултате за вредности *k* од *1* до *10*, при чему ће се за *k* узети вредност која даје добре перформансе.

Први и други експеримент су повезани јер се резултати експеримената користе за међусобну валидацију. Остали експерименти у овом раду нису повезани што значи да је извршена одвојена евалуација или анализа у зависности од типа експеримента.

5.2.3 Експеримент 3 - Комплексност предложених мета података

Да би се предложени мета подаци користили у аутоматизованим системима за машинско учење, неопходно је да задовољавају дефинисане критичне захтеве. Овај експеримент се бави евалуацијом критичних захтева везаних за скалабилност где је циљ да се за сваки предложени мета податак анализира комплексност.

Полазна хипотеза коју експеримент треба да валидира је:

- Предложени скуп мета података може да задовољи наведене критичне захтеве

Критични захтев везан је за скалабилност предложеног решења. Експеримент одређује теоријску комплексност функција коришћењем метода описаних у наставку. На основи прегледа отворене литературе, следећи концепти су коришћени за евалуацију предложеног решења. Велика *O* нотација је приступ који описује понашања функције кроз аспект граничних перформанси за вредности улаза који теже броју *n* или бесконачности у теорији. У пракси се овај приступ користи за одређивање комплексности извршавања алгоритма у односу на број података на улазу. Циљ коришћења велике *O* нотације у овом раду је провера да ли предложени мета подаци испуњавају критични захтев који се односи на скалабилност. Резултат експеримента представља теоријске граничне вредности комплексности предложених функција на основу величине улазних података. Комплексност функција је евалуирана појединачно и колективно. Појединачно евалуирање комплексности је заступљено у стандардном приступу примене ове нотације, док је евалуирање целе групе функција за израчунавање мета података неопходно у овом случају. Како се функције користе за карактеризацију мета података, није могуће парцијално користити скуп функција јер се таквим приступом може десити да се добију перформансе које не задовољавају остале критичне захтеве.

Алтернативни приступ за мерења комплексности функција који се користи у домену развоја софтвера јесте обострана информација из теорије података. Ова методологија мери количину информација које функција забележи процесирањем других информација. Овај приступ има сличности са ентропијом променљиве који мери количину информација садржаном у податку. Ако су променљиве зависне то значи да могу да дефинишу карактеристике аномалија у подацима и самим тим функције за израчунавање имају одређену комплексност. Прегледом отворене литературе показано је да се ова методологија примењује за алгоритме, док је предлог решења у овом раду представљен као скуп функција за које не постоји комплексно израчунавање. Још један приступ за мерење комплексности система представља условна ентропија. Условна ентропија се дефинише као количина информација која је потребна да опише резултат слободне променљиве на основу друге познате променљиве. Применом ове методологије је такође могуће одредити комплексност алгоритма. Прегледом отворене литературе показано је да се наведена методологија примењује за алгоритме искључиво, тако да није погодна за предложени скуп функција.

5.2.4 Експеримент 4 - Процена предложених мета података

Могућност процене предложених мета података представља једну од главних карактеристика предложеног решења. Процена предложених мета података представља карактеристику која није присутна у до сада ниједном предложеном решењу из отворене литературе. Како би се предложено решење евалуирало, неопходно је поредити га са постојећим решењима кроз аспекте од значаја.

Полазне хипотезе које експеримент треба да валидира су:

- Предложени скуп мета података може да задовољи наведене критичне захтеве
- Предложени скуп мета података је могуће проценити искључиво на основу доменског знања и без присуства података

Експеримент је дизајниран тако да валидира постављене критичне захтеве везане за једноставност. Једноставност дефинише да креатор података или доменски експерт треба да има могућност да израчуна мета податке у ситуацијама када подаци нису обележени или нису доступни. Оптимизационе метрике које се користе за евалуацију и поређење су наведене у поставци експеримената. Резултати експеримента треба да покажу да ли предложени скуп мета података може да се процени на основу доменског знања са одређеном грешком, и како грешка у процени утиче на перформансе.

Ако не постоји скуп података са обележеним аномалијама неопходно је оставити могућност да доменски експерт процени мета податке, како би систем могао да предложи оптимални алгоритам за дати проблем. Како доменски експерт може да претпостави да подаци имају аномалије и при томе одреди карактеристике аномалија, неопходно је евалуирати такав приступ са аспекта грешке коју систем прави приликом одабира алгоритма без обележених аномалија или присутних података. Експеримент уводи грешку приликом израчунавања мета података како би симулирао процену доменског експерта и затим евалуира перформансе система. На тај начин се креира функција грешке у односу на перформансе компоненте. Иако овај приступ није типичан за друге методе ненадгледаног учења, у домену детекције аномалија ово је неопходно јер често аномалије нису обележене.

Како би се креирао експеримент, неопходно је одредити који мета подаци могу да садрже грешку или који мета подаци из предложеног скупа мета података могу да се процењују са грешком. Прегледом предложених мета података утврђено је да само мета подаци који имају већу комплексност у претходно дефинисаном експерименту могу да се процене са грешком. Ти мета подаци су локалитет аномалија и количина аномалија у подацима. Остали мета подаци као што су тип података, домен података, као и димензионални простор аномалија у подацима могу једноставно да се процене применом простих функција и због тога се не евалуирају у овом експерименту. Као што је представљено у поставци експеримената, *f1* оптимизациона метрика даје добре резултате када су у питању не балансирани скупови података. Резултат овог експеримента се представљају као информација колико предложено решење добро ради када се мета подаци процењују.

5.2.5 Експеримент 5 - Евалуација пројектоване компоненте у различитим окружењима

Овај експеримент врши евалуацију пројектоване компоненте на различитим архитектурама за различита окружења. За разлику од претходних експеримената који

валидирају предложено решење, овај експеримент се бави евалуацијом пројектоване компоненте. Овај експеримент валидира критичне захтеве који се односе на скалабилност при чему претходни експерименти евалуирају тај критични захтев са теоријске стране, док овај експеримент евалуира тај критични захтев са имплементационе стране.

Полазна хипотеза коју експеримент треба да валидира је:

- Предложени скуп мета података може да задовољи наведене критичне захтеве

Експеримент евалуира имплементирану компоненту на различитим архитектурама са циљем да резултати експеримента покажу која архитектура је погодна за коју локацију извршавања кроз различите аспекте од интереса. Локације извршавања евалуиране у овом експерименти представљају решење у облаку, решење на крајњем уређају и хибридно решење које представља комбинацију претходна два решења. Карактеристике су представљене детаљно у претходним поглављима, тако да ће овде само да се представи кратак преглед анализираних карактеристика. Карактеристике од интереса приликом креирања експеримента представљају време извршавања компоненте, као и потрошња електричне енергије. Време извршавања се мери за фазу тренирања и тестирања како би се донели закључци за различите локације извршавања како и за хибридно решење. Потрошња електричне енергије представља други битан аспект за анализирање локација извршавања као и за одабир оптималне архитектуре за дато окружење. У окружењима где су ресурси ограничени, неопходно је одабрати архитектуру која захтева минимално ресурса за извршавање компоненте.

Одабир архитектура коришћених у овом експерименту је представљен и анализиран у поставци експеримената. Резултати експеримента представљају перформансе система приликом тренирања и закључивања на различитим архитектурама.

5.3 Резултати експеримената

У овом поглављу су представљени резултати експеримената. За сваки експеримент је кратко дат преглед полазних хипотеза које се проверавају. Након тога су представљени резултати експеримента и дискутовано је како они утичу на дефинисане хипотезе. На крају сваког експеримента су дате претње по валидност експеримената (енг. *Threats to validity*). Резултати су подељени по експериментима, али они могу да се поделе у групе за евалуацију експерименталних резултата предложеног решења, као и евалуацију имплементиране компоненте на различитим архитектурама. Прва група садржи прва четири експеримента који валидирају предложено решење и дају одговоре на све дефинисане полазне хипотезе. Друга група додатно анализира перформансе компоненте симулирањем различитих локација извршавања тако што евалуира имплементирану компоненту на различитим архитектурама које се користе на тим локацијама. Након добијања свих потребних резултата, последње поглавље даје одговоре на полазне хипотезе.

5.3.1 Резултати експеримента 1 - Евалуација предложених мета података

Циљ експеримента је да валидира предложено решење поређењем са постојећим решењима. Резултати експеримента показују у којим ситуацијама предложено решење даје исте или боље резултате у односу на постојећа решења. Полазне хипотезе које овај експеримент треба да валидира су да предложени скуп мета података може да задовољи наведене критичне захтеве и да предложени скуп мета података постиже исте или боље резултате у односу на постојеће скупове мета података за различите типове података,

различите локалитете аномалија, као и за различите области којима подаци припадају. Експеримент се састоји од 4 фазе које су описане у опису експеримента.

У табелама 18-21 су представљени резултати добијени у првој фази где се постојећа решења пореде кроз различите аспекте од интереса и за различите оптимизационе метрике. Табела је подељена по колонама на различите типове података, локалитет аномалија, као и домен података. По редовима су представљена поређена решења.

Табела 18: Поређење постојећих и предложеног решења кроз аспекте од интереса за *f1* оптимизациону метрику.

f1 оптимизациона метрика		Тип података		Локалитет аномалија			Домен података						
Мета подаци	Цео Репозиторијум	Вишедимензионални подаци	Временски подаци	Локалне аномалије	Глобалне аномалије	Микро-кластери	Производња	Транспорт	Економија	Медицина	Обрада текста	Софтвер	Графови
Прости мета подаци	0,74	0,55	0,74	0,75	0,83	0,66	0,66	0,87	1,00	1,00	0,50	0,83	0,90
Статистички мета подаци	0,68	0,55	0,90	0,68	0,83	0,66	0,66	0,62	1,00	1,00	0,50	0,83	0,90
Мета подаци засновани на теорији информација	0,69	0,55	0,70	0,75	0,66	0,66	0,66	0,75	0,83	1,00	0,50	0,76	0,90
Мета подаци засновани на доменском знању	0,84	0,44	0,90	0,86	0,83	0,66	0,66	0,62	1,00	1,00	0,50	0,93	0,90
Комплетан скуп мета података	0,73	0,55	0,90	0,68	0,83	0,33	0,33	0,87	1,00	1,00	0,50	0,66	0,80

Табела 19: Поређење постојећих и предложеног решења кроз аспекте од интереса за *accuracy* оптимизациону метрику.

accuracy оптимизациона метрика		Тип података		Локалитет аномалија			Домен података						
Мета подаци	Цео Репозиторијум	Вишедимензионални подаци	Временски подаци	Локалне аномалије	Глобалне аномалије	Микро-кластери	Производња	Транспорт	Економија	Медицина	Обрада текста	Софтвер	Графови
Прости мета подаци	0,47	0,55	0,63	0,37	0,77	0,66	0,33	0,62	0,66	0,50	0,50	0,76	0,80
Статистички мета подаци	0,53	0,55	0,90	0,51	0,83	0,66	0,33	0,62	0,66	0,50	0,50	0,93	0,90
Мета подаци засновани на теорији информација	0,52	0,55	0,63	0,55	0,72	0,33	0,33	0,75	0,66	0,50	0,50	0,60	0,80
Мета подаци засновани на доменском знању	0,84	0,44	0,90	0,42	0,83	0,66	0,33	0,62	0,50	0,50	0,50	0,93	0,80
Комплетан скуп мета података	0,52	0,44	0,90	0,48	0,83	0,66	0,33	0,62	0,66	0,50	0,50	0,40	0,90

Табела 20: Поређење постојећих и предложеног решења кроз аспекте од интереса за *precision* оптимизациону метрику.

precision оптимизациона метрика		Тип података		Локалитет аномалија			Домен података						
Мета подаци	Цео Репозиторијум	Вишедимензионални подаци	Временски подаци	Локалне аномалије	Глобалне аномалије	Микро-кластери	Производња	Транспорт	Економија	Медицина	Обрада текста	Софтвер	Графови
Прости мета подаци	0,69	0,55	0,65	0,51	0,50	0,33	0,66	0,87	1,00	0,00	0,50	0,80	0,70
Статистички мета подаци	0,58	0,22	0,61	0,66	0,50	0,33	0,66	0,75	1,00	0,00	0,50	0,73	0,70
Мета подаци засновани на теорији информација	0,61	0,44	0,67	0,75	0,50	0,33	0,66	0,75	0,66	0,00	0,50	0,66	0,70
Мета подаци засновани на доменском знању	0,84	0,55	0,70	0,60	0,50	0,33	0,66	0,62	1,00	0,00	0,50	0,80	0,70
Комплетан скуп мета података	0,58	0,33	0,63	0,66	0,50	0,33	0,33	0,75	1,00	0,00	0,50	0,66	0,70

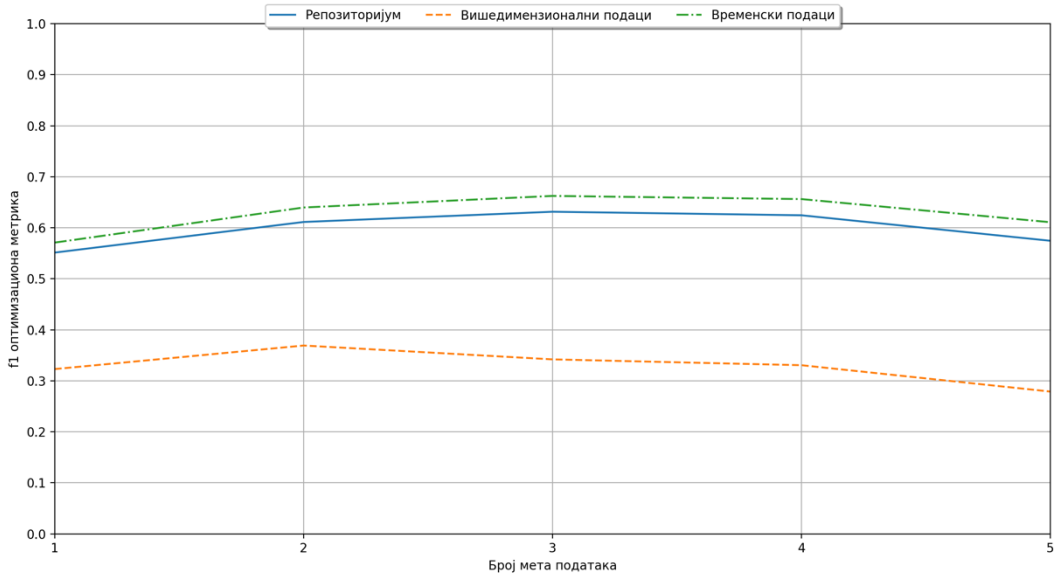
Табела 21: Поређење постојећих и предложеног решења кроз аспекте од интереса за *recall* оптимизациону метрику.

recall оптимизациона метрика		Тип података		Локалитет аномалија			Домен података						
Мета подаци	Цео Репозиторијум	Вишедимензионални подаци	Временски подаци	Локалне аномалије	Глобалне аномалије	Микро-кластери	Производња	Транспорт	Економија	Медицина	Обрада текста	Софтвер	Графови
Прости мета подаци	0,07	0,11	0,43	0,08	0,00	0,00	0,66	0,00	0,00	0,00	0,00	0,06	0,00
Статистички мета подаци	0,40	0,11	0,43	0,06	0,00	0,00	0,66	0,00	0,00	0,00	0,00	0,30	0,00
Мета подаци засновани на теорији информација	0,04	0,11	0,03	0,06	0,00	0,00	0,66	0,00	0,00	0,00	0,00	0,03	0,00
Мета подаци засновани на доменском знању	0,04	0,11	0,03	0,06	0,00	0,00	0,66	0,00	0,00	0,00	0,00	0,06	0,00
Комплетан скуп мета података	0,06	0,11	0,03	0,06	0,00	0,00	0,66	0,00	0,00	0,00	0,00	0,03	0,00

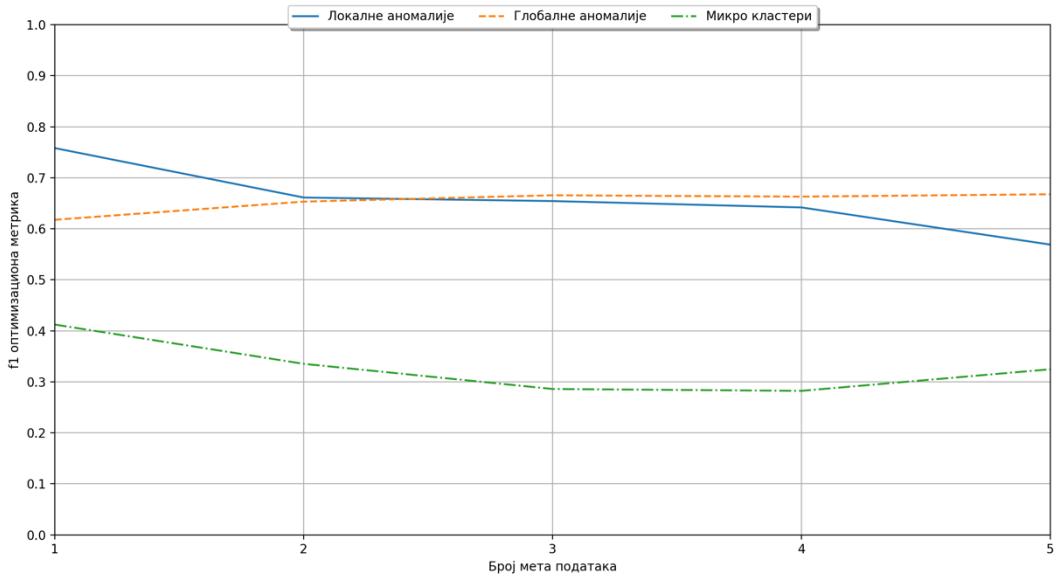
На основу добијених резултата за *f1* оптимизациону метрику предложено решење постиже тачност од 84% док комплетан скуп мета података постиже тачност од 73% за цео репозиторијум података од 63 скупа података. Такође, предложено решење постиже исте или боље резултате у поређењу са постојећим решењима за 55 временских скупова података док за 9 вишедимензионалних скупова података постиже лошије резултате за 10%. Слично томе, предложено решење постиже исте или боље резултате за све локалитете аномалија, као и за домене података, осим транспорта и софтверских записа. Скупови података могу да имају више од једног типа података или да припадају више домена. За *accuracy* и *precision* оптимизационе метрике, предложено решење постиже исте или боље перформансе за цео репозиторијум података, док у категоријама локалних аномалија и графова података постиже лошије перформансе. За *recall* оптимизациону метрику предложено решење постиже лошије перформансе за 1,5%. Примећује се да су перформансе предложеног решења и одређених постојећих решења лошије за ред величина због одабира алгоритма који не даје добре перформансе за ту оптимизациону метрику. На пример, алгоритам кластеризација методом К-средњих вредности даје добре перформансе за *recall* оптимизациону метрику али не постиже добре резултате.

Из табела може да се закључи да у већини случајева предложено решење даје исте или боље резултате у односу на постојећа решења за различите оптимизационе метрике. Као што је показано добијеним резултатима, за већину скупова података предложено решење даје исте или боље перформансе у односу на постојећа решења.

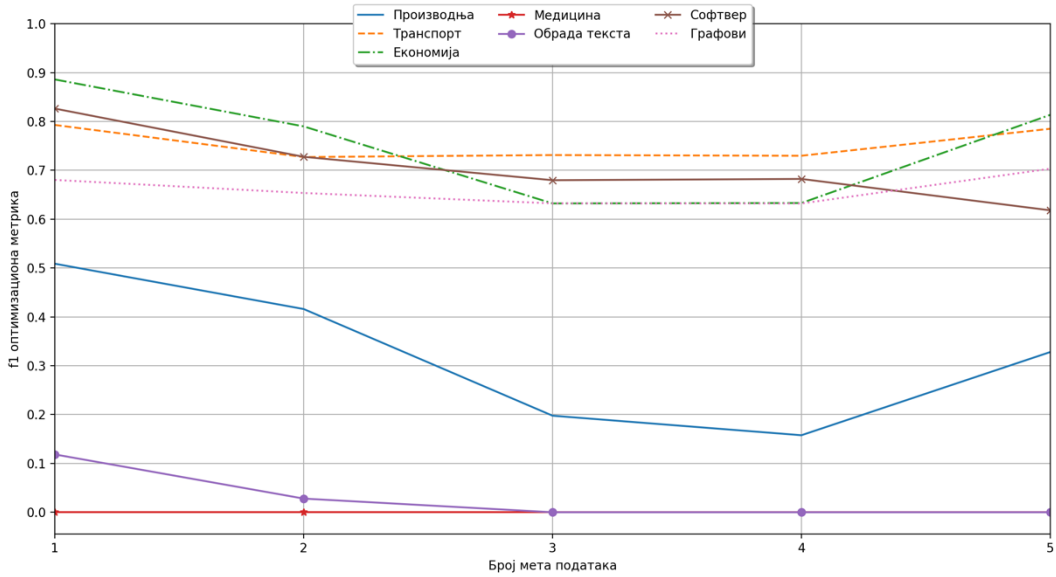
Свакако, оваквим поређењем, не може се потврдити да ли је то скуп који ће дати најбоље могуће резултате за постојећа и предложено решење. На основу поставке експеримента, друга фаза представља претрагу случајног простора за сва постојећа решења. Резултати такве претраге су представљени на сликама 24-26.



Слика 24: Претрага простора методом исцрпног претраживања за различити тип података. Број атрибута је вариран до *5* атрибута за проналажење минималног скупа. Приказани су резултати за *1000* итерација за сваку варијацију до *5* атрибута.

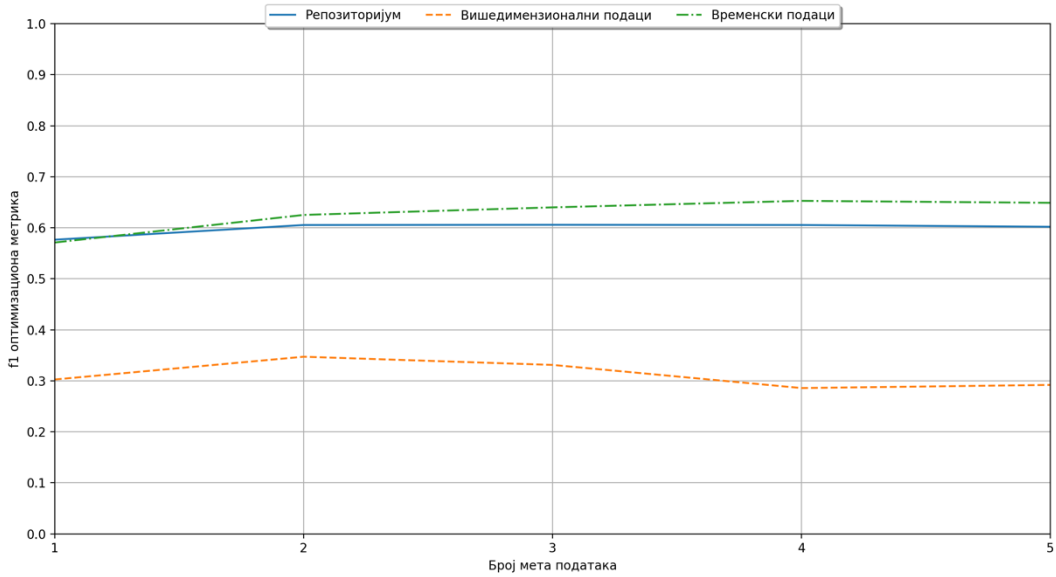


Слика 25: Претрага простора методом исцрпног претраживања за различити локалитет аномалија. Број атрибута је вариран до *5* атрибута за проналажење минималног скупа. Приказани су резултати за *1000* итерација за сваку варијацију до *5* атрибута.

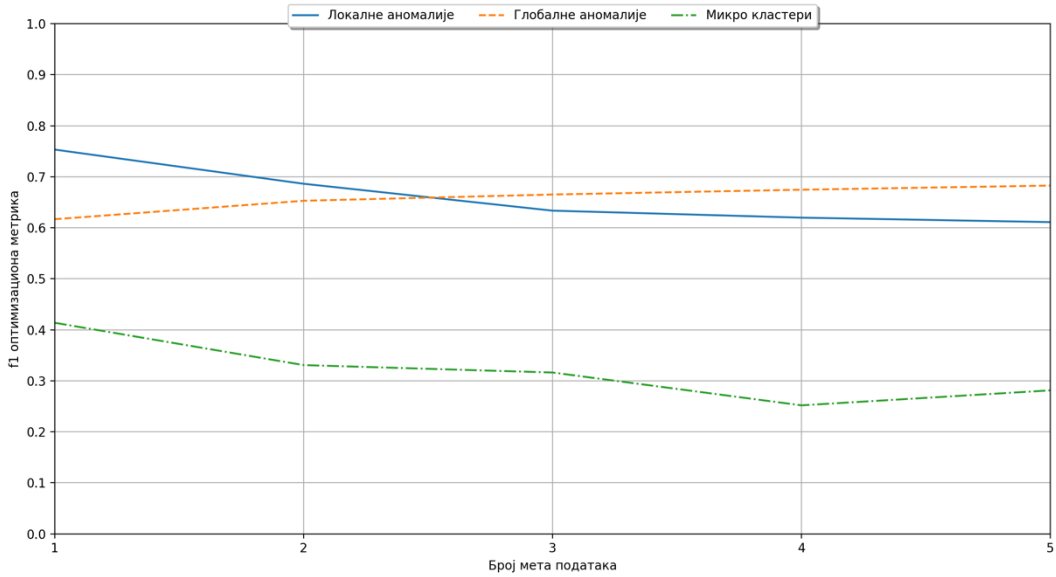


Слика 26: Претрага простора методом исцрпног претраживања за различити домен података. Број атрибута је вариран до *5* атрибута за проналажење минималног скупа. Приказани су резултати за *1000* итерација за сваку варијацију до *5* атрибута.

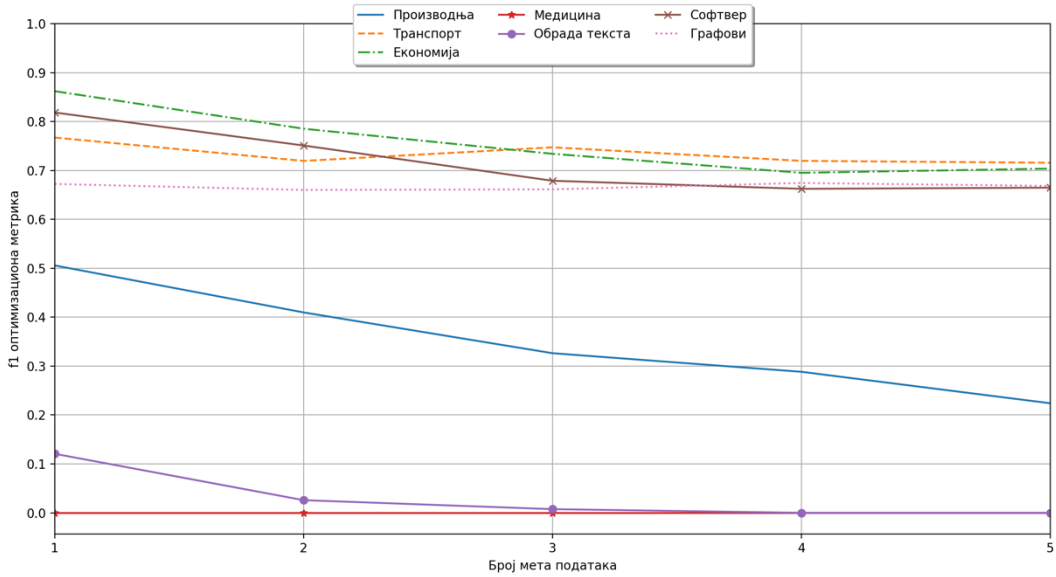
На основу добијених резултата показано је да комбинације могу да дају тачност у опсегу од 30% до 80%, што зависи од броја скупова података у репозиторијуму за одређену категорију. За мањи број скупова података, као што су доменски подаци од 2 скупа до 10 скупова података, показано је да комбинације дају тачност у опсегу од 0% до 90%. Добијени резултати представљају максималне вредности тачности за добијене комбинације, при чему се комбинације не понављају за различити број атрибута и категорије, не представљају решење које је добро за различите аспекте од интереса и при томе не постиже боље резултате за ред величине, па због тога нису даље анализирани. На основу поставке експеримента, трећа фаза представља претрагу случајног простора за сва постојећа решења која задовољавају дефинисане критичне захтеве. Резултати такве претраге су представљени на сликама 27-29.



Слика 27: Претрага простора методом исцрпног претраживања за мета податке који испуњавају дефинисане критичне захтеве и за различити тип података. Број атрибута је вариран до *5* атрибута за проналажење минималног скупа. Приказани су резултати за *1000* итерација за сваку варијацију до *5* атрибута.



Слика 28: Претрага простора методом исцрпног претраживања за мета податке који испуњавају дефинисане критичне захтеве и за различит локалитет аномалија. Број атрибута је вариран до *5* атрибута за проналажење минималног скупа. Приказани су резултати за *1000* итерација за сваку варијацију до *5* атрибута.



Слика 29: Претрага простора методом исцрпног претраживања за мета податке који испуњавају дефинисане критичне захтеве и за различити домен података. Број атрибута је вариран до *5* атрибута за проналажење минималног скупа. Приказани су резултати за *1000* итерација за сваку варијацију до *5* атрибута.

На основу добијених резултата показано је да комбинације могу да дају тачност у опсегу од 25% до 80%, што зависи од броја скупова података у репозиторијуму за одређену категорију. За мањи број скупова података, као што су доменски подаци од 2 скупа до 10 скупова података, показано је да комбинације дају тачност у опсегу од 0% до 90%. Добијени резултати представљају максималне вредности тачности за добијене комбинације, при чему се комбинације не понављају за различити број атрибута и категорије, не представљају решење које је добро за различите аспекте од интереса и при томе не постижу боље резултате за ред величине, па због тога нису даље анализирани. На основу поставке експеримента, четврта фаза представља претрагу случајног простора за предложено решење. Минимални скуп мета података предложеног решења су мета подаци без мета податка који описују домен података. У табелама 22-25 су представљени резултати добијени у четвртој фази где се минимални скуп предложеног решења и постојећа решења пореде кроз различите аспекте од интереса и за различите оптимизационе метрике. Табела је подељена по колонама на различите типове података, локалитет аномалија, као и домен података. По редовима су представљена поређена решења.

Табела 22: Поређење постојећих и предложеног решења кроз аспекте од интереса за *f1* оптимизациону метрику са редукованим скупом мета података.

f1 оптимизациона метрика		Тип података		Локалитет аномалија			Домен података						
Мета подаци	Цео Репозиторијум	Вишедимензионални подаци	Временски подаци	Локалне аномалије	Глобалне аномалије	Микро-кластери	Производња	Транспорт	Економија	Медицина	Обрада текста	Софтвер	Графови
Прости мета подаци	0,74	0,55	0,74	0,75	0,83	0,66	0,66	0,87	1,00	1,00	0,50	0,83	0,90
Статистички мета подаци	0,68	0,55	0,90	0,68	0,83	0,66	0,66	0,62	1,00	1,00	0,50	0,83	0,90
Мета подаци засновани на теорији информација	0,69	0,55	0,70	0,75	0,66	0,66	0,66	0,75	0,83	1,00	0,50	0,76	0,90
Мета подаци засновани на доменском знању	0,87	0,55	0,90	0,66	0,83	0,66	0,66	0,62	1,00	1,00	0,50	0,93	0,90
Комплетан скуп мета података	0,73	0,55	0,90	0,68	0,83	0,33	0,33	0,87	1,00	1,00	0,50	0,66	0,80

Табела 23: Поређење постојећих и предложеног решења кроз аспекте од интереса за *accuracy* оптимизациону метрику са редукованим скупом мета података.

accuracy оптимизациона метрика		Тип података		Локалитет аномалија			Домен података						
Мета подаци	Цео Репозиторијум	Вишедимензионални подаци	Временски подаци	Локалне аномалије	Глобалне аномалије	Микро-кластери	Производња	Транспорт	Економија	Медицина	Обрада текста	Софтвер	Графови
Прости мета подаци	0,47	0,55	0,63	0,37	0,77	0,66	0,33	0,62	0,66	0,50	0,50	0,76	0,80
Статистички мета подаци	0,53	0,55	0,90	0,51	0,83	0,66	0,33	0,62	0,66	0,50	0,50	0,93	0,90
Мета подаци засновани на теорији информација	0,52	0,55	0,63	0,55	0,72	0,33	0,33	0,75	0,66	0,50	0,50	0,60	0,80
Мета подаци засновани на доменском знању	0,84	0,55	0,90	0,86	0,83	0,66	0,33	0,62	0,50	0,50	0,50	0,93	0,80
Комплетан скуп мета података	0,52	0,44	0,90	0,48	0,83	0,66	0,33	0,62	0,66	0,50	0,50	0,40	0,90

Табела 24: Поређење постојећих и предложеног решења кроз аспекте од интереса за *precision* оптимизациону метрику са редукованим скупом мета података.

precision оптимизациона метрика		Тип података		Локалитет аномалија			Домен података						
Мета подаци	Цео Репозиторијум	Вишедимензионални подаци	Временски подаци	Локалне аномалије	Глобалне аномалије	Микро-кластери	Производња	Транспорт	Економија	Медицина	Обрада текста	Софтвер	Графови
Прости мета подаци	0,69	0,55	0,65	0,51	0,50	0,33	0,66	0,87	1,00	0,00	0,50	0,80	0,70
Статистички мета подаци	0,58	0,22	0,61	0,66	0,50	0,33	0,66	0,75	1,00	0,00	0,50	0,73	0,70
Мета подаци засновани на теорији информација	0,61	0,44	0,67	0,75	0,50	0,33	0,66	0,75	0,66	0,00	0,50	0,66	0,70
Мета подаци засновани на доменском знању	0,56	0,22	0,63	0,86	0,50	0,33	0,66	0,62	1,00	0,00	0,50	0,80	0,70
Комплетан скуп мета података	0,58	0,33	0,63	0,66	0,50	0,33	0,33	0,75	1,00	0,00	0,50	0,66	0,70

Табела 25: Поређење постојећих и предложеног решења кроз аспекте од интереса за *recall* оптимизациону метрику са редукованим скупом мета података.

recall оптимизациона метрика		Тип података		Локалитет аномалија			Домен података						
Мета подаци	Цео Репозиторијум	Вишедимензионални подаци	Временски подаци	Локалне аномалије	Глобалне аномалије	Микро-кластери	Производња	Транспорт	Економија	Медицина	Обрада текста	Софтвер	Графови
Прости мета подаци	0,07	0,11	0,43	0,08	0,00	0,00	0,66	0,00	0,00	0,00	0,00	0,06	0,00
Статистички мета подаци	0,40	0,11	0,43	0,06	0,00	0,00	0,66	0,00	0,00	0,00	0,00	0,30	0,00
Мета подаци засновани на теорији информација	0,04	0,11	0,03	0,06	0,00	0,00	0,66	0,00	0,00	0,00	0,00	0,03	0,00
Мета подаци засновани на доменском знању	0,04	0,11	0,03	0,06	0,00	0,00	0,66	0,00	0,00	0,00	0,00	0,03	0,00
Комплетан скуп мета података	0,06	0,11	0,03	0,06	0,00	0,00	0,66	0,00	0,00	0,00	0,00	0,03	0,00

На основу добијених резултата за *f1* оптимизациону метрику предложено решење постиже тачност од 87% док комплетан скуп мета података постиже тачност од 73% за цео репозиторијум података од 63 скупа података. Такође, предложено решење постиже исте или боље резултате у поређењу са постојећим решењима за 55 временских скупова података и 9 вишедимензионалних скупова података. Слично томе, предложено решење постиже исте или боље резултате за све локалитете аномалија осим локалних аномалија где постиже за 9% лошије перформансе у односу на мета податке засноване на теорији информација. За све домене података осим транспорта, предложено решење постиже исте или боље резултате, где комплетан скуп мета података постиже за 25% боље резултате. За различите домене података, предложено решење постиже исте или боље перформансе у односу на постојећа решења осим транспорта и софтверских записа. За *accuracy* оптимизациону метрику, предложено решење постиже исте или боље перформансе за цео репозиторијум података, док у категоријама транспорта, економије и графова података постиже лошије перформансе. За *precision* оптимизациону метрику, предложено решење постиже лошије перформансе за 1,03% за цео репозиторијум података. За *recall* оптимизациону метрику предложено решење постиже лошије перформансе за 1,5%. Примећује се да су перформансе предложеног решења и одређених постојећих решења лошије за ред величина због одабира алгоритма који не даје добре перформансе за ту оптимизациону метрику.

Показано је да предложено решење даје добре резултате за различите аспекте од интереса. Овако дефинисан експеримент даје добар преглед перформанси постојећих решења и предложеног решење, пореди са више приступа и комбинација што потврђује дефинисане полазне хипотезе за овај експеримент. Прво је поређено предложено решење са постојећим решењима где се показало да резултати предложеног решења могу да буду исти или бољи. Након тога се извршила претрага простора како би се дошло до бољих резултата. Међутим, претрага простора није дала добре резултате. Након тога се прешло на претрагу простора само постојећих мета података који испуњавају дефинисане критичне захтеве. Након њихове исцрпне евалуације, није се дошло до мета података који представљају минимални скуп мета података који постиже боље резултате за различите аспекте од интереса. Даље, извршена је претрага простора предложеног решења где су резултати показали да се предлог постојећег решења може редуковати искључивањем мета податка везаног за домен података. Евалуацијом таквог предложеног решења, дошло се до резултата који у већини случајева дају исте или боље перформансе. На основу тога се закључује да предложено решење даје позитивне резултат имајући у виду дефинисане полазне хипотезе везане за овај експеримент.

Приликом парцијалне претраге простора користила се метода случајног одабира комбинација за одређену величину минималног скупа мета података. Овакав приступ не може да се потврди да не постоји скуп мета података који је мањи од предложеног скупа и да при томе даје боље резултате за наведене аспекте од интереса. Начин тражење минималног скупа мета података може да се прошири другим методама како би се потврдили добијени резултати. Такође, приликом одабира алгоритама за детекцију аномалија није узет скуп свих постојећих алгоритама који могу да се користе.

5.3.2 Резултати експеримента 2 - Евалуација функција за мерење удаљености

Други експеримент се односи на дефинисање скупа функција за мерење удаљености које ће се користити за поређење предложеног и постојећих решења. Циљ експеримента је да се провери да ли постоји тип функција за мерење удаљености који ће дати боље резултате за

одређени тип мета података. Полазну хипотезу коју овај експеримент треба да валидира је да за предложени скуп мета података је могуће одредити тип функција за мерење удаљености између скупова података који у већини случајева даје боље резултате од осталих типова функција које се најчешће користе за мерење удаљености између скупова података.

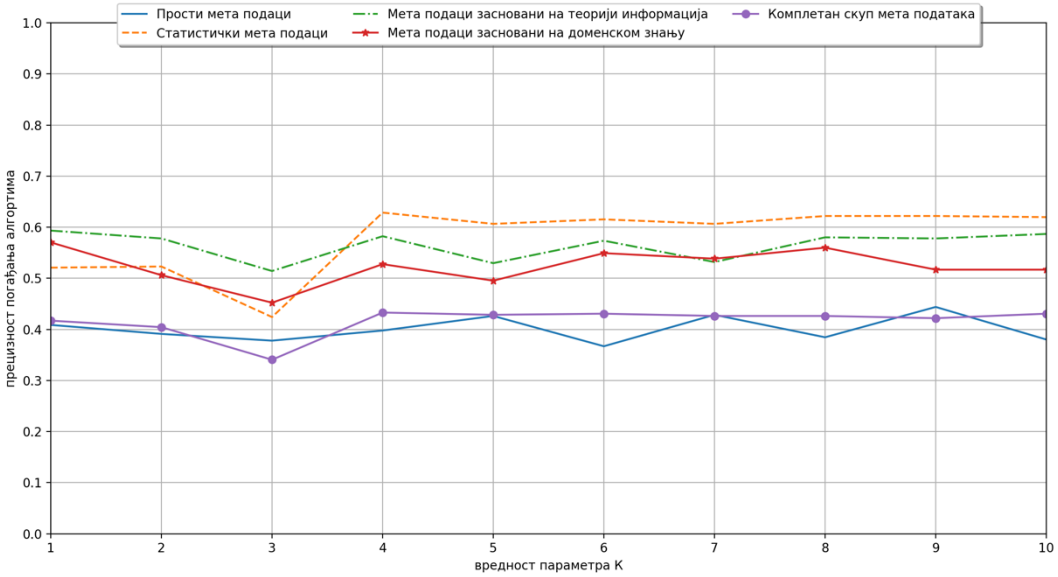
У табели 26 су представљени резултати добијени извршавањем овог експеримента. Табела је подељена по колонама на различите типове података, локалитет аномалија, као и домен података. По редовима су представљена поређена решења. Поређене функције за мерење удаљености су представљене у поставци експеримената. Поред стандардних функција за мерење удаљености, у овом експерименту су коришћени исти алгоритми за детекцију аномалија. Анализа резултате се односи искључиво за предложено решење како би се проверила једна од полазних хипотеза. Због тога ће се аналзирати добијени резултати искључиво за мета податке засноване на доменском знању. Преглед добијених резултата за постојеће решења може да се користи у даљим истраживањима у овој области.

Табела 26: Резултати експеримената за различите функције за мерење удаљености. Експеримент се извршава над целим репозиторијумом података за *f1* оптимизациону метрику.

Мета подаци	Еуклидска удаљеност	*Manhattan* удаљеност	Гаусова дистрибуција	Линеарна регресија	Робусна анализа главних компонената	Кластеризација методом K-средњих вредности	Аутоенкодер
Прости мета подаци	0,72	0,70	0,00	0,74	0,00	0,44	0,30
Статистички мета подаци	0,66	0,66	0,52	0,53	0,40	0,68	0,32
Мета подаци засновани на теорији информација	0,58	0,56	0,52	0,58	0,69	0,55	0,66
Мета подаци засновани на доменском знању	0,66	0,67	0,00	0,40	0,87	0,04	0,84
Комплетан скуп мета података	0,00	0,00	0,00	0,55	0,67	0,73	0,55

На основу добијених резултата, предложени скуп мета података коришћењем аутоенкодера и робусне анализе главних компоненти као функција за мерење удаљености постиже тачност од 84% и 87%. Коришћењем осталих функција за мерење удаљености, предложено решење постиже тачност до 67%. Закључује се да постоји веза између предложених мета података и функције које је погодно користити у тим случајевима. Показано је да функције за мерење удаљености на основу трансформација мета података у суб-димензионални простор и затим рачунања грешке приликом реконструкције дају добре резултате у већини случајева за предложено решење. Овим резултатима може да се потврди полазна хипотеза која дефинише оптимални тип функција за мерење удаљености за предложено решење. Такође, може да се закључи да постојећа решења која се користе за мерење удаљености а заснована су на простим функцијама не могу да добро израчунају сличност између скупова података.

У експериментима је приликом мерења удаљености између мета података узет број k као број најближих скупова података који се узимају у разматрање приликом одабира алгоритма. На слици 30 су представљени резултати одабира алгоритма за различите типове мета података и вредности k.



Слика 30: Одабир алгоритма коришћењем различитих типова мета података за различите вредности *k,* при чему је вредност у интервалу од 1 до 10.

На основу добијених резултата, показано је да у интервалу за k од 1 до 10 тачност различитих типова мета података над целим репозиторијумом података за одређивање оптималног алгоритма је у опсегу од 35% до 65%. На основу тога се закључује да повећање броја k не утиче на перформансе за различите типове мета података. Због тога је у претходним експериментима за вредност броја k узета вредност 1 како би се смањило време извршавања експеримената. Приликом одабира функција за мерење удаљености није узет скуп свих постојећих функција које се користе за мерење удаљености. Такав одабир не може да потврди да не постоји функција за мерење удаљености која даје боље резултате за предложено решење или постојећа решења. Опасност по резултате експеримената представља начин одабира функција за мерење удаљености између скупова података коришћењем мета података.

5.3.3 Резултати експеримента 3 - Комплексност предложених мета података

Циљ експеримента је да представи комплексност предложеног решења како би се показало да ли је могуће израчунати мета податке коришћењем мале количине ресурса и са малом комплексношћу. Полазна хипотеза коју овај експеримент треба да валидира је да предложени скуп мета података може да задовољи наведене критичне захтеве. Најбољи случај представља ситуацију када су аномалије обележене у подацима, познате димензије података, број аномалија у подацима, као и тип и домен података. Просечан случај представља ситуацију када су познате димензије података и број података, као и тип и домен података. Најгори случај представља познавање искључиво димензија података и присуство доменског експерта.

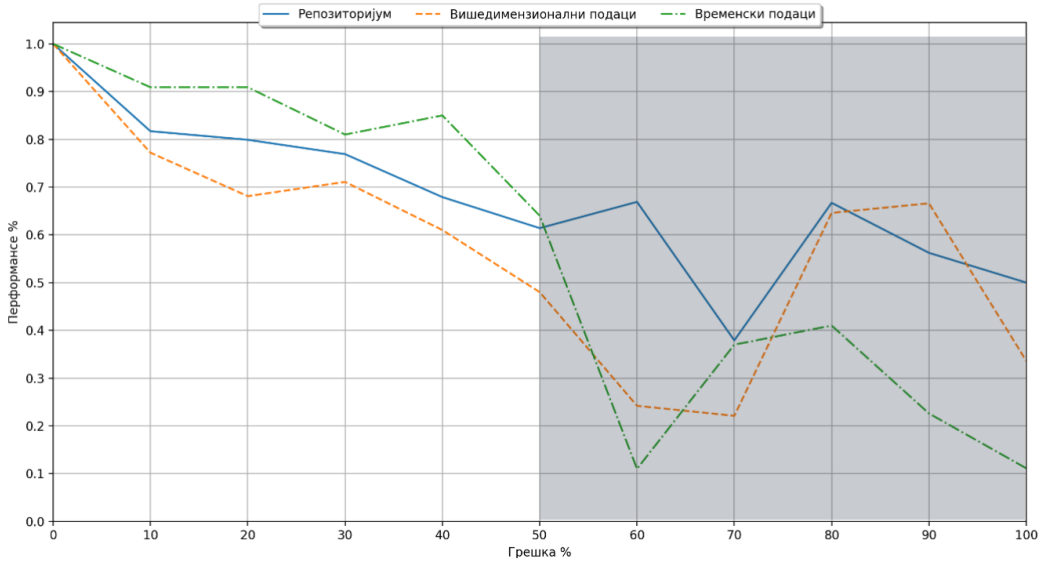
Први мета податак се односи на димензионални простор аномалија. Он се израчунава на основу броја атрибута у скупу података. У сва три случаја димензије података су познате па је време израчунавања овог мета податка увек константно $O(1)$. Следећи мета податак представља локалитет аномалија који захтева познавање карактеристика аномалија у подацима, тј. да су аномалије у подацима обележене. У најбољем случају, ако су аномалије обележене, проласком кроз аномалије могуће је одредити њихов тип и на тај начин одредити локалитет аномалија. У том случају, ако је a број аномалија у подацима и n број инстанци, онда је комплексност $O(an), a \ll n$. Апроксимацијом да је a мали број који тежи *0* у односу на n долази се до комплексности $O(n)$. У осталим случајевима, прво је потребно одредити аномалије па затим одредити њихов тип, где је комплексност $(an + n^2), a \ll n$. Апроксимацијом да је a мали број који тежи *0* у односу на n долази се до комплексности $O(n^2)$. Број аномалија је мета податак који може да се израчуна у константном времену ако је познат. У супротном, потребно је итерирати кроз податке и на основу одређене граничне вредности израчунати број аномалија у подацима у линеарном времену у односу на број података. На крају, мета подаци као што су тип и домен података се могу одредити у константом времену на основу доступних података или знања доменског експерта. У табели 27 је представљена анализа комплексности решења коришћењем велике *O* нотације. За сваки од мета података је представљена комплексност имплементације.

Табела 27: Комплексност предложених мета података коришћењем нотација за мерење временске комплексности функција, где n представља број података у једном скупу.

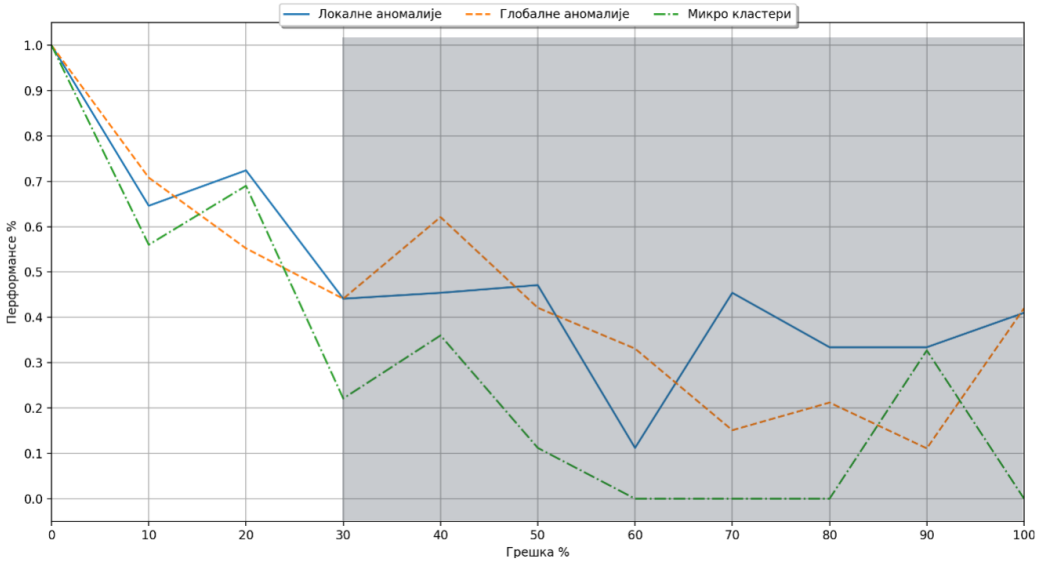
Предложени мета подаци	Најбољи случај	Просечан случај	Најгори случај
Димензионални простор аномалија	*O(1)*	*O(1)*	*O(1)*
Локалитет аномалија	*O(n)*	*O(n^2)*	*O(n^2)*
Број аномалија	*O(1)*	*O(n)*	*O(n)*
Тип података	*O(1)*	*O(1)*	*O(1)*
Домен података	*O(1)*	*O(1)*	*O(1)*

5.3.4 Резултати експеримента 4 - Процена предложених мета података

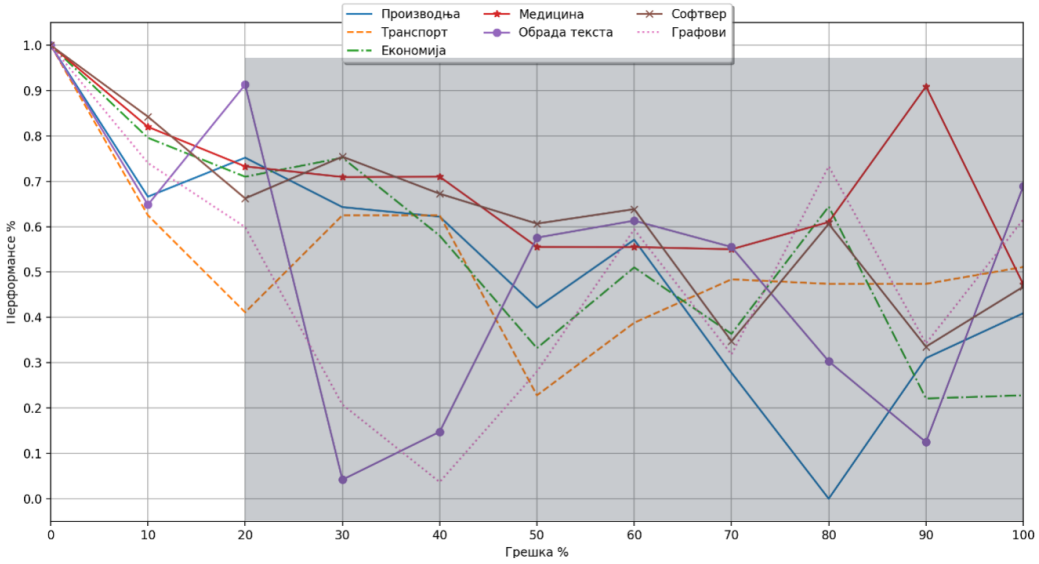
Када подаци нису доступни или доменски експерт није присутан, неопходно је на одређени начин проценити мета податке како би систем могао да врши закључивање. Циљ експеримента је да валидира предложено решење приликом процене мета података. Полазне хипотезе које овај експеримент треба да валидира су да предложени скуп мета података може да задовољи наведене критичне захтеве и да је предложени скуп мета података могуће проценити искључиво на основу доменског знања и без присуства података. Мета подаци који означавају локалитет аномалија и број аномалија у подацима имају већу комплексност у претходном експерименту и могу да се погреше приликом процене вредности. Вредности мета података су стандардизоване и на такве вредности је итеративно додавана грешка. Слике 31-33 представљају функцију грешке за локалитет аномалија кроз аспекте од интереса, док слике 34-36 представљају функцију грешке за број аномалија кроз аспекте од интереса. Аспекти од интереса су различити типови података, различити локалитети аномалија, као и различите области података.



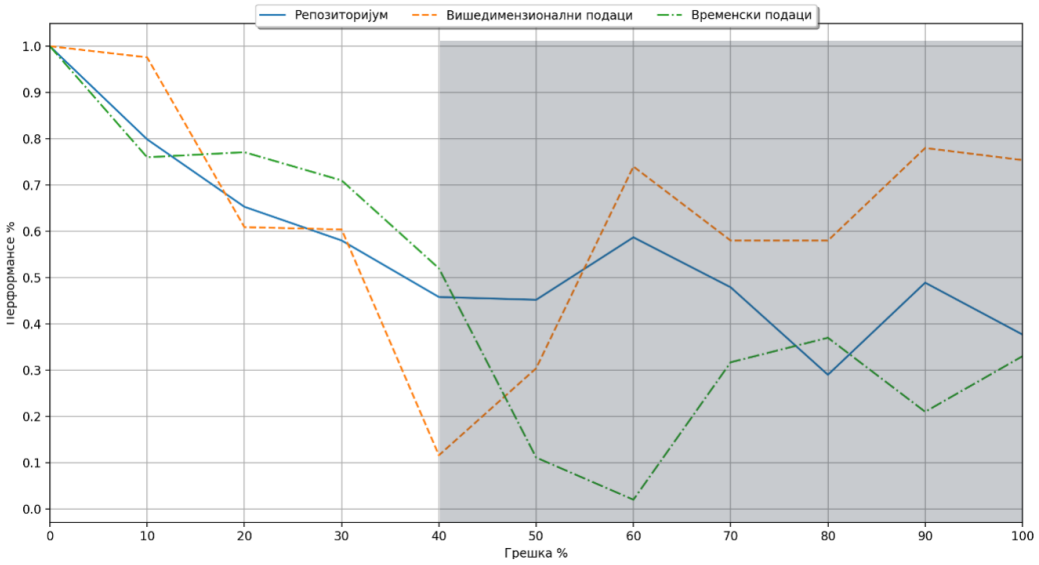
Слика 31: Грешка у перформансама приликом увођења грешке у процени локалитета аномалија за различите локалитете података. Функција грешке је нерастућа до грешке од 50%, што је обележено као бела зона.



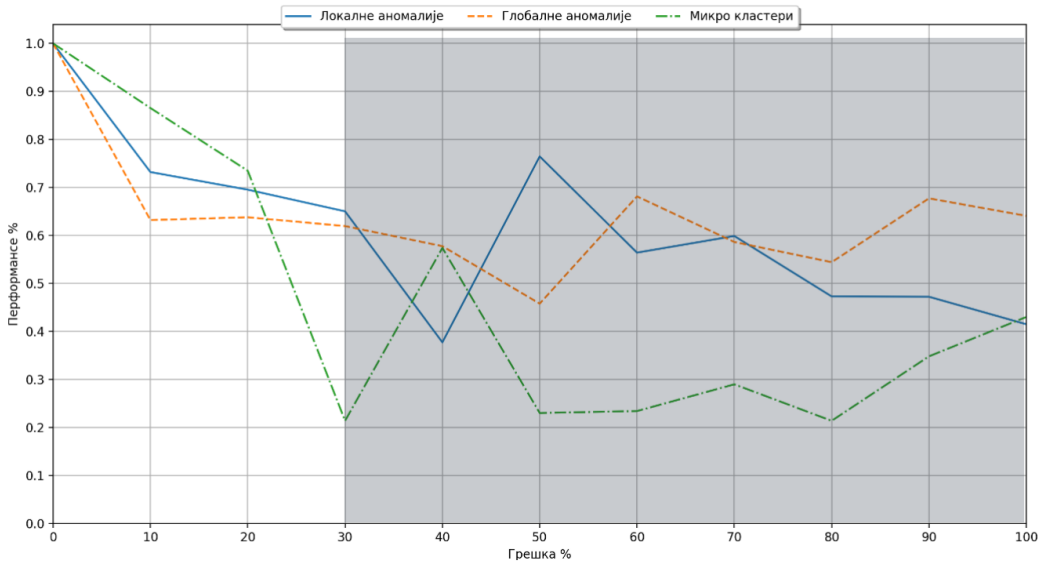
Слика 32: Грешка у перформансама приликом увођења грешке у процени локалитета аномалија за различити локалитет аномалија. Функција грешке је нерастућа до грешке од 30%, што је обележено као бела зона.



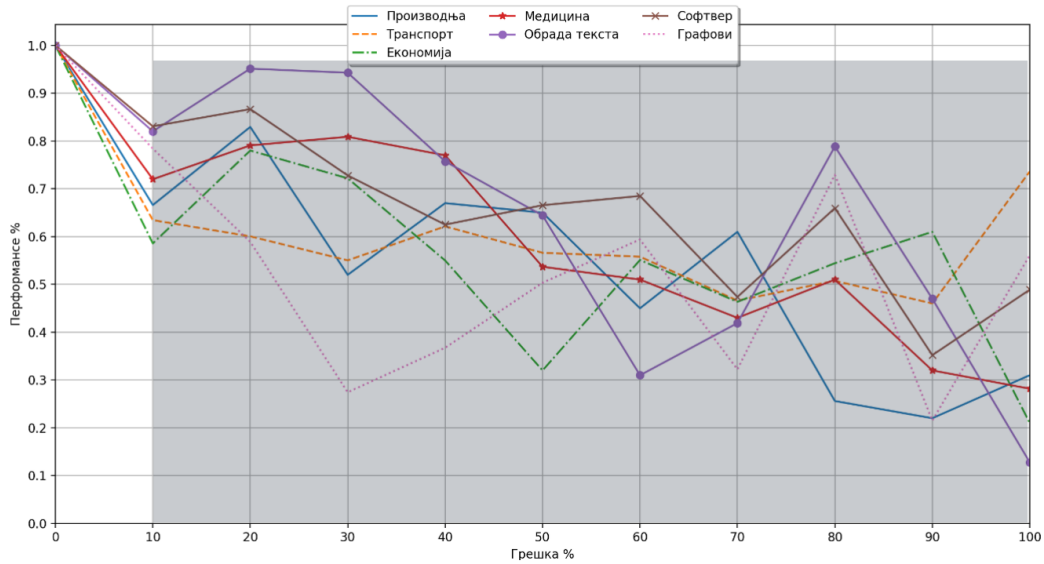
Слика 33: Грешка у перформансама приликом увођења грешке у процени локалитета аномалија за различите типове података. Функција грешке је нерастућа до грешке од 20%, што је обележено као бела зона. Показано је да перформансе не зависе од грешке већ се врши насумични одабир, ако је грешка преко 20%, због малих подскупова података у наведеним категоријама.



Слика 34: Грешка у перформансама приликом увођења грешке у процени броја аномалија у подацима за различите типове података. Функција грешке је нерастућа до грешке од 40%, што је обележено као бела зона.



Слика 35: Грешка у перформансама приликом увођења грешке у процени броја аномалија у подацима за различити локалитет аномалија. Функција грешке је нерастућа до грешке од 30%, што је обележено као бела зона.



Слика 36: Грешка у перформансама приликом увођења грешке у процени броја аномалија за различите домене података. Функција грешке је нерастућа до грешке од 10%, што је обележено као бела зона. Показано је да перформансе не зависе од грешке већ се врши насумични одабир, ако је грешка преко 10%, због малих подскупова података у наведеним категоријама.

На основу добијених резултата, показано је да је функција грешке за цео репозиторијум података од 63 скупа података нерастућа до одређене границе која зависи од броја скупова података у одређеној категорији. За цео репозиторијум података и различите типове података, функција грешке је нерастућа до грешке од 40% до 50%, што је обележено као бела зона. Након те вредности грешке сматра се да се одабир алгоритама врши насумично, што је обележено

као сива зона. За различите локалитете аномалија, функција грешке је нерастућа до грешке од 30%. За различите домене података, где категорије података садрже мали број скупова података, функција грешке је нерастућа до грешке од 10% до 20%. Код одређених скупова података као што су временски подаци, показано је да грешка може да има већи утицај у односу на остале типове података. Такође, грешка у процени за мањи број скупова података може да има већи утицај на резултате одабира алгоритма. Показано је да приликом већих грешака перформансе одабираних алгоритама могу доста да се разликују што доводи до закључка да случајним одабиром алгоритма могу да се добију боље перформансе. На основу свега показаног, закључује се да су дефинисане хипотезе за овај експеримент испуњене под одређеним условима. У зависности од мета податка који се процењују, до одређене границе грешка утиче предиктивно на перформансе одабраног алгоритма. Када се пређе та граница, грешка више није предиктивна за перформансе одабраног алгоритма. Даљом анализом би могло да се утврди колика би била кумулативна грешка код више различитих мета података.

5.3.5 Резултати експеримента 5 - Евалуација пројектоване компоненте у различитим окружењима

Евалуација пројектоване компоненте се врши имплементацијом и извршавањем компоненте на различитим архитектурама. Циљ експеримента је да евалуира имплементирану компоненту за различите архитектуре и локације извршавања кроз аспекте од интереса. Аспекти од интереса су потрошња електричне енергије и време извршавања за фазе тренирања и тестирања. На тај начин је могуће закључити које архитектуре су погодне за коришћење у различитим окружењима.

Полазну хипотезу коју овај експеримент треба да валидира је да предложени скуп мета података може да задовољи наведене критичне захтеве који се односе на скалабилност. Претходни експерименти који пореде предложено и постојеће решење се користе као оптерећење које ће се створити на одређеној архитектури. Како време извршавања може да зависи од имплементације алгоритама и коришћених скупова података, резултати су нормализовани и тако представљени. Циљ овакве репрезентације је приказивање односа резултата за различите архитектуре. Мерење електричне енергије потрошене није тривијалан задатак и није могуће прецизно урадити у окружењу које се користи за симулацију имплементације на различитим локацијама. Због тога је за потрошњу електричне енергије узет број транзистора као фактор који утиче на потрошњу електричне енергије. У табели 28 је представљена подела коришћених архитектура за различите фазе рада компоненте, као и за различите локације извршавања. За хибридно решење, прва наведена архитектура се користи за решење у облаку док се друга користи за решење на крајњем уређају.

Табела 28: Поставка различитих архитектура у зависности од фазе рада компоненте и различитих локација извршавања.

Архитектуре	Локације извршавања		
	Решење у облаку	Решење на крајњем уређају	Хибридно решење
Тренирање	*GPU*	*ASIC*	*GPU+ASIC*
Закључивање	*CPU*	*ASIC*	*CPU+ASIC*

У табелама 29 и 30 су представљени нормализовани резултати извршавања на различитим архитектурама и за различите фазе рада система. Нормализација се извршила скалирањем добијених резултата на исту средњу вредност како би се подаци поредили

независно од количине података и алгоритма. Аспекти од интереса који су представљени су брзина извршавања, број транзистора за сваку поређену архитектуру, као и брзина по транзистору како би се показале перформансе у односу на потрошњу електричне енергије. Карактеристике коришћених архитектура су дате у табели у поглављу 2.

Табела 29: Перформансе компоненте приликом тренирања за различите архитектуре извршавања. Дате вредности представљају нормализоване вредности како би се поредиле различите архитектуре.

Тренирање	Архитектуре		
	CPU	*GPU*	*ASIC*
Брзина извршавања	7,55	4,91	20,12
Број транзистора	7,20	7,10	2,10
Брзина по транзистору	0,95	1,44	0,10

Табела 30: Перформансе компоненте приликом закључивања за различите архитектуре извршавања. Дате вредности представљају нормализоване вредности како би се поредиле различите архитектуре.

Закључивање	Архитектуре		
	CPU	*GPU*	*ASIC*
Брзина извршавања	0,10	0,04	0,10
Број транзистора	7,20	7,10	2,10
Брзина по транзистору	0,01	0,01	0,05

На основу добијених резултата, брзина тренирања компоненте на *GPU* архитектури даје боље резултате у односу на *CPU* и *ASIC* архитектуре за 2,64 и 15,21 нормализованих јединица. Брзина тренирања по транзистору на *ASIC* архитектури даје боље резултате у односу на *CPU* и *GPU* архитектуре за 0,85 и 1,34 нормализованих јединица. Брзина закључивања компоненте на *GPU* архитектури даје боље резултате у односу на *CPU* и *ASIC* архитектуре за 0,06 нормализованих јединица. Брзина закључивања по транзистору на *ASIC* архитектури даје боље резултате у односу на *CPU* и *GPU* архитектуре за 0,04 нормализованих јединица. Показано је да архитектуре које садрже мањи број транзистора и раде на мањој фреквенцији могу да се користе у локацијама извршавања која немају пуно ресурса и захтевају да се извршавање система ради на том уређају. Са друге стране, решења која се користе у облаку показују боље перформансе са аспекта времена извршавања. Такође, показано је да коришћена *ASIC* архитектура у овом случају даје добре перформансе и у облаку, тако да може да се користи у више окружења. Специфично за коришћену *ASIC* архитектуру коришћену у раду је то што је оптимизована за алгоритме машинског учења па због тога троши мало ресурса при чему постиже добре перформансе.

На основу резултата може да се закључи да су комбинована решења добар приступ за будући развој аутоматизованих система за машинско учење, где се закључивање може вршити на крајњим уређајима, док се припрема података и тренирање извршава у облаку.

5.3.6 Провера хипотеза

У овом поглављу су дати одговори на хипотезе постављене у овом раду. Након детаљне анализе постојећих решења у овој области као и представљања компонената које су потребне за њихову имплементацију детаљно је представљено предложено решење. Затим је представљена имплементација компоненте, и дискутоване су различите локације извршавања система. Након тога је дата поставка експеримената и опис експеримената. На крају су представљени и дискутовани добијени резултати где су анализиране постављене полазне хипотезе. На основу свега тога, у овом поглављу је могуће дати одговоре на дефинисане полазне хипотезе као и опасности по експерименте који могу да утичу на добијене резултате.

Прва хипотеза се односи на могућност дефинисања скуп мета података заснованом на доменском знању за карактеризацију аномалија у подацима. У поглављу 3 је показано да је могуће дефинисати мета податке искључиво на основу доменског знања. Друга хипотеза се односи на начин дефинисања мета података тако да се задовоље критични захтеви. Експерименти су дефинисани тако да проверавају критичне захтеве где је неутралност и повезаност проверена у првом експерименту, скалабилност у трећем и петом експерименту и једноставност у четвртом експерименту. Трећа хипотеза се односи на могућност да тако дефинисани скуп мета података постиже исте или боље резултате у односу на постојећа решења за различите аспекте од интереса. Показано је да предложено решење постиже тачност од 87% и доследно испуњава дефинисане критичне захтеве и у одређеним случајевима даје боље резултате у експериментима са 63 скупа података заступљених у индустрији. Такође, предложено решење постиже исте или боље резултате у поређењу са постојећим решењима за 55 временских скупова података и 9 вишедимензионалних скупова података заступљених у индустрији. Предложено решење постиже исте или боље резултате у поређењу са постојећим решењима за све типове аномалија, осим за локалне аномалије где само мета подаци засновани на теорији информација постижу за 9% боље резултате. Предложено решење постиже исте или боље резултате у поређењу са постојећим решењима за све домене података осим за транспорт, где комплетан скуп мета података постиже за 25% боље резултате. Четврта хипотеза се односи на могућност процене мета података искључиво на основу доменског знања и без присуства података, што је показано у експерименту четири. Последња хипотеза се односи на могућност одређивања тип функција за мерење удаљености између скупова података који у већини случајева даје боље резултате од осталих типова функција које се најчешће користе за мерење удаљености између скупова података. Функције за рачунање удаљености засноване на грешкама реконструкције података постижу значајне перформансе за предложени скуп мета података.

Опасност по дефинисане експерименте представља начин одабира алгоритама и функција за мерење удаљености по степену трансформација над подацима, као и одабир скупова података за евалуацију. Предложено решење би могло да се даље потврди коришћењем додатних скупова података са означеним аномалијама заступљених у индустрији, као и поређењем предложеног решења са другим приступима мета учења који нису засновани на подацима.

Додатни проблеми који могу да утичу на добијене резултате експеримената представљају шум у подацима и поставка експеримената. Шум у подацима може дати неважеће резултате и оповргнути одговоре на истраживачка питања која су дата. Да би се елиминисали такви случајеви, експерименти су постављени тако да смање ову могућност. Прво, скупови података су прикупљени из различитих извора, што смањује могућност стварања шума у подацима које ствара исти извор података. Друго, предложени скуп мета

података се пореди са различитим типовима постојећих мета података коришћењем истих скупова података. Важно је напоменути да се овај рад не бави осталим компонентама аутоматизованих система за машинско учење, као што су припрема података, одабир атрибута, редукција и интеграција података и оптимизација параметара.

6 Закључак

Овај рад предлаже нови скуп мета података заснованих на доменском знању који може да се користи за одабир алгоритма за детекцију аномалија у аутоматизованим системима за машинско учење. Прво су дати типови и домени података који могу да садрже аномалије у подацима, као и типови аномалија који се јављају у подацима. Након тога су представљене групе алгоритама за детекцију аномалија и из сваке групе дат је по један репрезентативни пример који врши мању или већу трансформацију над подацима. Показано је како одабир алгоритама за одређени скуп података и оптимизациону метрику представља комплексан корак који зависи од карактеристика података. Прегледом отворене литературе се показало да аутоматизовани системи за машинско учење решавају проблем одабира алгоритма коришћењем мета учења. Дат је преглед постојећих аутоматизованих система за машинско учење са карактеристикама и компонентама. Након тога, представљено је мета учење, различити типови мета података и функције за мерење удаљености између мета података, које су неопходан модул у аутоматизованим системима за машинско учење.

Након увођења свих потребних појмова за разумевање проблема, дефинисана је потреба за новим скупом мета података, креирани су критични захтеви везани за скалабилност и перформансе мета података који треба да буду испуњени како би могли да се користе у аутоматизованим системима за машинско учење, и дефинисане су полазне хипотезе у раду. Након тога је предложено ново решење за израчунавање мета података на основу доменског знања које се односи на домен аномалија у подацима. Како би се предложено решење упоредило са постојећим решењима, било је потребно имплементирати компоненту аутоматизованог система која се односи на одабир алгоритма. Представљени су имплементациони детаљи и главни модули компоненте. На крају је урађена евалуација предложеног решења и имплементиране компоненте поставком и описом експеримената.

Анализом постојећих мета података и увођењем критичних захтева везаних за скалабилност и перформансе, показано је да постоји потреба за дефинисањем новог скупа мета података у домену мета учења за одабир алгоритама. Резултати експеримената су показали да предложено решење постиже тачност од 87%, док постојећа решења постижу тачност од 73%, и при томе има значајно мању комплексност и доследно испуњава дефинисане критичне захтеве над креираним репозиторијумом који се састоји од 63 скупа података заступљених у индустрији. Предложено решење је поређено са постојећим решењима за различите типове података, различите домене података и различите локалитете аномалија у подацима. Решење постиже исте или боље резултате у поређењу са постојећим решењима за све типове аномалија, осим за локалне аномалије где само мета подаци засновани на теорији информација постижу за 9% боље резултате. Решење постиже исте или боље резултате у поређењу са постојећим решењима за све домене података осим за транспорт, где комплетан скуп мета података постиже за 25% боље резултате. Дефинисани скуп мета података је заснован искључиво на доменском знању и простим мета подацима, што испуњава критичне захтеве за коришћење у аутоматизованим системима за машинско учење. Показано је да доменски експерт може да процени предложене мета податке са не растућом функцијом грешке искључиво у ситуацијама када постоји значајан број скупова података у систему и када је грешка у опсегу од 10% до 50%, у зависности од броја скупова података. Такође, показано је да су функције за мерење удаљености засноване на трансформацијама података у мањи димензионални простор погодне за мерење сличности између скупова података коришћењем предложеног скупа мета података.

На основу претходне анализе, могу да се изведу два скупа закључака. Један је везан за генерални приступ мета учењу и његовим предностима кроз различите аспекте, док је други скуп закључака везан за примену предложеног решења у системима за одабир алгоритама и аутоматизованим системима за машинско учење. У домену мета учења, овај рад предлаже нови приступ за израчунавање мета података на основу доменског знања, где мета подаци описују карактеристике аномалија у подацима. Предложено решење има значајно мању комплексност у односу на постојећа решења, испуњава критичне захтеве за примену у аутоматизованим системима за машинско учење и могуће је проценити мета податке искључиво на основу доменског знања. У домену аутоматизованих система за машинско учење, предложено решење се фокусира на побољшање компоненте за одабир алгоритма, које решава проблем са перформансама и скалабилношћу система, као и коришћења на различитим локацијама извршавања.

Први допринос у овом раду представља евалуацију постојећих скупова мета података који се користе у аутоматизованим системима за машинско учење и предлог њихове класификације са аспеката критичних захтева важних за проблем детекције аномалија и аутоматизованих система за машинско учења. Други допринос представља предлог новог скупа мета података заснованог на доменском знању који пружа боље карактеристике са аспекта критичних захтева важних за проблем одабира модела за детекцију аномалија. Трећи допринос представља креирање и карактеризацију новог репозиторијума који обухвата скупове података релевантне за евалуацију предлога модела у аутоматизованим системима за машинско учење на основу доступних података у отвореној литератури и индустрији допуњеним са мета подацима заснованим на доменском знању. Четврти допринос представља креирање софтверског система за одабир модела за детекцију аномалија на основу мета података погодан за коришћење у аутоматизованим системима за машинско учење, који је проширив новим скуповима мета података, алгоритмима за детекцију аномалија и функцијама за рачунање удаљености између скупова података. Пети допринос представља дизајнирање експеримената погодних за мерење перформанси предлога модела за детекцију аномалија на основу предложеног скупа мета података. Шести допринос представља евалуацију перформанси креираног софтверског система са становишта различитих локација извршавања и архитектура.

Предложено решење и добијени резултати у дисертацији имају практичну примену за одабир алгоритма у аутоматизованим системима за детекцију аномалија. Имплементирана компонента за одабир алгоритама представља целину у аутоматизованом систему за детекцију аномалија коју је могуће интегрисати у постојеће или нове система за детекцију аномалија. Приступ коришћен у раду за одређивање мета података на основу доменског знања може да се примени и на друге области како би се проверило да ли се доменским мета подацима могу карактерисати подаци тако да се врши предлагање алгоритама у аутоматизованим системима за машинско учење. Упоредна анализа алгоритама који се користе за детекцију аномалија, креирани репозиторијум података, предложене функција за мерење удаљености и предложени мета подаци представљају добру основу за даља истраживања у овој области.

Даља истраживања у овој области укључују проширивање скупа мета података за детекцију новим доменским и простим мета подацима, као и тражењем минималног скупа комбинацијом постојећих решења и предложеног решења. Такође, увођењем мета података заснованих на структурним и вредносним ограничењима скупова података може додатно да смањи комплексност израчунавања мета података и представља добар правац за будућа истраживања у овој области.

Литература

[1] A. Oussous, F. Z. Benjelloun, A. A. Lahcen и S. Belfkih, „Big data technologies: A survey,“ *Journal of King Saud University-Computer and Information Sciences,* т. 30, бр. 4, pp. 431-448, 2018.

[2] W. Günther, M. Mehrizi, M. Huysman и F. Feldberg, „Debating big data: A literature review on realizing value from big data,“ *The Journal of Strategic Information Systems,* т. 26, бр. 3, pp. 191-209, 2017.

[3] K. Wagstaff, „Machine learning that matters,“ *arXiv preprint,* бр. 1206.4656, 2012.

[4] M. Carletti, „Explainable machine learning in industry 4.0: Evaluating feature importance in anomaly detection to enable root cause analysis,“ у *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2019.

[5] M. Braun, P. Converse и F. Oswald, „The accuracy of dominance analysis as a metric to assess relative importance: The joint impact of sampling error variance and measurement unreliability,“ *Journal of Applied Psychology,* т. 104, бр. 4, pp. 593-594, 2019.

[6] R. Chalapathy и S. Chawla, „Deep learning for anomaly detection: A survey,“ *arXiv preprint,* бр. 1901.03407, 2019.

[7] M. Goldstein и S. Uchida, „A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data,“ *PloS One,* т. 11, бр. 2, 2016.

[8] J. Liu, J. Guo, P. Orlik, M. Shibata, D. Nakahara, S. Mii и M. Takáč, „Anomaly detection in manufacturing systems using structured neural networks,“ у *13th World Congress on Intelligent Control and Automation (WCICA)*, 2018.

[9] B. Lindemann, F. Fesenmayr, N. Jazdi и M. Weyrich, „Anomaly detection in discrete manufacturing using self-learning approaches,“ *Procedia CIRP,* т. 79, pp. 313-318, 2019.

[10] S. Wolfert, „Big data in smart farming–a review,“ *Agricultural systems,* т. 153, pp. 69-80, 2017.

[11] Y. Wang, K. LeeAnn и A. B. Terry, „Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations,“ *Technological Forecasting and Social Change,* т. 126, pp. 3-13, 2018.

[12] C. Finn, R. Aravind, K. Sham и L. Sergey, „Online meta-learning,“ у *International Conference on Machine Learning (PMLR)*, 2019.

[13] X. Shi, D. W. Yiik, C. Chen и Z.-F. L. Michael, „An automated machine learning (automl) method of risk prediction for decision-making of autonomous vehicles,“ *Transactions on Intelligent Transportation Systems,* т. 22, бр. 11, pp. 7145-7154, 2020.

[14] C. C. Aggarwal, „Outlier analysis,“ у *Data mining*, 2015.

[15] M. Hassan, A. Tizghadam и A. Leon-Garcia, „Spatio-temporal anomaly detection in intelligent transportation systems,“ *Procedia Computer Science,* т. 151, p. 852–857, 2019.

[16] Q. Lu, F. Chen и K. Hancock, „On path anomaly detection in a large transportation network,“ *Computers, Environment and Urban Systems,* т. 33, p. 448–462, 2009.

[17] N. Margaliot, *Systems and methods for derivative fraud detection challenges in mobile device transactions,* Google Patents, 2016.

[18] J. Awoyemi, A. Adetunmbi и S. Oluwadare, „Credit card fraud detection using machine learning techniques: A comparative analysis,“ у *2017 International Conference on Computing Networking and Informatics (ICCNI)*, 2017.

[19] J. Knights, Z. Heidary и J. Cochran, „Detection of Behavioral Anomalies in Medication Adherence Patterns Among Patients With Serious Mental Illness Engaged With a Digital Medicine System,“ *JMIR Mental Health,* т. 7, 2020.

[20] X. Wu и X. Zhu, „Mining with noise knowledge: error-aware data mining,“ *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans,* т. 38, p. 917–932, 2008.

[21] Y. Liu, W. Jun-Ming, A. Maxim и S. Si-Qi, „Multi-layer feature selection incorporating weighted score-based expert knowledge toward modeling materials with targeted properties,“ *Advanced Theory and Simulations,* т. 3, бр. 2, 2020.

[22] A. T. Fisch, E. Idris и F. Paul, „Subset multivariate collective and point anomaly detection,“ *Journal of Computational and Graphical Statistics,* 2021.

[23] J.-W. Kang, P. Hyeon-Jeong, R. Jong-Sum и J. Hyun-Kyo, „A strategy-selecting hybrid optimization algorithm to overcome the problems of the no free lunch theorem,“ *Transactions on Magnetics,* т. 54, бр. 3, pp. 1-4, 2018.

[24] E. M. Knorr и R. T. Ng, „Finding intensional knowledge of distance-based outliers,“ *Vldb,* т. 99, pp. 211-222, 1999.

[25] S. Zhao, L. Wenfeng и C. Jingjing, „A user-adaptive algorithm for activity recognition based on k-means clustering, local outlier factor, and multivariate gaussian distribution,“ *Sensors,* т. 16, бр. 8, 2018.

[26] M. Salehi и L. Rashidi, „A Survey on Anomaly detection in Evolving Data: [with Application to Forest Fire Risk Prediction,“ *ACM SIGKDD Explorations Newsletter,* т. 20, p. 13–23, 2018.

[27] W. Sun, Y. Gang, L. Jialin и Z. Dianfa, „Randomized subspace-based robust principal component analysis for hyperspectral anomaly detection," *Journal of Applied Remote Sensing,* т. 12, бр. 1, 2018.

[28] Z. Liangwei, J. Lin и R. Karim, „Adaptive kernel density-based anomaly detection for nonlinear systems," *Knowledge-Based Systems,* т. 139, бр. 1, pp. 55-63, 2018.

[29] C. Raghavendra, A. Menon и S. Chawla, „Anomaly detection using one-class neural networks," *arXiv preprint,* т. 1802, бр. 06360, 2018.

[30] S. Naseer, „Enhanced network anomaly detection based on deep neural networks," *IEEE Access,* т. 6, pp. 48231-48246, 2018.

[31] C. Z. Chai, K. Yeo, B. S. Lee и C. T. Lau, „Autoencoder-based network anomaly detection," у *Wireless Telecommunications Symposium (WTS)*, 2018.

[32] A. D и A. Bouchachia, „Detection of abnormal behaviour for dementia sufferers using Convolutional Neural Networks," *Artificial intelligence in medicine,* т. 94, pp. 88-95, 2019.

[33] X. He, K. Zhao и X. Chu, „AutoML: A Survey of the State-of-the-Art," *Knowledge-Based Systems,* т. 212, p. 106622, 2021.

[34] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari и J. Saeed, „A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction," *Journal of Applied Science and Technology Trends,* т. 1, бр. 2, 2020.

[35] S. W. Yahaya, A. Lotfi и M. Mahmud, „A consensus novelty detection ensemble approach for anomaly detection in activities of daily living," *Applied Soft Computing,* т. 83, p. 105613, 2019.

[36] Z. Yan и J. Zhang, „A machine learning model for detecting invasive ductal carcinoma with Google Cloud AutoML Vision," *Computers in Biology and Medicine,* т. 122, 2020.

[37] C. Arunima, A. Issak, K. Kate, Y. Katsis, A. Valente, D. Wang и A. Evfimievski, „AutoText: An End-to-End AutoAI Framework for Text," у *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[38] G. Xavier и Y. Bengio, „Understanding the difficulty of training deep feedforward neural networks," у *Artificial intelligence and statistics*, 2010.

[39] K. Milos, D. Bojic, M. Punt и V. Milutinovic, „Survey of deployment locations and underlying hardware architectures for contemporary deep neural networks," *International Journal of Distributed Sensor Networks,* т. 15, бр. 8, 2019.

[40] H. Yutao, X. Ma, X. Fan, J. Liu и W. Gong, „When deep learning meets edge computing," у *Иnternational conference on network protocols (ICNP)*, 2017.

[41] H. Xin, K. Zhao и X. Chu, „AutoML: A Survey of the State-of-the-Art,“ *Knowledge-Based Systems,* т. 212, 2021.

[42] K. Lars, C. Thornton, H. H. Hoos, F. Hutter и K. Leyton-Brown, „Auto-WEKA: Automatic model selection and hyperparameter optimization in WEKA,“ *Automated Machine Learning,* pp. 81-95, 2019.

[43] E. Nick, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li и A. Smola, „Autogluon-tabular: Robust and accurate automl for structured data,“ *arXiv preprint,* 2020.

[44] T. Anh, A. Walters, J. Goodsitt, K. Hines, B. Bruss и R. Farivar, „Towards automated machine learning: Evaluation and comparison of AutoML approaches and tools,“ у *International conference on tools with artificial intelligence (ICTAI)*, 2019.

[45] O. Randal и J. Moore, „TPOT: A tree-based pipeline optimization tool for automating machine learning,“ у *Workshop on automatic machine learning*, 2016.

[46] F. Luís, A. Pilastri, C. M. Martins, P. M. Pires и P. Cortez, „A Comparison of AutoML Tools for Machine Learning, Deep Learning and XGBoost,“ 2021.

[47] A. Rivolli, L. Garcia, C. Soares, J. Vanschoren и A. de Carvalho, „Towards reproducible empirical research in meta-learning,“ *arXiv preprint arXiv:1808.10406,* p. 32–52, 2018.

[48] H. S. Jomaa, L. Schmidt-Thieme и J. Grabocka, „Dataset2vec: Learning dataset meta-features,“ *arXiv preprint arXiv:1905.11063,* 2019.

[49] A. Cohen и N. Nissim, „Trusted detection of ransomware in a private cloud using machine learning methods leveraging meta-features from volatile memory,“ *Expert Systems with Applications,* т. 102, p. 158–178, 2018.

[50] G. J. Aguiar, E. J. Santana, S. M. Mastelini, R. G. Mantovani и S. B. Júnior, „Towards meta-learning for multi-target regression problems,“ у *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, 2019.

[51] R. S. Oyamada, L. Shimomura, S. B. Junior и D. Kaster, „Towards Proximity Graph Auto-configuration: An Approach Based on Meta-learning,“ у *European Conference on Advances in Databases and Information Systems*, 2020.

[52] Y. Zhang, R. Zhu, Z. Chen, J. Gao и D. Xia, „Evaluating and selecting features via information theoretic lower bounds of feature inner correlations for high-dimensional data,“ *European Journal of Operational Research,* 2020.

[53] J. Madrid и H. J. Escalante, „Meta-learning of Text Classification Tasks,“ у *Iberoamerican Congress on Pattern Recognition*, 2019.

[54] B. A. Pimentel и A. de Carvalho, „Unsupervised Meta-Learning for Clustering Algorithm Recommendation,“ у *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019.

[55] J. A. Sáez и E. Corchado, „A Meta-Learning Recommendation System for Characterizing Unsupervised Problems: On Using Quality Indices to Describe Data Conformations,“ *IEEE Access,* т. 7, p. 63247–63263, 2019.

[56] R. Andres, S. Conant, J. Ortiz и H. Terashima, „Selecting meta-heuristics for solving vehicle rout- ing problems with time windows via meta-learning,“ *Expert Systems with Applications,* т. 118, pp. 470-481, 2019.

[57] A. Correia, C. Soares и A. Jorge, „Dataset Morphing to Analyze the Performance of Collaborative Filtering,“ у *International Conference on Discovery Science*, 2019.

[58] Y. Zhang, F. Feng, C. Wang, X. He, M. Wang, Y. Li и Y. Zhang, „How to retrain recommender system? A sequential meta-learning method,“ у *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[59] I. Tanfilev, A. Filchenkov и I. Smetannikov, „Feature selection algorithm ensembling based on meta-learning,“ у *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017.

[60] L. Fan, „Revisit fuzzy neural network: Demystifying batch normalization and ReLU with generalized hamming network,“ *arXiv preprint,* 2017.

[61] X. Hang, W. Zeng, X. Zeng и G. Yen, „An evolutionary algorithm based on Minkowski distance for many-objective optimization,“ *IEEE Transactions on Cybernetics,* т. 49, бр. 11, 2018.

[62] G. Latifa, M. Jazouli, N. Es-Sbai, A. Majda и A. Zarghili, „Comparison between Euclidean and Manhattan distance measure for facial expressions classification,“ у *International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 2017.

[63] P. Kerschke, H. Hoos, F. Neumann и H. Trautmann, „Automated algorithm selection: Survey and perspectives,“ *MIT Press,* т. 27, бр. 1, pp. 3-45, 2019.

[64] M. Kotlar, M. Punt, Z. Radivojević, M. Cvetanović и V. Milutinović, „Novel Meta-Features for Automated Machine Learning Model Selection in Anomaly Detection,“ *IEEE Access,* т. 9, pp. 89675-89687, 2021.

[65] R. Abdulhammed, M. Faezipour, A. Abuzneid и A. AbuMallouh, „Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic,“ *IEEE sensors letters,* т. 3, бр. 1, pp. 1-4, 2018.

[66] A. Taha и A. Hadi, „Anomaly detection methods for categorical data: A review,“ *ACM Computing Surveys (CSUR),* т. 52, p. 1–35, 2019.

[67] M. Yoon, B. Hooi, K. Shin и C. Faloutsos, „Fast and accurate anomaly detection in dynamic graphs with a two-pronged approach,“ у *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[68] G. Campos, A. Zimek, J. Sander, R. Campello, B. Micenková, E. Schubert, I. Assent и M. Houle, „On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study,“ *Data Mining and Knowledge Discovery,* т. 30, p. 891–927, 2016.

[69] H. Bosman, G. Iacca, A. Tejada, H. Wörtche и A. Liotta, „Spatial anomaly detection in sensor networks using neighborhood information,“ *Information Fusion,* т. 33, p. 41–56, 2017.

[70] H. Song, Z. Jiang, A. Men и B. Yang, „A hybrid semi-supervised anomaly detection model for high-dimensional data,“ *Computational intelligence and neuroscience,* т. 2017, 2017.

[71] Y. Djenouri, A. Belhadi, J. C.-W. Lin и A. Cano, „Adapted k-nearest neighbors for detecting anomalies on spatio–temporal traffic flow,“ *IEEE Access,* т. 7, p. 10015–10027, 2019.

[72] M. Landauer, M. Wurzenberger, F. Skopik, G. Settanni и P. Filzmoser, „Dynamic log file analysis: An unsupervised cluster evolution approach for anomaly detection,“ *Computers & security,* т. 79, p. 94–116, 2018.

[73] W.-K. Wong, A. Moore, G. Cooper и M. Wagner, „Rule-based anomaly pattern detection for detecting disease outbreaks,“ у *AAAI/IAAI*, 2002.

[74] V. Vercruyssen, M. Wannes, V. Gust, M. Koen, B. Ruben и D. Jesse, „Semi-supervised anomaly detection with an application to water analytics,“ у *International Conference on Data Mining*, 2018.

[75] M. Kotlar, „Github,“ 2021. [На мрежи]. Available: https://github.com/kotlarmilos/meta-features-anomaly-detection. [Последњи приступ 27 12 2021].

[76] Z. Popovic, Kako napisati i objaviti naucno delo, Institut za fiziku, Beograd, 2004.

[77] D. Dheeru и C. Graff, „UCI machine learning repository,“ University of California, School of Information and Computer Science, 2017.

[78] M. Goldstein, *Unsupervised anomaly detection benchmark,* Harvard Dataverse, 2015.

[79] S. Ahmad, A. Lavin, S. Purdy и Z. Agha, „Unsupervised real-time anomaly detection for streaming data,“ *Neurocomputing,* т. 262, p. 134–147, 2017.

Скраћенице

AutoML - Аутоматизовани систем за машинско учење

CPU - Централна процесорска јединица

GPU - Графичка процесорска јединица

FPGA - Реконфигурабилни хардвер

ASIC - Специфично интегрисано хардвер

ROC - Карактеристике оператера пријемника

AUC - Простор испод криве

DRAM - Динамичка меморија са слободним приступом

SRAM - Статичка меморија са слободним приступом

BRAM - Блок меморија са слободним приступом

Слике

Табеле

Биографија аутора

Милош Котлар је рођен 07.11.1993. године у Београду. Електротехнички факултет уписао је 2012. године и дипломирао на модулу софтверско инжењерство 2016. године са просечном оценом 9,24, одбранивши дипломски рад код проф. др Вељка Милутиновића. Дипломске академске – мастер студије на Електротехничком факултету у Београду уписао је 2016. године и мастерирао на модулу софтверско инжењерство 2017. године са просечном оценом 9,50, одбранивши мастер рад код доц. др Марије Пунт. Докторске студије уписао је 2017. године на модулу рачунарска техника и информатика и положио је све испите са просечном оценом 10,00. Од 2016. године био је запослен као софтверски инжењер у фирми „*ABB*", Цирих, Швајцарска. Од 2019. године запослен је као водећи истраживач у фирми „*TraceLabs*", Љубљана, Словенија.

Милош Котлар је научно-истраживачко искуство започео 2016. године у сарадњи са Електротехничким факултетом у Београду на радовима у области *dataflow* рачунара и машинског учења. Даље искуство је стекао учествовањем на два пројекта као водећи истраживач у фирми „*TraceLabs*", финансираним од стране ЕУ у оквиру *H2020* програма. Пројекти у којима је учествовао припадају области машинског учења, проналажења скривеног знања, графова знања, дистрибуираних система, и архитектуре рачунара.

Први је аутор на три научна рада објављена у часописима са импакт фактором, један у категорији М21 и два у категорији М22, и такође је коаутор у још једном раду објављеном у часопису категорије М22, затим аутор је поглавља у књизи, као и у пет радова објављених на међународним и домаћим конференцијама категорија М33, М34 и М53. Такође, коедитор је две књиге издавача *Springer* и *IGI Global*.

образац изјаве о ауторству

Изјава о ауторству

Име и презиме аутора Милош Котлар

Број индекса 5003/2017

Изјављујем

да је докторска дисертација под насловом

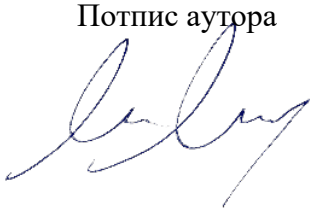
Детекција аномалија коришћењем мета података у аутоматизованим системима за машинско учење

резултат сопственог истраживачког рада;

- да дисертација у целини ни у деловима није била предложена за стицање другедипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину другихлица.

Потпис аутора

У Београду, 01.02.2022.



образац изјаве о истоветности штампане и електронске верзије докторског рада

Изјава о истоветности штампане и електронске верзије докторског рада

Име и презиме аутора Милош Котлар
Број индекса 5003/2017
Студијски програм Рачунарска техника и информатика
Наслов рада Детекција аномалија коришћењем мета података у аутоматизованим системима за машинско учење
Ментори доц. др Марија Пунт и проф. др Захарије Радивојевић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањивања у **Дигиталном репозиторијуму Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

Потпис аутора

У Београду, 01.02.2022.

образац изјаве о коришћењу

Изјава о коришћењу

Овлашћујем Универзитетску библиотеку „Светозар Марковић” да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

Детекција аномалија коришћењем мета података
у аутоматизованим системима за машинско учење

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

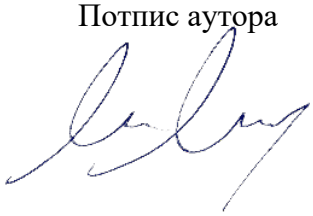
Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)
2. Ауторство – некомерцијално (CC BY-NC)
3. Ауторство – некомерцијално – без прерада (CC BY-NC-ND)
4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)
5. Ауторство – без прерада (CC BY-ND)
6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.
Кратак опис лиценци је саставни део ове изјаве).

Потпис аутора

У Београду, 01.02.2022.



1 **Ауторство**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2 **Ауторство – некомерцијално**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3 **Ауторство – некомерцијално – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4 **Ауторство – некомерцијално – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5 **Ауторство – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6 **Ауторство – делити под истим условима**. Дозвољавате умножавање, дистрибуцијуи јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.