



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

НОВИ САД



**Модел управљања просторно –
временским подацима у
паметном граду**

ДОКТОРСКА ДИСЕРТАЦИЈА

Ментор:
проф. др Миро Говедарица

Кандидат:
Младен Амовић

Нови Сад, 2023. године

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА¹

Врста рада:	Докторска дисертација
Име и презиме аутора:	Младен Амовић
Ментор (титула, име, презиме, звање, институција)	Проф. др Миро Говедарица, редовни професор, Факултет техничких наука, Универзитет у Новом Саду
Наслов рада:	Модел управљања просторно – временским подацима у паметном граду
Језик публикације (писмо):	Српски (ћирилица)
Физички опис рада:	Унети број: Страница 178 Поглавља 8 Референци 135 Табела 13 Слика 16 Графикона 0 Прилога 2
Научна област:	Геодетско инжењерство
Ужа научна област (научна дисциплина):	Геоинформатика
Кључне речи / предметна одредница:	Модел управљања паметним градом, геопросторни Big Data, Apache Spark SQL, Сензори, IoT
Резиме на језику рада:	У концепту паметних градова, користе се различите дигиталне технологије које пружају могућности прикупљања и процесирања геопросторних података. У том смислу потребно је податке добијене различитим технологијама, међусобно интегрисати у јединствену цјелину која је идентификована smart city концептом. Сходно томе, потребно је управљати структурираним, полуструктурираним и неструктурираним подацима. Успостављање система управљања који ће пружити могућности јединствене интеграције подразумијева потребу за рјешавањем тзв. 5V проблема. Проблеми система за управљање подацима се углавном се односе на: количину (volume) , различитост (variety), брзина (velocity), варијабилност (varibility) и вриједност (value). Предмет истраживања подразумијева успостављање новог концепта система управљања великим серијама просторно – временских података у оквиру паметног града који ће бити заснован на Big Data концепту трансформације, складиштења и процесирања геоподатака. Концепт користи приступ у рјешавању проблема употребом Big Data алгоритма паралелизације процеса, који обезбјеђују значајно боље перформансе система. Предложени модел је biG dAta sMart cItY maNagEment SyStem

¹ Аутор докторске дисертације потписао је и приложио следеће Обрасце:

5б – Изјава о ауторству;

5в – Изјава о истовјетности штампане и електронске верзије и о личним подацима; и

5г – Изјава о коришћењу.

Ове Изјаве се чувају на Факултету у штампаном и електронском облику и не кориче се са тезом.

	(GAMINESS). Модел управљања великим серијама просторно – временских података у контексту паметног града обезбијеђује: идентификацију стандарда под којим се врши дистрибуција података, прикупљање геоподатака који су у структурираној, полуструктурираној и неструктурираној форми, увођење комплексних типова података у потпуности заснованих на Big Data програмском окружењу Apache Spark, аутоматизоване процедуре превођења геоподатака у структурирану форму, успостављање интегралног система складиштења и управљања геоподацима и систем децентрализованог одлучивања у геосензорском пакету.
Датум прихватања теме од стране надлежног већа:	
Датум одбране: (Попуњава одговарајућа служба)	
Чланови комисије: (титула, име, презиме, звање, институција)	Председник: Члан: Члан: Члан:
Напомена:	

KEY WORD DOCUMENTATION²

Document type:	Doctoral dissertation
Author:	Mladen Amović
Supervisor (title, first name, last name, position, institution):	Prof. Miro Govedarica, Ph.D., full professor, Faculty of Technical Sciences, University of Novi Sad
Thesis title:	Management System of Smart City in the context of Big Data
Language of text (script):	Serbian language (cyrillic script)
Physical description:	Number of: Pages 178 Chapters 8 References 135 Tables 13 Illustrations 16 Graphs 0 Appendices 2
Scientific field:	Geodetic engineering
Scientific subfield (scientific discipline):	Geoinformatics
Subject, Key words:	Smart City, Geospatial Big Data, Apache Spark SQL, Sensors, IoT
Abstract in English language:	In the concept of smart cities, various digital technologies are used that provide opportunities to collect and process geospatial data. In this sense, it is necessary to integrate the data obtained by different technologies into a single entity identified by the smart city concept. Accordingly, it is necessary to manage structured, semi-structured and unstructured data. Establishing a management system that will provide opportunities for unique integration implies the need to solve the so-called 5V problems. Data management system problems mainly relate to: volume, variety, velocity, variability and value. The subject of the research involves the establishment of a new concept of a management system for large series of spatio-temporal data within a smart city, which will be based on the Big Data concept of transformation, storage and processing of geodata. The concept uses a problem-solving approach using Big Data process parallelization algorithms, which provide significantly better system performance. The proposed model is BIG DATA SMART CITY MANAGEMENT SYSTEM (GAMINESS). The model for managing large series of spatio-temporal data in the context of a smart city provides: identification of the standard under which data is distributed, collection of geodata in structured, semi-structured and unstructured form, introduction of complex data types based entirely on the Big Data programming framework Apache Spark, automated procedures for translating geodata into a structured form, establishing an integral system for storing and managing geodata and a decentralized decision-making system in the geosensor package.

² The author of doctoral dissertation has signed the following Statements:

56 – Statement on the authority,

5B – Statement that the printed and e-version of doctoral dissertation are identical and about personal data,

5r – Statement on copyright licenses.

The paper and e-versions of Statements are held at the faculty and are not included into the printed thesis.

Accepted on Scientific Board on:	
Defended: (Filled by the faculty service)	
Thesis Defend Board: (title, first name, last name, position, institution)	President: Member: Member: Member:
Note:	

ПРЕДГОВОР

У концепту паметних градова, користе се различите дигиталне технологије које пружају могућности процесирања геопросторних података коришћењем cloud технологија, интеграције различитих врста сензора, open data податка и слично. Подаци добијени различитим технологијама, морају међусобно да се повезују у јединствену цјелину која је идентификована концептом паметног града, а све у циљу превазилажења проблема које носе конвенционална рјешења представе и размјене геопросторних података. Размјена геоподатака коришћењем технологија заснованих на концепту паметних градова, обезбјеђује значајан пораст у коришћењу различито стандардизованих података који ће довести до бољег одрживог развоја животне средине и праћења свих појава које нас окружују у урбаним подручјима. Коришћење оваквих модерних геоинформационих технологија, захтијева адекватне опције складиштења података с напредним могућностима обраде тих истих. За управљање простором, често су доступни структурирани, полуструктурирани и неструктурирани подаци који се могу укључити у процес одлучивања. Стандардни системи управљања базама података имају проблем складиштења оваквих хетерогених података, као и њиховог укључивања у процесе одлучивања.

Успостављање система управљања који ће пружити ове могућности подразумијева потребу за рјешавањем тзв. 5V проблема. Проблеми система за управљање подацима се углавном се односе на: количину (volume) у контексту перформанси система приликом складиштења геоподатака као и количина просторно – временских серија геоподатака; различитост (variety) која се односи на колекцију геоподатака из извора који су базирани на размјени података на различитим стандардима; брзина (velocity) која се односи на перформансе реализације операција; варијабилност (variability) која подразумијева могућност извођења комплексних анализа над геоподацима из различитих извора, вриједност (value) која као и количина лимитира опције чувања геоподатака због перформанси платформе на којој ради.

Предмет истраживања подразумијева успостављање новог концепта система управљања великим серијама просторно – временских података у оквиру паметног града који ће бити заснован на Big Data концепту трансформације, складиштења и процесирања геоподатака. Ови проблеми се рјешавају употребом Big Data алгоритама

паралелизације процеса, који обезбјеђују значајно боље перформансе система. Предложени модел назван је biG dAta sMart cIty maNagEment SyStem (GAMINESS).

Модел управљања великим серијама просторно – временских података у контексту паметног града треба да обезбиједи: идентификацију стандарда под којим се врши дистрибуција података, могућност колекције геоподатака који су у структурираној, полуструктурираној и неструктурираној форми, дефинисање нових комплексних типова у потпуности заснованих на Big Data програмском окружењу Apache Spark, аутоматизоване процедуре превођења геоподатака у структурирану форму прописану новим системом управљања у оквиру концепта паметног града, успостављање интегралног система складиштења и управљања геоподацима, као и систем децентрализованог одлучивања у концепту геосензорских мрежа препознатог у систему паметног града.

PREFACE

In the concept of smart cities, various digital technologies are used that provide opportunities for processing geospatial data using cloud technologies, integration of different types of sensors, open data, and the like. Data obtained by different technologies must be connected to each other in a unique entity that is identified by the concept of a smart city, all with the aim of overcoming the problems caused by conventional solutions for the presentation and exchange of geospatial data. The exchange of geodata using technologies based on the concept of smart cities provides a significant increase in the use of differently standardized data that will lead to better sustainable development of the environment and monitoring of all phenomena that surround us in urban areas. The use of such modern geoinformation technologies requires adequate data storage options with advanced data processing capabilities. For space management, structured, semi-structured and unstructured data are often available that can be included in the decision-making process. Standard database management systems have the problem of storing such heterogeneous data, as well as their inclusion in decision-making processes.

Establishing a management system that will provide these opportunities implies the need to solve the so-called 5V problems. The problems of the data management system are mainly related to: the amount (volume) in the context of the performance of the system when storing geodata, as well as the amount of spatial-temporal series of geodata; variety, which refers to the collection of geodata from sources that are based on data exchange on different standards; speed (velocity) which refers to the performance of the implementation of operations; variability, which implies the possibility of performing complex analyzes on geodata from different sources, value, which, like the quantity, limits the possibilities of storing geodata due to the performance of the platform on which it works.

The subject of the research involves the establishment of a new concept of a management system for large series of spatio-temporal data within a smart city, which will be based on the Big Data concept of transformation, storage and processing of geodata. These problems are solved by using Big Data process parallelization algorithms, which provide significantly better system performance. The proposed model is called Big Data Smart City Management SyStem (GAMINESS).

The model for managing large series of spatio-temporal data in the context of a smart city should provide: identification of the standard under which data is distributed, the possibility of

collecting geodata in structured, semi-structured and unstructured form, defining new complex types based entirely on the Big Data programming framework Apache Spark, automated procedures for translating geodata into a structured form prescribed by the new management system within the concept of a smart city, establishing an integral system of storing and managing geodata and a decentralized decision-making system in the concept of geosensor networks recognized in the smart city system.

САДРЖАЈ

ПРЕДГОВОР.....	vii
PREFACE	ix
САДРЖАЈ	xi
ЛИСТА ПУБЛИКОВАНИХ ИСТРАЖИВАЊА	xiv
ПОПИС СКРАЋЕНИЦА И АКРОНИМА	xv
ПОПИС ТАБЕЛА	xvi
ПОПИС СЛИКА.....	xvii
ПОПИС ШЕМА.....	xviii
ПОПИС ПРИЛОГА.....	xix
I ТЕОРИЈСКО – ХИПОТЕТИЧКИ ДИО РАДА	1
1. УВОДНА РАЗМАТРАЊА	1
1.1. ПРОБЛЕМ И ПРЕДМЕТ ИСТРАЖИВАЊА	3
1.2. ХИПОТЕЗЕ С ОБРАЗЛОЖЕЊЕМ.....	7
1.3. ЦИЉЕВИ И ЗАДАЦИ ИСТРАЖИВАЊА.....	9
1.4. МЕТОДЕ ИСТРАЖИВАЊА.....	11
2. ДЕФИНИЦИЈА И ПРИМЈЕНА КОНЦЕПТА ПАМЕТНИХ ГРАДОВА У КОНТЕКСТУ ПРОСТОРНО – ВРЕМЕНСКИХ ПОДАТАКА	14
2.1. КОНЦЕПТ ПАМЕТНОГ ГРАДА	16
2.2. МОДЕЛ ПОДАТАКА ПАМЕТНОГ ГРАДА У КОНТЕКСТУ ПРОСТОРНО – ВРЕМЕНСКИХ ПОДАТАКА.....	22
2.2.1. ПРОСТОРНО - ВРЕМЕНСКИ МОДЕЛ ПОДАТАКА CITYGML.....	24
2.2.2. МОДЕЛ ПРОСТОРА CITYGML И INDOORGML СТАНДАРДА.....	30
2.2.3. ПРОСТОРНО - ВРЕМЕНСКИ МОДЕЛ ПОДАТАКА CITYJSON	33
2.3. ВРЕМЕНСКЕ СЕРИЈЕ ПОДАТАКА	34
2.4. СТАНДАРДИ ИЗ ОБЛАСТИ ГЕОПРОСТОРНИХ ПОДАТАКА.....	35
2.5. ИНФРАСТРУКТУРА ГЕОПРОСТОРНИХ ПОДАТАКА	36
2.5.1. ЕВРОПСКА ИНФРАСТРУКТУРА ПРОСТОРНИХ ПОДАТАКА – INSPIRE ДИРЕКТИВА	38
2.5.2. СТАНДАРДИ OPEN GEOSPATIAL CONSORTIUM	41
2.5.3. СЕРИЈА СТАНДАРДА ISO 19100.....	43

2.6. МЕТОДОЛОГИЈА ИЗГРАДЊЕ ПАМЕТНОГ ГРАДА.....	44
2.7. АНАЛИЗА ПОСТОЈЕЋИХ РЈЕШЕЊА ПАМЕТНОГ ГРАДА	50
2.8. ГЕОСЕНЗОРСКЕ МРЕЖЕ И ЊИХОВА ИНТЕГРАЦИЈУ У ИНФОРМАЦИОНОМ СИСТЕМУ ПАМЕТНОГ ГРАДА.....	52
3. ДЕФИНИЦИЈА И ПРИМЈЕНА КОНЦЕПТА BIG DATA ТЕХНОЛОГИЈА.....	56
3.1. ТЕХНОЛОГИЈЕ BIG DATA	58
3.2. ДОСАДАШЊА ИСТРАЖИВАЊА У ПРИМЈЕНИ КОНЦЕПТА BIG DATA ТЕХНОЛОГИЈА У РАДУ СА ПРОСТОРНО – ВРЕМЕНСКИМ ПОДАЦИМА И ПАМЕТНИМ ГРАДОВИМА	61
3.3. ОКРУЖЕЊЕ АРАСНЕ SPARK.....	73
4. АРХИТЕКТУРА GAMINESS СИСТЕМА ПРОСТОРНО – ВРЕМЕНСКИХ ПОДАТАКА.....	82
4.1. МЕТОДОЛОГИЈА ИЗРАДЕ GAMINESS СИСТЕМА	83
4.2. КОНЦЕПТУАЛНИ МОДЕЛ GAMINESS СИСТЕМА.....	87
4.2.3. КОНЦЕПТУАЛНИ МОДЕЛ GAMINESS СИСТЕМА СА ПРОШИРЕЊЕМ ЗА УНУТРАШЊУ ОРИЈЕНТАЦИЈУ	89
4.2.4. КОНЦЕПТУАЛНИ МОДЕЛ GAMINESS СИСТЕМА СА ПРОШИРЕЊЕМ ЗА СЕНЗОРЕ	91
5. РАЗВОЈ ПРОЦЕСНОГ И ИМПЛЕМЕНТАЦИОНОГ МОДЕЛА GAMINESS СИСТЕМА	96
5.1. РАЗВОЈ МОДЕЛА ТРАНСФОРМАЦИЈЕ У ОКРУЖЕЊЕ BIG DATA ПАРАДИГМЕ	99
5.2. ОКРУЖЕЊЕ ЗА РАЗВОЈ ИМПЛЕМЕНТАЦИОНОГ МОДЕЛА GAMINESS	106
5.3. РАЗВОЈ КОРИСНИЧКИ ДЕФИНИСАНИХ ТИПОВОВА ПОДАТАКА НА АРАСНЕ SPARK ОКРУЖЕЊУ	107
5.4. КОНВЕРЗИЈА МОДЕЛА У ПРОЈЕКТОВАНУ BIG DATA GAMINESS СТРУКТУРУ	112
5.5. ИМПЛЕМЕНТАЦИЈА ПАКЕТА ГЕОСЕНЗОРСКЕ МРЕЖЕ.....	115
5.6. РАЗВОЈ КОРИСНИЧКИ ДЕФИНИСАНИХ ФУНКЦИЈА У GAMINESS ОКРУЖЕЊУ	120
5.7. ТЕСТНО ОКРУЖЕЊЕ	123
5.8. ВЕРИФИКАЦИЈА МОДЕЛА	123
5.9. ДИСКУСИЈА.....	128

6. ЗАКЉУЧАК.....	134
7. ЛИТЕРАТУРА.....	136
8. ПРИЛОЗИ.....	146

ЛИСТА ПУБЛИКОВАНИХ ИСТРАЖИВАЊА

- [1] **Amović, M.**; Pajić, V.; Govedarica, M.; Vasiljević, S. Spatio-temporal types of data in big data paradigm. IFKAD 2016, Towards a New Architecture of Knowledge: Big Data, Culture and Creativity; Dresden, Germany, IFKAD, 2016; pp. 466–480, ISBN 978-88-96687-09-3 [Поглавље 2.4.4].
- [2] **Amović, Mladen** & Govedarica, Miro & Radulović, Aleksandra & Janković, Ivana. (2021). Big Data in Smart City: Management Challenges. Applied Sciences. 11. 10.3390/app11104557 [Поглавље 4.2.3, 4.2.4 и 5.8].
- [3] Pajić, V.; Govedarica, M.; **Amović, M.** Model of Point Cloud Data Management System in Big Data Paradigm. *Int. J. Geo-Inf.* 2018, 7, 265, doi:10.3390/ijgi7070265 [Поглавље 5.3].

ПОПИС СКРАЋЕНИЦА И АКРОНИМА

API	Application Programing Interface
BI	Business Intelligence
dwg	drawing
ГИС	Геоинформациони Систем
DBMS	Database Management System
DSL	Domain Specific Language
ETL	Extract, Transform and Load
ЕУ	Европска Унија
ISO	International Organization for Standardization
НИИП	Национална Инфраструктура Просторних Података
НИГП	Национална Инфраструктура Геопросторних података
OGC	Open Geospatial Consortium
pom	Project Object Model
RDBMS	Relational Database Management System
RDD	Resilient Distributed Dataset
РС	Република Српска
SDI	Spatial Data Infrastructure
shp	Shapefile
SQL	Structured Query Language
TB	Terabyte
UDF	User Defined Function
UML	Unified Modeling Language
HBase	Hadoop Base
HDFS	Hadoop Distributed File System
HQL	Hive Query Language
CQL	Cassandra Query Language
xml	Extensible Markup Language
WGS84	World Geodetic System 1984
WMS	Web Map Service
WFS	Web Feature Service
WPS	Web Processing Service

ПОПИС ТАБЕЛА

ТАБЕЛА 1. Основне карактеристике тематских модела стандарда CityGML.....	28
ТАБЕЛА 2. Дефиниција типова података за CityGML стандард зависно од окружења	29
ТАБЕЛА 3. Разлике између CityGML и CityJSON модела података	34
ТАБЕЛА 4. Досадашња истраживања у области Big Data Smart City концепта према 5V	61
ТАБЕЛА 5. Стандардни типови података у оквиру RDD шеме	78
ТАБЕЛА 6. Рад са дистрибуираним системима за складиштење заснованим на Apache Spark окружењу	80
ТАБЕЛА 7. Објекти и операције над њима у оквиру GeoSpark библиотеке.....	101
ТАБЕЛА 8. Предвиђене могућности Magellan библиотеке	102
ТАБЕЛА 9. Компаративна анализа Big Data техничких рјешења за рад са облаком тачака	105
ТАБЕЛА 10. Дефинисани типови геометријских примитива који су били предмет проширења	108
ТАБЕЛА 11. Карактеристике хардвера коришћених компјутера	123
ТАБЕЛА 12. Перформансе читавања	128
ТАБЕЛА 13. Понуђена рјешења у односу на досадашња истраживања у области Big Data Smart City концепта према 5V	129

ПОПИС СЛИКА

Слика 1. Различити типови агрегираних геометрија [92]	26
Слика 2. CityGML модел представљен кроз нивое детаљности [135].....	31
Слика 3. Примјер CityGML и IndoorGML простора [23]	32
Слика 4. Архитектура општег модела катастра према SDI [56].....	39
Слика 5. М2М шема сервиса у паметном граду [31]	45
Слика 7. Поређење између традиционалног модела паметног града и GAMINESS система [127]	84
Слика 8. Мапа пута методологије имплементације GAMINESS система управљања подацима [127]	85
Слика 9. Дијаграм компоненти архитектуре система GAMINESS	88
Слика 10. GAMINESS основни Abstract GML модел проширен IndoorGML класом за навигацију	90
Слика 11. Проширење CityGML пакета OGC Sensor Things елементима стандарда ...	92
Слика 12. Пакет Dynamizers са OGC Sensor Things проширењем [127].....	94
Слика 13. GAMINESS data-processing модули [127]	97
Слика 14. GAMINESS batch-processing модул	116
Слика 15. Мјерења процесирана као batch процеси са дефинисаним процедурама на сваком batch.....	117
Слика 16. Извод из излазне поруке мјерења са сензора у JSON формату	119

ПОПИС ШЕМА

Шема 1.	Кориснички дефинисана функција за тачку као прост примитив	110
Шема 2.	Проширење класе кориснички дефинисани тип	110
Шема 3.	UDT класе тачка [126].....	111
Шема 4.	UDT класе сензор	112
Шема 5.	Креирање стринга од постојећег DataFrame	113
	Обрнути процес је креирање шеме из JSON документа, гдје је као инстанца структурног типа узет jsonString као тип података (Шема 6).	113
Шема 6.	Креирање шеме из JSON документа.....	113
Шема 7.	Креирање DataFrame за систем GAMINESS из JSON документа.....	113
Шема 8.	Scala код за читање JSON документа	114
Шема 10.	Учитавање вектора из Apache Spark Machine Learning	118
Шема 11.	Binning уз помоћ Apache Spark Machine Learning библиотеке	119
Шема 12.	Encoding уз помоћ Apache Spark Machine Learning библиотеке	120
Шема 13.	Спајање уз помоћ Apache Spark Machine Learning библиотеке	120
Шема 14.	Класификација уз помоћ Apache Spark Machine Learning библиотеке .	120
Шема 15.	Кориснички дефинисана функција груписања и приказа података из SENSOR DataFrame	122

ПОПИС ПРИЛОГА

ПРИЛОГ 1. Извод из модела конверзије JSON шеме у DataFrames.....	146
ПРИЛОГ 2. Генерисање температурног DataFrame у предикцији	150

I ТЕОРИЈСКО – ХИПОТЕТИЧКИ ДИО РАДА

1. УВОДНА РАЗМАТРАЊА

Паметни град је концепт који превазилази ограничења традиционалног урбаног развоја коришћењем широког спектра могућности све већих количина геопросторних података и услуга које нуде дигиталне технологије (употреба Cloud технологија, Internet of Things или Open Data). Управљање и анализа података одувјек је представљала изазов у свим областима људског дјеловања. Најразличитије компаније као и јавне установе дуго су се бориле да пронађу прагматичан приступ за прикупљање информација о својим клијентима, производима и услугама. Да би опстали или стекли неку предност над корисницима у односу на конкуренцију, многе компаније су додавале нове линије производа те мијењале начин пружања услуга. Ови ефекти нису заобишли ни геодетску, геоинформатичку и картографску професију, тј. све оне који користе геопросторне податке. Потребне за новим и ефикаснијим производима у области геодезије расту из дана у дан. Готово да нема сфере живота у којој неки геопросторни параметри не фигуришу и не означавају важан фактор. Сходно томе, организације које се баве развојем и истраживањем у области геопросторних података имале су проблема да добију довољно рачунарске „моћи“ да би покренуле софистициране моделе за паралелне обраде података у реалном времену. Суочавамо се с много проблема када је ријеч о подацима. Неки подаци су структурирани и сачувани у релационим базама података, док су неки други подаци, укључујући документа, слике и видео записе, неструктурирани. Морамо да размотримо и нове изворе података које генеришу сензори, а који су базирани на различитим стандардима. Поред машина, извори информација су и они које генеришу људи као што су подаци из друштвених медија и click-stream подаци добијени с разних интернет страница. Доступност и прихватање нових, „паметних“ мобилних уређаја, уз сталан приступ глобалној мрежи доводе до нових извора података.

Скоро са свим изворима података може се независно управљати и претраживати. Тренутно, један од највећих изазова је пронаћи смислен пресјек свих тих података различитих типова и структура. Када постоји толико информација у свим различитим облицима, немогуће је размишљати о управљању подацима на традиционалан начин. Иако смо одувјек имали много података, разлика данас је само у врсти и начину обраде. Разне организације, више него икада раније, проналазе начин да искористе све информације које имају на располагању. Дакле, о управљању подацима мора се мислити

другачије и то је изазов, а уједно и шанса, за концепт назван Big data. Развојем технологија из подручја геоинформационих технологија, геоматике, геоинформатике, картографије и геодезије као резултат је добијено генерисање све већих количина геопросторних података. Стандардни приступи за њихово складиштење и обраду не могу, у овом моменту, да на задовољавајући начин одговоре на захтјеве које постављају толике количине података. Стога, предмет овог истраживања је модел управљања великим количинама просторно - временских података (*Big data*) који ће унаприједити постојеће приступе у складиштењу великих количина података прикупљених различитим геоинформационим технологијама.

Заједно, системи који интегришу овакве податке у контексту паметног града могу помоћи у повезивању различитих градских актера, побољшати укључивање грађана у систем одлучивања те понудити нове и побољшати постојеће услуге које су доступне у концепту дигиталног града са свим операцијама које се одигравају у њему. Употреба савремених геоинформационих технологија, укључујући сензоре за прикупљање различитих геопросторних података, захтијевају одговарајућу могућност складиштења тих података, како би се обезбиједило њихово прикупљање и даља анализа. У докторској дисертацији предложен је менаџмент систем *biG dAta sMart cItY maNagEment SyStem* (GAMINES) који је у потпуности заснован на концептима Big Data Apache Spark програмског окружења. У том контексту, истраживање обезбјеђује унапређење аналитичке обраде великих количина просторно - временских података и комбиновање различитих типова података у аналитичкој обради. Модел архитектуре нових типова просторно - временских података заснован је на Apache Spark Scala просторном упитном језику (*Spatial Query Language*) програмског модела, који представља широко прихваћен приступ за управљање и генерисање великих серија података с паралелно дистрибуираним алгоритмом на *cluster-y*³. Модул Spark SQL је дио Apache Spark програмског окружења који интегрише релационо процесуирање с Apache Spark функционалним програмирањем апликације. Постојеће библиотеке које на неки начин имају комуникацију с Apache Spark Scala програмским језиком и у којима су дефинисане методе и геопросторни типови проширене су новим типовима података. Основу предложеног модела чини архитектура рачунарског система, структура и организација података, као и правила за дефинисање алгоритама.

³ Cluster - Кластер је група међусобно повезаних рачунара или хостова који заједно раде како би подржали апликације и међуопрему (нпр. базе података). У кластеру се сваки рачунар назива „чвор“.

Предложени систем разликује се у односу на класичне базе података, јер пружа могућност смјештања и управљања огромним количинама структурираних, полуструктурираних и неструктурираних података у реалном времену, подижући перформансе система на виши ниво употребом концепата паралелизације процеса образложених кроз 5V принцип Big Data парадигме, о чему ће више бити ријечи у поглављу о концептима Big Data.

1.1. ПРОБЛЕМ И ПРЕДМЕТ ИСТРАЖИВАЊА

Развој сателитске и радарске даљинске детекције, глобалних навигационих сателитских система, аерофотограметријских камера, сензорских мрежа, ласерског скенирања и картографије доприноси експоненцијалном порасту количине прикупљених просторно-временских података. Последњих деценија, количине података су се значајно повећале. Системи за управљање базама података суочавају се са изазовом руковања тако великим количинама података, посебно са значајним модификацијама које су се догодиле код геопросторних података представљених, најчешће сложеним типом података. Доступност тако великих извора података, истиче потребу и значај великих количина података те подршку апликација и услуга у контексту паметног града. Професор Elman Al Nuaimi и сарадници истраживали су могућности за оптимизацију великих количина података у паметном граду [1]. Схватио је да главне проблеме Big Data у контексту паметног града могу се приказати кроз 5V дефиницију: Value, Velocity, Variety, Variability и Volume. Утврђено је да паралелна обрада великих серија геопросторних података може допринијети да њихова анализа траје неколико секунди умјесто неколико сати. Анализом доступне литературе, а према предложеном концепту идентификовани су **проблеми истраживања** за теоријску реализацију паметног града у концепту великих података:

- Подаци се генеришу у реалном времену, што доводи до огромних количина података које је потребно оптимизовати и адекватно их искористити у аналитичкој обради – ово представља проблем количине података (Volume) који је обрађен у следећим истраживањима [2], [3];
- Подаци се генеришу из различитих извора из којих се генеришу геопросторни подаци (паметни телефони, рачунари, платформе засноване на глобалним навигационим сателитским системима, геотехнички сензори, камере или подаци које у крајњем случају производе људи) – ово представља проблем

количине разноликости података (Variety) који је обрађен у следећем истраживању [4];

- Велике количине података се сортирају и складиште на различитим платформама што доводи до проблема у брзини приступа и одзива система – ово представља проблем брзине података (Velocity) који је обрађен у следећем истраживању [5]. Према концепту паметног града, захтијева се промјена у приступу извођења комплексних аналитичких операција над великим количинама података [6];
- У тако великим количинама доступних података, велики системи података морају да обезбиједу који су подаци који ће бити укључени у неке аналитичке процесе у фокусу побољшања апликација паметног града – ово представља проблем варијабилности података (Variability) који је обрађен у следећем истраживању [3];
- Потреба за трансформацијом начина управљања подацима у систем прикупљања великих количина података, базираном на пословним потребама, менаџменту и анализама – ово представља проблем количине података (Value) који је обрађен у следећим истраживањима [1], [7], [8].

На основу ових истраживања може се уочити да постоје бројни проблеми код система управљања паметним градовима. У поглављу анализа тренутног стања у области детаљно ће се анализирати доступна истраживања, али оно што је евидентно је да не постоји ниједно истраживање о доступним системима управљања паметним градом које је обрадило свих пет параметара у контексту Big Data. У контексту дефинисаних ограничења складиштења података, а која су повезана са концептом паметних градова у смислу релационих система база података дефинисан је предмет истраживања који подразумијева развој новог система управљања паметним градом у контексту Big Data. Акцент ће бити стављен на начин читавања и складиштења геопросторних података прикупљених из различитих извора података. **Предмет истраживања** огледа се кроз рјешавање сваког појединачног проблема:

- Количину (volume) – односи се на развој система управљања паметним градом који ће побољшати перформансе складишног система и перформансе платформе на којој је систем инсталиран;

- Разноликост (variety) – односи се на развој система управљања паметним градом којим ће се развити концептуални модел предложеног рјешења паметног града који ће имати могућност додавања различитих извора на предложену платформу будући да ради на различитим стандардима и протоколима;
- Брзина (velocity) – односи се на развој система управљања паметним градом којим ће се омогућити учитавање података из великог броја извора података за разлику од стандардних система за складиштење података. Проблем код стандардних система за управљање базама података је мања брзина у комплексним анализама у којима је потребно комбиновати податке из различитих извора података;
- Варијабилност (variability) - односи се на развој система управљања паметним градом који омогућава рјешавање проблема код извођења сложених анализа над подацима који се преузимају из различитих складишта података, а који су уједно различито структурирани;
- Количина (value) – односи се на развој система управљања паметним градом који рјешава проблем као и код проблема volume. Овим предметом истраживања рјешит ће се ограничења која се односе на перформансе система и платформе на којој се налазе подаци.

Апликације Big data захтијевају комбинацију различитих процесних техника, извора података и формата складиштења. Опције за оптимално складиштење и обраду података су се драстично прошириле. Технологије, као што су MapReduce и in-memory computing, обезбјеђују високо оптимизоване могућности за различите пословне намјене. Анализа података може бити извршена у реалном времену или веома близу реалног времена обрађујући цијели скуп података, а не репрезентативне узорке. Додатно, број опција за тумачење и анализирање података се такође повећао уз коришћење различитих технологија за визуелизацију. Све ово представља контекст на коме се базира Big data. Системи као што су Hive, Dremel и Shark дају предност декларативним упитима да обезбиједу већу аутоматску оптимизацију. Извођење операција над подацима различитих извора који могу бити у полуструктурираној или неструктурираној форми, захтијевају извођење напредних аналитичких обрада у релационим системима база података. Spark SQL представља модул у Apache Spark Scala програмском језику који

интегрише референтне обраде Spark функционалног програмирања апликације и прави паралелу са стандардним SQL код релационих база података [9]. Изграђен на Apache Spark модулу, Spark SQL омогућава Apache Spark велике предности приликом релационог процесуирања (нпр. декларативне упите и оптимизовано складиштење). Омогућава корисницима да позивају комплексне аналитичке библиотеке у Apache Spark (нпр. библиотеке базиране на машинском учењу). У поређењу са претходним системима, Apache Spark SQL нуди много прецизније интеграције између референтне и процедуралне обраде, кроз декларативни оквир апликације који се интегрише са процедуралним Apache Spark кодом. Укључује високо осјетљиве оптимизаторе, као што је нпр. Catalyst, коришћењем функција система Scala програмског језика⁴. То олакшава додавање нових правила, контролисање генерисања кода и дефинисање елемената проширења. Коришћењем Catalyst оптимизатора, дефинисане су бројне функције (нпр. шема за JSON преко које је имплементиран и GAMINESS, типове података који се користе у алгоритмима машинског учења и скупови упита за екстерне базе података) прилагођене за сложене потребе Big Data анализе података. Apache Spark SQL нуди генерални оквир за трансформисање структуре, коју користимо да изводимо анализе, планирање процеса и генерисање кодова у реалном времену. Кроз овај оквир Apache Spark SQL се проширује са новим изворима података, укључујући и полуструктуриране податке као што су подаци у JSON формату (у истраживању посебна важност се огледа јер се streaming подаци са сензора шаљу у овом формату) и „паметна“ складишта података над којим је могуће филтрирање (као што је HBase). Они располажу са кориснички дефинисаним типовима података и са кориснички дефинисаним функцијама, те доменима изведеним техникама машинског учења. Ови функционални језици су одговарајући за компајлирање и посредством њих се могу подићи перформансе оптимизације. На основу тога коришћењем нпр. Catalyst-а ефективно се додају нове могућности у Apache Spark SQL.

Програмски модел Apache Spark SQL пружа могућност дефинисања алгоритама за паралелну обраду великих серија података у оквиру дистрибуираног рачунарског система. Међу најпознатијим пројектима који је развијен на Apache Spark програмском окружењу је Apache Hadoop⁵, који представља софтверски систем отовреног кода који

⁴ Apache Spark се манифестује кроз R, Python или Scala програмски код.

⁵ Софтверска библиотека Apache Hadoop је оквир који омогућава дистрибуирану обраду великих скупова података кроз кластере рачунара користећи једноставне моделе програмирања. Дизајниран је за проширивање са појединачних сервера на хиљаде машина, од којих сваки нуди локално рачунање и складиштење.

омогућава дистрибуирану обраду података на cluster-у рачунара коришћењем Apache Spark SQL програмског модела. Посебан акценат се ставља на дефинисање нових типова података којима ће бити проширене постојеће библиотеке просторно-временских података којима су дефинисани основни геометријски примитиви. Њихова дефиниција као класе мора бити устројена тако да задовољи све елементе геометрије и топологије у складу са одговарајућим ISO стандардима у овој области.

Пројектовање логичког модела геопросторних података у фази дефинисања објеката и веза посједује низ специфичности у односу на ковенционалне податке. Једна од тих специфичности је идентификација могућих геометријских својстава објеката, односно атрибута којом ће та својства бити дефинисана. Типови података који се односе на геометријске типове података дефинишу сваки објекат или атрибут у зависности од њихове интерпретације у оквиру неке базе података. Геометријски типови података дефинисани су OGC и ISO стандардима.

1.2. ХИПОТЕЗЕ С ОБРАЗЛОЖЕЊЕМ

Као што је већ наглашено, истраживања су указала на важност концепта Big data и пословне интелигенције у функционисању великих система, којима се управља разним типовима података, као и на утицај ефикасног управљања комплексним типовима података која подразумева дугорочну одрживост и ефикасност тих типова у сложеним системима. На темељу изнесеног описа проблема оцјењено је научно релевантним истражити примјену концепта проширења постојећих просторно-временских типова података и предност овако дефинисаног рјешења које ће представљати основу проширења постојећих библиотека у Apache Spark SQL окружењу. Прегледом доступне научне грађе из подручја основних принципа и концепата проширења у Big data окружењу, као и из подручја проширења и дефинисања нових просторно-временских типова података у Big data окружењу, те успостављање адекватног модела управљања у контексту паметног града, заснованом на овим принципима, утврђено је како је мало сличних истраживања до данас спроведено и да ово свакако представља веома интересантну тему којом ће се у будућности морати посветити више пажње. Слиједом наведеног, посебно с обзиром на то да је 21. вијек – вијек информација и знања, а цијена прве информације јесте једнака цијени тржишног опстанка, намеће се потреба и оправданост за истраживањем примјене концепта проширења и дефинисања новог модела управљања просторно-временским подацима у контексту паметног града и

његовим односом с различитим форматима податка. Истраживачким хипотезама требала би се на темељу апликативног истраживања, потврдити њихова истинитост. Хипотезе су:

Општа хипотеза: Нови модел управљања просторно-временским подацима у паметном граду, у контексту Big data парадигме, омогућава рјешавање 5V проблема приликом геопроцесуирања великих количина просторно-временских података и њихову ефикаснију аналитичку обраду.

Посебне хипотезе:

1. Увођењем просторно-временских типова података омогућен је стандардизован приступ у Big data парадигми.
2. Проширењем CityGML стандарда за модел IoT, омогућена је далеко већа ефикасност у превазилажењу проблема варијабилности.
3. Дефинисање нових кориснички дефинисаних типова података, омогућава превођење података из полуструктуриране форме у структурирану форму модела.

Да би утврдили да ли нови модел управљања просторно-временским подацима у паметном граду у контексту Big data парадигме омогућавају геопроцесуирање огромних количина геопросторних података и њихову аналитичку обраду у кратким временским интервалима потребно је утврдити да ли модел који се добије има адекватан излаз, тј. да ли се нови типови података могу искористити у дефинисању операција у архитектури система у складу са унапријед постављеним захтијевима. Наведене хипотезе ће се анализирати, односно прихватити или одбацити на основу података који ће бити прикупљени апликативним истраживањем (научно истраживање). Практично рјешавање овог проблема има за циљ испитивање могуће примјене резултата утемељених на истраживањима или утврђивање нових метода и поступака за постизање жељеног циља.

Spark SQL користи угнијеждане моделе података базиране на Apache Hive⁶ за табеле и DataFrame-ове. Он подржава све основне SQL типове података, укључујући: *boolean*, *integer*, *double*, *decimal*, *string*, *date*, *timestamp* као и комплексне типове података: *structs*, *arrays*, *maps* и *unions*. Комплексни типови података могу бити угнијеждени заједно да креирају много функционалније типове података. И поред традиционалних система за управљање базама података, Spark SQL обезбјеђује најбољу подршку комплексним типовима података у креирању језика упита и апликација. Користећи овај систем, у

⁶ Софтвер складишта података Apache Hive™ олакшава читање, писање и управљање великим скуповима података који се налазе у дистрибуираном складишту користећи SQL.

могућности смо да прецизно моделујемо податке различитих извора и формата, укључујући Hive, релационе базе података, JSON, GeoJSON и изворне објекте у Java/Scala/Python што је неопходно за дефинисање модела управљања паметним градом.

1.3.ЦИЉЕВИ И ЗАДАЦИ ИСТРАЖИВАЊА

Полазећи од постављеног проблема истраживања те на темељу недовољног броја спроведених истраживања у овој области, поставља се питање како примјена концепта Big data доприноси већој ефикасности управљања просторно - временским подацима у предложеном моделу паметног града са аспекта дефинисања нових комплексних типова податка у Apache Spark SQL-у у односу на нека друга рјешења. У оквиру овог истраживања циљ је направити модел управљања просторно – временским подацима у контексту паметног града који ће пружити могућност руковања свим "паметним подацима" у једном интегралном моделу који их повезује и формира јединствено складиште података паметног града. Над тако структурираним моделом постоји могућност извођења сложених анализа и манипулација са подацима. Подаци који су прикупљени из различитих извора, а који улазе у категорију "паметних података", најчешће су прикупљени са различитих сензора који имају хетерогену структуру у погледу стандардизације према начину читања података, слања података и на крају њиховог чувања. Постоји неколико најчешће коришћених стандарда у контексту рада, прикупљања и складиштења података. Већина сензора ради под неким приједлозима заснованим на дефиницијама стандарда Internet of Things и SensorML стандарда [10].

Геопросторни подаци као Big data подаци се анализирају у погледу прикупљања, складиштења и обраде великих количина геопросторних података. Систем управљања у контексту процеса доношења одлука базираног на машинском учењу, представља додатни градивни блок у систему размјене и процесирања великих количина геопросторних података далеко ефикасније у односу на конвенционална решења [11]. Количина прикупљених података у великој мјери превазилази могућност њиховог складиштења на појединачним рачунарима, па рјешење представља складиштење података на кластеру рачунара [12]. Традиционални системи за управљање релационим базама података имају проблем да управљају свим овим подацима заједно, нарочито ако се разматрају у контексту Big data. Узимајући у обзир значај геопросторних података, мора се напоменути да 3D градски модели представљају саставни део паметног града [13].

У оквиру предложеног концепта система управљања просторно – временским подацима паметног града у контексту Big data у раду ће бити образложен нови модел biG dAta sMart cItY maNagEment SyStem (GAMINESS). Предложени систем базиран на Big data концептима пружа могућност рада са структурираним, полуструктурираним и неструктурираним подацима. Систем је описан кроз двије интегрисане компоненте које раде са извором геопросторних података и другим који омогућава интегрисање података са различитих сензора и додијеливање геопросторне компоненте. Систем GAMINESS на овакав начин омогућава повезивање различитих извора података без обзира на изворни стандард или формат улазног података. Ово рјешење посебно доприноси у рјешавању проблема варијабилности.

Циљеви овог истраживања представљају:

- Миграција стандардног концепта паметног града дефинисаног у оквиру релационих система у Big data окружење;
- Креирање концептуалног модела GAMINESS заснованог на стандарду CityGML 3.1.1 проширеног са елементима стандарда IndoorGML;
- Креирање концептуалног модела GAMINESS заснованог на стандарду CityGML 3.1.1 проширен са стандардом IoT;
- Креирање концептуалног модела GAMINESS заснованог на CityGML 3.1.1 стандарду проширен уз препоруке концепта паметног града уз спецификације дигиталних близанаца;
- Креирање концептуалног модела GAMINESS заснованог на стандарду CityGML 3.1.1 проширеног протоколима за повезивање за различите изворе података;
- Проширење стандардног CityGML-а за сложене типове података заснованог на изворима података;
- Израда GAMINESS модела који интегрише све улазне податке у јединствену структуру;
- Развој модела који је базиран на map–reduce парадигми Big data.

1.4.МЕТОДЕ ИСТРАЖИВАЊА

За потребе израде дисертације, спроведено је истраживање у ком су се користиле различите комбинације научних метода за потребе аргументације и евалуације постављених научних хипотеза.

Први дио истраживања усмјерен је на претраживање секундарних извора података чија је обрада послужила за израду теоријског дијела рада. Критички се анализира научна и стручна литература из подручја дефинисања нових типова података о простору и аналитичке обраде и геопроцесуирања великих количина података коришћењем модела Spark SQL. У овом истраживању најрелевантније су:

- **Метода анализе** којом се рашчлањивањем досадашњих концепата тврдњи успостављања традиционалних система управљања паметним градом стекла јаснија представа о објектима проучавања. Представља интуицију за рјешавање проблема захтијева познавања општих принципа управљања паметним градом, разумијевање конкретних механизма, усмјерену машту и анализу битних елемената, велико и широко искуство код управљања просторно – временским подацима.
- **Метода синтезе** којом су се повезале једноставније тврдње и закључци у сложеније и уопштеније тврдње са сврхом бољег и систематичнијег проучавања односа између објеката истраживања у концепту паметног града. У систему изучавања процеси анализе и синтезе се међусобно узајамно предпостављају и прелазе из једног у други. Најдубља веза која је у раду и описана је то што се ови процеси садрже један у другом у поступку успостављања новог система управљања паметним градом [14].
- **Методe индукције и дедукције** коришћене су за доношење закључака у теоријском и практичном дијелу рада приликом дефинисања проблема у систему управљања паметним градом. Ове методе се користе само са њене формално мисаоне стране приликом извођења општег става за дефинисање начина употребе Big Data приступа у успостављању паметног града установљеном из више посебних ставова којим је спозната објективна истина о коначној групи предмета истраживања.

- **Метода генерализације** која се користи са сврхом уопштавања од посебног појма до уопштенијег и схватања јединства општег и посебног (индивидуалног) у поступку рјешавања проблема варијабилности. Генерализују се заједнички елементи који се узимају као ошти оквир. Овом методом долазимо до појма који је по градацији виши од осталих појединачних појмова, с тим да је вјероватноћа добијеног појма постојана [15].
- **Метода дескрипције** за описивање појмова, законитости и поступака у оквиру успостављања система управљања паметним градом којим се на недвосмислен начин дефинишу односи између елемената цијелине као и правила на основу којих се успостављају. За овај дио коришћен је формални језик за специфицирање система UML.
- **Метода компарације** за потребе упоређивања истих или сличних чињеница, појава и процеса којим су објекти истраживања дефинисани са одговарајучим елементима упоредивости и дата одговарајућа оцјена њихових карактеристика чиме се недвосмислено може размотрити оправданост датог рјешења. Ова метода је коришћена у поступку верификације модела којим су перформансе система управљања паметним градом GAMINESS упоређене са процесуираним подацима на стандардним релационим системима података.
- **Метода моделовања** представља истраживачку процедуру током које је изграђен нови систем управљања паметног града GAMINESS (модел). Овом методом моделовања прво је дефинисан предмет који се истражује, друго означен је одређени начин одговара, треће дата је одговарајућа информацију о њему. На крају овим системом је, захваљујући наведеним својствима, могуће извршити експериментална истраживања и прорачуне или логичку анализу, како би се на овај начин постојећи типови података проширили на проучавану појаву и стекла поуздана знања. Овим теоријским моделом типова података одговорило се на реалним моделима.

Модел за управљање великим серијама просторно – временских података у паметном граду, имплементиран је на основу модела CityGML стандарда, проширеног за IoT елементе дефиниције сензора интегрисане у предложени модел, као и IndoorGML проширења везана за контекст путање. Реализација је изведена на Apache Spark open-source платформи на којој су дефинисани елементи складиштења и обраде великих серија

просторно – временских података на дистрибуираним рачунарским системима. Алгоритми за обраду геопросторних података дефинисани су према правилима Spark SQL програмског модела, а релационе операције на DataFrame-овима (специјализованим системом оквира података) коришћењем специфичног језика домена (domain – specific – language → DSL). Подаци су складиштени на екстерним складишним системима који подржавају нове типове података. Имплементација алгоритама је изведена у програмском језику Scala.

Испитивање модела за управљање великим серијама просторно - временских података у контексту паметног града, биће извршена над узорком који обухвата различите типове геопросторних података приказане у временским серијама. Узорак ће бити дефинисан на начин да се обухвате карактеристичне ситуације из праксе. Пошто је предмет истраживања управљање великим серијама просторно-временских типова података у паметном граду, узорак ће бити дефинисан тако да количина података по испитиваном скупу буде реда неколико десетина гигабајта.

2. ДЕФИНИЦИЈА И ПРИМЈЕНА КОНЦЕПТА ПАМЕТНИХ ГРАДОВА У КОНТЕКСТУ ПРОСТОРНО – ВРЕМЕНСКИХ ПОДАТАКА

Развојем информационих технологија последњих година, долази до великог развоја сервиса и апликација који доприносе аутоматизацији бројних процеса. Појавили су се бројни извори података који нуде различите информације, али које се врло често не могу адекватно искористити. Због тога, јавила се велика потреба за складиштењем и обрадом структурираних, полу-структурираних и неструктурираних података који нуде резултате различитим апликацијама у складу са њиховим захтјевима. Значајнија функционална побољшања јављају се као потреба да се доносе одлуке засноване на акумулираном знању кроз нови начин прикупљања и обраде информација неопходних за доношење одлука. Дигитализација природних феномена и давање геопросторних карактеристика појавама које нас окружују као и значајан пораст броја људи који живе у урбаном окружењу, довели су до потребе за оптимизацијом процеса у урбаном простору. Према истраживању Уједињених нација из 2018. године 55,3% људске попуације живи у урбаним просторима, са тежњом да пређе на 60,4% до 2030 [16], [17].

Ове чињенице као и хетерогеност просторне размјештености популације, указују да постоји потреба за новим приступом и стратегијама у планирању и обезбеђивања квалитетног живота. То се обезбијеђује кроз концепте планирања пољопривреде, здравствене његе, оптимизације саобраћаја и слично. На такав начин долази се до потребе за увођењем концепта паметног града. Развојем концепта паметног града пружа се могућност за вишеструким коришћењем добијених података који за крајњи циљ има боље управљање урбаним окружењем и животом. Паметни градови укључују велики број извора података (различите апликације са подацима који се налазе у различитим системима управљања базама података, сензори, разне врсте корисничких апликација) који имају различите принципе стандардизације у смислу структурирања, функционисања и испоруке података. Прикупљање различитих података из хетерогених извора даје бољи профил за доношење одлука, дефинисане кроз основне математичке принципе мултикритеријумске анализе. Геоинформационе и комуникационе технологије (GeoICT) све више се међусобно интегришу како би се подстакла урбана одрживост и структурирање паметних градова [18].

Концепт паметног града важан је концепт развоја живота за цијелу свјетску популацију. Паметни градови раде под добро дефинисаним и стандардизованим

моделима, углавном успостављеним коришћењем ISO серије стандарда којим се регулишу његове цјелине. Salina и други и Moniruzzaman и други у својим истраживањима су навели да постоји пуно извора података који се баве „паметним подацима“, али доступни модели паметних градова су углавном прилагођени да ријеше неке специфичне проблеме и дају повратне информације [19], [20]. Самим тим су потврдили да не постоји адекватно интегрално рјешење у овој области. Sivarañ и други су идентификовали визуелизацију као једно од ограничења паметних градова у контексту Big Data [4]. Визуелизација чини важан сегмент паметног града у процесу публиковања резултата, као и начина комуникације са корисником. Тежи се да се сваки феномен који се на неки начин идентификује и који је мјерљив у паметном граду, може и визуелизовати помоћу одговарајућих просторних референци. Већина просторних феномена може се представити основним геометријским примитивима, али је мали број рјешења који има довољан разрађен концепт визуелизације. Van Osteroom сматра да поред основних типова података који се најчешће користе у визуелизацији те развојем технологија за масовно прикупљање података, облак тачака може издвојити као засебна категорија података. Такође, сматра да се на овај начин може нагласити фотореалистичност и омогућити детаљнија анализа простора. Сходно овим тврдњама, као и препорукама стратегије Уједињених Нација о паметном граду у сврху одрживог развоја, облак тачка се може идентификовати као тип података којим се простор може дефинисати.

У оквиру истраживања подаци о простору биће учитани у овој структури, како би на што реалистичнији начин приказали податке о простору и омогућили вршење тродимензионалних анализа. У концепту паметног града идентификован је алат за геопросторну представу свих појава које се ту описују, односно свим параметрима даје се геопросторна конотација. Ту се долази до потребе да се изабере одговарајућа технологија којом се дефинише имплементациона форма паметног града у коме је приказана физичка структура екосистема који у њему егзистира. У контексту паметног града, постоји много различитих извора података који обезбјеђују податке за паметни град и процесе доношења одлука. Ти извори су ријетко исти, без обзира да ли се радило о изворима геопросторних или непросторних података, они су врло често ограничени у смислу компатибилности коришћених стандарда по којим функционишу. У процесу имплементације, паметни град се суочава са проблемом комбиновања тих, различитих извора података у контексту доступних модела података и алата који омогућавају ту

компатибилност. Стандард CityGML, омогућава ефикасно имплементирање архитектуре идентификованог система који обухвата геопросторне, али и податке других негеодетских извора података најчешће прикупљене путем различитих сензора. Стандард CityGML има велики потенцијал за проширења и адаптације за специфична ограничења која су предмет истраживања [21], [22]. Стандард CityGML има могућност геопросторног представљања реалног свијета на различитим нивоима детаљности, зависно од потреба презентације (Lod0 – Lod4). У оквиру модела података CityGML, омогућена је интеграција са различитим изворима података. Ту се као посебна категорија издвајају интеграције са различитим сензорима као мјерним уређајима и преузимању података из њихових структура. Наведени стандард има дефинисан пакет Dynamizer који у ту сврху препознаје сензор као објекат са карактеристикама уређаја и вриједностима мјерења.

Када је ријеч о управљању простором, као што је већ споменуто модел CityGML има могућност представљања паметног града на имплементационом нивоу у различитим нивоима детаљности. У оквиру своје дефиниције затворени простор са свим карактеристичним елементима се описују путем Lod4 нивоа детаљности. Ипак стандард CityGML са својим нивоима детаљности, има ограничења за неке важне анализе на овом нивоу детаља, као што је нпр. проблем навигације кроз затворени простор [23]. У својим истраживањима Kim и други [23] и Yuoo и други [23], [24] су за проблем навигације и дефинисања путања као адекватно рјешење предложили стандард IndoorGML који је предмет разматрања приликом дефинисања новог система управљања просторно – временским подацима у паметном граду.

2.1. КОНЦЕПТ ПАМЕТНОГ ГРАДА

Технологије представљају кључну улогу у одрживости града и дигитализацији система јавне управе. Оне представљају основу за успостављање паметног града. Нове технологије могу да дају робусна рјешења од којих корист могу да имају сви грађани. Циљ великих урбаних центара је да интегришу и уведу паметне системе у индустријске, инфраструктурне, образовне и социјалне елементе града. На овакав начин се подиже квалитет услуга које једна е-управа може да понуди својим грађанима. Сама идеја паметног града није у потпуности установљена, првенствено због великог броја области и технологија које се могу подвести под тај концепт. Концепт паметног града се објашњава кроз сервисе и услуге које он треба да понуди. Сходно томе неопходна су технолошка прилагођавања система како би обезбиједили тај ниво услуга. Самим тим се

разматра концепт архитектуре паметног града о коме ће бити ријечи у наредним поглављима. У том смислу разматра се архитектура паметног града у којој се дефинишу основне услуге које се морају обезбиједити у различитим областима и технологијама које је потребно да буду доступне. Те технологије подразумијевају платформе за прикупљање података из различитих извора (уређаја/сензора), могућност комуникације и преноса података између уређаја и система, претварање података у корисне информације и анализа тих информација у процесу доношења одлука. Сходно томе у поглављу Анализа постојећих рјешења разматране су платформе које се испитују у том погледу како би се добила повратна информација о захтјевима које треба да обезбиједе. Идентификовани су недостаци тих платформи и дат је увод за разраду предложеног GAMINESS система.

С обзиром на различите дефиниције концепта паметног града и његовог опсега, неопходно је успоставити типове услуга које паметни град треба да пружи грађанима. То ће помоћи да се циљеви појашњавају и да се схвате услови које град мора да испуњава да би био сматран паметним. Ово знање ће омогућити процјену различитих аспеката града и провјерити да ли може бити класификован као паметни. Упркос недостатку потпуно дијелене и глобално прихваћене дефиниције, могуће је описати најчешће карактеристике паметног града. Такође, дефинишу се услуге које могу да побољшају степен или ниво „интелигенције“ и у том погледу најважније аспекте. Дефиниција концепта „паметног града“ мора бити специфицирана да би могла да се врши та процјена. Различити аутори су се сложили око три главна аспекта приликом дефинисања овог концепта [25], [26]:

- Ефективност се односи на способност града да ефикасно пружа јавне и приватне услуге, као што су услуге према грађанима, компанијама и непрофитним организацијама.
- Предности у аспекту животне средине, односе се на побољшање квалитета животне средине у великим градовима. Један од главних стубова паметног града је спречавање деградације животне средине. Да би се то остварило, неопходно је спровести унапређења система у погледу потрошње енергије, загађености ваздуха и воде или регулације саобраћаја. Зато се у концепту паметних градова посебна пажња посвећује на ова рјешења како би се очувао квалитет животне средине.

- Иновације значе да паметни град мора примјењивати најнапредније технологије да би побољшао квалитет својих главних компоненти, како би се пружале боље услуге. Зато технологије представљају централни аспект паметних градова.

Сходно томе, мјерљивост степена паметног града може се побољшати ако се трансформише у ефективне и иновативне алате који нису штетни за животну средину. На тај начин је обезбијеђена и јавна вриједност. Међутим, ова три аспекта нису у потпуности довољна за повећање вриједности степена паметног града. Као што је Моор предложио [27], крајњи циљ треба да буде стварање паметнијег града гдје је захтијев да сви пројекти и иницијативе буду усмјерени на грађане. Концепт јавне вриједности је сложен и укључује неколико димензија :

- стварање економских и социјалних вредности, које је тешко ујединити и понекад улазе у конфликт једне са другима,
- креирање вриједности за различите заинтересоване стране, које могу имати различита очекивања која нису увек компатибилна једна са другим,
- стварање вриједности у погледу различитих димензија живота у граду, што такође подразумијева разумијевање стварних потреба и приоритета.

Наведеним је неопходно да се велики скуп промјенљивих уобличи кроз добро дефинисани свеукупни оквир да би се креирао концепт паметног града. Предложени оквир мора бити у стању да обезбиједи потребе, очекивања и перцепције грађана у погледу тога шта очекују од паметног града у свакодневном животу. Да би се ријешило овај проблем, неопходно је да се објаве услуге које нуди овај општи оквир. У овом оквиру, неопходно је идентификовати услуге које се често нуде друштву на структуриран начин, класификујући их у складу са њиховим доменом. Класификација услуга које паметни град треба да обезбиједи дефинисана је према основним показатељима:

а) Природни ресурси и енергија

- Паметне мреже - сервиси који подижу услугу на искуствени ниво од коришћења услуга електричне мреже на нивоу оптимизације гдје се у обзир узимају навике потрошача, одрживост, оправданост и сигурна дистрибуција енергије [28].

- Расвјета - градска расвјета са свјетилкама које пружају додатне погодности контроле загађености ваздуха или Wi-Fi конекцију пружајући могућност да се софтверски смањи потрошња базирана на читавом низу критеријума [29].
 - Обновљива енергија - експлоатација природних ресурса који су регенеративни или неискључиви, као што су топлота, вода или ваздух [30], [31].
 - Управљање отпадом - прикупљање, рециклажа и одлагање отпада коришћењем метода који спречавају негативне ефекте на животну средину или неадекватно управљање отпадом [31].
 - Управљање водом - анализа и управљање количином и квалитетом воде која се користи у пољопривреди, градским или индустријским намјенама [32].
 - Храна и пољопривреда - као што је коришћење геосензорних мрежа за управљање жетвеним процесом и мониторинг усјева [21].
- b) Транспорт и мобилност:
- Градска логистика - унапређење логистике градова ефикасно интегрисаним пословним потребама са условима саобраћаја и географским и еколошким питањима [33].
 - Информације о мобилности - дистрибуција и коришћење динамички изабраних информација, како прије завршетка путовања тако и током путовања, са циљем да се побољша саобраћајна и транспортна ефикасност, као и да се обезбиједи искуствене информације из саобраћаја [28].
 - Мобилност људи - коришћење различитих иновативних и одрживих начина да се обезбиједи превоз људима у градовима, као што су развој јавних саобраћајних режима и возила на зелено напајање, сви подржани напредним технологијама и проактивним понашањем грађана [34].
 - Услуге коришћења и интегрисања дистинктних информационих модела - Модели који су специфични за домен који укључују моделе података, Building Information Model (BIM), Geographic Information Model (GIS) и System Information Model (SIM) [28], [35].
- c) Паметне зграде:
- Управљање објектима - чишћење и одржавање урбаних објеката [25].

- Грађевинске услуге - коришћење услуга као што су електричне мреже, лифтови, противпожарни системи и телекомуникације [29], [36].
 - Побољшање квалитета становања - аспекти који се односе на квалитет живота у стамбеним зградама, као што су удобност, осветљење, гријање, вентилација и климатизација. Ова категорија укључује све што се односи на повећање нивоа задовољства људи у њиховом животном простору [37].
- d) Свакодневни живот:
- Забава - начини стимулација туризма, пружање информација о слободним догађајима, приједлози за слободно вријеме и ноћни живот [38].
 - Туризам - способност града да угости студенте, туристе и друге нерезиденте нудећи одговарајућа рјешења за њихове посебне потребе [39].
 - Контрола загађења - контрола емисија гасова и воденог отпада из употребе различитих уређаја. Доношење одлука о побољшању квалитета ваздуха, воде и животне средине уопште [30].
 - Јавна безбједност - заштита грађана и њихових ствари заснована на активном укључивању јавних организација, полиције, па чак и самих грађана. Прикупљање и праћење информација за превенцију злочина [40].
 - Здравље - превенција, дијагноза и лијечење болести подржаних информационим и комуникационим технологијама [41], [42].
 - Благостање и социјална инклузија - побољшање квалитета живота стимулацијом друштвеног учења и учешћа. Одређене групе грађана захтјевају посебну пажњу, попут старијих особа и особа са инвалидитетом [42].
 - Култура - ширење информација о културним активностима и мотивисање грађана да се укључе у њих [18].
 - Управљање јавним просторима - брига, одржавање и активно управљање јавним просторима ради побољшања атрактивности града и рјешења која посјетиоцима пружају информације на мјестима туристичке атракције у граду [43].

е) Влада:

- Е-управљање - дигитализација јавне управе кроз управљање документима и формуларима помоћу дигиталних алата, у циљу оптимизације рада и пружања брзих и нових услуга грађанима [44].
- Е-демократија - употреба информационих и комуникационих система за управљање гласовима [45].
- Транспарентност - омогућавање грађанима приступа службеним документима на једноставан начин и смањење шансе за злоупотребу овлашћења која могу користити систем за своје интересе или ускратити релевантне информације властима [45].

ф) Економија и друштво:

- Иновације и предузетништво - мјере за промоцију иновационих система и урбаног предузетништва [46].
- Управљање културним наслеђејем - употреба дигиталних система може посјетиоцима локалитета културног наслеђеја пружити нова искуства. Системи за управљање информацијама о имовини се користе за одржавање историјских зграда [46].
- Дигитално образовање - опсежна употреба методологија и дигиталних алата у школама [47].
- Управљање људским капиталом - политике које побољшавају инвестиције у људски капитал и привлаче и задржавају таленте, избјегавајући бјекство људског капитала, познато као одлив мозгова [48], [49].

Иако постоји више класификација, ова представља једну од најпотпунијих у погледу домена и поддомена у којима паметни град може бити у могућности да пружа услуге. С обзиром на велику разноликост постојећих услуга, која се еволуцијом технологије повећава готово свакодневно, предложена архитектура мора се користити као основа за имплементацију нових услуга, способних да подрже хетерогени скуп технолошких решења.

2.2.МОДЕЛ ПОДАТАКА ПАМЕТНОГ ГРАДА У КОНТЕКСТУ ПРОСТОРНО – ВРЕМЕНСКИХ ПОДАТАКА

Подаци прикупљени из апликација паметних градова могу помоћи грађанима да добију информације о догађајима који се дешавају у различитим дијеловима града. Већина рјешења за анализу података паметних градова усредсређена је на догађаје и појаве града у цјелини, што отежава разликовање тачног мјеста и времена последица одређеног догађаја. Сходно томе потребне су адекватне методологије модела у граду у контексту простора и времена. Из тих разлога је потребно имати одговарајуће платформе које подржавају учитавање и процесуирање просторно – временских података, најчешће усаглашених са принципима геосензорских мрежа. У савременом свијету модерне технологије за прикупљање, управљање и приказ геопросторних података могу се посматрати као монета будућности. Међутим, прикупљање података и информација и даље може бити тешко, поготово када су подаци динамички и просторно дистрибуирани. Једно од рјешења за овај проблем јесте примјена концепта Big data у колекцији и анализи овако великих количина података. Значајну улогу у тим процесима има развој геоинформационих технологија и услуга. Садашњу епоху карактерише изузетно брз пораст дигиталних географских садржаја, као последица развоја технологија за прикупљање, управљање и приказ геопросторних података. Савремене технологије уз развијену инфраструктуру просторних података окренуте су кориснику, при чему омогућавају брз и једноставан приступ релевантним просторним подацима уз све предности ефеката визуелизације. Наведене технологије доприносе спознаји географске стварности унутар сложеног комплекса међудјеловања човјека и простора. Доступност просторних података омогућава да се јасније сагледају проблеми и доносе ефикасније одлуке везане за стратегијска питања јавне и националне безбједности, одрживог развоја, заштите животне средине, приступа природним ресурсима, енергетске безбједности и многа друга питања.

Годинама су различите организације и компаније сакупљале трансакционо структуриране податке при чему су користиле batch обраду података, како би ставиле репрезентативне узорке у традиционалне релационе базе података. Анализа оваквих података је ретроспективна и истраживање се врши на скуповима података. Последњих неколико година, нове технологије омогућавају побољшано сакупљање, складиштење и анализу података по нижим комерцијалним условима, првенствено због пуно већег избора нових технологија у смислу колекције и обраде података. Различите интересне

групе сада могу сакупити више података из много више неструктурираних извора података (блогови, аудио и видео фајлови). Опције за оптимално складиштење и обраду података су се прошириле као последица развоја нових технологија. Због тога алгоритми MapReduce и in-memory computing, обезбеђују високо оптимизоване могућности за различите опције складиштења и управљања подацима. Анализа података може бити извршена у реалном времену или веома близу реалног времена обрађујући цијели скуп података, а не репрезентативне узорке.

Сходно овим сазнањима потребно је дефинисати концепт дигиталног близанца према коме просторно – временски модел паметног града треба бити успостављен. Дигитални близанци служе за апстрактну представу моделовања паметног града. Треба да побољша пословне процесе, смање ризик, оптимизују оперативну ефикасност и утиче на процес доношења одлука аутоматизацијом процеса и предвиђањем исхода. Дигитални близанци пружају већи контекст за рјешавање пословних изазова стварањем односа и поједностављењем токова рада. Технологија географског информационог система (ГИС) је основа сваког дигиталног близанца. Ове геопросторне технологије повезују информације, системе, моделе и понашања са просторним контекстом, стварајући холистички дигитални приказ окружења, имовине, мрежа и градова. Дигитални близанац је виртуелна представа стварног свијета, укључујући физичке објекте, процесе, односе и понашање. Геоинформациони системи стварају дигиталне близанце природног и изграђеног окружења и јединствено интегришу многе врсте дигиталних модела уважавајући временску компоненту. У том контексту геопросторни подаци и системи засновани на геопросторним технологијама побољшавају прикупљање и интеграцију података, омогућавају бољу визуализацију у реалном времену, пружају напредну анализу и аутоматизацију будућих предвиђања и омогућава размјену информација и сарадњу. Да би то било могуће, неопходно је идентификовање временске компоненте елемената система. Коришћењем концепта дигиталних близанаца модернизује се начин на који организације прикупљају и приказују податке, интегришу различите системе и врше анализу података. Неке од основних предности коришћења концепта просторно – временских података у концепту дигиталних близанаца су:

- моделовање података,
- системска интеграција и управљање,
- обрада у реалном времену,

- моделовање геометрије објеката у просторно – временском контексту и
- ток рада и пословни системи.

Водећи се овим принципима анализираћемо два најзаступљенија стандарда који служе за имплементацију паметног града уз уважавање просторно – временских карактеристика, а ријеч је о CityGML и CityJSON стандарду који и јесу основа за израду система управљања просторно – временским подацима у контексту Big Data. О мањкавостима изабраног стандарда и проширењима, детаљно ће се разматрати у поглављу везаном за имплементацију GAMINESS система.

2.2.1. ПРОСТОРНО - ВРЕМЕНСКИ МОДЕЛ ПОДАТАКА CITYGML

Стандард CityGML представља заједнички информациони модел за 3D урбане објекте и обезбјеђује обимну и детаљну представу објеката. Заснован је на стандарду ISO 19118 као отворени формат модела података који је базиран на XML језику. Служи за складиштење и размјену виртуелних градских 3D модела. Представља апликациону шему за верзију Geography Markup Language 3.1.1. (GML3), тј. проширење стандарда за просторну размјену података прописан од стране Open Geospatial Consortium (OGC) и ISO TC211. Кроз дефинисану структуру описан је геометријски и тематски модел. Овај стандард је детаљно описан кроз CityGML спецификације [50], [51]. Израда концептуалне структуре коришћењем UML дијаграма, подразумијева уважавање принципа објектно – оријентисаног моделовања. То подразумијева дефинисање спецификација над концептуалним моделом CityGML кроз вишеструкост атрибута, зависност и типове веза и дефинисање начина рекурзије, и то:

- а) Вишеструкост атрибута подразумијева да дефинисана варијабла може бити приказана у 0-*n* случајева (*). Атрибути су дефинисани типом података који омогућавају чување вишеструког резултата као ред или као предефинисана вриједност елемената. На овакав начин је дефинисан опсег броја објеката укључен у релацију.
- б) Зависности и типови веза подразумијевају да у CityGML моделу гдје постоје *n*:*n* односи, мора постојати додатна табела. Ова табела је састављена од примарних кључева табела оба елемента који креирају композитни примарни кључ. Уколико се веза може превести на 1: *n* или *n*:1 везу, креирање додатне табеле може бити

избјегнуто. Композитним кључевима се поједностављују зависности и релације између елемената који у њима учествују.

- с) Рекурзије у CityGML су заступљене у неким од односа унутар дефинисаног модела података. У оквиру ових процедура, неопходно је обезбиједити добре перформансе. Имплементација рекурзивних асоцијација ствара двије додатне колоне у којима се чува ID надређеног и ID подређеног елемента.

У оквиру CityGML концептуалног модела података, основни типови података су проширени кориснички дефинисаним комплексним типовима података који пружају могућност ефикасног представљања учитаних података. Просторне карактеристике геометријских елемената представљене су као објекти у оквиру дефинисаног GML 3 геометријског модела који је базиран на стандарду ISO 19107 „Просторна шема“ [52]. За 2D и 3D површинске геометрије, дефинисани су типови код којих се геометрије чувају као полигони, а који су даље агрегирани у структуре: MultiSurfaces, CompositeSurfaces, TriangulatedSurfaces, Solids, MultiSolids и CompositeSolids.

Стандард CityGML је написан као концептуални модел представљен UML дијаграмима како би се адекватно могли разумијети елементи структуре овог стандарда, неопходни за даља проширења [53]. У том контексту потребно је схватити који су услови под којим се могу правити измјене и прилагођавања модела новом концепту са разликама у односу на основни CityGML стандард. Сви елементи структуре модела представљени су и стандардизовани као дијаграм класе описан UML језичком нотацијом. Модели су подијелени у:

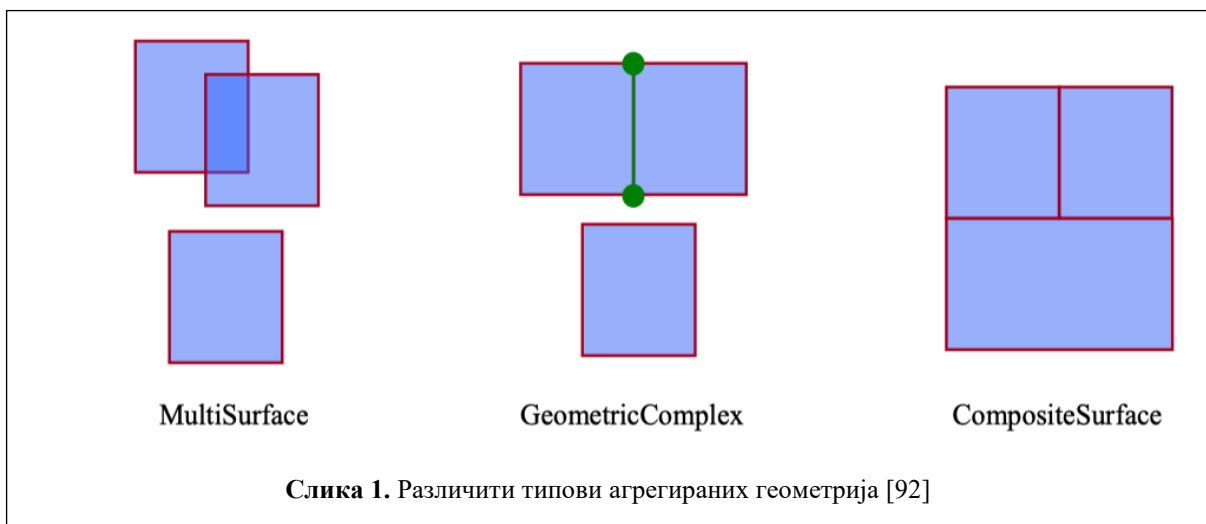
- геометријско - тополошки модел,
- модел представе (изгледа),
- тематски модел који представља проширење CityGML основног (Core) модела са елементима који се односе на: модел објекта, модел моста, градску инфраструктуру, дигитални модел терена, вјештачке објекте са пратећим атрибутима, начин коришћења земљишта, објекте транспорта приказане у 3D равни, модел тунела, модел хидрографије и модел објеката вегетације.

Детаљан опис модела дат је у следећим поглављима:

а) Геометријско - тополошки модел

Геометријски модел CityGML представљен је као цјелина примитива, који се комбинују како би формирали комплексне, композитне геометрије или агрегационе геометрије. Нула - димензионисани објекти су моделовани као тачке док су једно - димензионисани моделовани као криве. Криве су ограничене да буду представљене као сет правих линија, па се од верзије GML 3 користи само LineString тип геометрије. За потребе приказивања вишеструких површи, могуће је вршити комбиновање геометрије као агрегација, комплекса и композитних примитива (Слика 1).

Код агрегације, просторни однос између компоненти није ограничен. Оне могу бити раздвојене, преклапати се, додиривати се или бити искључене из структура. У оквиру стандарда GML 3, дефинисан је посебан тип агрегације за сваку димензију претходно наведених примитива. Ове агрегације су дефинисане као: MultiPoint, MultiCurve, MultiSurface или MultiSolid. За креирање сложених структура, посебна важност се огледа у јасно уређеном тополошком структурирању модела. Сегменти ових структура су раздвојени, при чему се не смију преклапати. Постоји могућност додиривања сегмената по границама или заједничким дијеловима граница. Од стављања на снагу стандарда GML 3, уведени су посебни комплекси дефинисани као композитне геометрије. Композитне геометрије подразумевају да елементи морају да буду састављени од геометрија истих димензија, при чему морају бити раздвојени, али и тополошки спојени по међусобним границама елемената. Композитне геометрије могу бити: CompositeSolid, CompositeSurface или CompositeCurve. На имплементационом нивоу модел је дефинисан релационом шемом гдје се геометрије записују у табели SURFACE_GEOMETRY.



б) Модел изгледа

Фотореалистичност и изглед површина у смислу видљивих карактеристика површина, представља један од саставних дијелова виртуелних градских модела. Често визуелизација и изглед површине могу бити брз и једноставан начин представљања резултата неких анализа. Важно је напоменути да текстуре морају бити означене неким географским референцама. У оквиру модела CityGML, сваки ниво детаљности може имати појединачне приказе. Независно од основне стурктуре, сваки објекат у оквиру модела може чувати своје податке о изгледу. Текстуре се дефинишу у оквиру класа CityObject и CityModel. Свака тема која се додијељује објекту, дефинисана је сопственим идентификатором, гдје се изглед према задатој теми дефинише скупом објеката класе изгледа (Appearance) упућивањем на атрибут теме. Објекат Appearance прикупља податке о површини релевантне за одређену тему преко релације SurfaceDataMember.

У том смислу текстуре се дефинишу као материјали који се додијељују структури геометрије. CityGML препоручује да се COLLADA формат користи за дефинисање текстура. Структура текстура се дефинише коришћењем RGB скале у којој су основне боје дефинисане у опсегу од 0 до 1. Основна класа текстура приказана је помоћу AbstractTexture. Овдје су текстуре увијек 2D текстуре засноване на растеру. Текстуре се могу квалификовати помоћу атрибута типа текстуре, разликујући текстуре које су специфичне за одређени објекат и прототипске текстуре типичне за ту површину објекта или као непознате. Спецификација текстуре усвојена су из дефиниције COLLADA докумената. Текстуре су параметризоване према локацији на површини којом се текстуре уклапају у текстурни простор.

с) Тематски модел

Састоји се од дефиниција класа за најважније типове објеката у виртуелним 3D моделима градова. Тематске класе се изводе из основних класа Feature и FeatureCollection према основним појмовим дефинисаних у ISO 19109 и GML3 за представљање обиљежја геометрија. У том смислу су приказане карактеристике просторних и непросторних атрибута мапираних у својства одговарајућих типова података. Геометријска својства су представљена као асоцијације на класе геометрије док тематски модел обухвата различите врсте међуодноса између класа као што су агрегације, генерализације и асоцијације. Навођењем тематских концепата и њихове семантике, заједно са њиховим пресликавањем на UML и GML3, различите апликације могу се ослонити на добро

дефинисан скуп типова атрибута и типова података са стандардизованим значењем или тумачењем. Да би омогућио размјену објеката и/или атрибута који нису експлицитно моделовани у CityGML, уведени су концепти *GenericCityObjects* и *GenericAttributes*. У табели 1 дат је преглед основних модела (пакета) дефинисаних стандардом CityGML.

ТАБЕЛА 1. Основне карактеристике тематских модела стандарда CityGML

Редни број	Тип модела	Основне карактеристике
1	Основни модел (<i>Core Model</i>)	Представља основну класу у оквиру тематских класа CityGML модела података представљену као <i>CityObject</i> . Има могућност представљања у свим нивоима детаља. Обезбјеђује креирање и брисање временских одредница објекта, преко ког се врши верзионисање модела и екстерно повезивање на друге одговарајуће објекте и сетове података.
2	Модел објеката (<i>Building Model</i>)	Модел објеката може бити представљен у пет нивоа детаљности од LoD0 – LoD4. Представља једноставне објекте који су састављени од једне компоненте или комплекса који чине дијелови објеката. У том контексту постоје класе <i>Building</i> и <i>BuildingPart</i> као дијелови класе <i>AbstractBuilding</i> . Зависно од комплексности и нивоа детаљности објекти се представљају геометријом <i>Solid</i> , <i>MultiSurface</i> и <i>MultiCurve</i> зависно од сегмената приказа. <i>Solid</i> и <i>MultiSurface</i> за објекте или дијелове објеката, док је <i>MultiCurve</i> за нпр. пресијек објекта и терена.
3	Модел мостова (<i>Bridge Model</i>)	Развијен је као аналогија моделу објеката са нагласком на структуру и атрибуте. Омогућава приказ тематских, просторних и визуалних елемената моста или дијелова моста. Као и код објекта постоји <i>Bridge</i> и <i>BridgePart</i> дијелови класе <i>AbstractBridge</i> . Представља се у LoD0 – LoD3 нивоу детаљности. Изузетно могу се приказивати у LoD4 нивоу детаља инсталације моста.
4	Модел урбаног намјештаја (<i>CityFurniture Model</i>)	Представља објекте као што су свјетлосна сигнализација, саобраћајни знакови, билборди, аутобуска стајалишта и друго. Класа има атрибуте класа, функција и употреба. Атрибут класе класификује објекат чему припада (нпр. саобраћајни знак). Атрибут функција дефинише којој тематској области објекат припада (нпр. транспорт). Атрибут употреба реферише стварну намјену објекта.
5	Дигитални модел терена (<i>Digital Terrain Model</i>)	Дигитални модел терена је посебна тематска цјелина у оквиру кога је омогућено комбиновање различитих модела представе (грид, тин, преломне линије, тачке) зависно од нивоа детаљности. Представљен је класом <i>ReliefFeature</i> и дефинише се LoD0 нивоом детаља. Може се представљати у различитим координатним системима, што представља значајну предност у размјени дигиталних модела терена у овом формату.
6	Генерички објекти и атрибути (<i>Generic Objects and Attributes</i>)	Овим моделом се описује чување и размјена 3D објеката, који немају експлицитно додијељену класу у оквиру модела или ако су потребни додатни атрибути. Слично као <i>CityFurniture</i> има исте атрибуте специфициране <i>gml:CodeType</i> .
7	Модел начина коришћења земљишта (<i>LandUse Model</i>)	Свака површ има свој начин коришћења. Слично као и у катастарским евиденцијама ова класе се користи за парцеле у 3D приказу. Сваки објекат има своје атрибуте класу, функцију и употребу.

8	Модел транспорта (<i>Transportation Model</i>)	Представља мултифункционалан и мултискалабилан модел који се фокусира на тематске, фунционалне и геометријско/тополошке аспекте. На LoD0 нивоу детаља описан је као линеарна мрежа, док на следећим нивоима детаља приказују се као 3D објекти и површине. Представљен је основном класом <i>TransportationComplex</i> којом се описују путеви, пруге, шине и сл.
9	Модел тунела (<i>Tunnel Model</i>)	Блиско је повезан са моделом објекта. Подржава приказивање тематских и просторних карактеристика тунела и дијелова тунела у нивоу детаљности LoD1 – LoD4. Описан је класом <i>AbstractTunnel</i> која у себи има садржане елементе тунела и дијела тунела.
10	Модел вегетације (<i>Vegetation Model</i>)	Описује вегетационе објекте као што је дрвеће и вегатационе области, приказује биотопе као што су шуме итд. Вегетациони објекат као јединка се моделује класом <i>SolitaryVegetationObject</i> , док се области представљају класом <i>PlantCover</i> . <i>PlantCover</i> елементи се дефинишу површинским геометријама. Јединка представљена класом <i>SolitaryVegetationObject</i> дефинише се у LoD1 – LoD4 координатама објекта или имплицитно геометријом.
11	Модел водних тјела (<i>WaterBodies Model</i>)	Представља тематску цјелину 3D геометрије ријека, канала, језера и сл. У нивоу детаља LoD2 – LoD4 водене површине су ограничене јасном површи. Постоје три основна елемента класе који су обавезни и то: <i>WaterSurface</i> (дефинисан као граница између воде и ваздуха), <i>WaterGroundSurface</i> (дефинисан као граница између воде и подземних површи) и <i>WaterClousureSurface</i> (дефинисан као виртуелна граница између различитих водних тијела и краја моделованог региона).

На основу сва три приказана модела, може се уочити да се ради о изузетно комплексном моделу који у имплементационом смислу када се преводи у релациону структуру међусобно повезаних табела, често је доста једноставнији, јер су бројне класе мапиране у једну табелу. Будући да изабрани модел CityGML је основа за изградњу Big Data GAMINESS модела, важно је анализирати који су то основни типови података на којим је модел описани. Сходно томе прави се анализа који су типови података подржани и који су то кориснички дефинисани типови података који ће се морати превести у структуру DataFrame (Табела 2).

ТАБЕЛА 2. Дефиниција типова података за CityGML стандард зависно од окружења

UML	Oracle	PostgreSQL/PostGIS
String, anyURI	VARCHAR2, CLOB	VARCHAR, TEXT
Integer	NUMBER	NUMERIC
Double, gml:LenghtType	BINARY DOUBLE	DOUBLE PRECISION
Boolean	NUMBER(1,0)	NUMERIC
Date	DATE, TIMESTAMP WITH TIME ZONE	DATE, TIMESTAMP WITH TIME ZONE
Primitive Type (Color, TransformationMatrix, CodeType)	VARCHAR2	VARCHAR
Enumeration	VARCHAR2	VARCHAR
GML Geometry, textureCoordinates	SDO_GEOMETRY	GEOMETRY
GML RectifiedGridCoverage	SDO_GEORASTER and SDO_RASTER	RASTER

2.2.2. МОДЕЛ ПРОСТОРА CITYGML И INDOORGML СТАНДАРДА

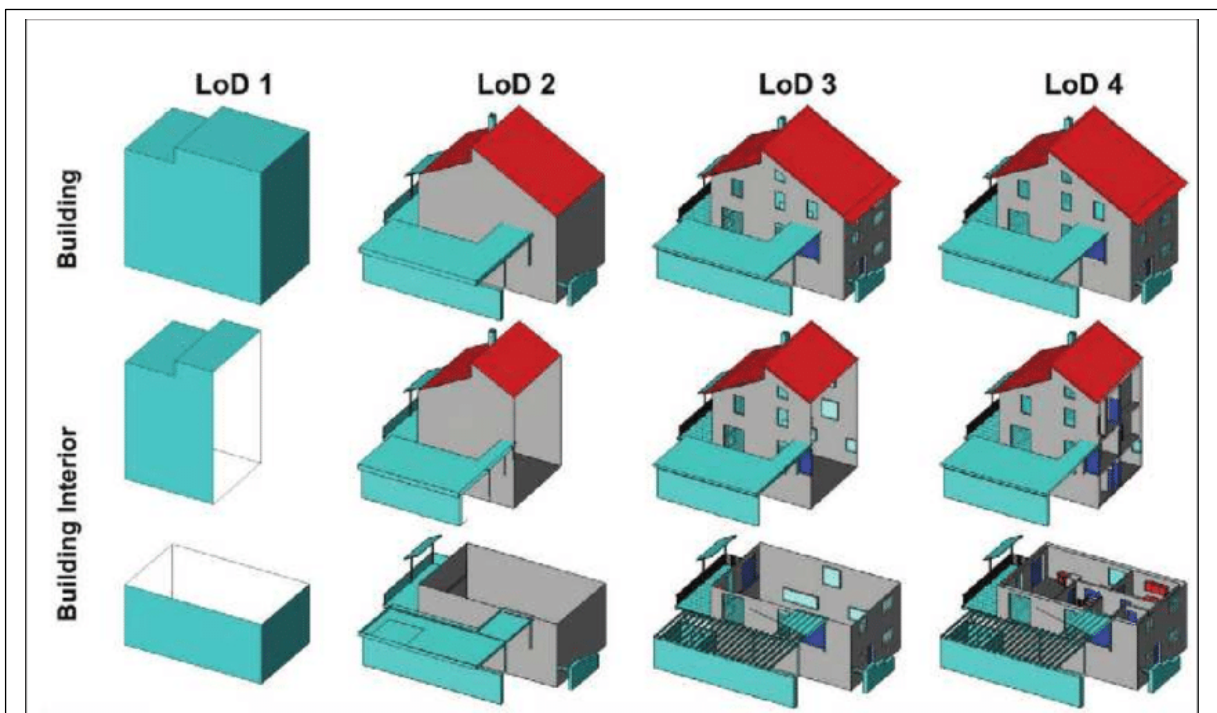
Стандард CityGML је свеобухватни стандард који дефинише основне ентитете простора са атрибутима и везама унутар генералног 3D модела града. Он обезбјеђује заједнички објектни (feature) модел за 3D градско окружење. Када је ријеч о 3D моделима, најчешће се простор представља као дигитални модел висина (DEM), дигитални модел терена (DTM) и дигитални модел површина (DSM). Дигитални модел висина представља модел терена са основном информацијом о висинама. Дигитални модел терена представља статичку представу континуалне површи земљишта на основу великог броја изабраних тачака добијених неком од метода прикупљања геопросторних података са познатим X,Y,Z координатама у произвољном координатном систему. За разлику од дигиталног модела висина садржи, информације о карактеристикама терена (експозиција, осунчаност итд.). Дигитални модел површи представља проширени дигитални модел терена који садржи и информације о трајним и вјештачким објектима, вегетацији и слично. Као што је већ и наведено у претходним поглављима CityGML представља свеобухватан информациони модел и као такав ће бити анализиран. Према основном моделу, детаљност приказа простора у оквиру стандарда CityGML је дефинисана помоћу концепта нивоа детаљности. Простор је представљен помоћу LoD0 - LoD4 нивоа детаљности (Слика 2):

- LoD0 ниво детаљности подразумијева представу 2,5 димензионалних дигиталних модела терена (DTM-Digital Terrain Model), на који се може поставити растер.
- LoD1 ниво детаљности је модел блокова који садржи зграде облика квадра представљене равним крововима. У оквиру LoD1 модел зграде садржи геометријску представу Building volume.
- LoD2 ниво детаљности је градски модел са фасадама и различитим кровним структурама са могућношћу представе вегетационих објеката. Спољне фасаде се семантички представљају помоћу класа: BoundarySurface – који описује дио вањског дијела зграде помоћу функционалности: зид (WallSurface), кров (RoofSurface), дио терена (GroundSurface). BuildingInstallation – који се употребљава за дијелове зграда (балкони, димњаци и сл).

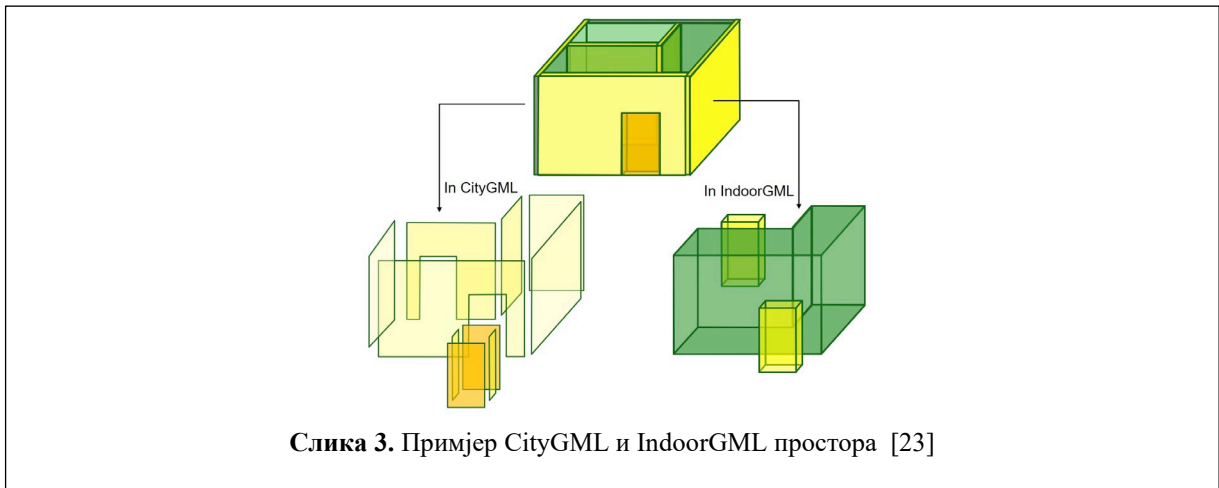
- LoD3 ниво детаљности класификује се као модел града односно модел детаљне архитектуре (структуре кровова, фасаде, балкони, увале и слично), а компоненте овог модела су саобраћајни знаци и објекти вегетације са детаљима.
- LoD4 представља модел ентеријера, односно архитектонски модел, додавањем унутрашњих структура 3D објектима, кроз који се може прошетати.

Као што се може видјети из претходног поглавља информациони модел CityGML пружа заједнички модел простора. Међутим у погледу LoD4 нивоа детаљности постоје основна ограничења у погледу контекста путања и навигације. У том погледу треба посматрати стандард IndoorGML као информациони модел који пружа свеобухватнији ниво у овом смислу.

Основни ентитет у IndoorGML моделу је ћелија гдје IndoorGML не прати само карактеристике сваке ћелије, већ и тополошке везе између ћелија како би објаснио структуру унутрашњег простора. У том смислу CityGML не пружа довољно карактеристика о унутрашњем простору, осим интерпретације равни са једним просторним слојем [12]. Ово ограничење представља основну разлику и намјену између ова два стандарда у погледу feature моделовања и моделовања простора.



Слика 2. CityGML модел представљен кроз нивое детаљности [135]



Основни ентитет IndoorGML је ћелија и IndoorGML не садржи само карактеристике сваке ћелије већ и тополошке везе између ћелија како би била дефинисана структура унутрашњег простора. На слици 3. приказана је разлика између CityGML Feature модела и IndoorGML ћелијског модела. Према приказаном моделу CityGML сет укључује колекцију features које су углавном површи, док унутрашњи простор у IndoorGML је декомпозиција сета непреклапајућих ћелија гдје су ћелије објекти које су распоређене по површи, док врата повезују ћелије.

Просторни модел CityGML описује карактеристике унутрашњег простора као равну интерпретацију јединственог просторног слоја, стога ћелијска представа у овом контексту боље представља карактеристике унутрашње геометрије. У IndoorGML моделу тополошки однос између ћелија је дефинисан експлицитно, гдје ивице и чворови мреже представљају ћелије, док врата представљају везу између ћелија у унутрашњем простору. Код CityGML топологија представља релације између површи које могу у унутрашњем простору бити веза између двије или више просторија. Предност CityGML се може посматрати у ClosureSurface елементима који су неправилне структуре као што су нпр. елементи који имају степенице. Ови елементи су дио унутрашњег намјештаја који није саставни елемент IndoorGML дефинисаног модела. Сходно резултатима истраживања [23] анализирани су предности и мане CityGML и IndoorGML на примјеру унутрашњег простора. Овом анализом предност је дата CityGML LoD 4 Feature моделу у контексту визуелизације и геометријске анализе док је IndoorGML ћелијски простор показао параметре који су боље оцијењени у проналажењу ћелија простора, хијерархијској презентацији простора, анализи рута и контекста путање.

2.2.3. ПРОСТОРНО - ВРЕМЕНСКИ МОДЕЛ ПОДАТАКА CITYJSON

Формат CityJSON представља формат за размјену података код кога је кодирање засновано на JavaScript Object Notation (JSON) синтакси. Он је подскуп модела података CityGML (од верзије 2.0.0) који представља стандардизовани модел података те формат за размјену 3D градских модела и околине. CityGML је усвојен као стандард од стране OGC као формат за размјену градских модела. Модел CityJSON дефинише већину уобичајних 3D урбаних карактеристика простора и објеката. Ту се као и у случају CityGML стандарда мисли на зграде, путеве, ријеке, мостове, вегетацију и намјештај, као и на међусобне односе између њих. Такође, елементи су дефинисани на основу различитих нивоа детаља (LoD) за 3D објекте, чиме је омогућено представљање објеката у различитим резолуцијама, намјенама и апликацијама. Циљ овог модела је да се понуди алтернатива GML кодирању CityGML, које је детаљно и сложено. Овај модел је једноставан за читање скупова података, њихово креирање и манипулацију. Проширив је, тако да се лако прилагођава апликацијама. Објекат CityJSON, који је дио града, је „раван“, сходно хијерархији CityGML која је спљоштена и у том дијелу имплементирају се само градски објекти који су „листови“ ове хијерархије. На овакав начин је олакшано складиштење градских модела.

Проширење модела је омогућено на структуриран начин кроз дефинисање нових објеката са сложеним атрибутима. Такође, дат је нови приступ у 3D моделовању градова, гдје тренутно постоји већ значајан број софтверских алата који подржавају овај формат. Моделом је омогућено чување геометрије, атрибута и семантике различитих врста 3D градских објеката који се могу употпунити различитим текстурама. Односи између различитих објеката се дефинишу уз помоћ CityGML. Ипак постоје одређени проблеми и огреничења објеката који се представљају уз помоћ CityGML, а нису могући уз помоћ CityJSON. Виљески и други напомињу да већина доступних 3D градских модела садржа више геометријских и тополошких грешака [54]. Најчешће су некомплетне површине, дупли чворови, самопресијћуће површине и слично. У табели 3 су приказане основне разлике и недостатци CityJSON модела података у односу на CityGML. У релационим базама података подаци су организовани у табеле. У CityGML моделу, једна или више класа UML дијаграма често се мапирају у једну табелу, где назив табеле одговара имену класе. Класе се комбинују у једну табелу према односима класа описаним у дијаграму класа UML модела.

ТАБЕЛА 3. Разлике између CityGML и CityJSON модела података

CityGML модули и подржане карактеристике	CityJSON модули и карактеристике које нису подржане
<ul style="list-style-type: none"> • CityGML основни пакет (<i>CityGML Core</i>) • Објекти (<i>Building</i>) • Мост (<i>Bridge</i>) • Тунели (<i>Tunnel</i>) • Урбани намјештај (<i>CityFurniture</i>) • Начин коришћења земљишта (<i>LandUse</i>) • Релјеф (<i>Relief</i>) • Транспорт (<i>Transportation</i>) • Група објеката (<i>CityObjectGroup</i>) • Вегетација (<i>Vegetation</i>) • Водна тијела (<i>WaterBody</i>) • Генерички модел (<i>Generics</i>) • Адресе (<i>Address Apperance</i>) 	<ul style="list-style-type: none"> • нема подршку за ADEs додатке • нема подршку за тополошке везе које могу бити дефинисане, као што су релативно у односу на терен (<i>relativeToTerrain</i>) или релативно у односу на воду (<i>relativeToWater</i>) • нема ClosureSurface елементе геометрије • нема подршку за произвољне координатне системе. Само координатни системи са листе EPSG могу бити коришћени. Све геометрије које су препознате у датом CityJSON документу морају користити исти координатни систем. • у CityGML већина објеката могу имати ID (најчешће <i>gml:id</i>), гдје један објекат има ID, а сваки 3D примитив формира своју геометрију која има ID. У CityJSON, само City Object типови могу имати ID, као и сваки семантички површински објекат.

2.3. ВРЕМЕНСКЕ СЕРИЈЕ ПОДАТАКА

Просторно-временске серије података су скупови података који повезују информације о времену и простору. Ове врсте података се користе за анализу и предвиђање трендова, развој догађаја и феномена у времену и простору. Примјери просторно-временских серија података укључују метеоролошке податке, податке о саобраћају, податке о кретању људи, податке о трговини и много других.

Анализа просторно-временских серија података може пружити значајне увиде и помоћи у доношењу одлука. На примјер, анализа метеоролошких података може помоћи у предвиђању временских услова, што је важно за сектор пољопривреде, авијације и туризма. За анализу просторно-временских серија података потребно је користити статистичке методе и алате за анализу података, као што су регресионе анализе, кластер анализе и моделирање временских серија. Паметни градови користе напредне технологије за побољшање квалитета живота становника и управљање градом. Big data концепт се односи на велике количине података које се стварају и користе у различитим секторима, укључујући управљање градовима. Просторно-временске серије података зато представљају важан дио биг дата контекста у паметним градовима.

Паметни градови користе просторно-временске серије података за анализу и управљање различитим аспектима града, укључујући саобраћај, енергију, здравље, сигурност и слично. Анализа просторно-временских серија података може помоћи у идентификовању трендова и промјена у граду, што омогућује градским властима да донесу боље одлуке у управљању градом. Међутим, употреба просторно-временских

серија података у паметним градовима ствара нове изазове. Један од изазова је питање приватности и сигурности података. Просторно-временске серије података укључују податке о понашању људи, што може довести до заштите приватности. Други изазов је управљање великим количинама података и примјена одговарајућих алата и техника за анализу. У свјетлу ових изазова, паметни градови морају проводити мјере за заштиту приватности и сигурности података, као и јачати капацитет за управљање и анализу просторно-временских серија података. Ова мјера ће омогућити паметним градовима да искористе потенцијал. Постоји неколико стандардних метода за приказивање временских серија, укључујући:

- Графикони временских серија - Ово је најједноставнији и најчешће кориштени начин приказивања временских серија. Графикони временских серија показују промјене у вриједностима варијабле током времена на једноставан начин.
- Стохастичка моделовања - Ова метода користи математичке моделе да би се описала сложеност временских серија. Модели кориштени у стохастичком моделовању укључују ARIMA, SARIMA и GARCH моделе.
- Дезаксонализација и десеасонализација - Ове методе користе се за исправљање временских серија које садрже сезонске утјецаје. Дезаксонализација користи се за исправљање временских серија које имају цикличне учинке, док се десеасонализација користи за исправљање временских серија које имају сезонске учинке.
- Тренд и сезонске анализе - Ове методе користе се за идентификовање трендова и сезонских утицаја у временским серијама. Тренд анализа помаже у идентификовању дуготрајних промјена у вриједностима варијабле, док сезонска анализа помаже у идентификовању краткотрајних цикличних промјена.
- Компонентна анализа - Ова метода користи се за анализу сложености временских серија. Компонентна анализа раздваја временске серије у неколико компоненти, укључујући тренд, сезонске и цикличне учинке.

2.4. СТАНДАРДИ ИЗ ОБЛАСТИ ГЕОПРОСТОРНИХ ПОДАТАКА

Стандардизација је процес утврђивања и примјене одређених правила ради сређивања и регулисања активности у одређеној области у корист и уз учешће свих заинтересованих страна, а нарочито ради остварења свеопштих оптималних уштеда, узимајући у обзир

функционалну намјену и захтијеве техничке безбједности. Стандард је документ у коме се дефинишу правила, смијернице или карактеристике за активности или њихове резултате ради постизања оптималног нивоа уређености.

У области просторних информација стандардизација представља кључан аспект за успјешно креирање, коришћење и дистрибуцију података. Стандарди дефинишу шеме потребне за опис геопросторних података и сервиса. Стандардима се повећава доступност података, могућност вишеструке употребе истих података, интегрисање сетова података и креирања нових информација, просторна анализа података, разумијевање података и смањују се трошкови креирања и одржавања података. Стандарди укључују спецификације, формалне стандарде и документоване праксе. За област просторних података битни су следећи нивои стандарда:

- Национални стандарди – стандарди на нивоу једне државе.
- Интернационални стандарди – стандарди на нивоу више држава. Циљ интернационалних стандарда је да се обезбједи структура за опис дигиталних геопросторних података. Припрему и креирање, као и публикавање интернационалних стандарда врше ISO (The International Organization of Standardisation) технички комитети. Поред ISO, у изградњи стандарда учествују и бројне интернационалне владине и невладине организације. Примарне групе стандарда и спецификација у области геопросторних података су ISO/TC 211 и OpenGIS. ISO TC 211 (Technical Committee ISO/TC 211, Geographic information/Geomatics) је технички комитет задужен за стандарде из области геопросторних информација [55]. Ови стандарди су апстрактни и представљају темељ за имплементацију. Open Geospatial Consortium (OGC) је задужен за креирање имплементационих спецификација.

2.5.ИНФРАСТРУКТУРА ГЕОПРОСТОРНИХ ПОДАТАКА

Развијени дио свијета увелико је у фази преласка из индустријског у информатичко друштво кроз трансформацију класичне технологије у информационо – комуникациону технологију, националне економије у свјетску, децентрализовану у централизовану, хијерархије у мрежу, институционалне помоћи у самопомоћ и сл. Наведени процеси могу се усвојити као основни принципи успостављања инфраструктуре просторних података (Spatial Data Infrastructure - SDI). Постоји више дефиниција о томе шта је то SDI. Једна од дефиниција инфраструктуре просторних података подразумијева означавање

технологија, политика, норми и људских ресурса потребних за прикупљање, обраду, складиштење, дистрибуцију и унапређење употребе просторних података . Такође, она обухвата скуп технологија, полиса и институционалних споразума који обезбеђују лакши приступ геопросторним подацима [56]. Инфраструктура просторних података подразумева сетове просторних података, метаподатке и сервисе просторних података, мрежне сервисе и технологије, споразуме о дијелењу, приступу и употреби, механизме координације и надзора, процесе и процедуре, успостављене, вођене или стављање на располагање са акцентом на питања заштите животне средине.

Национална инфраструктура просторних података – НИПП (National Spatial Data Infrastructure – NSDI) као посебан термин, омогућава повезивање и размјену хармонизованих података и сервиса из различитих извора од различитих власника података и чини их лако доступним за кориснике путем Интернета у циљу повећања добробити за друштво у цијелини. Овај систем чини скуп технологија, правила, стандарда и људских ресурса неопходних за прикупљање, обраду, смјештај, дистрибуцију и ефективно коришћење просторних података. Сврха интегрисане инфраструктуре је постизање боље расположивости геоинформација за кориснике и омогућавању јавним институцијама и осталим партнерима да размијењују податке на ефикасан начин [57]. Национална инфраструктура просторних података спада у групу стратешких докумената информационих система који би требао да постоји у свакој држави ради бржег и одрживог економског развоја. Може се рећи да НИПП представља најважнију карику у хијерархијском моделу SDI. На нивоу НИПП успоставља се институционални оквир као кључна компонента за организацију, финансирање, координацију и контролу инфраструктуре просторних података .

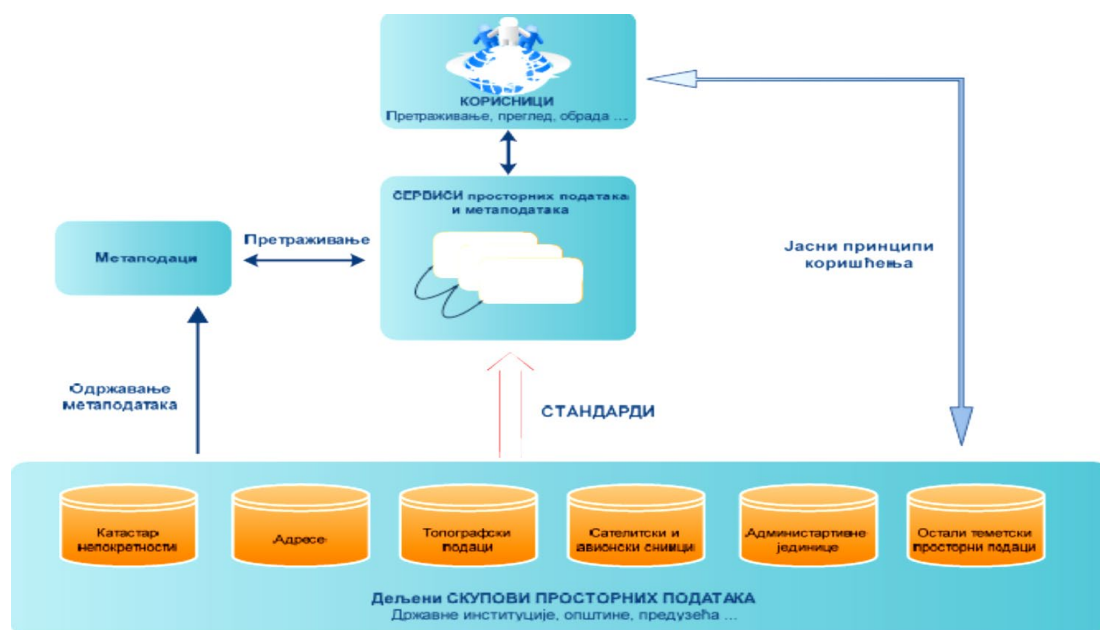
Национална инфраструктура просторних података треба да обезбиједи чување података о простору, њихову доступност и одржавање на одговарајућем нивоу, могућност комбиновања просторних података из раличитих извора и њихово дијелење између корисника и апликација. Подаци прикупљени на једном нивоу власти треба да се дијеле са другим административним органима, а проналажење и приступ расположивим просторним подацима треба да буде брзо и лако. Национална инфраструктура просторних података треба да се формира у складу са принципима дефинисаним у европској INSPIRE директиви. Основни концепт је да корисник може приступити лако геоинформацијама које су му потребне. Свако ко жели да приступи подацима мора то учинити кроз техничке компоненте.

2.5.1. ЕВРОПСКА ИНФРАСТРУКТУРА ПРОСТОРНИХ ПОДАТАКА – INSPIRE ДИРЕКТИВА

Потреба за INSPIRE директивом проистекла је из општег стања и недефинисаних правила везаних за просторне податке у Европи. То су: подијељеност и недостатак одговарајуће доступности података, мањак усклађености између података различитих географских размера, редувантност прикупљања, подијељеност и преклапање података, недовољно коришћење стандарда – некомпатибилни подаци и информациони системи, недостатак координације између управљачких нивоа који су одговорни за прикупљање података – од локалног до државног нивоа, недостатак одговарајућих података, рестрикције у дистрибуцији података – ауторска права, интелектуалне својине, приступ подацима, накнаде за коришћење и друго [56].

Директива садржи 35 генералних одредби, главни текст је подељен на 7 делова и чине га 26 чланова са три анекса (Анекси I, II и III), а у последњем 26. члану пише: „Ова Директива је намењена државама чланицама, Стразбур, 14. март 2007. године!“. Основне компоненте будуће просторне информационе инфраструктуре требало би да буду: а) метаподаци; б) мрежни сервиси и технологије, укључујући европски геопортал; в) интероперабилност и хармонизација низова просторних података и сервиса; г) споразуми за приступ, размијену, коришћење и дијелење просторних података и сервиса; д) механизми координације, мониторинга и извјештавања. Наведене компоненте ће омогућити реализацију усвојених INSPIRE принципа, изведених из основних принципа инфраструктуре просторних података [56]: просторне податке треба једном прикупити, а затим одржавати на нивоу гдје се то може радити на ефикасан и економичан начин; потребно је омогућити комбиновање просторних података из различитих извора широм Европе и дијелити их између великог броја корисника и апликација; просторни подаци прикупљени на једном нивоу управљачке структуре треба да буду доступни на свим нивоима; просторни подаци потребни за ефикасно управљање не би требали да имају ограничења у смислу масовног коришћења; лак приступ и проналажење доступних просторних података треба да буде праћен једноставном процјеном да ли су погодни за жељену употребу и под којим условима; приказ просторних података треба да буду асоцијативан, једноставан за разумијевање уз отворену могућност за додатне визуелизације у одговарајућем контексту сходно потребама корисника.

Изградња база просторних података на различитим нивоима и јединственог интегрисаног система у кохерентну инфраструктуру просторних података ЕУ, јесте основни циљ INSPIRE директиве (Слика 4). Директива предвиђа да се у првој фази ускладе и документују метаподаци и изграде механизми који ће ту документацију учинити доступном. Друга фаза, односи се на стварање услова за једноставан приступ просторним подацима, њихову једноставну анализу, без обзира на извор података и на коју тему се односе. Трећа, води ка апсолутној стандардизацији модела просторних података у одговарајућем домену. Прикупљени просторни подаци, мапирају се у заједнички модел који омогућава напредне анализе података, координацију и визуелизацију. Директива не захтијева прикупљање нових просторних података, већ се односи на постојеће сетове просторних података који испуњавају следеће услове: а) односе се на подручје гдје држава чланица има или извршава правосудну надлежност; б) у дигиталном (електронском) су формату; в) чувају се од стране јавне управе или треће стране; г) односе се на појмове наведене у Анексима I, II и III. Четврта фаза, предвиђа операционализацију услова за приступ актуелним метаподацима у реалном времену кроз модел интегрисаних просторних података из различитих извора, од локалног до европског нивоа, на бази јединствених стандарда и протокола. У спровођењу директиве посебну пажњу треба посветити моделу финансирања и политике цена за просторне податке[56]. Модел финансирања и креирање цијена за просторне податке и услуге значајно утиче на однос и приступ корисника просторним подацима. Један од услова за успешну имплементацију директиве јесте образовање тијела за координацију, главног



Слика 4. Архитектура општег модела катастра према SDI [56]

менаџмента на нивоу ЕУ и менаџмената на националним нивоима. За изградњу *ESDI* је одговорна Европска комисија, а координација и менаџменти на националним нивоима обезбјеђују функционалност и интероперабилност са *ESDI*.

Директива треба да понуди ријешење за баријере које ствара више од 20 језика, више од 100 различитих просторних референтних система, неколико хиљада високо квалитетних локалних и националних база просторних података, моделованих на различите начине и са различитим стандардима“[56]. Потребно је учинити напор, у смислу јединствених дефиниција метаподатака на свим језицима ЕУ, унификације каталога објеката и спецификација, уз уважавање националних и културних аспеката. Прихватање *INSPIRE* директиве води ка успостављању заједничког профила метаподатака, који обухвата модел метаподатака и формате метаподатака у функцији њиховог проналажења, приступа и коришћења. Интеграција националних инфраструктура у *INSPIRE* ће се реализовати кроз приступ истим инфраструктурама преко геопортала ЕУ којим руководи Комисија. *INSPIRE* геопортал представља Интернет сајт који омогућава приступ услугама. Ограничење у приступу, у смислу услуга откривања које омогућавају тражење сетова просторних података и услуга на основу садржаја одговарајућих метаподатака и приказа садржаја метаподатака, предвиђа се за случајеве када би такав приступ утицао на међународне односе, јавну безбедност и националну одбрану. Ограничење на услуге увида, преузимања, трансформације и активирања других услуга за просторне податке су предвиђена у случајевима потенцијално негативног утицаја на: а) на повјерљивост правног поступка; б) међународне односе, јавну безбиједност и националну одбрану; в) повјерљивост комерцијалних или индустријских информација; г) права на интелектуалну својину; д) повјерљивост личних података, заштиту животне средине, и) интересе или заштиту свих лица која су добровољно дала тражену информацију без законске присиле да то учине, осим ако то лице није сагласно са објављивањем такве информације. Анекси I, II и III Директиве, дефинишу појмове који се односе на сетове просторних података. Циљ директиве јесте да се утврде општа правила оснивања инфраструктуре просторних података у ЕУ. Имплементација националних инфраструктура треба да буде прогресивна, а питањима везаним за просторне податке треба додијелити одговарајући ниво приоритета. Директива има обавезујући карактер за све државе чланице ЕУ, а за државе кандидате представљају важан дио преговора у процедури стицања пуноправног чланства. Обавезе држава чланица ЕУ јесу да први извештај о имплементацији

Директиве поднесу Комисији до 15. маја 2010. године, а почевши од 15. маја 2013. извештај по одређеним питањима државе чланице достављаће сваке три године. Такође, треба поменути и PSI директиву која има за циљ подстицање поновне употребе података јавног сектора. Дефинише пет категорија геопросторних сетова података које треба консултовати приликом израде националне инфраструктуре просторних података. Интегрални модел за управљање просторно – временским подацима, мора бити успостављен на принципима дефинисаним INSPIRE директивом и општим правилима дијелења података. Општи проблем INSPIRE директиве је што је већина активности била усмјерена на 2D геопросторне податке.

2.5.2. СТАНДАРДИ OPEN GEOSPATIAL CONSORTIUM

Међународна организација за стандардизацију – OGC је непрофитна међународна организација која развија стандарде постигнуте консензусом својих чланова. Основни задатак OGC конзорцијума је развој отворених стандарда који омогућавају интероперабилност и интеграцију просторних информација, софтвера за њихову обраду и просторних сервиса. Просторне информације и њихова обрада обухватају географске информационе системе (ГИС), даљинску детекцију, премјер и картирање, навигацију, локацијски базиране сервисе, приступ просторним базама података, сензорски web, и друге просторне технологије и изворе информација. У OGC консензусном процесу, преко 250 владиних, приватних и академских организација кооперативно дефинише, развија, тестира, документује, валидира и одобрава интерфејсе и стандарде за кодирање који превазилазе проблеме интероперабилности. Имплементациони стандарди разликују се од апстрактних спецификација по томе што су писани за технички аудиторijум и спецификују детаље око структуре интерфејса између софтверских компоненти. За спецификацију интерфејса сматра се да је на имплементационом нивоу детаља ако, када се имплементира од стране два различита произвођача независно један од другог, резултујуће компоненте интерагују на том интерфејсу. OGC стандарди односе се на референтни модел, Web Map Service, Web Feature Service, Web Coverage Service, Web Processing Service, Web Catalog Service, GML, KML и GeoXACML. На бази ових сервиса развијен је и CityGML модел који представља основу за развој GAMINESS модела. Према основној OGC дефиницији, стандардни сервис за дијелење геопросторних података су:

- *Web Map Service (WMS)* је скуп спецификација интерфејса који дају униформан приступ од стране web клијената картама рендерованим на *map* серверу на Интернету. WMS је спецификација интерфејса сервиса који омогућава динамичку конструкцију карте као слике, као серија графичких елемената, или као скупа географских објеката. WMS укључује рендеревање геопросторних података. WMS одговара на основне упите о садржају карте. Може да информише друге програме о картама које може да произведе и над којима од њих се могу даље вршити упити. Основни WMS захтеви су *GetCapabilities* – враћа метаподатке о сервису, *GetMap* – враћа карту и *GetFeatureInfo* – враћа информације о објектима на карти на задатим координатама[58].
- *Web Feature Service (WFS)* [59] је web сервис који допушта кориснику да објави геопросторне објекте на интернету заједно са дефиницијом њихове структуре. WFS је интерфејс сервиса који описује манипулацију подацима о гео-објектима. Операције манипулације подацима укључују могућност додавања, брисања, ажурирања, добављања и вршења упита над гео-објектима на бази просторних и алфанумеричких ограничења. WFS може бити основни или трансакциони. Основном WFS може се приступити у *read only* моду, док трансакциони WFS (WFS-T) додаје механизам закључавања гео-објеката и подржава трансакције. За разлику од WMS који доставља податке у виду слике, WFS одржава директну експлоатацију и приступ подацима на webу. Основни WFS захтеви су *GetCapabilities* – враћа метаподатке о сервису, *DescribeFeatureType* – враћа GML апликативну шему која описује класу просторних објеката и *GetFeature* – враћа просторне податке у GML формату (према GML апликативној шеми).
- *Web Processing Service (WPS)* пружа стандардни интерфејс који поједностављује извршавање како једноставних тако и сложених просторних процеса као web сервиса. Ови сервиси укључују све просторне процесе који се могу наћи у ГИС алатима, као и специјализоване процесе за просторно и временско моделовање и симулације. WPS обезбеђује робустан, интеоперабилан и свестран протокол за извршавање процеса у оквиру web сервиса. Основни WPS захтеви су *GetCapabilities* – даје метаподатке о процесу (опис, како приступити процесу), *DescribeProcess* – потпун опис процеса (улази и излази процеса) и *Execute* – извршавање процеса. Неки од процеса који се могу извршити на web-у су бафер, унија, пресјек и слично [60].

2.5.3. СЕРИЈА СТАНДАРДА ISO 19100

Технички комитет ISO/TC211 специјализован је за стандардизацију у области дигиталних геопросторних информација. Циљ овог комитета је креирање низа стандарда који се односе на објекте и феномене који су директно или индиректно придружени некој локацији релативној у односу на Земљу. Ови стандарди дефинишу методе, алате и сервисе за управљање геопросторним подацима, укључујући њихово дефинисање и опис, прикупљање, процесуирање, анализирање, приступ, презентацију и трансфер просторних података у електронској форми између различитих корисника, система и локација. ISO/TC 211 описује ISO 19100 серију стандарда. Ови стандарди обезбеђују оквир за развој апликација које користе географске податке. ISO/TC 211 не описује имплементацију спецификација за различите платформе и софтвере, већ дефинише модел података високог нивоа за јавни сектор, као што су: влада, федералне агенције и професионалне организације. У наставку ће бити наведени стандарди који су коришћени за израду кориснички дефинисаних типова геопросторних података који су коришћени у GAMINESS окружењу. Неки од стандарда су:

- ISO 19101 – Референтни модел географских информација (ISO 19101 Geographic information – Reference model) даје оквир за ISO 19100 серију стандарда. Описује окружење у којем се врши стандардизација географских информација, фундаменталне принципе који ће се примјењивати и архитектуру оквира за стандардизацију [61].
- ISO 19103 – Стандард (ISO 19103 Geographic information – Conceptual schema language) даје правила и смернице за употребу језика концептуалне шеме у оквиру стандарда ISO 19100 серије. Одабрани језик концептуалне шеме је Unified Modeling Language (UML)[62].
- ISO 19107 – Просторна шема (ISO 19107 Geographic information – Spatial schema) дефинише просторну шему географских података. Просторни објекат (feature) је апстракција феномена из реалног света. Feature назива се географским ако је повезан за локацију релативну у односу на земљу. Векторски подаци састоје се од геометријских и тополошких примитива којима се представљају просторне карактеристике географских feature-а. Растерски подаци базирани су на подијели информације у мале цијелине мозаика и на додијели вриједности свакој цијелини. Овај стандард се бави само векторским подацима[63].

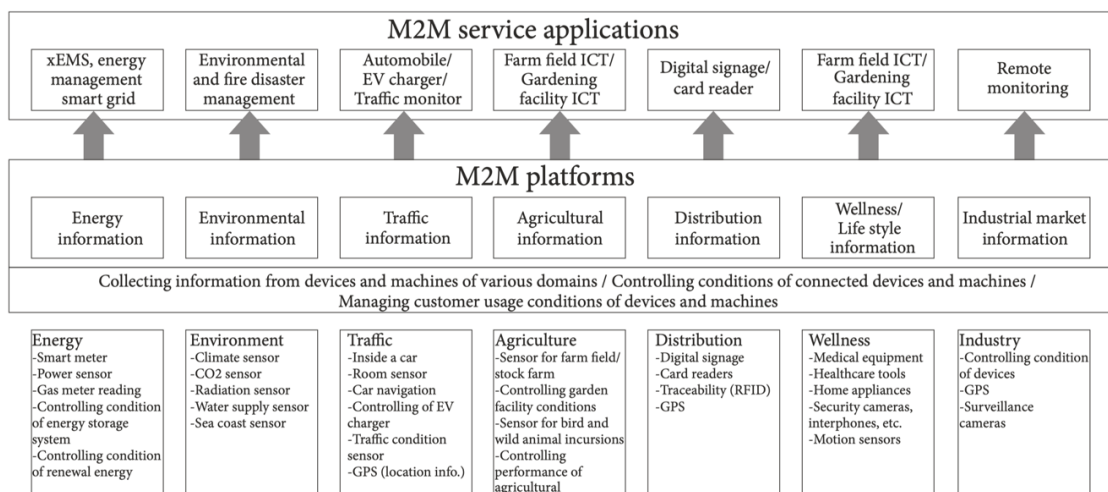
- ISO 19108 – Временска шема (ISO 19108 Geographic information – Temporal schema) дефинише концепте потребне за опис темпоралних карактеристика географских информација. Стандардизована концептуална шема за темпоралне карактеристике повећава могућност географских информација да се користе у апликацијама за потребе предиктивног и симултаног моделовања[64].
- ISO 19109 – Правила за дефинисање апликационе шеме (ISO 19109 Geographic information – Rules for application schema) дата су у оквиру овог стандарда. Како би географски подаци били разумљиви и корисницима и рачунарским системима структуре информација за приступ и размјену података морају бити документоване у цијелости. Зато, интерфејси за размјену између ових система морају бити дефинисани у складу са стандардним методама датим у овом стандарду. Апликативна шема даје формални опис структуре података и садржаја коју захтијевају једна или више апликација. Апликативна шема садржи опис географских података и опис других повезаних података. Основни концепт географских података је feature. Овај стандард дефинише правила за креирање и документовање апликативних шема укључујући принципе за дефинисање feature[65].
- ISO 19111 – Дефинише концептуална шему за опис просторних референци координатама (ISO 19111: Spatial referencing by coordinates), опционо проширене за друге просторно-временске референце. Описује минимум потребних података да дефинишу једно, дво и тродимензионални просторни координатни референтни системи са проширењем просторно-временског референтног система. Омогућава додатне описне информације да буду обезбијеђене. Такође, описује информације потребне да се промјене координате из једног координатног референтног система у други[66].
- ISO 19115 – дефинише шему потребну за описивање географских информација и сервиса (ISO 19115: Metadata). Обезбјеђује информације о идентификацији, при ширењу, квалитету, просторној и темпоралној шеми, просторном референцирању и дистрибуираним дигиталним географским подацима [67].

2.6. МЕТОДОЛОГИЈА ИЗГРАДЊЕ ПАМЕТНОГ ГРАДА

У области методологије пројектовања паметних градова, Internet of Things (IoT) је представљен као алат који пружа низ специфичних услуга које пружају подршку

различитим апликацијама које се нуде грађанима у овом окружењу [68]. Усвајање IoT концепта у контексту паметног града преузело је значајну улогу у том систему. Узимајући у обзир велики технолошки напредак као што су бежичне комуникације и значајна стандардизација комуникационих протокола мале снаге, омогућено је добијање података са сензора готово свуда и у било које време. Главни циљ IoT је да повеже све ствари и да гарантује да су све те ствари интелигентне. IoT је представљен као парадигма која омогућава стварима (Things концепту) да комуницира у окружењу, путем Интернета, представљено као нека врста рачунарске платформе [49]. Са друге стране може се дефинисати као ствар или објекат попут уређаја, сензора и паметних телефона који су способни да међусобно комуницирају и сарађују са интелигентним компонентама ради постизања заједничких циљева [69]. Оно што се чини очигледним је да када се говори о одређеним услугама које паметни град мора понудити, концепт IoT игра релевантну улогу, посебно у прикупљању информација из окружења путем сензора, као и у извршавању одређених радњи.

Други концепт који је блиско повезан са IoT је M2M платформе (машина до машине). M2M се посебно користи у паметним градовима, гдје веза није само између технологије и људи, већ и између машина и гдје било који објекат може постати дио мреже. Тако велики број елемената се може повезати, па је потребна нека врста механизма који подржава и обим веза и врсту комуникације. Овако се управља подацима на стандардизован и стабилан начин. Омогућено је комбиновање више врста веза и услуга (Слика 5). Сврха постојања оваквих система је управљање подацима на поуздан начин уз одговарајућу стандардизацију пословања. Концепт M2M система, пружа могућност



Слика 5. M2M шема сервиса у паметном граду [31]

интегрисања различитих технологија сензорских мрежа као што су ZigBee, Bluetooth, Wi-Fi, ViMak, PLC, GSM / GPRS, 6LoWPAN, EnOcean or Z-Wave.

Са проширењем IoT, број објеката повезаних путем Интернета ће се повећавати и самим тим је доступно више сензора за коришћење и укључивање у систем доношења одлука [70]. Тренутно је употреба неких сензора ограничена на врсту објеката у које су уграђени као и платформе са којих се прате ти сензори. Наведени модел се може користити чак и ако сензор није физички објекат и није неопходно да има придружени хардвер. Постоје могућности креирања виртуелних сензора који подразумевају софтверски систем који се користи као сензор. Такви системи су сврсисходни јер имају могућност прикупљања података. У том систему онај ко располаже сервером на коме је интегрисан сензор, прикупља податке и дијели их по потреби. У овом концепту постоје четири основна слоја:

- Слој карактеристика сензора и сензор као објекат: сензор представља концепт ентитета (софтвера или хардвера) који детектује или мјери физичке величине, записује их и преноси. Подаци са сензора могу бити јавни уколико такве податке јавне институције одлуче да објавље. За распоређивање јавних или приватних сензора потребна је дозвола власника. Власници сензора одлучује да ли да објави сензор у cloud.
- Слој издавача сензора: открива доступност сензора, комуникације са власницима сензора и добијање дозволе за објављивање података помоћу cloud технологија.
- Слој проширених услуга подразумева коришћење података са сензора као улаз за неке друге параметре који се пружају према корисницима.
- Ниво потрошача сензора представља апликације које користе податке које обезбеђују сензори. Све комуникације са корисницима података обавља се помоћу сервиса са сензора према корисницима.

Овакав модел омогућава управљање свим ентитетима града који треба да буду уврштени као извори података (сензори), од прикупљања података до примјене апликација које пружају коначне услуге грађанима. Инфраструктура архитектуре паметног града дозвољава комуникацију између повезаних објеката или ентитета, што омогућава добијање информација прикупљених од стране различитих хетерогених ентитета. Пошто се број ентитета повезаних са инфраструктуром може разликовати, као и обим генерисаних података, рачунарске инфраструктуре морају бити у стању да се

прилагоде складишту, количини и брзини потреба за одговором. Да би се обезбедило рјешење за потребе тамо где су ти фактори непознати, појављује се технологија Cloud Computing. Концепт Cloud Computing је прилично широк. Односи се и на апликације које се нуде у виду услуга преко Интернета, као и на хардверске и софтверске системе који пружају те услуге у *Data* центрима. Предност корисника у коришћењу оваквих рјешења се огледа у оптимизацији рачунарских ресурса према потребама система, без специфичних захтјева за инфраструктуру.

Концепт паметног града треба да обезбиједи дистрибуирану инфраструктуру. Ова инфраструктура мора бити у стању да прилагоди ресурсе потребне за пружање различитих услуга у реалном времену, без обзира на њихове захтјеве за обраду. Неопходно је утврдити како се информације требају обрађивати у овом окружењу. У том смислу, велики дио информација може бити одбачен зато што се сматра нерелевантним или старим након неког времена пошто те информације више нису релевантне. Нове технологије појавиле су се последњих година да би се пружило рјешење за проблеме везане за велике количине података. NoSQL базе података су једно од рјешења за овакве структуре.

NoSQL базе података, у поређењу са релационим базама података, нуде још један скуп различитих опција за складиштење великих количина података као дио концепта Big Data. Постоји више модела NoSQL база података који нуде различите функције и у зависности од потребе користе се за управљање, експлоатацију и успостављање релација у оквиру база података. Тренутно постоје различити модели у којима се могу класификовати NoSQL базе података:

- Базе података према кључним вриједностима које представљају најједноставније NoSQL базе података. Оне складиште вриједност (пандан реду у релационим базама података) која садржи кључ и његову вриједност која је потпуно независна у бази података. Овај тип базе података се користи када се подацима приступа помоћу примарног кључа и може да изврши једноставне операције као што су складиштење, упити и брисање евиденција. Основна предност ових база података је да су њихове перформансе веома високе упркос њиховој једноставности. Оне нуде врло једноставну скалабилност.
- Колона оријентисане базе података су структуриране. Оне се базирају на колонама које складиште податке у *"column family"* концепту, што би било упоредиво са

табелама у релационом моделу база података. Свака од ових структура садржи један ред са више колона међусобно повезаних са кључем. Поред тога, у овом моделу, редови не садрже исте колоне. Овај тип базе података користи софтициране кључеве, у којој вриједности више нису стриктне унутар базе података већ се формирају према дефинисаном обрасцу. Највећа предност овог типа база података је могућност анализе великих података у реалном времену.

- Графички оријентисане базе података су модели база података засноване на теорији графика. Они користе чворове веза да представе сет сачуваних података. Чворови складиште ентитете, док су различите везе моделиране везама које им се придружују. Овај тип база података се тренутно углавном користи у рекурзивним претрагама на више нивоа или оним конципираним за друштвене мреже.
- Базе података оријентисане на документе, засноване су на складиштењу докумената и то на пару кључних вредности. Документ је структура података у облику стабла која може да се састоји од мапа, колекција и скаларних вредности које се могу ускладиштити у текстуалним датотекама типа XML (Extensible Markup Language), JSON (JavaScript Object Notation), BSON (Binary JSON) итд. Није неопходно да документи буду идентичне структуре. Главна предност ових система је висока хоризонтална скалабилност података, што га чине веома ефективним типом базе података за складиштење великих количина података заснованог на Big Data концепту.

Ово је више врста архитектура које су доступне у овом концепту и које се могу бирати како би се што боље прилагодили траженом концепту. Свака архитектура нуди различиту структуру за организовање информација, али ове базе података омогућавају раздвајање начина на који су информације организоване из структуре самих информација. У оваквим концептима употреба модела NoSQL базе података је најпогоднија. Сходно томе масовно генерисање дигиталних података у актуелним општим мрежама и одређеним концептима паметних градова, отворила је врата новим технолошким трендовима који траже додатну вриједност кроз анализу ових дигиталних података. Један од главних разлога константног генерисања дигиталних података је то што мобилни уређаји постају све сличнији рачунарима, уз брзу Интернет везу и геолокацијске системе. Све то значи да све радње које су предузете на мобилним телефонима, рачунарима и повезаним објектима, оставља одређени траг. Све ово

генерише велику количину информација које се константно повећавају и акумулирају док чекају да буду анализирани. Поред тога, неке од нових апликација које су се недавно појавиле имале су велики утицај на друштво, као што су отворени извори података, друштвене мреже, појава IoT. Анализа информације прикупљених на овај начин отвара врата новим приликама, што може донијети велике предности: штедњу трошкова, повећану зараду, стварање радних места итд. Аналитике могу да помогну у побољшању управљања предузећима или градовима, омогућавањем предвиђања у низу поља или откривању односа узрока и ефекта између података који до сада нису могли да се схвате. У том контексту Big Data има значајну улогу у концепту паметних градова. Истраживања у Лабораторији за компјутерске науке и вештачку интелигенцију, на Институту за технологију Масачусетс, предложили су нови систем под називом Data Science Machine, софтвер заснован на вјештачкој интелигенцији који је способан да пронађе шаблоне у односима између података и да се предвиђања формирају као обрасци у кратким временским интервалима [71]. Један од основних фактора за аутономно издвајање знања из великог скупа података је идентификација промјенљивих што представља концепт машинског учења. Ипак као што је наведено и у поглављу у којем су анализирани Big data проблеми и ограничења, потребно је превазићи комплекс процеса прикупљања информација. Он има ограничење у хетерогености података која се огледа кроз отворене и затворене системе извора податка, широк спектар формата података и различит квалитет доступних података праћен различитом тачношћу прикупљања података. Постоји више проблема у подацима који компликују процес акумулирања знања, као што је губитак података из различитих извора праћен широким спектром и структуром формата (као што су текст, слике или видео записи), квалитет и тачност података. У том погледу, долази се опет до закључка да без обзира да ли се у концепту паметног града користиле методологије засноване на NoSQL базама података, машинско учење или неки други приступ, неопходно је вршити анализу система кроз 5V Big Data концепт.

Према дефиницији IBM, Big Data је начин рјешавања обраде или анализе великих обима информација које по њиховој неструктурираној природи се не могу анализирати у прихватљивом времену користећи традиционалне процесе и алате пословне интелигенције [68]. Дефиниција концепта Big Data, описана је кроз функције самих података, анализе података и презентације резултата анализе података. Сходно томе циљ Big Data је да из анализе добије само корисне информације и искуства која се могу примјенити у алгоритмима система, примјењујући та рјешења на проблеме доступности

података, протоку, великим количинама, структурираности, форматима и квалитету података. Док није било доступно толико количина дигиталних информација, подаци су долазили из традиционалних информационих система. Уз нове технике анализе, доступни су нови типови података који се могу анализирати као што су документи, фотографије, трансакциони подаци и сл. Ти интерни подаци, складиштени у некој структури морају имати могућност да се не једноставан начин укључују у заједничке анализе са подацима добијеним из јавних извора података или друштвених мрежа. Основни проблеми који се морају ријешити у предложеном концепту су: огромна сложеност интегрисања извора података, лош квалитет података, управљања генерисаним подацима у реалном времену, недостатак искуственог учења и избор праве архитектуре. У том контексту је потребно створити систем који ће масивне технике управљања анализом података адаптирати за технике анализе података великих количина са акцентом на оптимизацију перформанси времена извршавања операција.

2.7. АНАЛИЗА ПОСТОЈЕЋИХ РЈЕШЕЊА ПАМЕТНОГ ГРАДА

Постоје бројни пројекти и рјешења која се баве концептом паметног града. Анализираћемо нека која су креирана као подршка услугама оријентисаним на концепт паметног града. Већина њих користи технологије које су образложене у претходним поглављима. Описујући различите постојеће платформе, важно је напоменути да постоје платформе које користе само дио скупа информација и које су оријентисане на специфичне градове и окружења и оне системе који су опште намјене. Поред овог приступа, ова окружења се могу класификовати као платформе за јавни приступ подацима или платформе за посебне врсте корисника (најчешће комерцијалне), зависно од доступности коришћења услуга према специфичним категоријама корисника:

- Sentilo је дио архитектуре који издваја апликације које су развијене да искористе информације "генерисане од стране паметног града" коришћењем слоја сензора распоређеног широм града да прикупљају и емитују ове информације. Пројекат се реализује последњих 10 година. Ово рјешење користи град Барселона, Тераса и Реус. Креиран је као софтвер отвореног кода са властитим складишним системом. Дјелимично користи концепте Big data.
- Пројекат SmartSantader предлаже експерименталну урбанистичку платформу засновану на објектима. Подржане су различите апликације у оквиру овог екосистема и примјењују се типичне услуге паметног града. Пројекат је обухватио

Санандер, Београд, Гилдфорд и друге градове. Софтвер нуди бесплатан систем приступа информацијама за пројектанте који га користе и обезбеђује интеграцију података из различитих апликација за доношење одлука. Није базиран на концептима Big data.

- IBM интелигентни оперативни центар је приватна платформа у власништву компаније IBM заступљена у више паметних градова. Нуди окружење које пружа различите алате којим се интегришу подаци из различитих извора, а који су прилагођени специфичним захтјевима. Дјелимично користи концепте Big data.
- Пројекат CitySDK има за циљ да се обезбеди програмска структура за распоређивање система паметног града, који је тестиран у 8 градова у Европи: Амстердаму, Барселони, Хелсинкију, Истанбулу, Ламији, Лисабону, Манчестеру и Риму, а који укључују преко 5 приватних компанија у сарадњи са 5 универзитета. Они омогућавају интеграцију нових градова, али уз предефинисано коришћења апликација које се интегришу на овај систем. Дјелимично користи концепте Big data.
- Open Cities је платформа која омогућава корисницима да користе податке складиштене на њему, а који омогућавају пружање услуга у концепту паметног града. То је комерцијални систем генеричне природе (који није оријентисан на одређене градове). Није базиран на концептима Big data.
- i-SCOPE је платформа која пружа три врсте услуга : а) унапређивање укључивања и мобилности грађана са системима усмјеравања и сигнализацијом баријера у граду; б) оптимизовање потрошње енергије; в) еколошка контрола. Није базирана на концептима Big data.
- Reople је платформа која пружа услуге у контексту паметног града. Базирана је на принципу софтвера отвореног кода. Омогућава да заједница користи и дијели услуге које развија. Користе га Билбао, Бремен и Терми. То је јавни пројекат, али се може користити само у одређеним срединама. Дјелимично користи концепте Big data.
- IoT Open платформе су иницијативе које обезбеђује скуп библиотека, техничку документацију, веб услуге и протоколе за коришћење од стране цијеле развојне заједнице. Један такав пројекат је VITAL-OS Smart City Platform. Дјелимично користи концепте Big data.

- 3DCityDB је платформа развијена на принципу стандарда CityGML која прије свега пружа могућност проширења за специфичне задатке у оквиру концепта паметног града. Примјењује се у Берлину, Хамбургу, Франкфурту, Минхену, Дрездену, ХанOVERу, Постдаму, Лаипцигу, Лудвигсбургу, Гелсенкирхену, Кемптену, Салзбургу, Бечу, Сингапуру, Хелсинкију, Цириху, Ротердам, Хагу, Истанбулу, Познању и Токију. Није базиран на концептима Big data.
- Hexagon Smart City рјешења која се баве аутоматизацијом процеса и интеграцијом сензора у ова рјешења. Углавном развијена за специфичне задатке и није општег карактера. Дјелимично користи концепте Big data.
- SuperMap развија просторно – временску платформу за паметни град. Тренутно је у фази развоја иновативних технологија које интегришу велики број сектора и индустрија, обезбјеђујући комплексне податке и информације. Обавезно укључује просторне податке у свој систем. Дјелимично користи концепте Big data.

Може се закључити да постоје бројне платформе које користе концепт паметног града са услугама које су оријентисане према њему. Већина платформи су затвореног карактера иако омогућавају приступ подацима. Стога је неопходна општа и отворена платформа која укључује скуп услуга високог нивоа које корисници могу да искористе.

2.8.ГЕОСЕНЗОРСКЕ МРЕЖЕ И ЊИХОВА ИНТЕГРАЦИЈУ У ИНФОРМАЦИОНОМ СИСТЕМУ ПАМЕТНОГ ГРАДА

Геосензорске мреже представљају дистрибуиране системе који се састоје од геодетских и геотехничких сензора међусобно повезаних комуникационом мрежом. Посредством геосензорске мреже, уређаји (logger) прикупљају податке и достављају их кориснику. Према Вртунском [72] архитектуру геосензорских мрежа за мониторинг терена и објеката чине четири основне компоненте:

1. Аквизициона компонента – ову структуру чине уређаји (сензори) помоћу којих се врши прикупљање (аквизиција) геопросоторних података. Елементе ове компоненте чине све подкомпоненте система које обезбијеђују прикупљање неких геопросоторних података који се шаљу ка кориснику, односно ка систему. Зависно од специфичности захтјева мониторинга, врши се и избор одговарајућих сензора, те њиховог међусобног распореда у пројектовању геосензорске мреже за специфичан задатак. Према Вртунском у пројектовању мрежа праћења

деформација објеката препорука је да се призме постављају на објекте гдје се очекују највеће деформације, тј. на најкритичнијим тачкама, како се не би пратиле само деформације, већ и вршила и детекција лажних аларма, погрешних мјерења, неисправност сензора, те ширење деформација.

2. Комуникациона компонента – је дио структуре геосензорске мреже који треба да омогући комуникацију између аквизиционе компоненте и хардвера (рачунара који су дио система). Комуникациона компонента није униформног карактера, већ на њену конфигурацију утичу елементи који чине аквизициону компоненту, као и сама конфигурација терена који се осматра. Постоје различите могућности у погледу трансмисије података и то путем аналогних сигнала, дигиталних сигнала, кабловске везе и бежичне везе. Према Вртунском, важно је одговорити на захтјеве који су постављени пред комуникациону компоненту:

- комуникационе везе морају бити робусне,
- комуникационе везе морају обезбиједити двосмјерну комуникацију,
- треба свести на минимум кашњење преноса порука,
- треба постојати могућност накнадног додавања сензора у структуру,
- сви прекиди у комуникацији морају бити детектовани и евидентирани.

3. Процесорска компонента – представља дио инфраструктуре система који прима податке мјерења послате са аквизиционе компоненте путем комуникационе и процесира их. То процесирање подразумијева процесе складиштења података, обраду према дефинисаним условима и на крају визуелизацију. Структура ове компоненте може бити дистрибуирана, са напоменом да дијеловим хардверске компоненте са инсталираним софтвером могу бити постављени у близини сензора услед потребе за смањивањем растојања до сензора.

4. Дистрибутивна компонента – представља дио инфраструктуре који обрађен податак и финални производ дистрибуира ка крајњим корисницима. Елементе ове компоненте чине софтверска рјешења која су базирана на web технологијама.

Најчешће су базирана на OGC SWE спецификацијама услед постизања већег степена интероперабилности између различитих компоненти система. Геосензорску мрежу сачињавају: чворови, комуникациона мрежа која повезује чворове и управља протоком информација и вриједности параметара околине који се прате преко чворова мреже. Када

су у питању геопросторне компоненте овог типа мрежа у neighborhood-based моделу, дате су једино информације о сусједним чворовима, док код extended spatial модела сваки чвор посједује информације и о својој локацији. У том контексту се посматрају апсолутне локације (екстерни референтни системи) и релативне локације (локација у односу на локацију другог чвора). У оквиру компоненте аквизиције прате се промјенљиви феномени или промјенљивост мрежа. Три основне промјене које се идентификују су:

- Динамизам околине у смислу праћења геопросторних феномена са временском димензијом,
- Мобилност чворова у смислу могућности просторне мобилности чвора, и
- Промјенљивост чворова у смислу активације, деактивације и реактивације.

Сврсисходност ових система се првенствено и огледа у додавању временске димензије и праћењу одређених феномена кроз вријеме.

Активности у оквиру геосензорских мрежа се одвијају у складу са одређеним функционалностима које су описане алгоритмима. Они имају четири компоненте[72]:

- Ограничења које намеће окружење у којем се алгоритам извршава,
- Догађаји који се дешавају на чворовима,
- Акције које се спроводе као одговор на различите догађаје, и
- Стања чворова која омогућавају задржавање знања о претходним интеракцијама.

У оквиру система геосензорских мрежа чворови као основни елементи реагују на одређени тип догађаја који региструју, и то:

- Регистровање порука које су стигле са сусједног чвора,
- Trigger догађаји који представљају већ унапријед дефинисан образац када се читавају одређене вриједности, и
- Спонтани импулси генерисани изван система геосензорске мреже.

У структурама геосензорских мрежа посебна пажња се мора посветити такозваним deep learning алгоритмима помоћу којих чворови стичу ”искуство” и задржавају знање на основу претходних догађаја, првенствено како би могли да изводе различите акције за исте догађаје, зависно од осталих услова. У контексту елемената геосензорских мрежа

важно је идентификовати да нису сви чворови они на којима се врши мјерење. У том смислу разликујемо: сензоре (чворове који врше мјерења одређених природних феномена) и комуникационе чворове (служе за комуникацију и повезивање сензора и на њима се не врши аквизиција). У оквиру ове структуре комуникација се одвија тако што сензори комуникацијом чворовима шаљу информације мјерења, док комуникациони чворови одговарају сензорима које акције да изводе. У том контексту, зависно од типа сензора комуникација се обавља аналогно (напонски или струјни сигнали) и дигитално (низови битова записани у одређеном формату). Код геодетских сензора као што су ГНСС пријемници, роботизоване тоталне станице, дигитални нивелири и сл. комуникација је дигитална, док код геотехничких постоје оба типа веза. У контексту датог истраживања, важно је нагласити да комуникациони чворови који комуницирају са осталим чворовима у геосензорској мрежи преко разних модула повезују у јединствену структуру рутере који омогућавају бежичну комуникацију, рачунаре на којим се налазе софтвери за обраду и специфичне модуле као што су data logger. Они раде на принципима аналогног сигнала и треба да обезбиједи конверзију у дигитални, меморишу вриједности према одређеним интервалима и шаљу их даље комуникационим чворовима.

3. ДЕФИНИЦИЈА И ПРИМЈЕНА КОНЦЕПТА BIG DATA ТЕХНОЛОГИЈА

Убрзан развој науке и технологије доприноси развоју нових рјешења у различитим областима људског дјеловања. Све те области захтијевају неке системе управљања који су засновани на разним рачунарским платформама које обезбјеђују њихову функционалност. Њихов основни захтјев темељи се на осавремењавању њиховог дјеловања те константном проширивању њихових могућности. Велики и комплексни сетови података, код којих традиционалне апликације за обраду података нису примјенљиве означава основни појам Big data. Те скупове података карактеришу разноврсност формата, велике брзине обраде и приступа, и велики обим информација. Основне активности укључују пројектовање и реализацију инфраструктуре и сервиса за складиштење великих количина података, њихову претрагу, анализу, дијелење и визуелизацију. Термин Big data често се односи на употребу предикативне аналитике или других напредних метода за издвајање вриједности из података, а не само на одређену величину скупа података.

Концепт Big data карактерише прелазак са релационих на нерелационе базе података. То су на примјер: Google Bigtable и Amazon Dynamo. Једно од рјешења за инфраструктуру Big data је Apache Hadoop, софтверско рјешење овореног кода. Модели Big data пружају могућност обраде података у реалном времену, а претрага се врши коришћењем Map reduce алгорита. На примјер, резултати претраге у Google претраживачу добијају се у милисекундама управо захваљујући овим технологијама.

Опсег онога што се данас посматра као Big data је широк, а дефиниције су нејасне чак и контрадикторне. Најшире прихваћена дефиниција термина Big data проистекла је из анализе Meta Grupe која је рађена 2001. године и по тој дефиницији под појмом Big data се подразумијева информациони ресурс велике количине, велике брзине и велике разноврсности података који захтијева нове и иновативне методе обраде и оптимизације информација, побољшање увида у садржај података и доношења одлука [73]. Ово је дефиниција 3V димензије Volume, Variety, Velocity (количина, разноврсност, брзина) гдје количина података иде и до реда петабајта, а разноврсност података који се обрађују у реалном времену је широка (структурирани и неструктурирани подаци као што су текстуални и мултимедијални подаци). Неки аутори дефиницију Big data проширују на још једну димензију 4V veracity - истинитост (поузданост, предвидљивост, тачност података), а неки и на 6V гдје додају димензије viscosity - дјелотворност и value –

вриједност [74]. Дефиниција коју је потребно издвојити, која је покушај да се обједини више дефиниција – „Big data“ представља појам који означава смјештање и анализу великих, комплексних структура података коришћењем низа техника, укључујући, али не и ограничавајући се на технологије NoSQL, MapReduce и машинског учења [75]. Иза дефиниција не крију се само проблеми обраде веће количине расположивих података, као што сам термин Big data може да асоцира. Оне настају као резултат промјена у информационам технологијама и подразумевају и унапређење аналитичких капацитета свих података који би се користили у пословним процесима одлучивања.

Концепт Big data не представља јединствену технологију, већ комбинацију нових и старих технологија које помажу да се стекне дјелотворан увид у обрађене податке. Представља могућност управљања великим количинама различитих података разумном брзином и у одговарајућем временском оквиру да би се омогућила анализа тих података у реалном времену. Као што је раније напоменуто, за Big data карактеристични су:

- Количина – Много фактора доприноси увећању обима података (транзакциони подаци складиштени годинама, текстуални подаци који константно надолaze са друштвених мрежа, итд.). У прошлости је прекомјерна количина података стварала проблеме око складиштења, али са данашњим цијенама меморијских уређаја то више не представља проблем. Ипак, јављају се и други проблеми, укључујући одређивање важности одређених података у великим количинама доступних података.
- Брзина – Брзина обраде података представља двије ствари. Прва је брзина производње и генерисања података, а друга је брзина којом подаци морају бити обрађени да би задовољили одређене критеријуме. Правовремено реаговање и брза обрада података представљају велики изазов и за највеће компаније на свијету.
- Разноврсност – Данас се подаци налазе у великом броју различитих формата. Постоје стандардни системи управљања базама података, текстуални фајлови, e-mail, видео, аудио, геопросторни подаци и сл. Према неким процјенама око 80 % података није нумеричког типа, али они и даље морају бити укључени у процедуре анализе и доношења одлука у вези са њима.

Када говоримо о карактеристикама, битно је напоменути још двије битне димензије:

- Промјенљивост – Као додаток великим количинама и брзинама обраде података, ток података може постати прилично неправилан са временом. То се може објаснити неком популарном појавом у средствима јавног информисања, где се један исти податак понавља небројено пута. Овакви изузеци јако су тешки за обраду, поготову када се узме у обзир раст популарности друштвених мрежа.
- Сложеност – Када се бавимо великим количинама података, они уобичајено долазе из различитих извора. У великом броју случајева ризично је упаривати, прочишћавати и трансформисати те податке на било који начин. Ипак, неопходно је извршити повезивање односа између података и хијерархија података, јер у супротном адекватно управљање великим количинама података не можемо исконтролисати.

Подаци се добијају из различитих извора и налазе се у различитим облицима. С „експлозијом“ развоја сензора, паметних уређаја и друштвених мрежа подаци су постали сложени првенствено зато што сада не укључују само традиционалне структуриране податке, већ и неструктуриране или полуструктуриране податке. Када говоримо о структурираним подацима, ту првенствено мислимо да они описују податке који су груписани у релационе шеме (гдје су редови и колоне структурирани у оквиру стандардних база података). Организација ових података даје могућност извршавања једноставних упита који могу вратити употребљиве повратне информације. Под полуструктурираним подацима сматра се да они представљају податке за које се не може рећи да су груписани у неку фиксирану шему. Подаци су често нераздвојиви и садрже ознаке које помажу при хијерархијском организовању оваквих података. Неструктурирани подаци, углавном су подаци које је тешко убацити у релационе табеле база података ради анализе или извршавања упита над њима. Подаци оваквог типа представљају слике, аудио и видео фајлове.

3.1. ТЕХНОЛОГИЈЕ BIG DATA

Почетак примјене Big data технологије, прије десетак година, везује се за компаније као што су Google и Yahoo, које су имале велике количине података генерисане на Интернету, да би 2008. године Hadoop постао Apache пројекат отвореног кода највишег нивоа, што је дало пресудан утицај за развој ове технологије. Hadoop и MapReduce технологије осмишљене су како би се ефикасно могле обрадити велике количине

података коришћењем више рачунара везаних у кластер. Технологија MapReduce заснива се на окружењу у којем се проток података разбија у двије фазе: фазу мапирања и фазу редукције. У фази мапирања, групе података се обрађују изоловано посебним процесима који се означавају као „мапирање“, гдје се појединачни елементи дијеле у уређене парове, тј. кључ - вриједност парове. Резултат рада мапирања прослијеђује се процесима редукције, који комбинују ове уређене парове у мање скупове што представља, коначне резултате. За потребе складиштења података, подршку овој технологији базираној на кластеру рачунара, пружа Hadoop дистрибуирани фајл систем (Hadoop Distributed File System - HDFS) [3].

Подаци у Hadoop кластеру дијеле се у мање дијелове (тзв. блокове) и дистрибуирају се кроз кластер, те се на тај начин омогућава да се процеси мапирања и редукције раде на мањим количинама података, чиме се значајно добија на скалабилности. Сама идеја била је да се користе сервери у кластеру са јефтиним интерним дисковима, који ако се користе појединачно, представљају проблем због брзине и доступности, односно губитка података. Циљ је да се MapReduce процеси извршавају на серверима гдје се налазе подаци који се обрађују, тзв. принцип локалних података. Како HDFS приликом смијештања податка, дијели на мање блокове, копије ових блокова смјештају се и замјењују на различите сервере у оквиру кластера (минимално два сервера). На такав начин се добија већа доступност података и обрада на свим серверима. Цјелокупна логика управљања подацима обавља се на серверу који се означава као NameNode. Овај сервер је као нека врста сервера с мета подацима у којима се држе информације о фајловима, односно блоковима података на Hadoop дистрибуираном фајл систему. Све информације које има NameNode сервер држе се у меморији што омогућава брзу манипулацију с подацима који су на фајл систему или се уписују на њега. Са друге стране овај приступ, представља критичну тачку прекида, јер би се губитком метаподатака изгубиле све информације о подацима у кластеру. Осим што овај сервер треба да биуде робуснији на прекиде од сервера у кластеру, потребно је обезбиједити његов backup (резервну копију). Од верзије Hadoop 0.22, имплементирана је Backup Node функционалност која обезбјеђује одржавање ажурне копије NameNode у меморији. Пет елемената чини Big data технологију:

- Аналитичка база података;
- Меморијски систем управљања базама података (In-memory database management system);
- Обрада токова података у реалном времену;
- Hadoop; и
- Софтвер и хардвер на ком се систем смјешта.

Као подршка за ову технологију у оквиру Apache пројекта развија се више пројеката отвореног кода повезаних са Hadoop-ом: Ambari, Avro, HBase, Cassandra, Chukwa, Mahout, Hive, Pig, Spark и други. Ambari је Apache пројекат који пружа подршку за једноставније управљање (креирање, покретање, надгледање, конфигурирање) Hadoop кластером кроз web базиран интерфејс. Avro је систем за серијализацију података. Cassandra је Hadoop дистрибуирана база података (database system) отвореног кода који користе eBay, Twiter, Esri и много других компанија чија је карактеристика да имају велике количине активних података. Највећи познати Cassandra кластер има преко 300 TB података на више од 400 сервера [5]. HBase је, такође, Hadoop дистрибуирана, скалабилна база за смјештање великих количина података. HBase и Cassandra су врста „NoSQL“ дистрибуираних база података. Појам „NoSQL“ означава да база података није релациона база која подржава SQL (structured query language) као свој примарни језик. Chukwa је систем прикупљања података за праћење великих дистрибуираних система. Mahout пројекат има за циљ изградњу скалабилне библиотеке машинског учења. Hive омогућава упите и управљање великим скуповима података који се налазе у дистрибуираном систему. Pig је, заједно са инфраструктуром која то подржава, платформа за анализе великих количина података. Apache Spark је окружење за брзу обраду података, док Impala, окружење за паралелну обраду података заобилази MapReduce како би се обезбиједила могућност упита у реалном времену, а самим тим омогућава и аналитику, односно приступ алатима пословне интелигенције над подацима смјештеним у Hadoop у реалном времену.

3.2. ДОСАДАШЊА ИСТРАЖИВАЊА У ПРИМЈЕНИ КОНЦЕПТА BIG DATA ТЕХНОЛОГИЈА У РАДУ СА ПРОСТОРНО – ВРЕМЕНСКИМ ПОДАЦИМА И ПАМЕТНИМ ГРАДОВИМА

Концепт паметног града покрива различите природне феномене које укључује у процес одлучивања како би добио одговарајуће повратне информације. Програмом Уједињени паметни градови подстакнутом од стране UNECE⁷ и других партнера, кључне области које је потребно анализирати у контексту паметних градова су: урбана мобилност, одрживи живот, чиста енергија, управљање отпадом и ICT [18]. У оквиру свог истраживања Lim предлаже пет параметара које би требало размотрити у претварању градова у паметне, користећи концепт Big data:

- i. положај постојећих услуга у систему вођења и коришћења података;
- ii. искуство корисника у вези са подацима и информацијама;
- iii. оријентација података у дизајнирању апликација према услугама које треба да пружа;
- iv. синергије и конфликти између заинтересованих страна везаних за податке; и
- v. интеграција различитих перспектива података и апликација[76].

Развој концепта Интернета ствари (Internet of Things - IoT), развој паметних телефона, рачунара и других уређаја мобилне комуникације довео је до великих проблема са подацима и створио проблеме у погледу брзине, структуре, цијене, вриједности, безбједности, приватности и интероперабилности података. Традиционалне методе нису довољно ефикасне када се суочавају са Big data проблемима, због њиховог недостатка у погледу скалабилности и ефикасности. Сви ти проблеми у контексту паметног града могу се сагледати кроз 5V проблеме. У табели 4 је дат преглед досадашњих истраживања у области успостављања концепта паметног града у Big Data парадигми. Из наведеног може се уочити да не постоји јединствено рјешење у погледу 5V које је засновано на Big Data парадигми, а које је успостављено према концепту паметног града.

⁷ United Nations Economic Commission for Europe (ECE или UNECE) је једна од пет регионалних комисија под јурисдикцијом [United Nations Economic and Social Council](#). Успостављена је да промовише економску интеграцију и сарадњу између земаља чланица ([member states](#)).

ТАБЕЛА 4. Досадашња истраживања у области Big Data Smart City концепта према 5V

ПРОБЛЕМ	ОПИС	ИСТРАЖИВАЊЕ У КОНТЕКСТУ BIG DATA SMART CITY	РЈЕШЕЊЕ У КОНТЕКСТУ BIG DATA SMART CITY ЗА НАВЕДЕНИ ПРОБЛЕМ	ОГРАНИЧЕЊЕ У КОНТЕКСТУ BIG DATA SMART CITY	ИМПЛЕМЕНТАЦИЈА НА АРАСНЕ SPARK ОКРУЖЕЊУ	РАД СА ПРОСТОРНО-ВРЕМЕНСКИМ ПОДАЦИМА
Проблеми са количинама података (volume)	Анализе података у контексту Big data спроводе се над великим сетовима података за које креирање напредних аналитичких алгоритама захтјева велике ресурсе и процесну моћ. Стандардне технике извођења упита над гопросторним подацима, утичу значајно на перформансе система. Стандардни релациони системи управљања базама података имају проблем у погледу скалабилности.	Zhou и други [76]	Проширено је Apache Hadoop окружење за просторне компоненте како би се подигле перформансе просторне аналитике.	Нема могућност учитавања и рада са комплексним геометријама.	ДА	НЕ
		Boehm и други [77]	Модул Apache Spark је коришћен за извођење паралелизације процеса над сложеним типовима просторних података.	Могућност учитавања и процесирања облака тачака само над основним класама.	ДА	НЕ
		Yu и други [78]	Apache Spark као Big data окружење повезан је са релационим	Користи Big data окружење само за дио аналитичких процеса. Нема	ДА	НЕ

	системима за управљање базама података у процесирању геопросторних података.	могућност рада са комплексним геометријама.		
Bibri и други [12]	Коришћењем GeoSpark библиотеке, стандардни облик Resilient Distributed Dataset - RDD, проширен је у просторни RDD (Spatial Resilient Distributed Dataset - SRDD) образац. Формиран је метод за индексирање и чување података.	Нема могућност учитавања облака тачака и рада са тим скупом података.	ДА	НЕ
Liu и други [79]	Библиотека GeoMesa обезбеђује просторно-временско индексирање приликом процеса складиштења података основних геометријских примитива. Овај систем омогућава креирање Spark RDD и DataFrame-ова података ускладиштених у GeoMesa и серијализацију једноставних функција коришћењем алата	Нема могућност учитавања и рада са комплексним геометријама.	ДА	ДА

		Кгую. Тренутно је подржана са четири главна система база података заснованих на Hadoop: Akumulo, Apache, HBase и Google Cloud BigTable.			
	Kang и други [80]	Код рада са облаком тачака предложено је рјешење Apache Accumulo и Apache HBase складишта која пружају подршку за out-of-the-box дистрибуиране key-value изворе података. За индексирање података у жељено складиште кључних вриједности, користи вишеструке координатне линије за попуњавање простора (Space Filling Curves - SFC).	Нема могућност раслојавања облака тачака у структуре класа које су описане у складу са CityGML стандардом.	ДА	НЕ
	Voehn и други [81]	Развили су метод за убацивање облака тачака у структурама података на Apache Spark окружењу.	Нема могућност учитавања и раслојавања облака тачака у структуре класа које су описане у складу са CityGML стандардом.	ДА	НЕ
					НЕ
	Alis и други [82]	Извршили су индексирање облака	Нема могућност учитавања и	ДА	НЕ

			тачака засновано на принципима SFC.	раслојавања облака тачака у структуре класа које су описане у складу са CityGML стандардом.		
		Dean и други [83]	Развили су једноставан метод за процесирање на великим кластерима.	Нема могућност учитавања и рада са комплексним геометријама.	ДА	НЕ
						НЕ
		IQUMULUS [84]	Развијена је фузиона и аналитичка платформа за покривеност и волуметријске скупове облака тачака коришћењем Apache Spark технологије.	Нема могућност учитавања и раслојавања облака тачака у структуре класа које су описане у складу са CityGML стандардом.	ДА	НЕ
Проблеми са разноликошћу података (variety)	У контексту паметног града, подаци се генеришу из више извора и структурирају се у различитим форматима, који у односу на предложене стандарде могу бити у структурираној, неструктурираној и полуструктурираној форми. Складиштење и обрађивање таквих података није могуће коришћењем конвенционалних софтвера.	Kim и други [24]	Kim и остали су предложили метод за аутоматско издвајање IndoorGML података из CityGML Lod4 скупа података користећи спољне референце из инстанце IndoorGML до објекта у CityGML подацима.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	НЕ

У контексту овог истраживања упоредићемо CityGML Lod4 и IndoorGML модел. Као резултат овог поређења CityGML је објектни модел, уз подржану визуализацију, изврсне перформансе за анализу геометрије, лоше перформансе за анализу рута и средње перформансе за анализу контекста трака. IndoorGML је ћелијски просторни модел, са лошијим перформансама визуализације, лошим перформансама геометријске анализе и одличним перформансама анализе руте и контекста трака. Према овим поређењима, оба стандарда имају неке предности.

Claridades и други [85]

Claridades и други предлажу приступ интегрисаног IndoorGML са Indoor Point of Interest подацима проширивањем IndoorGML појма о

Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.

НЕ

НЕ

		простору са тополошким односима.			
Hashem и други [6]		Предлаже разматрање креирања сложених формата података у контексту паметног града како би омогућили складиштење података у структурираној, неструктуриран и полуструктурираној форми.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	НЕ
OGC Sensor Web Enablement Initiative [55]		Предлажу укључивање технологија стандардизованих према OGC Sensor Web Enablement Initiative (SWE) у концепт паметног града.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	НЕ
Sundaram и други [86]		Концепт Internet of Things (IoT) техника у контексту паметног града омогућавају прикупљање великих количина података коришћењем великог броја различитих сензора	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	НЕ
Chaturvedia и други [87]		Предлаже комбинацију отворених стандарда (API) који олакшава испоруку	Рјешење није базирано на Big Data концептима, већ користи стандардне	НЕ	ДА

		геопросторних функција и мјерења са сензора на кохерентан начин, па самим тим подржава интероперабилне и мрежне услуге унутар паметних градова.	релационе системе.		
	Huang и други [88]	Дефинисао је да стандардни CityGML модел има ограничења у представљању својства зависних од времена, а која су уједно и динамичка својства. Поменути модел омогућава складиштење статичких вриједности.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	НЕ
	Gaur и други [89]	Користили су Apache Spark као алат за обраду сетова податка на акцелерометерима паметних телефона.	Коришћен као окружење само за обраду дијела података.	ДА	ДА
	Saraswathi и други [90]	Користио је Apache Spark као велики систем за праћење података у реалном времену како би предвидјели укупан број протока података у разним каналима како би се смањило губљење података у саобраћају.	Коришћен као окружење само за обраду дијела података.	ДА	ДА
	Freedman и други [91]	За концепт паметног	Рјешење није	НЕ	НЕ

				града сматра неопходним успостављање алата са streaming методама, а које су развијене да би се оствариле интеракције са многим Интернет и Веб услугама.	базирано на Big Data концептима, већ користи стандардне релационе системе.		
Проблеми брзином података (velocity)	са	Данас постоји неколико ДБМС система који се користе за складиштење података који структурирају паметне градове.	Вилјески и други [54]	Указују да већина доступних 3D градских модела садржи многе геометријске и тополошке грешке. Најчешће грешке су непотпуне површине, дупликати вертикалних елемената, површи које се међусобно сијеку итд. Поред тога, постоји проблем са складиштењем података прикупљених са различитих сензора. Све то доводи до успорене обраде и процесирања података.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	НЕ
			Уао и други [92]	Интеграција City Data, ускладиштен као CityGML.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе	НЕ	НЕ

			системе.		
	Нijazi и други [93]	Интеграција City Data, ускладиштен као CityGML.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	НЕ
	Zhu и други [94]	Код доступних система паметног града развојени су елементи визуелизације и система за складиштење података. Доводи до спорије презентације обрађених података.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	НЕ
	52 ° North [95]	Једно од рјешења за интеграцију сензора у јединствен систем чувања података обезбјеђује пројекат 52° North гдје постоји JavaScript библиотека која помаже визуелизацију семантичног модела града сачуваног као CityGML и OGC SOS сервис опажања са 52 ° North.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	ДА
	Ledoux и други [96]	Модел CityJSON је формат који на основу JSON кодирања нуди размјену 3D градских модела, који се такође	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе	НЕ	ДА

			називају дигиталне макете или дигитални близанци коришћењем тих стандарда.	системе.			
Проблеми са варијабилношћу података (variability)	са	Проблем скалабилности система.	Liu и други che Spark, заснован на MapReduce парадигми, уводи нову платформу за обраду на кластеру рачунара који ради на једном од кластера оперативних система. Apache Spark је осмишљен тако да одржава скалабилност и толеранцију за MapReduce алгоритме коришћењем cloud computing технологија.	Нема развијену структуру за учитавање просторно – временских података.	ДА	ДА	
Проблеми количином података (volume)	са	Да би ријешили овај проблем, традиционални СУБП је додао још централних јединица за обраду (или CPU) или више меморије систему за управљање базама података када би се требала подесити побољшања система у вертикалном смислу. Већина података долази у полуструктурираном	DABAMOS [97]	Омогућава укључивање различитих типова геодетске опреме која обезбеђује огромне количине перманентних података.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	ДА

<p>или неструктурираном формату из друштвених медија, аудио, видео записа, текстова и е-порука. Геопросторни подаци најчешће су структурирани као растерски или векторски тип података, те у последње вријеме као сложени тип, облак тачака. Big data технологије користе се за оптимизацију складишта података, обраду, упите и комплексне анализе за све типове интелигентних апликација и услуга.</p>	Engel и други [98]	<p>Пројекат OpenADMS Node је чвор преко кога се складиште огромне количине података мјерења са сензора у релациону базу података PostgreSQL у кратким временским интервалима уз дефинисане перформансе система.</p>	<p>Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.</p>	НЕ	ДА
	Bibri и други [12]	<p>Апликације паметних градова као што су Smart City Map, City Drop, Jobs – Nearby,</p>	<p>Рјешење није базирано на Big Data концептима, већ користи</p>	НЕ	ДА

	Hexagon apps нуде фузију аутоматизације и IoT објеката са могућностима аналитике у реалном времену.	стандардне релационе системе.		
InfluxDB [99]	Инфраструктура складиштења података за рјешења паметних градова, уважава различитости и динамичности извора података и узимања у обзир временских серија.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	НЕ	ДА
Druid [100]	Инфраструктура складиштења података за рјешења паметних градова, уважава различитости и динамичности извора података и узимања у обзир временских серија. Користи Big Data Apache Hadoop.	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	ДА	ДА
NodeRed [101], [102]	Node-RED је Big Data алат за повезивање хардвера, апликација и online сервиса која ради са великим сетовима података.	Има само обезбијеђен дјелимичан сервис заснован на Big Data окружењу.	НЕ	ДА
Kodali и други [102]	PubNub је Data Stream Network преко кога се подешава архитектура система у Big Data окружењу	Има само обезбијеђен дјелимичан сервис заснован на Big Data	НЕ	ДА

	за стриминг података.	окружењу. Нема подршку за просторне податке.		
Lin и други [103]	PubNub је Data Stream Network преко кога се подешава архитектура система у Big Data окружењу за стриминг података.	Има само обезбијеђен дјелимичан сервис заснован на Big Data окружењу. Нема подршку за просторне податке.	НЕ	ДА
Ammaг и други [68]	AWS IoT платформа која омогућава повезивање уређаја и интеракцију са њима када су изван мреже за управљање великим количинама података.	Има само обезбијеђен дјелимичан сервис заснован на Big Data окружењу. Нема подршку за просторне податке.	НЕ	ДА

3.3. ОКРУЖЕЊЕ АРАСНЕ SPARK

Окружење софтверског модела Apache Spark представља кластер генералне намјене рачунарског система апликација у Scala, Java и Python програмским језицима. Пружа подршку библиотекама за streaming, графичко процесуирање и машинско учење које су компатибилне са овим окружењем. Објављен је 2010. године и према досадашњим искуствима је један од најшире коришћених система са „језичком интеграцијом“ апликација. Представља најактивнији систем отвореног кода за Big data процесуирање [104]. Apache Spark подржава и Windows и UNIX – рачунарске системе (нпр. Linux, Mac OS). Apache Spark је подржан на Java 7+, Python 2.6+ и R 3.1+ библиотекама. У дисертацији је коришћен Apache Spark 2.4.8 са синтаксом везаном за верзију Scala 2.12. За успјешно компајлирање неопходно је користити компатибилну верзију у овом случају, Scala (2.12), а сходно томе и све библиотеке које она пружа. Apache Spark кластер модел даје опис кључних концепата процесирања на кластеру рачунара. Apache Spark може да буде покренут или самостално или преко неколико постојећих кластер менаџера:

- Amazon EC2: EC2 скрипте омогућавају покретање на кластеру у временском интервалу до 5 минута;
- Standalone Deploy Mode: најједноставнији начин за развој Apache Spark на приватном кластеру (коришћен у реализацији истраживања);
- Apache Mesos; и
- Hadoop YARN.

За потребе истраживања окружење је подешено на локалној мрежи. Apache Spark покреће програме до 100 пута брже него Apache Hadoop MapReduce у меморији или десет пута брже него на брзим дисковима рачунара [105], [106]. Има напредну логику која подржава циклични ток података и меморијско процесуирање. Apache Spark обезбјеђује функционално програмирање апликација слично другим актуелним системима, гдје се корисницима пружа могућност процесуирања кроз операције као што су: map, filter и reduce. Апликације користе функције у програмском окружењу и шаљу их на чвор кластера [107], [108]. Кодом апликације креира се еластично дистрибуирани подаци (Resilient Distributed Dataset – RDD), стрингови представљени линијама које се читају из HDFS фајла. Трансформишу се коришћењем filter функције како би обезбиједио

извршавање следећег RDD. Ово представља процедуру процесуирања података у Apache Spark окружењу у генеричком смислу. Еластично дистрибуирани подаци (RDD) отпорни су на грешке и такав систем може да поврати изгубљене податке коришћењем графичког RDD (поновним покретањем функција као што је filter за обнову платформе). Подаци могу бити експлицитно кеширани у меморији или на диску, како би се могло подржати ово итеративно извршавање апликације. Сваки RDD представља логички план да процесуира сет података, али Apache Spark чека да одређене излазне операције, као што је count, покрену процесуирање. Ово омогућава машини да изводи неке једноставне оптимизације, као што су каналске операције. Таквим приступом приликом процесуирања не мора да материјализује средишње линије процеса и да обрађује грешке. Иако је таква оптимизација изузетно успјешна, она има и својих ограничења. Машине не разумију структуру података у RDD (што је произвољно да ли је Java/Python објект) или семантику корисничких функција (која садржи произвољни код). Apache Spark омогућава интеграцију и коришћење великог броја библиотека и апликација. Један од начина управљањем пројектом који користи Apache Spark окружење је употребом рјешења GitHub складишта, за који има подршку. Apache Spark структуриран је у следећим пакетима [109]:

а) Инфраструктурних пројеката:

- Spark Job Server –интерфејс за управљање и REST interface for managing за слање сервиса помоћу Apache Spark на исти кластер употребом REST сервиса;
- SparkR – R оптимизован за Apache Spark;
- MLbase – Machine Learning библиотека за машинско учење;
- Apache Mesos – Кластер менаџмент систем који подржава рад Apache Spark;
- Tachyon – Меморијски оријентисан дистрибутивни систем који подржава процесуирање Apache Spark;
- Spark Cassandra Connector – Једноставно читавање Cassandra података у Apache Spark и Apache Spark SQL из Datastax репозиторијума;
- Elasticsearch – Apache Spark SQL проширење интерфејса;
- Spark-Scalding –транзициони каскадни код за Apache Spark;

- Zeppelin – ipython библиотека за Apache Spark. Ту су такође и Ispark и Spark Notebook; и
- IBM Platform Application Service Controller – Кластер менаџмент софтвер интегрисан са Apache Spark.

б) Апликације које користе Spark:

- Apache Mahout – претходно на Hadoop MapReduce, додао је Apache Spark као другу опцију;
- Apache MRQL – упитно процесуирање и оптимизација система за велике размјере дистрибуираних анализа података, изграђене на Apache Hadoop, Hata и Apache Spark;
- BlinkDB – систем за управљање базама података који омогућава масивне упите над машинама изграђене у Shark и Apache Spark;
- Spindle – Spark/Parquet-базиран на интернет аналитици упита;
- Thunderain – оквир за комбиновање stream процесуирања са старим подацима коришћењем алгоритама Lambda архитектуре; и
- DataFrame из Ayasdi Pandalas – имплементација оквира података за Apache Spark.

3.3.1. ПРЕТХОДНИ РЕЛАЦИОНИ СИСТЕМИ НА SPARK ПЛАТФОРМИ

Први покушај да се креира релациони интерфејс на Apache Spark је био Shark , који је модификовао Apache Hive систем да се покрене на Apache Spark окружењу. Овај систем је тежио да имплементира оптимизоване стандардне релационе системе за управљање базама података, као што је процесуирање по колонама, кроз Apache Spark окружење. Док је Shark показивао добре карактеристике и добре прилике за интеграцију са Apache Spark програмима, десиле су се три битне промјене. Прво, Shark је једино могао бити коришћен да се врши упит над екстерним подацима складиштеним у Hive каталогу и такав није био користан за релационе упите над подацима унутар Apache Spark програма (нпр. на грешке у креирању RDD-а од стране корисника). Једини начин да се позове Shark програм из Apache Spark програма је био да се стави заједно са SQL стрингом, који је непогодан и склон грешкама да ради у оквиру модуларног програма.

Hive оптимизатор је устројен за MapReduce и није погодан за проширења. Самим тим не препоручује се да гради нове карактеристике система као што су проширења за нове типове података за машинско учење или подршка за нове изворе података.

Apache Spark је меморијски дистрибуиран процесни алат. Не захтијева упаривање са Hadoop, али како је Hadoop један од најпопуларнијих Big data процесних алата, Apache Spark је прилагођен да ради добро у том окружењу [110]. На примјер Hadoop користи HDFS (Hadoop Distributed File System) да складишти своје податке, па је Apache Spark у могућности да чита податке из HDFS документа и да чува измјене у HDFS [6]. Apache Spark чува податке у меморији. Популарна складишта осим HDFS су и Hbase и Cassandra база података [111]. Једном уčitан у меморију, Apache Spark може покретати више трансформација на сету података за рјешавање постављеног проблема. Када се изврши процес, крајњи резултат пише се у трајно складиште. На овакав начин, ово рјешење представља алтернативу за Hadoop и може бити много бржи у одређеним ситуацијама. На примјер multi-pass map reduce операције могу бити значајно брже на Apache Spark него Hadoop map reduce пошто је import/export на Hadoop диску избјегнуто. Apache Spark може да чита податке форматиране за Apache Hive [112], па Apache Spark SQL може бити значајно бржи него да се користи HQL (Hive Query Language). Систем за управљање базама података Cassandra има свој изворни упитни језик назван CQL (Cassandra Query Language), што представља проширење за SQL, али је прилично слаба за операције као што су агрегација или ад хок упити. Када је Apache Spark упарен са Cassandra, нуди много богатији упитни језик и пружа могућност да ради разне анализе над подацима који изворни CQL не може да обезбиједи. Такође, Apache Spark се користи за stream процесуирање. Apache Spark може бити подешен да узима долазеће податке у реалном времену и процесуира их у микро серије и онда сачува као резултат сталног складиштења, као што је HDFS, Cassandra, итд. Стога Apache Spark је самосталан у меморији система и може бити упарен са више различитих дистрибуираних база података и системских фајлова да дода карактеристике. Нуди много комплетнију SQL имплементацију.

3.3.2. УПИТНИ ЈЕЗИК SPARK SQL

Модел Apache Spark SQL⁸ и Apache Spark интерфејс омогућавају рад са структурираним и полуструктурираним подацима. Структурирани подаци су сви подаци

⁸ Spark SQL је нова и брзо напредујућа компонента Apache Spark. Представља окружење Apache Spark за вршење упита.

који имају шему то јест, познат скуп поља за сваки запис. Када имамо ову врсту података, Apache Spark SQL чини их лакшим и ефикаснијим за учитавање и вршење упита. Посебно, Apache Spark SQL обезбеђује три главне могућности:

- Може учитати податке из различитих структурираних извора (нпр. JSON, Hive и Parquet).
- Омогућава вршење упита над подацима користећи SQL, како унутар Apache Spark програма тако и спољним алатима који повезују Apache Spark SQL кроз стандардне конекторе базе података (JDBC/ODBC).
- Када се користи у оквиру програма написаних на Apache Spark, Apache Spark SQL обезбеђује интеграцију између SQL и стандардног Python/Java/Scala кода, укључујући и способност да се придруже RDD и SQL табеле, прикажу прилагођене функције у SQL, и још много тога. Ово рјешење представља у пракси најпримјењенији начин рада, који је и коришћен у реализацији овог истраживања о чему ће више бити речено у дијелу који си односи на имплементацију GAMINESS модела.

За реализацију ових могућности, Apache Spark SQL даје посебну врсту RDD под називом шема RDD. Шема RDD је скуп RDD редова објеката, гдје сваки представља одговарајућу вриједност. Шема RDD интегрише шему својих редова. Док шема RDD изгледа као стандардни RDD елемент, интерно она чува податке на ефикаснији начин, користећи своју интерну шему објекта. Поред тога, пружа нове операције које нису доступне на RDD, што представља могућност за покретање SQL упита. Шеме RDD могу бити креиране из екстерних извора података, од резултата упита, или и из стандардних RDD [104], [113]. У практичним примјерима дисертације приказана је употреба Шема RDD у стандардним Apache Spark програмима за учитавање и вршење упита над структурираним подацима. У раду ће бити описан Apache Spark SQL сервер са JDBC конектором, који омогућава покретање Apache Spark SQL-а на заједничком серверу и повезивање кластера. У методама које су кориштене приказују се SQL упити између RDD и стандардних табела у релационим системима. Модел Spark SQL је новија компонента Apache Spark који је еволуирао у стандардни пакет Apache Spark. У оквиру стандардног модела имплементирана је документација за најновије информације о Apache Spark SQL и шеми RDD.

3.3.3. СИСТЕМИ ЗА СКЛАДИШТЕЊЕ ПОДАТАКА ЗАСНОВАНИ НА BIG DATA ПАРАДИГМИ

Као што је и наведено у претходном поглављу, постоје различити Big Data прилагођени облици читавања података у тражене структуре. Учитани подаци и извршени упити враћају шему RDD. Шема RDD је слична табелама у стандардним базама података и представља одговарајући оквир за манипулацију подацима. Шема RDD је RDD састављен од редова објеката са додатним информацијама шеме о типовима података у свакој колони. У новијим верзијама Apache Spark, име шема RDD је промјењено у DataFrame, али је дата напомена због схватања концепта модела података у Big Data структури. Шема RDD представља стандардни RDD, тако да је могуће радити са њима користећи постојеће RDD трансформације као што су `map()` и `filter()`. Једна од важнијих одлика оваквих система је пријавити одговарајући DataFrame као привремену табелу⁹ на којој је могуће вршити упите преко `HiveContext.sql` или `SQLContext.sql` оператора. То се извршава коришћењем `SchemaRDD registerTempTable()` методе. DataFrame може сачувати неколико основних типова података, као и структуре и редове ових типова. (Табела 5). Последњи тип података, структуре, се приказује као други ред у Spark SQL-у. Сви ови типови могу бити угниједени један у други, што представља један од предмета истраживања.

ТАБЕЛА 5. Стандардни типови података у оквиру RDD шеме

SparkSQL/HiveQL тип	Scala тип	Java тип	Python тип
TINYINT	Byte	Byte/byte	int/long
SMALLINT	Short	Short/short	int/long
INT	Int	Int/int	int/long
BIGINT	Long	Long/long	Long
FLOAT	Float	Float/float	float
DOUBLE	Double	Double/double	float
DECIMAL	Scala math Big Decimal	Java math Big Decimal	Decimal/ Decimal
STRING	String	String	String
BINARY	Array[Byte]	byte[]	bytearray
BOOLEAN	Boolean	Boolean/boolean	Bool
TIMESTAMP	java.sql.TimeStamp	java.sql.TimeStamp	datetime. datetime
ARRAY<DATA_TYPE>	Seq	List	list, tuple, array
MAP<KEY_TYP,VAL_TYPE>	Map	Map	dict
STRUCT<COL1:COL1 TYPE,...>	Row	Row	Row

⁹ Привремене табеле (Temp Table) су локалне табеле које се користе путем HiveContext или SQLContext и нестају из меморије по завршетку апликативне процедуре.

У контексту DataFrame објекти редова представљају евиденције у тој наведеној шеми RDD-ова, фиксних вриједности низова поља. У Scala/Java програмском окружењу, објекти реда дефинисани помоћу Apache Spark Scala језика, имају велики број добијених функција (getter) да обезбиједи вриједност сваком пољу датог индекса. Стандардни getter, добија и узима број колоне и враћа тип објекта који је одговоран за промјену Boolean, Byte, Double, Float, Int, Long, Short и String типа, гдје постоји getType() метода, која враћа тај тип. Метод, get String(0) ће вратити поље 0 као стринг. Пошто су познати типови сваке колоне, Apache Spark је у могућности да ефикасније складишти податке. Када се кешира табела у Apache Spark SQL окружењу, путем SQLContext.sql оператора, онда се подаци постављају у меморији, у структури колоне. Ове кеширане табеле остају у меморији само током трајања "driver" програма, па прекидом програма, за следеће активности, поново се морају кеширати подаци из кеш меморије. Као и код RDD, кеширају се табеле када се очекује да буде покренуто више задатака или упита у односу на исте податке. Могуће је кеширати табеле коришћењем HiveQL/SQL израза. За кеширање или одкеширање табеле користе се CACHE TABLE tableName или UNCACHE TABLE tableName. Ово се најчешће користи са апликацијама које су повезане на друге системе за управљање базама података преко JDBC сервиса. Кеширан DataFrame у Apache Spark апликацијама посматра се као и сваки други RDD.

Apache Spark SQL подржава велики број података структурираних извора, омогућавајући да из њих добијемо објекте реда без икаквог компликованог процеса читавања. У табели 6, даћемо напомене за најчешће коришћене изворе података као што су: HDFS, Hive табеле, JSON и Parquet фајлови. Вршење упита над овим изворима података, коришћењем Apache SQL, подразумијева само одабир подскупа поља. Модул Apache Spark SQL може да скенира подкуп података за дефинисана поља, умјесто да скенира све податке као на примјер Spark Context.hadoopFile [105]. Поред ових извора података, могуће је претворити стандардни RDD у програму у шема RDD додијеливањем шеме за учитани документ без обзира на његову изворну структурираност. То чини да се у овом окружењу онда могу једноставно извршавати SQL упити, чак и када су основни подаци Python или Java објекти. Често, SQL упити су концизни кад процесуирамо веће количине података одједном.

ТАБЕЛА 6. Рад са дистрибуираним системима за складиштење заснованим на Apache Spark окружењу

Модел чувања података	Опис система	Могућност рада са просторно – временским серијама података	Ограничења у контексту просторно – временске серије података
HDFS	Hadoop Distributed File System (HDFS) је дистрибуирани системски документ. Дизајниран је да се може покретати на системима са лимитираним хардверским могућностима. Погодан је за апликације које раде са великим сетовима података. Ради са POSIX захтијевима како би омогућио стриминг приступ системским документима. Представља Apache Hadoop подпројекат [110]. Базиран је на master/slave архитектури. HDFS кластер је састављен од једног чвора, а мастер сервер управља фајл системом и регулише приступ подацима од стране клијената. HDFS је дизајниран да чува велике количине података на машинама у кластеру. Чува сваки документ као секвенцу блока, а сваки блок у засебан документ. Блокови докумената су реплицирани и високо осјетљиви RDD.	ДА	Постоје истраживања у којима је складиштен облак тачака у овој структури, али без развијених могућности анализа временске компоненте. Основно ограничење у постојећим рјешењима код учитавања комплексних типова геометрија.
APACHE HIVE	При учитавању податаке из Hive структуре, Spark SQL подржава било који Hive подржаван формат чувања: текстуалне фајлове, RC фајлове, ORC, Parquet, Avro, и Protocol Buffers [71]. Да повеже Apache Spark SQL са постојећом Hive инсталацијом, неопходно је обезбиједити Hive конфигурацију. Може се обезбиједити копирањем hive-site.xml фајла у Spark./conf/ директоријум. Уколико је потребно, локална Hive складишта метаподатака користе се ако hive-site.xml није подешен и могуће је једноставно учитати податке у Hive табелу и извршити упит над њима.	ДА	Досадашња истраживања обухватила рад само са простим геометријама.
PARQUET	Је изузетно популаран колона оријентисан формат за чување података који може да ефикасно чува податке са угнијеженим пољима. Често је коришћен алат у системима и подржава све типове података у Apache Spark SQL. Модул Apache Spark SQL обезбјеђује методе за читање података директно у и из Parquet	ДА	Постоје истраживања у којима је складиштен облак тачака у овој структури, али без развијених могућности анализа временске компоненте.

	<p>фајлова. За читавање података, могуће је користити HiveContext.parquetFile или SQLContext.parquet [72].</p>		
JSON	<p>Иако је JSON првобитно заснован на неструктурираном подскупу javascript језика за скриптовање¹⁰, најчешће је коришћен, иако представља језички независан формат. Код за рашчлањивање и генерисање JSON података лако је доступан за велики број програмских језика. Ако постоји JSON фајл са евиденцијом уклопљеном у исту шему, Apache Spark SQL може извести закључак о шеми скенирањем фајла и пуштањем приступа пољима по називу. Ако се догоди да имамо велики директоријум JSON евиденција, Apache Spark SQL шема закључује врло ефикасан пут за започињање рада са подацима без уписивања неког специфичног кода читавања. Да би читали JSON податке, све што је потребно је да се позове jsonFile() функција у hiveCtx. Модул Apache Spark DataFrames чини једноставним читање различитих формата података, укључујући JSON. Он аутоматски закључује JSON шему.</p>	ДА	<p>Погодан за рад у геосензорским мрежама, јер је већина произвођача стриминг засновала на овом формату.</p>

¹⁰ стандард ЕСМА-262 треће издање-децембар 1999. године

4. АРХИТЕКТУРА GAMINESS СИСТЕМА ПРОСТОРНО – ВРЕМЕНСКИХ ПОДАТАКА

У концепту паметних градова се користе бројне дигиталне технологије као што су Cloud Computing, Internet of Things или Open Data у циљу превазилажења традиционалног представљања и размјене геопросторних података. Овај концепт осигурава значајно повећање употребе података за успостављање нових услуга које доприносе бољем одрживом развоју и праћењу свих појава које се дешавају у урбаним подручјима. Употреба савремених геоинформационих технологија, попут сензора за прикупљање различитих геопросторних и сродних података, захтијева одговарајућу могућност складиштења за даљу анализу података. У истраживању је предложен нови систем за управљање просторно – временским подацима назван biG dAta sMart cItY maNagEment SyStem (GAMINESS), који је заснован на Apache Spark Big Data окружењу. Модел управљања системом GAMINESS заснован је на принципима моделовања Big Data података који се концептуално разликује у односу на стандарде релационе системе управљања просторно – временским подацима.

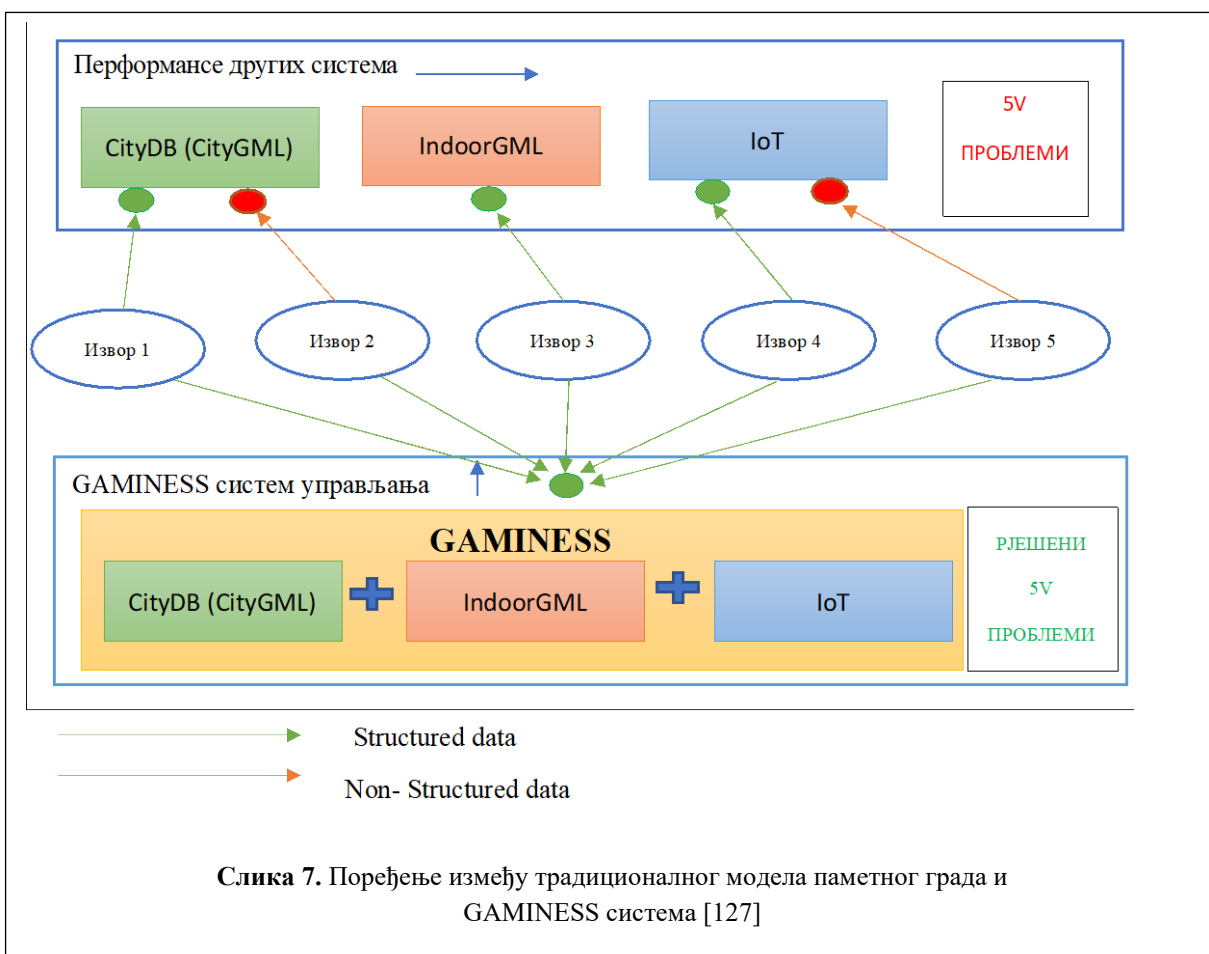
Овај приступ пружа могућност складиштења и управљања огромним количинама структурираних, полуструктурираних и неструктурираних података у реалном времену. Перформансе система се повећавају на виши ниво коришћењем паралелизације процеса објашњене кроз 5V принципе Big Data парадигме. Постојећа рјешења заснована на принципима 5V фокусирана су само на визуализацију података, а не на саме податке. Таква рјешења често су ограничена различитим механизмима складиштења и способношћу извођења сложених анализа на великим количинама података са очекиваним перформансама. Систем управљања GAMINESS треба да ова ограничења сведе на нижи ниво и омогући превазилажење ових недостатака, претварањем просторно -временских података прикупљених у паметном граду у одговарајућу структуру без додатних ограничења у вези са форматима података или коришћеним стандардом. Предложени модел садржи двије основне компоненте: геопросторну компоненту и геосензорску компоненту које су међусобно повезане. Оне су засноване на стандардима CityGML и SensorThings. У следећим поглављима, биће образложено проширење постојећих стандарда и његова интеграција у постојећи модел GAMINESS који пружа могућност размјене података без обзира на коришћени стандард или формат података у предложеном Apache Spark Data Framework шему.

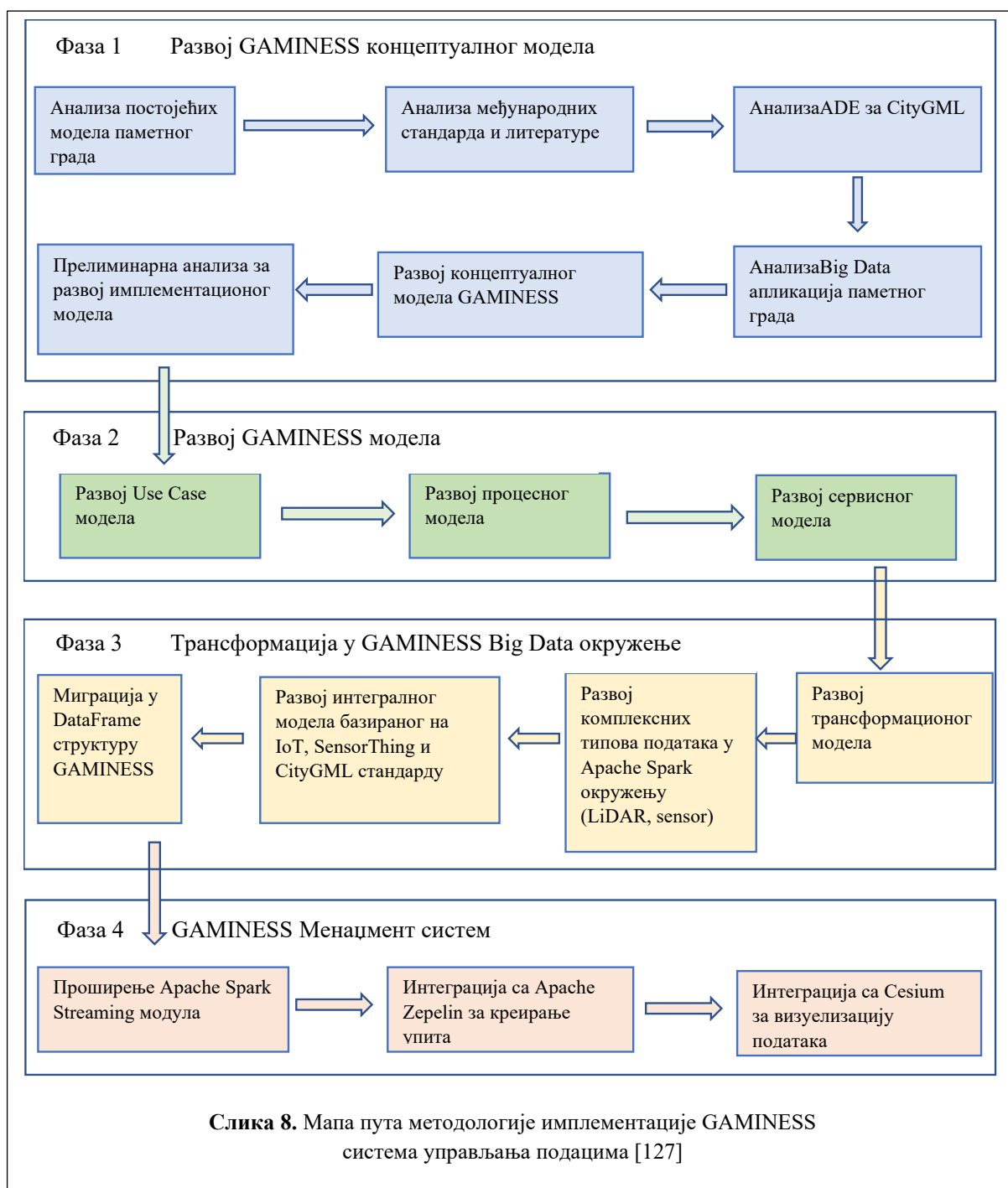
4.1. МЕТОДОЛОГИЈА ИЗРАДЕ GAMINESS СИСТЕМА

Технологије великих података не представљају јединствену технологију, већ одговарајућу комбинацију нових и старих технологија, које омогућавају управљање великим количинама различитих просторно - временских података у структурираном, полуструктурираном и неструктурираном облику. Као што је наведено, у дисертацији се разматра нови систем управљања просторно – временским подацима који треба да омогући превазилажење ограничења проузрокованих употребом различито структурираних података на лимитираном хардверском окружењу. На слици 7 приказан је концепт модела GAMINESS са рјешењима и елементима система на којим је базиран. У данашње вријеме гдје су доступни бројни подаци и гдје скоро сваки уређај представља својеврсан сензор за прикупљање података, врло важан аспект је њихова искористивост и могућност укључивања у систем доношења одлука. У том смислу је потребно пронаћи адекватно рјешење које ће моћи складиштити и структурирати те податке. Будући да је основна тема истраживања управљање просторно – временским подацима у контексту Big Data парадигме, важно је уочити проблеме који су идентификовани у контексту ових података. Као основни проблем који се намеће је проблем варијабилности гдје као што је и наведено постоје бројни извори података који на неки начин прикупљају перманентно геопросторне податке (мреже перманентних станица, класична терестичка мјерења са чешћим интервалима понављања мјерења, геосензорске мреже, па чак и паметни телефони као својеврсни сензори који пружају неку врсту геопросторних информација). У контексту паметног града и бољег управљања простором, неопходну основу треба да представља имплементациони ниво приказа структуре града који је материјализован приказом структуре терена, објеката, унутрашњости објеката и слично. У том контексту разматрани су најчешћи стандарди према којима се врши пресликавање геопросторних података у наведене структуре. Будући да су у раду разматрани подаци који се добијају различитим технологијама за аквизицију геопросторних података и да се разматрају осим основних растерских и векторских типова података и подаци заснованих на ласерском скенирању. У контексту коришћења свих ових типова података се разматрају стандарди који се најчешће користе за записивање ових података. Уколико разматрамо стандарде CityGML и IndoorGML за описивање простора, те Internet of Things као најприсутнији стандард и протокол по коме се врши размјена података са сензора у систем, долази се до потребе да управо на бази ових стандарда се изврши

реализација новог система који ће превазићи ограничења проузрокована њиховим коришћењем у конвенционалној обради података и релационим системима база податка.

Пројектовани систем карактерише обрада великих података, коришћењем кластерске мреже, која вертикално убрзава перформансе и побољшава укупне перформансе система. Систем GAMINESS је рјешење које пружа другачији приступ подршке за пет карактеристика Big Data у поређењу са најчешће коришћеним системима управљања базама података попут Postgresql и Oracle. Предложено рјешење за разлику од конвенционалних система има могућност учитавања података у један јединствен систем за разлику од стандардних релационих система што се може видјети на слици 7. Управо у овом смислу у традиционалним базама података постоји ограничење у погледу интегрисања неструктурираних података и њиховог укључивања у систем одлучивања. Ово је карактеристично за NoSQL базе података, које са друге стране имају ограничење у погледу стандардних формата геопросторних података. У овом контексту, коришћењем алгорита заснованих на Big Data обради података и њиховим интегрисањем у већ дефинисане моделе обраде геопросторних података, ствара се оквир GAMINESS система чија предност је уочљива у односу на стандардне системе.





На слици 8. је приказана мапа пута методологије којом је успостављен систем управљања просторно – временским подацима GAMINESS. Предложени модел, заснован на концептима Big Data парадигме, коришћењем алгоритама паралелизације процеса за дистрибуиране изворе даје значајну ефикасност над великим количинама ускладиштених података (петабајти и већи). Предност Big Data система огледа се првенствено у обради велике количине података, док се на мањој количини података та предност не примјећује, па стога нема посебне потребе за кориштењем. Дијелењем података путем кластера мрежних система, ови алгоритми обезбјеђују већу брзину

обrade, што ће детаљније бити објашњено у резултатима истраживања, где је извршена упоредна анализа истих количина података у окружењу GAMINESS и 3DCityDB Postgres / Oracle. Као што се може и видјети на Слици 8. у оквиру истраживања је направљена детаљна анализа постојећих модела паметних градова, са имплементацијом стандарда на којима су они успостављени. Посебан значај се даје CityGML стандарду који најдетаљније описује компоненте просторно – временских података у контексту геопросторних апликација паметних градова. У том смислу, посебна анализа је дата елементима проширења CityGML Application Domain Extension (ADE) који представљају елементе концептуалног проширења за специфичне случајеве. У контексту паметног града урађен је посебан осврт на постојећа рјешења паметних градова заснована на Big Data окружењу, како би се што боље установили постојећи недостаци и потенцијална рјешења која могу бити искоришћена за реализацију овог истраживања. На бази CityGML стандарда и резултатима анализе креира се GAMINESS концептуални модел. Овај модел је послужио као добра основа за имплементацију предложеног менаџмент система и тражење адекватног Big Data окружења на коме ће се модел развити. У контексту реализације развијени су дијаграми случајева употребе са описом процеса за потребе успостављања сервисно – оријентисане архитектуре. У оквиру истраживања развијен је модел трансформације који модел CityGML преводи у изабрано Big Data окружење. Да би преписивање било омогућено, развијени су комплексни типови података. Оваквим приступом нуде се рјешења проблема варијабилности, будући да су ови комплексни типови препознати у већини сличних апликација које раде са оваквим подацима. На имплементационом нивоу све класе предложеног модела се имплементирају као одговарајући DataFrames пакета. За потребе апликативног успостављања система у коме ће бити могућа интеграција са различитим изворима података који шаљу податке константно (сензори), неопходно је на бази изабраног Big Data окружења, проширити Apache Streaming модул. На основу приједлога дефинисане мапе пута приказане на Слици 8, визуелни ”query tool” се може реализовати коришћењем Apache Zepelin рјешења које у основи користи параметре Apache Spark SQL модела који је пандан SQL у релационим базама података. Дакле овим моделом се само формирају одговарајуће предефинисане функције које су доступне у оквиру система.

У системима за обраду података који се односе на паметне градове, према моделу DigitalTwins, подаци се појављују више пута са детаљима временске валидности. То доводи до сложеније обраде података. У систему управљања GAMINESS овај проблем

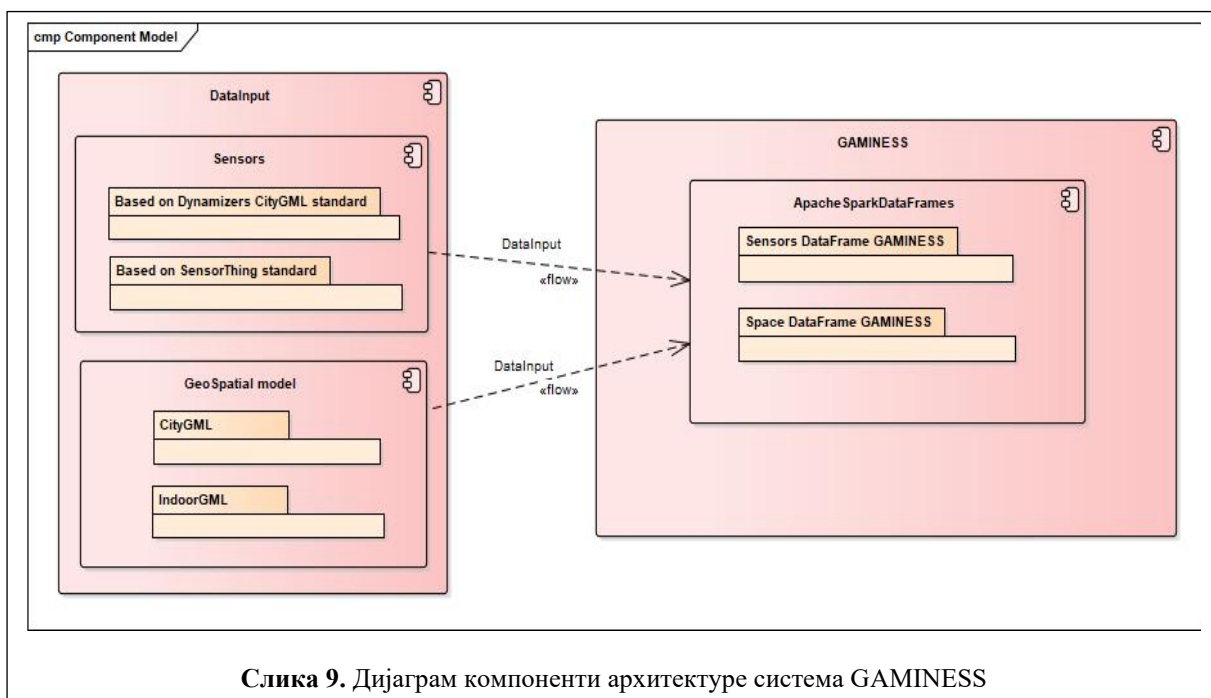
је рјешен примјеном *lifespan* атрибута, у складу са препорукама дефинисаним *Inspire* директивом. На такав начин је обезбјеђено верзионисање модела. Предложено рјешење дефинисано на слици 7, пружа могућност трансформација података из хетерогених извора података у предложену структуру, узимајући у обзир однос између података и хијерархије података у циљу контроле управљања тако великим количинама података. Ово раслојавање дефинисано је моделом података у којем постоји веза између података који су геопросторно одређени (облак тачака у комбинацији са сензорским подацима интегрисаним у модел града). У *GAMINESS* моделу, дате су двије цјелине које представљају модел учитавања геопросторних података и података прикупљених са различитих сензора који не прикупљају иницијално геопросторне податке. Дакле, приликом превођења података у предложену структуру, врши се раслојавање и слање дијела података тачно оном *DataFrame* за који се тај дио података односу. Тако, се врши дијељење комплетне структуре, стандардизација и аналитика према дефинисаним сегментима. У оквиру предложеног система дефинисани су нови кориснички дефинисани типови. Најтипичнији примјери су за класе облака тачака и интегралног пакета сензора. Типови су у потпуности засновани на *Big Data* концепту и написани у *Apache Spark Scala* програмском језику. На овакав начин је рјешен проблем сложености података.

4.2. КОНЦЕПТУАЛНИ МОДЕЛ GAMINESS СИСТЕМА

Концептуални модел система управљања *GAMINESS* дефинисан је као проширење постојећег *CityGML* модела. Овај стандард је детаљно описан кроз спецификације *CityGML* [92], [114], [115]. Концептуални модел структуриран је у два подмодела, геопросторни и геосензорски међусобно повезане. Геопросторни подмодел заснован на концепту модела *CityGML* стандарда, са свим преузетим елементима описа простора, има проширење за параметре унутрашње оријентације дефинисане стандардом *IndoorGML*. У основи има идентичну структуру до *LoD4* нивоа детаљности. Овај ниво детаљности је допуњен претходно споменути проширењем. Геосензорски подмодел, заснован је на концепту подмодела *Dinamizers CityGML* стандарда, који је проширен за параметре адаптације *IoT* и *SensorThings* стандарда. Проширење овог концепта, у великој мјери рјешава проблем варијабилности. Стандард *CityGML* је уобичајен информациони модел за 3D урбане објекте и пружа свеобухватан и детаљан приказ градских модела. Стандард је дефинисан као XML формат за складиштење објеката на основу ISO 19118 са основном функциом дијељења виртуелних 3D модела градова.

Предложена проширења дефинисана системом управљања GAMINESS су у складу са OGC (Geography Markup Language – GML3) и дефиницијом CityGML 3.0 концептуалног модела за складиштење и размјену 3D градских модела коју су представили Kutzner и др.[116]. Геометријски и тематски модели описани су кроз дефинисану структуру модела. За геопросторни подмодел који представља настанак 3D структуре града, општи приказ је могућ коришћењем уноса података без обзира на технологију прикупљања података (LiDAR, даљинска детекција, УАВ, ГНСС, класичне геодетске терестичке методе итд.). За разлику од класичних система за складиштење, GAMINESS пројектује могућност увоза података облака тачака подијелених у више датотека. Сходно томе, систем лоцира датотеке на кластерима и обезбеђује процес паралелизације. Подаци се могу учитату у формату LAS и delimited text layer (XYZ). На овај начин се рјешава проблем процесуирања великих количина података. На слици 9 приказан је модел компоненти система и главних пакета.

Други проблем који је рјешаван је проблем варијабилности на улазу. Рјешен је у геопросторном подмоделу. Читач података очитава облак тачака, трансформише га и нормализује улазне податке у геопросторну структуру израчунавањем одговарајућих Мортон кодова. За читање таквих података кориштени су резултати истраживања Пајић и други у коме су дефинисане процедуре и кориснички дефинисани типови података засновани на Apache Spark архитектури [117]. Подмодул Геосензор заснован је на моделу Dynamizers.



Слика 9. Дијаграм компоненти архитектуре система GAMINESS

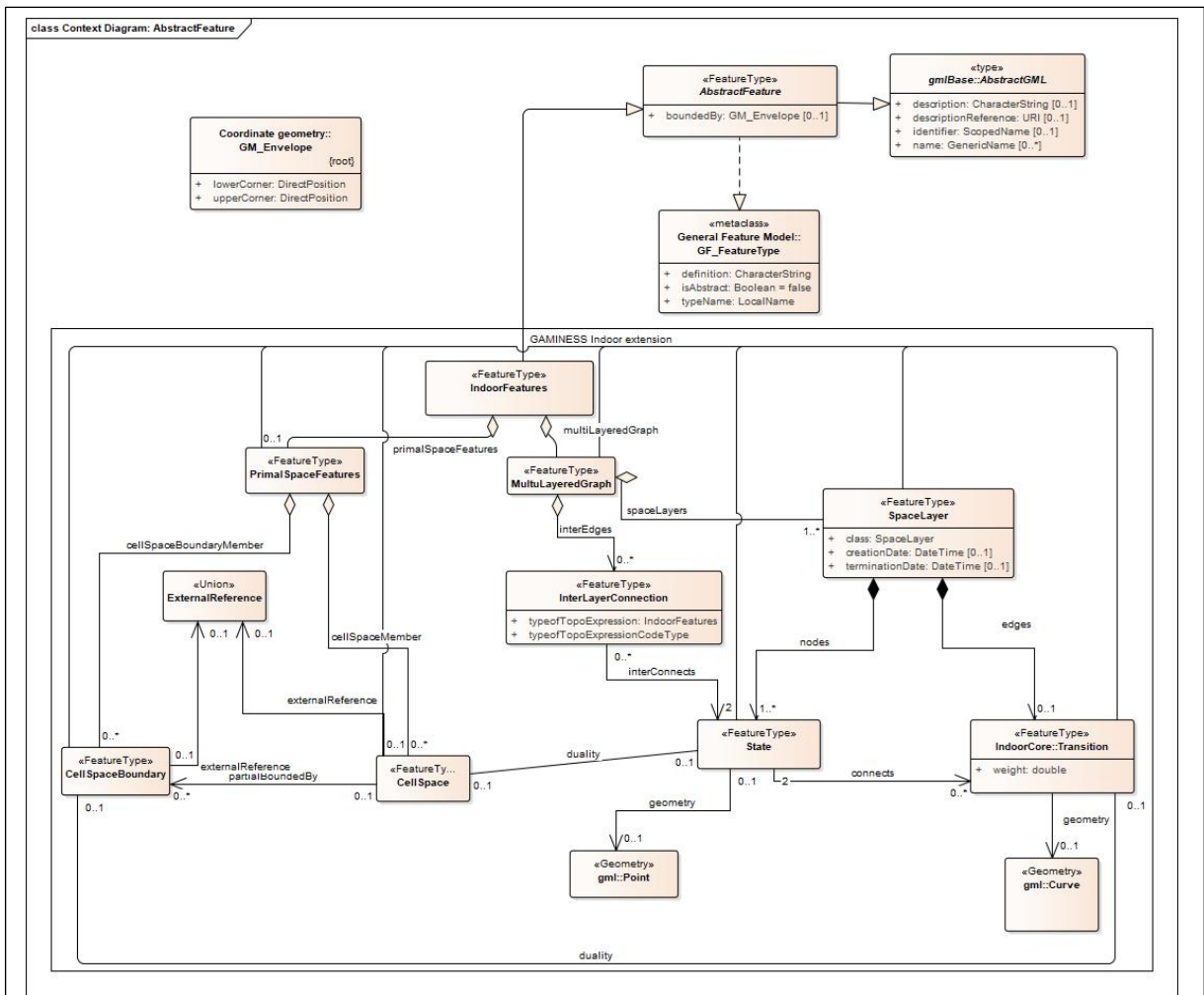
Предложена проширења која ће детаљно бити описана у поглављу GAMINESS логички модел са сензорским проширењима, омогућиће стандардизовано читање и превођење података које прикупљају сензори. Систем управљања GAMINESS омогућава повезивање између сензора и система у реалном времену. Да би се решио проблем варијабилности, основни модел Dynamizers је проширен елементима IoT и SensorThings стандарда, који представљају стандарде по којима већина новијих сензора ради, који раде на принципима бежичне комуникације. У складу са дефинисаним параметрима комуникације између сензора и GAMINESS система, прикупљени подаци се шаљу у JSON формату, без обзира на то који тип сензора је на улазу (аутоматизоване тоталне станице, нивелири, тоталне станице, геотехнички сензори итд.). Као комуникациона платформа између GAMINESS система и самог сензора, користи се OpenADMS платформа која има стандардизовано повезивање са различитим типовима сензора и преноси податке са сензора као JSON поруке у структурираној форми. Даље путем тог протокола се као WFS подаци похрањују у систему. На овај начин се обезбијеђено је рјешење за варијабилност. Van Oosterom и др. упоредили су могуће системе управљања облаком тачака у стандардним релационим базама података, дистрибуираним NoSQL базама података и алтернативима заснованим на датотекама [52]. Коришћењем резултата тог истраживања предложени оквир проширује стандардну mapreduce парадигму за обраду у меморији у односу на стандардно кориштени Hadoop.

Да би се касније извршила анализа и визуелизација облака тачака потребно је одабрати најефективнији приступ у дефинисању опште логике. Richter и сарадници [118] су предложили приступ анализе и визуелизације облака тачака скалирањем језгра. Да би то било могуће потребно је дефинисати параметре који ће омогућити успостављање такве врсте функционалности, приликом креирања записа у систему. У наредним поглављима детаљно ће бити описан модел концептуалног проширења GAMINESS система у односу на стандардни CityGML модел.

4.2.3. КОНЦЕПТУАЛНИ МОДЕЛ GAMINESS СИСТЕМА СА ПРОШИРЕЊЕМ ЗА УНУТРАШЊУ ОРИЈЕНТАЦИЈУ

Да би искористили предности и једног и другог модела, извршено је проширење унутрашњег простора дефинисаног основном класом feature моделовања AbstractFeature за карактеристике ћелија као CellSpace. У том смислу омогућена је сегментација унутрашњег простора са просторним слојевима са могућностима унутрашње селекције карте, просторног проналажења, стајање или могућност помјерања са локације

(Слика 10). Обје опције нуде могућност основе засноване на GeneralSpace параметрима са проширењем постојећег модела као хелија IndoorFeature и кретањем по хелијама које су дефинисане класом која уважава тополошке односе, TransitionSpace. Овим путем се ријешава проблем варијабилности, тако што је омогућено учитавање елемената заснованих на IndoorGML стандарду. Основна класа GML дефинисана је као Abstract Feature и садржи генерални Feature модел преко кога је описана основна геометријска структура. У овом контексту поред основних координатних геометрија модел је проширен за Indoor Feature Types. Елементи структуре су описани вишеслојном графиком која представља агрегацију више просторних слојева. Оне су међусобно повезане међуслојним повезницама заснованим на тополошким структурама и односима између слојева. Елементи структуре су структурирани као жични модел који је дефинисан GML линијама и GML тачка типом геометрије.



Слика 10. GAMINESS основни Abstract GML модел проширен IndoorGML класом за навигацију

Просторни Feature се трансформише у ћелије преко којих се дефинише контекст и путања у оквиру IndoorGML модела као ћелијског модела.

4.2.4. КОНЦЕПТУАЛНИ МОДЕЛ GAMINESS СИСТЕМА СА ПРОШИРЕЊЕМ ЗА СЕНЗОРЕ

Паметни градови пружају интеграцију физичких и дигиталних система који су у међусобном односу у изграђеном окружењу које дефинишемо као урбано подручје. Процеси се у овим окружењима одвијају посредством утицаја људског фактора. Предност оваквих система је вишеструка, а посебно се огледа у могућностима креирања и симулирања модела различитих појава у урбаним цјелинама. Конвенционални приступ подразумијева да су градски 3Д модели статички и не подржавају динамичке карактеристике појава које су мјерене путем сензора. У том контексту потребно је направити одговарајуће проширење како би се обезбиједиле карактеристике простора које ће у обзир узимати комплексне обрасце дефинисане аналитичком обрадом мјерења прикупљених са сензора. У том контексту предложени CityGML модел са својим пакетом Dynamizers је проширен за SensorThings и SWE параметре стандарда.

Сензор Web представља дистрибуирану мрежу која повезује различите типове сензора. Уз постојање довољне енергије, теренска и даљинска мјерења могу континуирано прикупљати мјерења са инструмената. Предуслов за дистрибуцију овако великих количина мјерења је постојање адекватне информационо – комуникационе технологије засновану на различитим концептима стандарда. Последњих година су препознати бројни стандарди у овој области који омогућавају примјену Сензор Web-а у контексту паметних градова.

Стандардом CityGML препознат је специјални карактер Feature објекта који је дефинисан као Dynamizers. Успостављен је на бази ISO 19108 стандарда према дефиницији TM_Position. У оквиру његове основне структуре дефинишу се два основна елемента:

- Структура података како би се представиле временски зависне вриједности у различитим и генеричким представама података
- Модел којим се уанпређује статичка структура модела града са динамичким, временски зависним компонентама.

Динамички подаци који се прикупљају путем сензора користе стандарде OGC Sensor Observation Services (SOS) и OGC SensorThings API [87]:

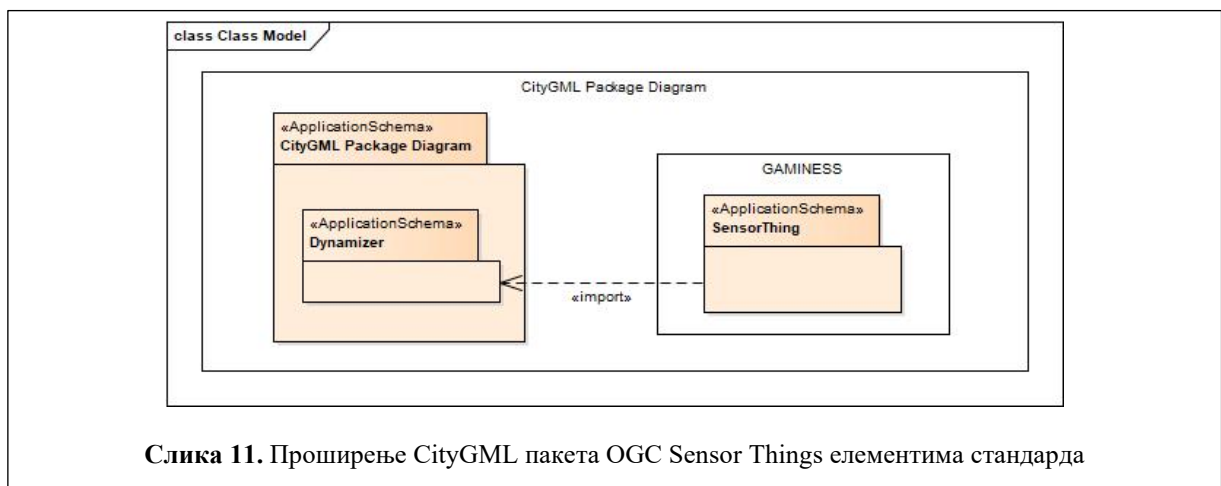
a) OGC Sensor Observation Services (SOS)

- Представља отворени стандард који је дио OGC Sensor Web Enablement (SWE).
- Допушта упите у реалном времену на подацима са сензора и временским серијама.
- Одговори опажања су декодирани путем O&M стандарда.

b) OGC SensorThings API

- Флексибилан стандард за повезивање Internet of Things објеката.
- Уређаји, подаци и апликације су повезани путем web мреже.
- Изграђен на OGC SWE и O&M стандарду.
- Обезбијењује REST сервисе и декодирање података у JSON формату.

Основни CityGML Dynamizers модел кроз своје класе подржава динамичку представу мјерених величина. Омогућено је повезивање екстерних сензора у интегрисан систем GAMINESS. Сензори врше мјерења и уз мјерења чувају карактеристике самих сензора, уз информацију о локацији сензора као јединственог објекта у предложеном концепту паметног града. Порастом количине података који се прикупљају са сензора, као и све јефтинијом и доступнијом мрежном инфраструктуром и широкопојасним интернетом омогућена је једноставна могућност перманентног прикупљања мјерених величина и њихово складиштење у базама података, те укључивање у рад у другим системима.

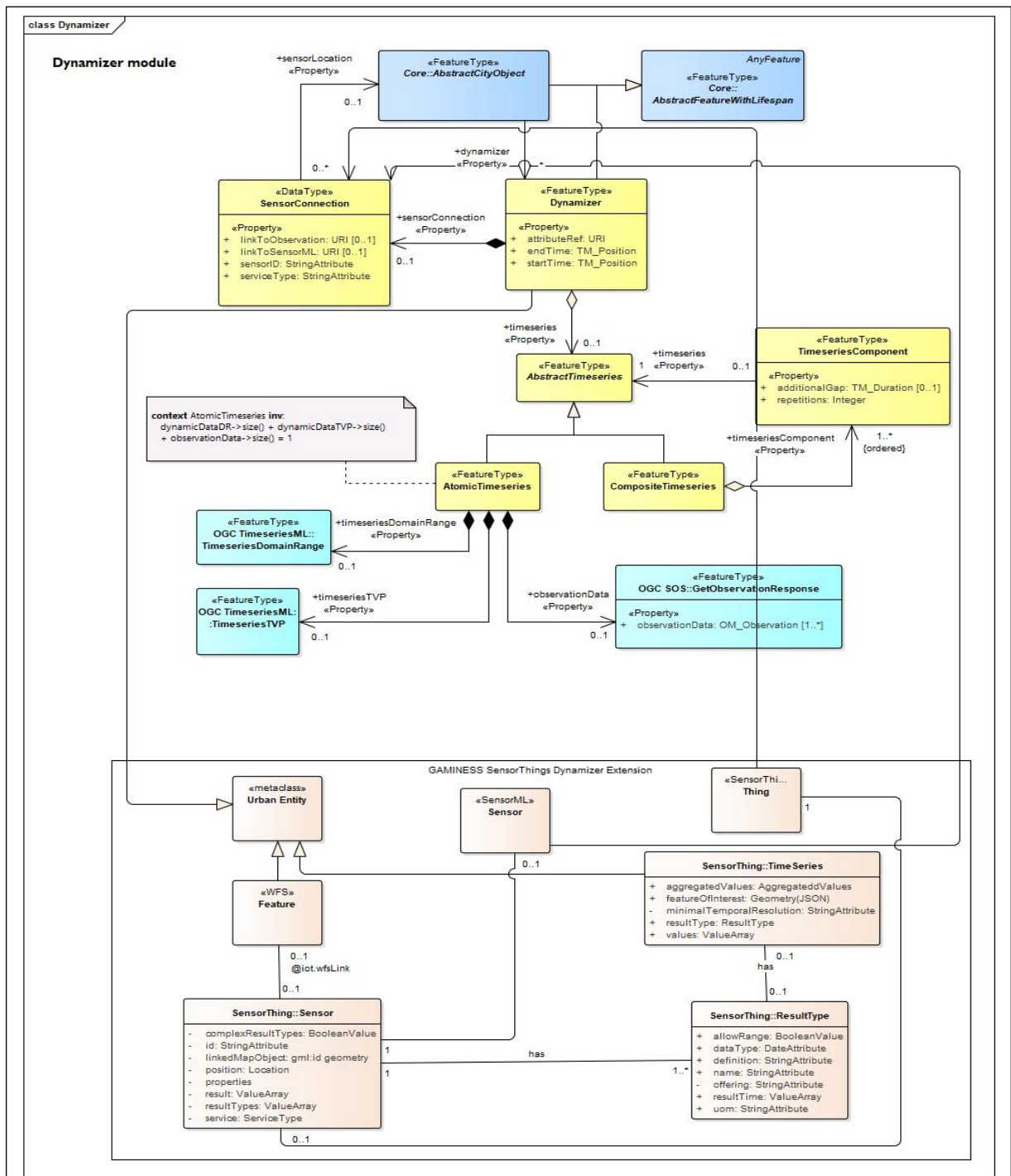


Слика 11. Проширење CityGML пакета OGC Sensor Things елементима стандарда

Основни CityGML модел је првобитно дефинисао укључивање сензора према дефинисаном оквиру представљеном кроз Dynamizers пакет. Са порастом прикупљања сензорских података заснованог на IoT-у, постоји потреба за ревизијом модела за прикупљање посматраних података. OGC Sensor Things је отворен, геопросторно омогућен и уједињен начин за међусобно повезивање уређаја, података и апликација Интернета ствари преко мреже. У ту сврху концептом GAMINESS је извршено проширење основног Dynamizers пакета са OGC Sensor Things карактеристикама (Слика 11), како би било омогућено повезивање сензора у систем који раде на принципима размјене података који је дефинисан путем *Thing* карактеристика.

Иако је то OGC Sensor Things стандард, он се заснива на богатом скупу доказаних и широко прихваћених отворених стандарда, као што су веб протоколи и стандарди за омогућавање веб сензора OGC (SWE), укључујући ISO/OGC модел података за опажање и мерење (OGC). 10-004p3 и ISO 19156:2011). У овој структури, сваки уређај је моделован као стварни ентитет. Он користи локацијске ентитете да омогући информације о тренутној локацији у виду атрибута, али задржава могућност давања информација и историјским локацијама ентитета везано за старе позиције и евентуална кретања[87]. Ентитет може имати један или више токова података, при чему је сваки ток података логичка агрегација мјерења сензора, које производи управо сензор. Група токова података је колекција мјерења која региструју величине коју биљежи сензор монтиран на ентитет. Ток података има име, опис ентитета и JSON објекат који садржи кључ-вриједност парове дефинисане као јединица мјере. Вриједности јединице мјере прате јединствени код за јединицу мјере и тип посматрања, који сервис користи за кодирање мјерења. У предложеном моделу, ствар (ентитет) је физички сензор са именом, описом и JSON објектом који је представљен датотеком карактеристика тог сензора. Он има локацију, ток података и историјске локације које се биљеже и анализирају. Тренутни семантички 3Д модели града су статички по природи и немају у својим моделима записана својства која су зависна од времена. У ту сврху према претходно описаним елементима модел Dynamizers је проширен, како би се креирале динамичке карактеристике и својства која омогућавају пружају вишедимензионалну динамичку представу GAMINESS модела паметног града проширеног подацима из динамичких извора података који имају изворе засноване на различитим стандардима (Слика 12). На овај начин дефинисана су проширења којима се рјешава проблем варијабилности.

Динамизер омогућава моделовање и интеграцију динамичких својстава унутар семантичких 3Д модела града. Објекти Динамизер успостављају експлицитне везе између података сензора/мјерења и одговарајућих својстава објеката модела града који се мере сензорима. У моделу је успостављена експлицитна веза између урбаног ентитета и семантике података са сензора.



Слика 12. Пакет Dynamizers са OGC Sensor Things проширењем [127]

Тај однос је имплицитно дефинисан у оквиру модела града. Моделом је предвиђено да објекат (Feature) има успостављену мрежну комуникацију са ентитетом сензора базираном на SensorThing стандарду путем web feature service (WFS). Овим путем се омогућава слање карактеристика агрегираних у Sensor класи гдје је дефинисан: идентификациони број сензора, локација, карактеристике, резултати, сервис. Поменути проширењем се може примјетити да за сензоре који користе елементе пакета Dynamizers и оне који користе елементе SensorThing проширења метаподаци се агрегирано уписују као информације урбаног ентитета. У смислу мјерења као што се генерализоване информације из Dynamizers класе прослијеђују класи која дефинише конекцију између екстерног уређаја и класе објекта у основном GML моделу, тако се на исти начин мапирају подаци са Thing агрегиране класе на ту исту класу која обезбијеђује конекцију ка објекту. Поменуто проширење у коначном прегледу нам даје могућност два улазна канала, сензора који ће бити подешен и структуриран према основној Dynamizers параметризацији или према SensorThing карактеристикама прописаним тим стандардом. У оба случаја без обзира да ли комуникациони протокол подразумијевао директну везу или WFS сервис подаци се на јединствен начин мапирају кроз SensorConnection протокол за комуникацију. Овим путем је на концептуалном нивоу дато рјешење за проблем варијабилности дефинисан Big Data концептом.

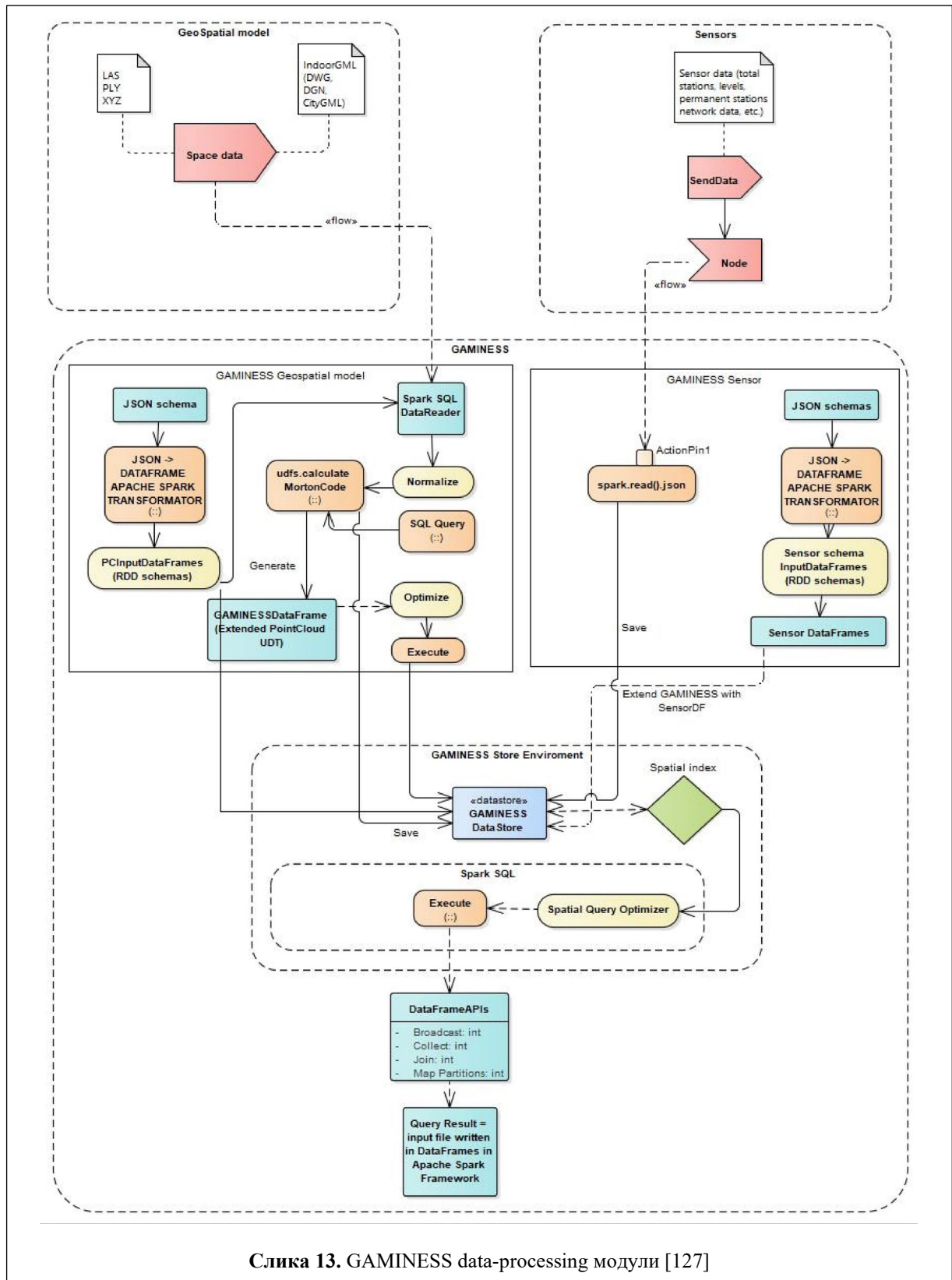
5. РАЗВОЈ ПРОЦЕСНОГ И ИМПЛЕМЕНТАЦИОНОГ МОДЕЛА GAMINESS СИСТЕМА

Систем управљања паметним градом GAMINESS, представља свеобухватан систем који је базиран на концептима просторно – временског управљања информацијама и структуром паметног града. Као основа за израду овог модела, кориштен је CityGML модел са својим основним пакетима који су проширени и оптимизовани за потребе рјешавања 5V проблема. Овај стандард је општеприхваћено и свеобухватно рјешење које представља тзв. open data model за чување и размјену градских модела. Као основа геопросторне компоненте у предложеном моделу користи се шема стандарда Geography Markup Language верзија 3.1.1 (GML3) заснованог на Open Geospatial Consortium (OGC) и ISO TC211 стандардима [119].

У пракси постоји више рјешења за управљање паметним градом, али најчешће су конципирана за управљање само неким од дефинисаних проблема. Врло често су то рјешења која интегришу у свој систем карактеристике одређеног произвођача сензора и пружају управљање и визуелизацију само за те податке. Ипак, постоје и друга рјешења која имају сличну поставку као предложени GAMINESS систем, а која су такође заснована на CityGML стандарду. Управо та рјешења ће бити мјеродавна да се врши упоредна анализа између предложеног концепта и постојећих рјешења. Као основа за први слој, слој складиштења података, постојећи модели су углавном имплементирани на неком од већ познатих система за управљање базама података. Једно од најзначајних рјешења је пројекат 3DcityDB модел који користи систем за управљање базама података Oracle и PostgreSQL/PostGIS. 3DcityDB систем је свеобухватан модел који је у потпуности заснова на CityGML стандарду. Када је ријеч о геосензорској компоненти или компоненти која се односи на унутрашњи простор и навигацију, елементи су задржали оне описе који су дефинисани одговарајућим пакетима CityGML стандарда (Dynamizers и AbstractGML). Као засебан дио се издвајају параметри IndoorGML стандарда који нису имплементирани у оквиру 3DcityDB модела. У овом смислу постоје појединачни случајеви гдје су вршена проширења постојећег модела у сврху имплементације SensorThings стандарда које је описао Chaturvedi [87] (Слика 13).

На основу описаних елемената у поглављима у којима је дефинисано проширење AbstractGML ентитета параметрима унутрашње оријентације преузетим из IndoorGML стандарда, те успостављањем подмодела геосензорског ентитета који проширује

Dynamizers пакет за IoT и SensorThings стандарде у наредним поглављима биће дате напомене на који начин је имплементирано то проширење у Big Data окружењу.



Слика 13. GAMINESS data-processing модули [127]

У оквиру истраживања успостављен је систем који је базиран на Big Data концепту чиме је оправдана општа хипотеза. За рјешавање овог проблема коришћена је метода анализе и метода дескрипције гдје је модел паметног града дефинисан одговарајућим дијаграмима класа којим је формално специфициран систем. Конвенционални приступ и коришћење традиционалних рјешења система управљања базама података за реализацију имплементационог модела GAMINESS система, отпада као могуће рјешење. Дакле, да би задовољили све наведене потребе неопходно је креирати окружење које ће бити флексибилно и лако прошириво за имплементацију захтјева GAMINESS система. За трансформацију из традиционалних оквира дефинисаних објектно – релационим системима база података у сасвим нови оквир базиран на RDD као ћелији управљања GAMINESS системом.

Као што је споменуто GAMINESS као систем управљања треба да обезбиједи управљање 3D подацима у динамичком смислу. Модел користи основу CityGML стандарда за размјену и управљање 3D градским моделима. Комплетна структура подржава правила прописана стандардима Open Geospatial Consortium (OGC) и ISO TC211 [119]. До сада су развијана бројна рјешења за специфичне случајеве, али се посебно истиче рјешење 3DcityDB које је успостављено на Oracle и PostgreSQL/PostGIS платформи, на принципима објектно – оријентисане инфраструктуре система заснованог на CityGML стандарду. Овај модел може служити као упоредни систем за предложену структуру GAMINESS система управљања.

Систем управљања GAMINESS пружа бољу функционалност система заснованог на програмском језику Apache Spark у приступу дистрибуиране обраде података засноване на MapReduce парадигми, приказаној кроз концепт 5V. У Big Data окружењу, основни концепт MapReduce укључује раздвајање процеса на два подпроцеса: фазу редукције и фазу мапирања података. У систему се предлажу одговарајући алгоритми који ће извршити спајање података без губитака. Један од најпознатијих система за складиштење података у овом контексту Big Data је Hadoop, који нема развијену геопросторну компоненту и стога није адекватан за употребу у овом истраживању. У процесу редукције података постоје компоненте HDFS (Hadoop distributed file systems) који разбијају објекте на мање под-елементе, који се налазе на кластерима мрежних система. Креирањем блокова врши се дистрибуција кроз кластер, те се на тај начин смањују изворни подаци и врши одговарајуће мапирање у систему. Основни апстрактни феномен дефинисан у окружењу Apache Spark је постојање одговарајућег дистрибуираног скупа

података (RDD), који је скуп елемената који су партиционисани кроз чворове кластера и мапирани у предложену структуру, што на јединствен начин пружа могућност паралелизације. У окружењу GAMINESS, RDD је креиран из Scala колекције улазних података пресликаних у структуру Apache Spark. Scala има пуну подршку за функционално програмирање и веома чврст систем статичких типова података. Ово омогућава да програми написани у Scala програмском језику буду врло концизни и самим тим мањи по величини од других програмских језика опште намене [120]. Систем омогућава постојање RDD у системској меморији, што пружа могућност поновног коришћења поменутог RDD кроз операције паралелизације. Употреба тако редукованих системских датотека резултује скалабилношћу и скраћивањем времена обраде података, што је један од највећих доприноса у структурирању података о паметним градовима у овом моделу.

Уз додатак геопросторне компоненте у овом систему управљања, паметни градови могу укључивати различите геопросторне податке који се налазе у различитим структурама, нпр. подаци прикупљени са геотехничких сензора. У потпоглављу имплементације приказан је примјер кориштења температурних сензора. Измјерене вриједности се мапирају у елементе који су део подмодула Геосензор. Метода читања, структурирања, складиштења и управљања сензорским подацима у дистрибуираном окружењу заснована је на Apache Spark оквиру. Систем обезбјеђује смањење и скалабилност тродимензионалног простора у једнодимензионални низ уз очување просторне локације тачака са одговарајућим атрибутима без обзира на њихов положај. Ово пружа могућност повезивања и управљања великим бројем различитих сензора са уносом великих количина података на малом простору.

5.1. РАЗВОЈ МОДЕЛА ТРАНСФОРМАЦИЈЕ У ОКРУЖЕЊЕ BIG DATA ПАРАДИГМЕ

Развојем технологије, долази до све већег пораста употребе геопросторних података. Чак 80% данашњих информација имају неку врсту геопросторне одреднице. Скоро све данашње апликације имају геопросторне компоненте. Стога све те огромне количине доступних информација је потребно каналисати на одређени начин, одбацити сувишне од корисних за одлучивање према неком критеријуму. Такве енормне количине информација треба посматрати са аспекта Big Data информација. Са растућим потребама прије свега друштвених мрежа и развијања алгоритама који треба да схвате навике корисника апликација, долази до развоја бројних техничких рјешења и различитих језика

који су базирани на Big Data концептима. Са друге стране ако посматрамо геопоросторне податке, велика експанзија развоја аутомобилске индустрије која нагиње ка аутономној возњи је довела до растућих потреба за развојем алата који ће огромне количине података обрадити у реалном времену. Сходно томе долази до потреба за развојем технолошких рјешења која омогућавају миграцију геопросторних података на Big Data платформе. У том смислу не постоји јединствено рјешење које ради са свим типовима геопросторних података и које се користи за развој модела геопросторних података сложеног типа заснованог на концептима Big Data парадигме. Да би се добио адекватан модел коришћена је метода генерализације којом је уопштен проблем варијабилности у оквиру паметног града и моделован систем који ће генералисти улазне податке у структурирану форму.

У раду са геопросторним подацима сусрећемо се с енормно великим бројем различитих формата за чување геопросторних података. Ради се о више од 200 различитих, што векторских што растерских формата од којих већина датира из 80-их и 90-их година прошлог вијека. Ни један од њих не може одговорити потребама и захтијевима данашњег брзорастућег и широко распрострањеног коришћења различитих архитектура софтверских система. Међународна организација за стандардизацију је прописала услове и начине дефинисања правила и модела којим ће растерски и векторски типови података бити прецизно описани. На бази ових спецификација дошло је до развијања нових класа и метода за различите формате података који би требали да попуне празнину у лепези геоиндустријских норми и смањили конфузију и проблеме који настају у раду са толиким бројем различитих формата и сталним конверзијама из формата у формат. Потребно је било створити отворене, апликацијске и платформски независне формате, као и преносива и интероперабилна складишта за просторне податке. Такође потребно јестворити формате који у исто вријеме омогућавају чување геометрије, топологије и атрибутивних података (метаподатака) са могућношћу коришћења јединственог упитног језика SQL-а према тим подацима. Неке од најзначајних библиотека које раде с геопросторним подацима описане су у наредним поглављима, а која су развијана на концептима Big Data парадигме:

а) Библиотека GeoSpark

Библиотека GeoSpark развијена је у Apache Spark програмском окружењу. Представља један од пројеката DataSys Lab на Arizona State University. Окружење је употпуности развијено као библиотека отвореног кода. Омогућава компјутерском

систему ефикасно процесуирање великих количина просторних података. На имплементационом нивоу има развијене елементе екстензије стандардног RDD елемента Big Data шеме DataFrame за просторну (spatial – SRDD) ћелију. Овај просторни SRDD препознаје основне геометријске примитиве и пружа могућност вршења просторних упита коришћењем просторног упитног процесног слоја GeoSpark класе над објектом који се налази у SRDD. Ради искључиво са векторским типом података [121].

Као што је наведено библиотека GeoSpark проширује RDD у форму просторног RDD (SpatialRDDs) и ефикасно раздваја SRDD просторне елементе кроз кластере и представља паралелизоване просторне операције. То су најчешће трансформације и акције (за SRDD) које обезбијеђују операције засноване на концептима машинског учења намјене корисницима за потребе аналитичких анализа над програмима. У потпуности је успостављен према OGC стандардима.

Проширени SRDD лејер служи да изведе просторне упите (нпр. оквирни упит, KNN упит и Join упит) над великим количинама просторних сетова података. Након што су геометријски објекти преузети у Spatial RDD лејер, корисници могу да позову процесне операције просторних упита обезбијеђене у Spatial Query Processing Layer-у GeoSpark-а који се покреће преко меморијског кластера. На овом ниову се одлучује о складиштењу и индексирању просторних објеката коришћењем SRDD. Ова библиотека пружа једноставну могућност употребе и интеграције са другим окружењима које нуде могућност анализе над SRDD. У табели 7 су приказане основне геометријске операције које су омогућене изворним кодом ове библиотеке. Такође, GeoSpark пружа боље перформансе у погледу брзине у односу на Hadoop базиране системе у апликацијама просторних анализа као што је просторно спајање, агрегација и просторна ко-локација.

ТАБЕЛА 7. Објекти и операције над њима у оквиру GeoSpark библиотеке

Подржавани објекти	Кратке операције
Објекти	Point, Rectangle и Polygon
Просторни индекси	R-Tree и Quad-Tree
Геометријске информације	Минимални гранични правоугаоник, Унија полигона и Преклапање/Унутрашњост (самоспајање)
Операције просторних упита	Оквир просторних упита, просторни join упит и просторни KNN упит

б) Библиотека Magellan

Библиотека *Magellan* је библиотека отвореног кода компаније *Geospatial Analytics*-а која користи Apache Spark као основну машину. Библиотека Magellan специјализована је за рад са основним геометријским примитивима и користи Catalyst оптимизацију за

извођење просторних упита и одређених опрација као што су *Intersects*, *Contains* и *Within* доста ефикасније него стандардни оператори. Библиотека је написана у *Python* и *Scala* верзијама *Apache Spark Scala* програмског језика. Пружа могућност обављања основних геометријских операција као што су *union*, *distance*, и *area* и тополошких анализа кроз дефинисане операције као што су: *intersects*, *overlay*, итд. Развијена је само за 2D геометријске примитиве [122]. Модел *SparkSQL* користи напредне операторе за геометријске упите у изворном *DSL*-у и *Pyspark Python* интеграцији како би се обезбиједило *Python* спајање. Библиотека подржава *Esri* формат фајлове. Циљ развоја ове библиотеке је да подржи цијелокупну гарнитуру *OpenGIS* једноставних примитива са *SQL* просторним функцијама и операторима уз одговарајуће тополошке анализе (Табела 8).

ТАБЕЛА 8. Предвиђене могућности *Magellan* библиотеке

Геометријски примитиви	<i>Point</i> , <i>LineString</i> , <i>Polygon</i> , <i>MultiPoint</i> , <i>MultiLineString</i> , <i>MultiPolygon</i> , <i>GeometryCollection</i>
Предвиђене функције	<i>Intersects</i> , <i>Touches</i> , <i>Disjoint</i> , <i>Crosses</i> , <i>Within</i> , <i>Contains</i> , <i>Overlaps</i> , <i>Equals</i> , <i>Covers</i>
Операције	<i>Union</i> , <i>Distance</i> , <i>Intersection</i> , <i>Symmetric Difference</i> , <i>Convex Hull</i> , <i>Envelope</i> , <i>Buffer</i> , <i>Simplify</i> , <i>Valid</i> , <i>Area</i> , <i>Length</i>

в) Библиотека *GeoMesa*

Библиотека *GeoMesa* је библиотека отвореног кода. Библиотека *GeoMesa*, која пружа могућност управљања дистрибуираним просторно – временским подацима у основи користи *Apache Accumulo* структурно управљање табелама. Имплементирани су геоалати који обезбјеђују геопросторну функционалност на великим сетовима података. Интегрисан је као додатак који је могуће користити уз геосерверске компоненте, будући да подаци који су складштитени у *Accumulo* се могу дијелити путем стандардних *OGC HTTP* сервиса. Омогућен је кластер мониторинг кроз администраторски интерфејс.

Проширењем *Apache Accumulo* платформе овом библиотеком, омогућено је чување геопросторних података. У основи је написана у *Scala* програмском језику. Има развијене основне концепте рада у *Big Data* парадигми. Операције се изводе на принципима коришћења *MapReduce* парадигме [123], [124]. *GeoMesa* пружа операторе за индексирање изворних података у оквиру *Accumulo* структуре. Методе које користи ова библиотека за увоз података из различитих складишта, читање шема тих структура, индексирање и слично су преузете из *GeoTools* библиотеке. Будући да је библиотека написана у *Scala* програмском језику, она је доступна као *core* директоријум и на *mvp* репозиторијуму. Обавезујуће је да се користи заједно са библиотекама *JavaConversions* и *JavaConverters*. Без обзира на начин учитавања ове библиотеке се дистрибуирају у

Accumulo сервере lib/ext директоријуме или на HDFS гдје Accumulo's VFSCClassLoader мора бити изабран. Уколико је библиотека подешена на овакав начин онда подаци складиштени у стандардним Accumulo табелама могу користити подршку дијелења кроз стандардне геосервисе WFS и WMS у оквиру Geoservera. JAR фајл назван geomesa-plugin--geoserver-plugin.jar се укључује у GeoServer копирањем у geoserver/WEB-INF/lib/ и на такав начин обезбјеђује читање података из ове структуре. Ту се чува GeoHash имплементација и друге функције генералне библиотеке повезане са Accumulo који садрже помоћне алате за GeoMesa библиотеку. Неки од ових алата као што је GeneralShapefileIngest имају Map/Reduce компоненте. Ова библиотека пружа алате са дистрибуираним окружењем. Омогућени су CQL упити над RDD структурама у оквиру GeoMesa.

г) Библиотека Geowave

Библиотека Geowave, користи Apache Accumulo платформу. Она омогућава вишедимензионално индексирање. Ради искључиво са векторским типом података. Она служи за складиштење, индексирање и претрагу више-димензионалних података према кључним вриједностима у оквиру базе података. Укључује посебно кројене имплементације које имају напредну подршку за OGC просторне типове (до три димензије) и за ограничене и неограничене временске вриједности [124].

Има тенденцију да креира вишедимензионални индексни слој који може бити додат у операцију било ког кључ-вриједност складишта. Модуларне је архитектуре и самим тим је омогућена једноставна интегрисаност у друге платформе. Библиотека GeoWave, такође, означава као приоритет чување конфигурације података, формат и друге информације неопходне за манипулисање подацима у бази података. Ово омогућава софтверу да програмски испитује све податке складиштене у појединачном сету GeoWave инстанци без потребе за дијелом конфигурације од клијената, апликативних сервера и других спољних складишта података. Додавање вишедимензионалног индексирања у Accumulo даје подршку геопросторним објектима и операторима. Додатак за GeoServer омогућава геопросторним подацима да кроз Accumulo буду дијелени и визуелизовани преко OGC стандардних сервиса. MapReduce парадигма је могућа и на улазним и на излазним компонентама и форматима за дистрибуирано процесирање и анализу геопросторних података.

д) Библиотека GeoTrellis

Библиотека GeoTrellis је написана у Scala програмском језику. Чита, пише и оперише са растерским подацима изузетно великим брзинама. Има функционалност да искриви (промјени резолуцију и оквир) растера на учитавању и кроз низ опција [125]. Имплементира више картографско - алгебарских операција као што су вектор у растер или растер у вектор. Пружа могућност извођења геопросторних операција над растерским подацима на најбржи могући начин, без обзира на размјеру. GeoTrellis обезбјеђује алат да рендерује растер у PNG формат слике за потребе креирања нпр. web картографских апликација или да конвертује информације о растерима у JSON формат. Оно што последња истраживања показују је да GeoTrellis обезбјеђује процесирање растера са web брзином (до 10 ms). То брзо процесирање растерских података засновано је на Big Data парадигми. Библиотека је имплементирана у оквиру FOSS. Примјер је алат за сијечење и спајање ивица геопросторних растера на Hadoop платформи у брзом процесирању великих улазних података. Према концепту операција и трансформација улазних растера, припада категорији Big Data окружења.

ђ) Библиотека Iqmulus

Један од ријетких пројеката који је радио на поступку успостављања структуре која ће омогућити рад са комплексним типом података облаком тачака у потпуности ослоњеном на Big Data окружењу је Spark SQL Iqmulus библиотека. У оквиру ове библиотеке су развијени методи за читање и записивање облака тачака као колекцију из PLY, LAS, и XYZ формата и њихово структурирање кроз Spark SQL. Окружење је прилагођено за Apache Spark верзију 1.6.2 [9]. Библиотека обезбјеђује алате за 3D стилизацију облака тачака за 2D статичке визуелизације које унапријеђују главну анализу компоненти у смислу локалне геометријске анализе. Прескакањем стандардне процјене грешака система висина земљишта, библиотека обезбјеђује концепт по коме је рендеровање робусно за неравне површине уз дефинисање висинских граница робусности. У оквиру библиотеке је имплементирана одговарајућа линија рендеровања која омогућава подизање на произвољно велике скупове података коришћењем Spark оквира и његовог дистрибуираног скупа података (RDD) и апстракције DataFrame[104], [113].

На основу прегледа могућности неких од библиотека које раде са геопросторним подацима, а користе концепте Big Data парадигме извршена је упоредна анализа како би

се изабрало најадекватније окружење које ће бити коришћено у сврху креирања основе за израду имплементационог оквира GAMINESS система. У компаративној анализи у контексту Big Data могућности посматрано је за које врсте примитива постоји подршка, за које типове података и да ли има могућност подршке за комплексне типове, прије свега облак тачака који је основа за учитавање 3D градских модела. Следећа карактеристика која је предмет интересовања је постојање подршке за JSON формат, будући да ће поруке које ће се достављати са сензора записивати у овом формату. Будући да систем треба да обезбиједи управљање системом града, стога су неопходне операције просторног индексирања. Као језик који је у потпуности заснован на Big Data парадигми је изабран Apache Spark те је због тога неопходна подршка за ово окружење са акцентом на одређене верзије овог програмског језика. Највише коришћена платформа која има обезбијеђен имплементациони оквир заснован на DataFrame и RDD користи се Hadoop 2.x. Будући да је податке града потребно повезати са одговарајућим регистрима, посебно се наглашава важност могуће трансформације података кроз координатне системе. Као погодност се гледа могућност имплементације кроз Scala верзију Apache Sparka. За рад у Big Data парадигми неопходно је омогућити рад у кластеру рачунарске мреже па се као погодност гледа могућност партиционисања података. Додатно, анализирале су се могућности геостатистичких анализа као извјештаја из система. Сумарно резултати анализа, приказани су у табели 9.

ТАБЕЛА 9. Компаративна анализа Big Data техничких рјешења за рад са облаком тачака

Карактеристике	Окружење					
	Geospark	Magellan	Geomesa	Geowave	GeoTrellis	Iqumulus
Типови податка (примитиви) - растер	+	-	+	+	+	-
Типови податка (примитиви) вектор	+	+	+	+	+	+
Рад са облаком тачака	-	+	-	-	-	+
Подршка за json формат	+	+	+	+	+	+
Просторни упити	+	+	+	+	+	+
Имплементација у Apache Spark програмском језику	+	+	+	+	+	+
Подршка за Apache Spark 3.x верзију	+	+	-	+	+	-
Могућност имплементације на Apache Spark Scala верзији	+	+	+	+	+	+
Hadoop 2.x	+	+	+	+	+	+
Партиционисање	+	+	+	+	+	+
Трансформације координатних система	-	+	+			
Геостатистика(просторне интерполације и сл.)	-	-	-	+	+	-

Резултати анализе указују да је између ових 6 окружења по 13 мјерених параметара најбоље оцјењена библиотека Magellan са 84,6 % задовољених услова. Као додатна потврда избору овог окружења служе имплементирани оквири дефинисани у истраживању [109], [126], [127], а који се односе на креиране кориснички дефинисане типове података за облак тачака са основним операцијама које се изводе над њима. Управо елементи из библиотеке Magellan ће се користити за формирање DataFrames из креираног концептуалног модела који је успостављен као JSON шема.

5.2. ОКРУЖЕЊЕ ЗА РАЗВОЈ ИМПЛЕМЕНТАЦИОНОГ МОДЕЛА GAMINESS

Анализом постојећих окружења која су наведена у претходном поглављу дошло се до најоптималнијег рјешења гдје ће библиотеке Magellan бити коришћена као основа имплементационог модела GAMINESS система управљања базом података. Предложено окружење прије свега је базирано на концептима парадигме великих података и као таквим су и развијане компоненте овог система. Библиотека Magellan написана је у верзији Apache Sparak 2.1+ и верзији Scala програмског језика 2.11. Верзије ове библиотеке су у потпуности компатибилне са OGC стандардима. Подржане су конверзије и могућност учитавања података из структура које су посебно значајне за развој окружења паметног града које се пројектује, а то су: Esri Shapefile, GeoJSON, OSM-XML, WKT. На овакав начин је задовољена могућност претварања свих геометрија у одговарајућу Spark структуру геометријских примитива. Стандардне геометрије које су подржане су: тачка, линија, полигон, мултитачка и мултиполигон. Поред основних могућности читања и записивања геометрија, интегрисане су методе за тополошке анализе као што су: пресијек (intersects), садржи (contains), налази се у (within) и слично.

Библиотека Magellan која је предмет проширења је саставни дио Spark пакета која се налази у његовом репозиторијуму. При покретању основног Spark Shell, библиотека Magellan се учитава као и сваки други Apache Spark пакет уз навођење његових основних карактеристика као што је верзија библиотеке. Коришћена је верзија 1.0.5 која је написана у 2.11 верзији Scala програмског језика.

Елементи ове библиотеке су коришћени као база за изградњу имплементационог модела GAMINESS окружења и развој метода које ће омогућити укључивање других библиотека и аналитику над учитаним подацима. Да би се могао извршити препис података из основне структуре која се односи на 2D/3D геометрију записану у изворном

формату као Esri Shapefile, GeoJSON, OSM-XML или WKT непоходно је омогућити одређени број пакета који се налазе у оквиру основне Magellan библиотеке као што су стандардни изрази, SQL типови података и функција и структурирање основних геометријских примитива, примјер за учитавање тачака и полигона (Шема 8). Ово је неопходно како би се од објеката могао направити одговарајући DataFrame.

У оквиру постојећих пакета омогућено је читање података који су структурирани као JSON запис. Ова функционалност је значајна зато што поруке које се шаљу са сензора су записане у овој структури. Посебна категорија у коју се ови подаци преводе уз додавање геолокација је GeoJSON. Тај формат прије преласка у GAMINESS структуру је главни начин комуникације у оквиру геосензорске мреже паметног града. Учитавање се врши тако што се за подешавање варијабле DataFrame у коју ће бити учитани подаци са сензора прво подешава контекст гдје се као формат контекста бира GeoJSON ()

Приликом учитавања геометрија, потребно је омогућити функције геопросторног индексирања. Ово је важан сегмент код креирања комплексних типова података. Библиотека са уграђеним функцијама пружа могућност индексирања по Z осиг, кроз тзв. листу ZOrder Curves према прецизности учитане геометрије.

5.3. РАЗВОЈ КОРИСНИЧКИ ДЕФИНИСаниХ ТИПОВОВА ПОДАТАКА НА АРАСНЕ SPARK ОКРУЖЕЊУ

Модел Apache Spark SQL користи угњежене моделе података који су базирани на Hive за табеле и DataFrame оквира података. Подржава основне SQL типове података као што су: boolean, integer, double, decimal, string, date и timestamp. У истраживању је коришћен DataFrame због сличности са структурираним библиотекама података које ће бити коришћене. У предложеном рјешењу служимо се већ дефинисаним типовима података и библиотекама који ће бити проширене за неке нове функције у складу са одговарајућим стандардима. Оно што карактерише комплексне типове података је могућност заједничког угњежавања различитих типова података, како би се створили комплекснији типови податка који ће моћи одговорити на постављене захтијеве коришћењем Spark SQL. Као што је и наглашено у претходном поглављу, за развој нових компоненти система коришћене су неке већ постојеће дефиниције типова података и метода над њима. Коришћена је Apache Spark платформа која у себи има интегрисану библиотеку Apache Spark SQL 3.12, компатибилну са верзијом Scala 2.12. За развој функција у Scala програмском језику, коришћења су правила дефинисана стандардом

ISO 19107. У ту сврху је коришћена библиотека *esri-geometry*, верзија 1.2.1 која је проширена са новим типовима. Посебну улогу у дефинисању овог модела који је коришћен у раду имала је библиотека *Magellan*, аутора *Ram Sriharsha*¹¹ [122]. У оквиру овог истраживања у практичном дијелу даће се одговарајуће рјешење које ће одговорити основним типовима података који треба да дефинишу и геометрију и топологију сваког коришћеног примитива. У овом поглављу објашњена је израда основног проширења комплексног типа основних геометријских примитива, који се користи за креирање кориснички дефинисан тип облак тачака који користи дефиницију геометријских примитива из ове библиотеке написане у *Apache Spark Scala* програмском језику, чиме се уклапа у концепт система изграђеног на *Big Data* окружењу. Функције које су коришћене представљају апстракцију феномена стварног свијета који је геопросторно дефинисан и који је повезан са неком локацијом у односу на Земљу. Векторски подаци састоје се од геометријских и тополошких примитива коришћених, одвојено или у комбинацији, за изградњу објеката који изражавају просторне карактеристике геопросторним карактеристикама. Кроз стандард ISO 19107 [63] је дефинисан тај приступ којим се описују типови примитива као што су: *Point*, *Multipoint*, *Line*, *Polyline*, *Polygon* и *Multipolygon*. Сваки од ових примитива има дефинисан референтни систем у коме се налази, геодетски датум и параметре геометрије који их описују (Табела 10).

ТАБЕЛА 10. Дефинисани типови геометријских примитива који су били предмет проширења

Тип примитива	Начин на који је дефинисан геометријски примитив
Point	Дефинисан као SQL кориснички дефинисан тип гдје је проширена основна класа <i>Esri point</i> додјелјивањем произвољној тачки трансформације у тражени облик.
Line	Дефинисан као SQL кориснички дефинисан тип гдје је приказан као линијски сегмент између двије тачке. Проширена је основна класа <i>Esri line</i> трансформацијом крајњих тачака линије у тражени облик .
Polyline	Дефинисан као SQL кориснички дефинисан тип гдје је приказан као уређени сет тјемена која садрже један или више дијелова спојених секвенци двије или више тачака. Тачке могу, а и не морају бити спојене или се сијећи Проширена је основна класа <i>Esri polyline</i> трансформацијом спојених или сијекућих тачака у тражени облик.
Polygon	Дефинисан као SQL кориснички дефинисан тип који садржи један или више прстенова који представљају повезане секвенце четири или више тачака. Могу бити затворени са несамосијекућом петљом која може да садржи вишеструке спољашње прстенове. Редослијед прстенова и оријентација указује која страна прстена је унутрашња. Сусједни иду око прстена у смјеру тачака у односу на онај у полигону. Крајње тачке прстена дефинишу рупе у полигону у смјеру казаљке на сату. Прстенови полигона су повезани са свим његовим дијеловима. За дефинисање полигона проширена је основна класа <i>Esri polygon</i> трансформацијом тачака прстенова у тражени облик задовољавајући претходно наведене услове.

¹¹ *Ram Sriharsha* је инжењер и члан развојног тима једне од најмоћнијих компанија у силицијумској долини, *Yahoo*. Један је од учесника развојног програма *Mapbox* и аутор бројних *open source* рјешења у области *Big data*. У последње вријеме истиче се пројектима из области аналитичког управљања геопросторним подацима коришћењем развојних производа *Apache Spark* платформе.

Просторне карактеристике описане су са једним или више просторних атрибута чија вриједност даје геометријски објекат (GM_Object) или топологију објекта (TP_Object). На шеми 1 је дат примјер кориснички дефинисаног типа података тачка коришћењем библиотеке Magellan [122]:

```
import com.esri.core.geometry.{Point => ESRIPoint}
import org.apache.spark.sql.Row
import org.apache.spark.sql.catalyst.expressions.GenericMutableRow
import org.apache.spark.sql.types._
import org.json4s.JsonAST.JValue
import org.json4s.JsonDSL._

@SQLUserDefinedType(udt = classOf[PointUDT])
class Point(val x: Double, val y: Double) extends Shape {
  def this() {this(0.0, 0.0)}
  override private[magellan] val delegate = {
    val p = new ESRIPoint()
    p.setX(x)
    p.setY(y)
    p}
  override final val shapeType: Int = 1

  def equalToTol(other: Point, eps: Double): Boolean = {
    math.abs(x - other.x) < eps && math.abs(y - other.y) < eps}
  override def equals(other: Any): Boolean = {
    other match {
      case p: Point => x == p.x && y == p.y
      case _ => false}}
  override def hashCode(): Int = {
    var code = 0
    code = code * 41 + x.hashCode()
    code = code * 41 + y.hashCode()
    Code}
  override def toString = s"Point($x, $y)"
  @param fn
  @return
  override def transform(fn: (Point) => Point): Point = fn(this)
  override def jsonValue: JValue =
    ("type" -> "udt") ~
    ("class" -> this.getClass.getName) ~
    ("pyClass" -> "magellan.types.PointUDT") ~
    ("x" -> x) ~
    ("y" -> y)}
private[magellan] class PointUDT extends UserDefinedType[Point] {
  override def sqlType: DataType = Point.EMPTY
  override def serialize(obj: Any): Point = {
    obj match {
      case p: Point => p
      case Array(t: Int, x: Double, y: Double) => new Point(x, y)
      case _ => ???}}
  override def userClass: Class[Point] = classOf[Point]
  override def deserialize(datum: Any): Point = {
    datum match {
      case row: Row => {
        row(0).asInstanceOf[Point]}
      case p: Point => p
      case null => null
      case Array(t: Int, x: Double, y: Double) =>
Point(x, y)
```

```
    case _ => ???}}
    override def pyUDT: String = "magellan.types.PointUDT"}
object Point {
    val EMPTY = new Point()
    private[magellan] def fromESRI(esriPoint: ESRIPoint): Point = {
        new Point(esriPoint.getX(), esriPoint.getY())}
    def apply(x: Double, y: Double) = new Point(x, y)}
```

Шема 1. Кориснички дефинисана функција за тачку као прост примитив

Дефинисана тачка је коришћена као основа за дефинисање кориснички дефинисаног типа Point који је структуриран за тачку као сегмент облака тачака. Класа UserDefinedType представља дио пакета org.apache.spark.sql.types. Овом класом се дефинишу кориснички типови (<UserType>). Имплементирани су интерфејси из java.io.Serializable. На овакав начин омогућено је проширење стандардних типова података за кориснички дефинисане типове података (Шема 2).

```
public abstract class UserDefinedType<UserType>
extends DataType
implements scala.Serializable
```

Шема 2. Проширење класе кориснички дефинисани тип

Развијени интерфејси функција омогућавају кориснику да креира властите класе које су интероперабилне са SparkSQL, креирањем кориснички дефинисаних типова података. На овакав начин је омогућено креирање DataFrame који има претходно дефинисану класу у шеми. У процесу конверзије у Apache Spark Scheme из JSON Scheme, кориснички дефинисани типови података имају анотацију на SQLUserDefinedType, како би Apache SparkSQL могао препознати и читати кориснички дефинисан тип. Конверзија серијализацијом се изводи формирањем DataFrame из другог RDD, док се десеријализацијом изводи директним читањем из DataFrame. За потребе конверзије JSON шеме у структуру шеме GAMINESS конверзија се одвија десеријализацијом. Читање подразумијева метод формирања објеката класа за кориснички дефинисан тип. За формирање одговарајуће шеме у оквиру модела уведен је кориснички тип података тачка (User defined Type (UDT) class point). У дијелу пакета шеме Geospatial GAMINESS, омогућава се конверзија стандардног формата облака тачака у структуру предвиђену овим моделом. У ту сврху кориштени су резултати објављени у истраживању Пајић и остали [126]. Ова класа подразумијева проширење постојеће класе тачка, која врши структурирање у PointUDT објекат шеме ишчитане из ц приказано на шеми 3:

```
@SQLUserDefinedType(udt = classOf[PointUDT])
case class Point (x: Double,
                  y: Double,
                  z: Double,
                  normx: Integer,
```

```

        normy: Integer,
        normz: Integer,
        var mortonCode: Long,
        intensity: Int,
        classification: Short,
        red: Short,
        green: Short,
        blue: Short) {
class PointUDT extends UserDefinedType[Point] {
    def dataType = StructType(Seq(
        StructField("x", DoubleType),
        StructField("y", DoubleType),
        StructField("z", DoubleType),
        StructField("normx", IntegerType),
        StructField("normy", IntegerType),
        StructField("normz", IntegerType),
        StructField("mortonCode", LongType),
        StructField("intensity", IntegerType),
        StructField("classification", ShortType),
        StructField("red", ShortType),
        StructField("green", ShortType),
        StructField("blue", ShortType)
    ))

```

Шема 3. UDT класе тачка [126]

Након што је омогућено регистровање овог типа података, тачка се препознаје као изворни објекат који представља предмет конверзије у DataFrame помоћу Spark SQL. UDT[Point] је дефинисан према логичкој шеми која је објављена у истраживању Пајић и остали. Уз податке о тачкама у оквиру шеме дефинисана је структура за метаподатке облака тачака.

Још један примјер комплексног типа је дефинисан у оквиру Geosensor пакета шеме GAMINESS. Развијен је посебан кориснички дефинисан тип податка SensorUDT. Усклађен је са структуром модела проширења пакета Dynamizer. Објекат који ће се креирати као DataFrame из шеме структурира се као Sensor (SensorThing) класа са својом шемом (Шема 4).

```

@SQLUserDefinedType(udt=classOf(SensorUDT))
case class Sensor(id: String,
    complexResultType: Boolean,
    mapObject: IdPointUDT,
    time: Timestamp,
    position: PointUDT,
    properties: String,
    result: Array,
    service: ServiceType){
class SensorUDT extends UserDefinedType(PointUDT) {
    def dataType = StructTypeSeq{
        StructField("id", StringType),
        StructField("complexResultType", BooleanType),
        StructField("mapObject", IdPointUDTType),
        StructField("time", TimestampType),
        StructField("position", PointUDTType),
        StructField("properties", StringType),

```

```
StructField("result", ArrayType),  
StructField("service", ServiceType)}}
```

Шема 4. UDT класе сензор

Приликом стриминга података, модел треба да обезбиједи мапирање података у наведену структуру и њихово складиштење у одговарајући дефинисани DataFrame. Подаци мјерења се записују као `ArrayType` и структурирају у шеми објекта. Класа пружа могућност ишчитавање потребних метаподатака којим се обезбјеђују параметри повезивања сензора са структуром шеме GAMINESS система за управљање подацима. Подаци су конвертовани у систем класом `Sensor` преко које се креира објекат из наведеног DataFrame. Овај објекат је предмет даље анализе развијене кроз кориснички дефинисане функције које омогућавају напредну аналитику над системом и враћање информација систему заснованих на напредним системима логике одлучивања. Структурирање података кроз овакав кориснички дефинисан тип, омогућило је учитавање података са различитих платформи, заснованих на различитим стандардима у јединствену GAMINESS структуру. Дефиниције сензора и модел повезивања структурира се путем JSON објекта складиштеног у оквиру `properties stringa`.

Да би се омогућили концепти просторно – временске идентификације и историјског праћења структуре паметног града, дефинисани елементи временске и просторне одређености сваког мјерења кроз записивање ове врсте метаподатака у облику временског отиска сваког мјерења. На овакав начин ће бити задовољен услов прописан стандардом DigitalTwins.

5.4. КОНВЕРЗИЈА МОДЕЛА У ПРОЈЕКТОВАНУ BIG DATA GAMINESS СТРУКТУРУ

Модел GAMINESS је у основи базиран на CityGML стандарду. У оквиру система управљања укључени су пакети дефинисани структуром града представљеном стандардом CityGML модела са структурним и модуларним описом елемената који га чине. У дискусији у оквиру описа концептуалног модела GAMINESS система, дате су напомене о проширењима која се односе на употребу различитих стандарда чиме је дато рјешавање проблема варијабилности, а који се првенствено односе на различите комуникационе и трансмисионе стандарде на којима су сензори конфигурисани као и на укључивање параметара контекста простора. Модел који укључује ова проширења је развијен и публикован у форми JSON шеме. У основи креирање JSON шеме и њено превођење у DataFrame је једноставан поступак. Конверзија се врши методом

`read.json()` или превођењем сета у `Dataset [String]` или у `JSON file`. Овакав `JSON file` није типичан `JSON` документ, већ представља структурни документ који садржи раздвојене и валидиране `JSON` објекте.

Примитиван начин креирања стринга из објекта описаног у `DataFrame` структури је представљен на шеми 5:

```
val schema = df.schema
val jsonString = schema.json
```

Шема 5. Креирање стринга од постојећег `DataFrame`

Обрнути процес је креирање шеме из `JSON` документа, гдје је као инстанца структурног типа узет `jsonString` као тип података (Шема 6).

```
import org.apache.spark.sql.types.{DataType, StructType}
val newSchema = DataType.fromJson(jsonString).asInstanceOf[StructType]
```

Шема 6. Креирање шеме из `JSON` документа

Документ у `JSON` формату чува податке који су једноставне структуре података и објекте у конкретном `JSON` формату, који представља стандард за размјену података. Помоћу `Spark SQL` омогућен је `spark.read.json("path")` којим се чита једнолинијски и вишеллинијски `JSON` документ у структуру `Apache Spark DataFrame` и којим се дефинише `dataframe.write.json("path")` за чување и записивање документа. Конвертором се дефинишу основни типови података који ће бити коришћени у оквиру конверзију. Посебну категорију у конверзији чине претходно дефинисани комплексни типови података који су у принципу угнијеждане табеле преведене у `RDD` структуру података. У оквиру ове структуре се описују типови података и како ће бити мапиран његов тип у `DataFrame` структури. Примјер је дат као Шема 7 што представља извод из Прилога 1:

```
val SchemaFieldName = "name"
val SchemaFieldType = "type"
val SchemaFieldId = "id"
val SchemaStructContents = "properties"
val SchemaArrayContents = "items"
val SchemaRoot = "/"
val Definitions = "definitions"
val Reference = "$ref"
val TypeMap = Map(
  "string" -> StringType,
  "number" -> DoubleType,
  "float" -> FloatType,
  "integer" -> LongType,
  "boolean" -> BooleanType,
  "object" -> StructType,
  "array" -> ArrayType
  "pointcloud" -> PointCloudType
```

Шема 7. Креирање `DataFrame` за систем `GAMINESS` из `JSON` документа

Након дефинисања типова, евидентно је да се комплетна конверзија из новоформиране JSON шеме врши по принципу омогућавања стриктних типова за варијабле система. Као Прилог 1 дат је општи модел конверзије у DataFrame за структуре које садрже основне типове података који су проширени са комплексним типовима података који су дефинисани у претходним поглављима. На овакав начин се прави општи модел конверзије у структурални DataFrame. Комплексни типови података у конверзији се посматрају као угнијеждене структуре које су у основи састављене из DataFrame-а који чине атрибути дефинисани овим истим основним структурним типовима података. На овакав начин је извршено и проширење основне библиотеке Spark SQL за извршавање операција над подацима. Овим приступом омогућиће се формирање шеме помоћу које се екстрактују и декодирају атрибути и вриједности JSON стринг објекта у DataFrame као колоне уређене претходно дефинисаном шемом.

Над креираном структуром система модул Apache Spark DataFrames чини једноставним читање различитих формата података, документовано читавање за стриминг. Он аутоматски закључује JSON шему. Код учитаних података, приступ појединачним пољима се врши прво читањем појединачних поља вишег нивоа као и код JSON редова са угњежденим објектима (Шема 8).

```
>> val temperatura = sqlContext.read.json("temperatura.json")
temperatura: org.apache.spark.sql.DataFrame
```

Шема 8. Scala код за читање JSON документа

Уколико је потребно провјерити структуру постојеће шеме података, потребно је позвати резултујућу шему RDD (Шема 9).

```
//Input files
{"name": "Sema"}
{"name":"temperatura", "vrijednost":true, "knows":{"temp":
["Sema "]}}
```

```
// Учитавање JSON-а са Spark SQL у Scala
val input
input = hiveCtx.jsonFile(jsonFile);
// Учитавање JSON-а са Spark SQL у SchemaRDD
|-- knows: struct (nullable = true)
|   |-- temp: array (nullable = true)
|   |   |-- element: string (containsNull =
false)
|-- vrijednost: boolean (nullable = true)
|-- name: string (nullable = true)
```

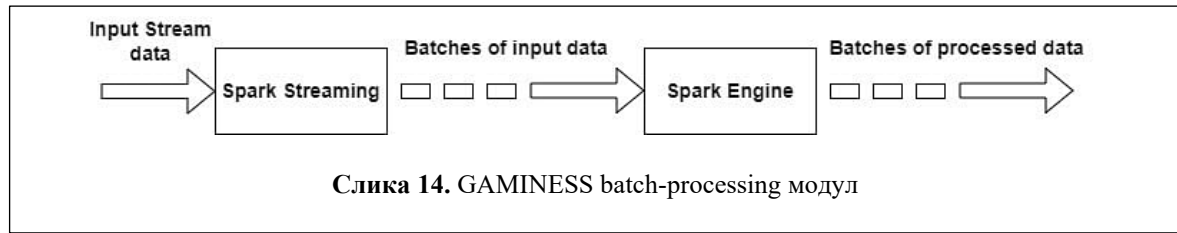
Шема 9. Учитавање резултујуће RDD шеме

JSON шема је спецификација за формате засноване на JSON-у за дефинисање структуре JSON података. JSON шема пружа гаранције за шта су JSON подаци потребни у датој апликацији и како се могу мијењати. Она има за циљ да обезбиједи валидацију документације и контролу интеракције са JSON подацима. Базирана је на концептима XML Schema, RelaxNG и Kwalify, али је и намењена да се базира на JSON нотацији, како би се JSON подаци у форми шеме могли користити за валидацију JSON података [105]. Исти алати за серијализацију/десеријализацију могу се користити за шему и податке. Стога модел података GAMINESS користи структуру модификоване JSON шеме преко које се читавају подаци.

5.5. ИМПЛЕМЕНТАЦИЈА ПАКЕТА ГЕОСЕНЗОРСКЕ МРЕЖЕ

Структура паметног града описана је у поглављу 4.2.4. у којој се сензори базирани на различитим стандардима јединствено укључују у систем одлучивања у паметном граду и враћају информације ка централној јединици. Структура модела подразумева формирање окружења у контексту Big Data стандардизованог DataFrame за регистровање мјерења са сензора и њихово повезивање са моделом објеката. Успостављен је одговарајући Sensor DataFrame у који се читавају подаци са сензора препознати као JSON поруке. Поруке се читају и праве се одговарајући низови гдје се атрибути из порука раздвајају и мапирају у одговарајуће колоне. Да би правовремено давао информације од интереса на основу одређених мјерења, неопходно је у систему дефинисати тригер функције којим ће систем самостално одлучивати у одређеним ситуацијама како да се понаша. За те потребе кориштени су Decision trees Apache Spark модули. Ове гране одлучивања представљају метод који Apache Spark обезбијеђује за обављање задатака класификације и регресије засноване на принципима машинског учења. Ти модели су искоришћени јер су једноставни за интерпретацију и управљање категорисаним објектима. Овдје се једноставно проширују вишекласна подешавања која не захтијевају скалирање података. Систем на овакав начин је омогућио регистровање нелинеарности и интеракција објеката.

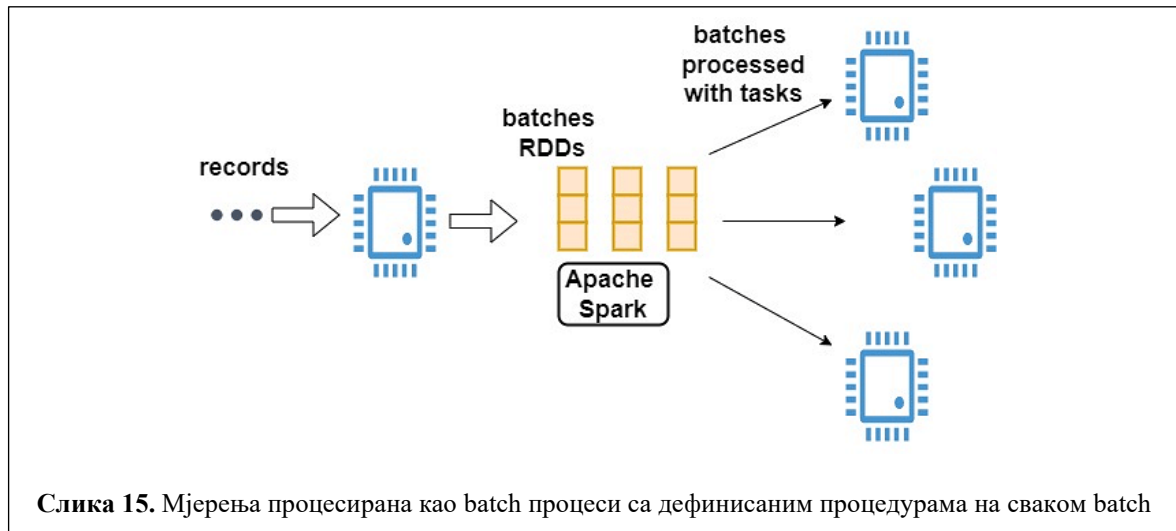
Библиотека `spark.mllib` коришћена је за вишекласну класификацију и регресију. Имплементирани су партиције података према редовима, гдје је омогућено дистрибуирано тренирање сета података на великом броју инстанци/чворова. Основу рада овог система чини пакет Spark Streaming који омогућава процесирање података у реалном времену путем Spark SQL.



У основи систем функционише да се подаци који су регистровани као JSON порука, а који представљају stream дијеле у мање подскупове (chunks) који представљају серије (batches) за извођење тзв. процесирања података у batches (Слика 14).

Уз помоћ Spark Streaminga и његових способности процесуирања података једноставно је омогућено за се користи јединствено окружење да задовољи све процесне захтјеве. За разлику од овог модела, успостављена је платформа која уз помоћ Mllib библиотеке омогућава тренирање податка ван мреже, а онда њихово процесирање путем мреже и Spark Streaming. Развијени модел који у основи користи ову библиотеку, задржава њену способност да анализира податке прикупљене у реалном времену и постпроцесира их. У пројектованој структури постоји неколико сегмената које је потребно напоменути као што је Streaming ETL који омогућава да подаци буду екстрактовани из различитих структура и стандардизовани у складу са прописаним моделом, те након тога агрегирани прије самог учитавања у пројектовани извор за складиштење података. Такође, модел је омогућио додатно структурирање података који подразумева да се подацима у реалном времену придружују нове информације како би се могле вршити што комплексније анализе у систему. Последњи сегмент који је важан у овом систему су тригери. У реалном времену помоћу одговарајућег SQL упита детектује се нестандартно понашање система и пружа се могућност реакције система на реакцију из уређаја сензора.

Пакет Spark Streaming врши дискретно стримовање података у мале микро batch структуре. Оне омогућавају извођење различитих функције на нивоу тих различитих процеса на различитим batch-овима (Слика 15). За разлику од стандардних сензорских мрежа у овој архитектури се задаци додијељују и изводе динамички зависно од доступних ресурса и њихове локације. Сваки batch процес користи RDD који је и омогућио стриминг података и процесирање кроз Spark библиотеке у GAMINESS систему.



Слика 15. Мјерења процесирана као batch процеси са дефинисаним процедурама на сваком batch

Када су у питању гране одлучивања оне представљају алгоритме који изводе рекурзивно партиционисање простора. Свака партиција је одређена најоптималнијим издвајањем на основу свих могућих издвајања с циљем да извуче најоптималнији сет информација из чвора гране. Овај приступ је посебно важан код комплексних система као што је нпр. аутономна возња код које је потребно да се на основу великог броја података донесе одлука у реалном времену. За предиктивно управљање подацима прикупљеним са сензора у геосензорском GAMINESS систему користе се концепти Apache Spark машинског учења. У склопу концепта надгледаног машинског учења изводе се два типа задатака и то класификација и регресија, на основу којих је циљ успоставити систем предвиђања на основу података. У оквиру предиктивног одржавања кроз регресију се поставља теза у којој је потребно одредити преоставли корисни вијек трајања до неке екстремне вриједности. Класификацијом се предвиђа хоће ли до квара доћи у неком временском периоду. Нпр. На овакав начин је могуће предвидјети да ли на сензору може доћи до отказивања преноса података. Регресија је више информативног карактера и њом се процјењује оквир када може доћи до екстремног оступања, не егзактан тренутак, Концепт регресије се одлично уклапа у BIG DATA поставку, јер за функционисање овог концепта у коме је потребно одредити скривене обрасце варијација и тачна предвиђања неопходне су велике количине података и инстанци да би модел уопште могао бити истрениран. За системе у којима је мањи број инстанци (сензора), предност се даје концепту класификације.

Регресиона процјена је тип мултиваријантног проблема временске серије јер сензори прикупљају информације током времена. У обрађеном примјеру сензор мјери температуру у сваком циклусу. Најједноставнији приступ који се користи је да тачке

података ниу у корелацији у временским корацима и информације о прошлим циклусима мјерења не узимају се у обир осим информација о једном циклусу мјерења преузетог у том тренутку. Овај приступ представља стандардни оквир за надгледано машинско учење, гдје се априори све варијабле узимају као статичне, па се тако екстремни предвиђају само на основу тренутног стања.

Apache Spark има богате могућности за извођење напредне аналитике. Садржи Mllib библиотеку која се користи да изгради „end-to-end“ конекторе машинског учења. Основна структура која је коришћена у овом систему за моделовање помоћу Apache Spark Machine Learning библиотеке је DataFrame добијен кроз Apache Spark SQL модул. На овакав начин искориштавају се комплексно структурирани упити и стриминг процесирање. За разлику од стандардног DataFrame, Apache Spark Machine Learning користи Vector тип податка, што је у ствари и формат за објекте у DataFrame. Сви модели учења захтијевају да се објекти споје у једну колону која је Vector (Шема 10).

```
from pyspark.ml.linalg import Vectors
```

Шема 10. Учитавање вектора из Apache Spark Machine Learning

Дефинисање тока алгорита врши се помоћу везе (Pipeline), која представља секвенцу нивоа која треба бити извршена. Сваки ниво је или трансформатор или естиматор. Трансформатори конвертују улазни DataFrame у други додајући колону/е. Естиматори су алгоритми учења који су тренирани на улазним DataFrame. Сама веза је естиматор коме се обезбијеђује DataFrame за производњу модела везе. Он представља обучени трансформатор који производи предвиђања. Предност оваквог модела је успостављање хиперпараметара овог система одједном (фитуризација, архитектура модела и учење) и бирање најбољег модела за задатак за који га тренирамо. За обучавање модела је коришћена дистрибуирана обрада. За те потребе је потребно да се пакети са више сензора јединствено уклапају у једну машину обезбијеђену Геосензорским пакетом. Основни задатак у предиктивном систему је обрада сирових података, а не израда карактеристика или подешавање модела предвиђања. То представља задатке као што су: анализа података, валидација и уклањање грешака. Примјери ове предикције укључују спајање табела из различитих извора података, преобликовање табела, уклањање грешака, довођење променљивих у исти формат вриједности и типова података дефинисане колоне. Apache Spark SQL представља модул за структурирану обраду података који се може успјешно користити за ову врсте задатака. Библиотека Apache Spark Machine

Learning и Apache Spark SQL су компатибилне и обје се ослањају на исту структуру DataFrame, тако да се могу обављати почетни задаци обраде података у Apache Spark SQL, а затим развити решење за машинско учење са Apache Spark MLlib.

Подаци коришћени у експерименту били су класификовани LiDAR подаци облака тачака и виртуелизовани подаци температурног сензора обезбијеђени бежичним ИОТ температурним сензором са логером података. Сензор је мерио температуру у интервалу од 1 s и биљежио податке према дефинисаној шеми GAMINESS модела (слика 16). Подаци из логера су путем WFS сервиса послати у GAMINESS структуру.

```
ID,mapObject,Time,Position,properties,result,service
1,sensor1,2020-05-21 15:56:25.843575+00,44.773692 17.210762 196.155456,7F8383EC-D3EC-495C-A8CF-B8BBE85C2920,19.6615830013373,TRUE
2,sensor1,2020-05-21 15:56:26.843575+00,44.773692 17.210762 196.155456,7F8383EC-D3EC-495C-A8CF-B8BBE85C2920,19.6615830013374,TRUE
3,sensor1,2020-05-21 15:56:27.843575+00,44.773692 17.210762 196.155456,7F8383EC-D3EC-495C-A8CF-B8BBE85C2920,19.6615830013373,TRUE
4,sensor1,2020-05-21 15:56:28.843575+00,44.773692 17.210762 196.155456,7F8383EC-D3EC-495C-A8CF-B8BBE85C2920,19.6615830013375,TRUE
5,sensor1,2020-05-21 15:56:29.843575+00,44.773692 17.210762 196.155456,7F8383EC-D3EC-495C-A8CF-B8BBE85C2920,19.6615830013374,TRUE
6,sensor1,2020-05-21 15:56:30.843575+00,44.773692 17.210762 196.155456,7F8383EC-D3EC-495C-A8CF-B8BBE85C2920,19.6615830013374,TRUE
```

Слика 16. Извод из излазне поруке мјерења са сензора у JSON формату

Скуп података се састоји од само 5 инстанци исте машине при чему је прва имала грешку после 45 циклуса. Типичан скуп података садржи много машина истог типа, више необрађених карактеристика и дужу историју од покретања до отказа (нпр. редове од стотина циклуса) (Прилог 2). Техника binning података претвара ред према једном атрибуту из континуирану у категоричку тако што се подаци дијеле у корпе. Крајњи резултат је и даље једна колона, али са модификованим карактеристикама вектора. Овај приступ подразумијева губитак информација због дискретизације и мора се пажљиво користити. У систему предиктивног одржавања, непрекидне необрађене варијабле често могу бити са шумом и имају искривљену дистрибуцију. Мерење температуре неких компоненти може бити веома искривљено јер је већина тачака података концентрисана око оптималног радног опсега. Релативно је мало читавања изван овог опсега, што ове области чини недовољно заступљеним у узорку и доводи до искривљености (Шема 11).

```
From pyspark.ml.feature import QuantileDiscretizer
bucketizer = QuantileDiscretizer (numBuckets=20, inputCol="result",
                                outputCol="result_binned")
df_binned = bucketizer.fit(df).transform(df)
```

Шема 11. Binning уз помоћ Apache Spark Machine Learning библиотеке

На овакав начин се прије трансформације температуре изван дефинисаног опсега не приказују, па тако сваки квантил има исти број узорака. У моделу све карактеристике дате моделу учења морају бити дефинисане у овом случају као векторски формат како би се могле декодирати категоричке варијабле (Шема 12).

```
from pyspark.ml.feature import StringIndexer
indexer = StringIndexer(inputCol="model_variant",
outputCol="model_variant_idx"=
indexed_df = indexer.fit(df).transform(df)
```

Шема 12. Encoding уз помоћ Apache Spark Machine Learning библиотеке

У оквиру овог поступка доступне су и друге функције које олакшавају развој. У оквиру ове библиотеке доступне су стандардне функције као што су скалирање, нормализација и импутер за управљање вриједностима које недостају. У овом окружењу се користи трансформатор VectorAssembler [9] који спаја вишеколумне вриједности у једну колону као Vector тип који је захтијеван овим Spark моделом. Индексирани DataFrame енкодира вриједности које се спајају (Шема 13).

```
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(inputCols=["ID", "mapObject", "Time",
"Position", "properties", "result", "service",
outputCol="features")
assembled_df = assembler.transform(indexed_df)
```

Шема 13. Спајање уз помоћ Apache Spark Machine Learning библиотеке

DataFrame предвиђања чува колону предвиђања са класама додељеним опажањима из скупа тестова. На основу резултујућих предвиђања, могу се израчунати различите метрике да би се процијенио квалитет модела прије него што се примјени (Шема 14).

```
from pyspark.ml.classification import RandomForestClassifier
rf = RandomForestClassifier (featuresCol='features',
labelCol='label', numTrees=10,maxDepth=4)
rfModel = rf.fit(df_train)
predications = rfModel.transform(df_test)
```

Шема 14. Класификација уз помоћ Apache Spark Machine Learning библиотеке

На овакав начин је приказан најпримитивнији начин одлучивања на нивоу инстанце. Модел је омогућио да се користе различити хиперпараметри учења на овом моделу, тако да је GAMINESS окружење омогућило јединствено учитавање из различитих извора у DataFrame који је јединствен и који омогућава трансформацију у Apache Spark Machine Learning Vector колону.

5.6. РАЗВОЈ КОРИСНИЧКИ ДЕФИНИСАНИХ ФУНКЦИЈА У GAMINESS ОКРУЖЕЊУ

3DCityDB користи дефиницију Dynamizers за читање података са сензора. Основни модел Dynamizers има низ проблема који се огледају кроз дефиницију стандарда коришћеног сензора. У студији случаја коришћен је температурни сензор прилагођен за

3DcityDB окружење према Dynamizers правилима. Систем GAMINESS обезбјеђује повезаност сензора преко JSON чвора на основу имплементације два стандарда поменута у поглављу 4.2.4. У том контексту користи се Batch обрада. Међутим, она није најбоља могућност за увид у реалном времену. Из тих разлога, за информације и обраду у реалном времену, користили смо Apache Spark Streaming за брзе повратне информације о апликацији засноване на догађајима. Коришћени сензори су генерисали стримовање података у облику JSON датотеке, који су обрађени помоћу Apache Spark и ускладиштени у DataFrame. Сенсор DataFrame је користио Spark Streaming као директан приступ. Процесирао је као колекција порука гдје су догађаји организовани у категорије. Сензори су сортирани по произвођачима и изворима података према шеми предложеној у моделу. Подаци за стримовање су подељени ради скалабилности. Скалабилност се постиже распоређивањем оптерећења на партиције. За аналитику преко стримованих података коришћена је Apache Spark SQL библиотека. Apache Spark SQL омогућава SQL операције на RDD и екстерним изворима података увођењем DataFrame API који је колона приказ RDD налик на базу података. RDD конструкција се анализира и оптимизује током рада. Да би се омогућило читање скупа стримованих датотека као Apache Spark Dataframe, креиран је модул који је трансформисао податке из JSON у RDD. Имплементациона шема GAMINESS као DataFrame креирана је трансформацијом JSON шеме разрађене у GAMINESS логичком моделу података. Направљен је као процес препознавања JSON поља и мапирања у адекватна поља структуре RDD са типовима података и кориснички дефинисаним типовима у зависности од типа структуре (Прилог 1). Концептуални модел GAMINESS је извезен као JSON шема. Један од циљева система управљања GAMINESS је да подржи интегритет улазних података приликом читавања JSON података у модел написан у Apache Spark Scala програмском језику. У ту сврху постоји могућност читања постојеће датотеке JSON шеме, рашчлањивања JSON шеме и прављења Spark DataFrame шеме. Генерисана шема се користи приликом читавања JSON података у Apache Spark GAMINESS окружење. Ово потврђује да су улазни подаци у складу са датом шемом и омогућава филтрирање оштећених улазних података. На овај начин, подаци се могу груписати да би се читали паралелно, што је кључ за високе перформансе приликом скалирања.

Обрада података у реалном времену помоћу Spark Streaming врши стримовање и batch обраду заједно користећи Spark core Api преко Spark конектора. На овај начин систем обезбеђује Big Data циљеве количине и вриједности. Основни Spark Streaming дијели

токове података на више токова података, који су секвенце RDD. Сваки RDD садржи записе из интервала серије. RDD као дистрибуирана колекција елемената се простира на више чворова у кластеру. Подаци садржани у RDD се партиционирају, а операције се обављају паралелно над подацима који су кеширани.

Свака операција примењена на ток података преводи се у операције на основном RDD, који примењује трансформације на елементе RDD. GAMINESS подржава брзо писање и скалирање где се табеле аутоматски дијеле на кластер према опсегу кључева. Табела има функцију писања и читања, где су у меморији доступни подаци за читање, писани подаци и кеширане фамилије колона. За најбољу оптимизацију процеса, систем обезбеђује партиционисање користећи Spark Core кеширање и минимизирање времена за читање и уписивање помоћу Spark Streaming.

У моделу је описано како треба конфигурисати датотеке параметара сензора. Ова конфигурација треба да обезбиједи слање порука за партиционисање. У овом кораку, GAMINESS обезбеђује својства конфигурације пара кључ-вриједности за поруке које се шаљу. Запис са сензора је пар кључ-вриједност и шаље се у GAMINESS. Састоји се од назива теме, броја партиције, кључа и вредности поруке. Параметри типа записа се поклапају са својствима серијализације. За функционалност GAMINESS повезаног сензора, врши се иницијализација Spark Streaming контекстног објекта. Он ствара ток података. У DataStream, сваки запис је написан као ред текста. Вриједности порука се рашчлањују у сензорске објекте, са операцијом мапе на dStream. Ова операција мапирања примјењује функцију Sensor.parseSensor на RDDs у dStream, што резултира RDDs сензорских објеката који генеришу RDDs тока података сензора (Шема 15).

```
sensorDStream.foreach RDD {rdd=> val sqlContext =
SQLContext.getOrCreate (rdd.sparkContext)
rdd.toDF().registerTempTable("Sensor")
val res=sqlContext.sql (" SELECT id, mapObject, position,
properties, result, service FROM Sensor GROUP BY id")
res.show() }
```

Шема 15. Кориснички дефинисана функција груписања и приказа података из SENSOR DataFrame

Систем парсира JSON податке у класе Spark сензора. Обрађени RDD регистрован као DataFrame је табела која пружа систему могућност да користи следеће SQL изразе над атрибутима сензора. Сензорски RDD објекти се могу филтрирати, претварати у објекте, а затим уписати у систем за складиштење. Пријем података мора бити покренут преко

функције start () у Streaming контексту. Завршетак зауставља овај процес, а затим чекамо да се стримовање заврши са процесом.

5.7. ТЕСТНО ОКРУЖЕЊЕ

Тестирање је извршено 2021. године у ГИСЛАБ (ГИС лабораторији) на Архитектонско – грађевинско – геодетском факултету Универзитета у Бањој Луци. Експеримент је изведен на 4 рачунара конектована са 10 гигабитном мрежом. За потребе тестирања инсталирана је 3DCityDB на PostgreSQL систему за управљање базама података (верзија 10.12) и Apache Spark (верзија 3.0.2). PostgreSQL је инсталиран на једном, док је Apache Spark инсталиран и конфигуриран на три рачунара спојена у кластер мрежом. У кластеру један је рачунар мастер рачунар (главни) и два су радна рачунара. Конфигурација хардвера је приказана у Табели 11. У оквиру овог окружења извршно је поређење између перформанси GAMINESS модела и стандардних релационих система за управљање базама података. Основни тест се огледао на поређењу између перформанси два система приликом учитавања података из различитих извора.

ТАБЕЛА 11. Карактеристике хардвера коришћених компјутера

CPU	RAM	HDD
INTEL Core i7-7700 3.6 GHz	8 GB, DDR 4, 2400 MHz	1 TB, 7200 rpm

5.8. ВЕРИФИКАЦИЈА МОДЕЛА

У раду је предложен нови систем управљања GAMINESS заснован на принципима Big Data. Концептуални модел GAMINESS је направљен коришћењем CityGML стандарда, који је проширен за стандардне елементе IndoorGML и SensorThings стандарда. Недостатак релационих база података је што не могу да категоришу неструктуриране податке. Постоји потреба да се ти хетерогени подаци прикупљају и структурирају у једном моделу. У том смислу, геопросторни подаци се разматрају заједно са новим техникама за прикупљање геопросторних података. Евидентно је да постоји огроман пораст тих података. Поред тога, растерски подаци прикупљени коришћењем различитих техника обезбеђују географске ознаке високе прецизности и позиционе резолуције. Уз њих су векторски подаци, који се обично користе за просторну анализу. У последњој деценији користе се технике масовне аквизиције геопросторних података, где су подаци облака тачака најчешћи тип за представљање геопросторних модела и комплексне анализе. Сви ови подаци захтијевају значајне рачунарске ресурсе, а постоји

и потреба да рачунари високих перформанси убрзају вертикалну обраду, што се одражава на техничке и финансијске аспекте апликација које раде са овим подацима. На нивоу паметног града, дигитални и имплементациони модели града могу бити повезани Интернетом ствари, формирајући тако интегрисани сајбер-физички простор [77], [118].

По дефиницији, паметни градови имају неку физичку репрезентацију простора са збирком података који описују различите појаве. Те природне појаве могу се описати подацима прикупљеним од различитих сензора, као што су геотехнички сензори. За разматрање концепта паметног града направљена је интеграција два подмодела: унутрашња навигација и укључивање сензора без проблема са разноврсношћу у процесу доношења одлука. У анализи поређења између CityGML LoD4 модела карактеристика и IndoorGML модела ћелијског простора, главна предност IndoorGML репрезентације је дата за дефиницију обрасца кретања и анализу контекста. Предности CityGML LoD4 модела су што има већу могућност визуелизације и геометријске анализе, али IndoorGML модел има боље проналажење ћелија, хијерархијску репрезентацију, дефиницију руте, анализу руте и анализу контекста [23]. За хостовање карактеристика геопросторних података, Oracle Spatial Graph платформа има уграђен механизам за геокодирање и рутирање, омогућавајући брзу имплементацију прилагођених решења у области рутирања и навигације. Вриједи напоменути да су Oracle Spatial и уграђени механизам за рутирање успешно коришћени за рутирање унутар IndoorGML (MLSEM) скупа података у сродним истраживањима[128]. IndoorGML изворни подаци би у почетку могли да се трансформишу помоћу FME. OpenTripPlanner (OTP) је био OSM механизам за рутирање одабран за употребу са нашим доказом имплементације концепта. У свом истраживању Јовановић и остали [129], [130] препоручили су 3DCityDB као платформу за складиштење трансформисаних и усвојених IndoorGML података, која обезбеђује добру повезаност са другим платформама за визуелизацију (ГИС и виртуелна реалност).

За имплементацију подмодула геосензора разматрана су два стандарда (IoT и SWE). SWE је скуп стандарда који не само да омогућава моделовање описа и запажања сензора, већ и специфицира веб сервисе за размену описа и запажања сензора на интероперабилан начин. Комбинација отворених стандарда (API) олакшава испоруку геопросторних карактеристика и сензорских опсервација на кохерентан начин и на тај начин подржава интероперабилне и међудомене градске услуге [23]. CityGML модел није у стању да представи временски зависна и динамичка својства. Омогућава чување својстава као статичких вриједности. Својства која варирају у времену могу бити варијације

просторних својстава, или варијације тематских атрибута, или варијације у односу на сензорске или податке у реалном времену. Постоји технолошки јаз када је у питању представљање променљивих својстава града или градских објеката [88]. Постоји много постојећих отворених стандарда дефинисаних за обједињавање интерфејса SW-IoT сервисног слоја. Хуанг ет ал. предложена отворена и интероперабилна SW-IoT архитектура од краја до краја заснована на OGC SensorThing API. Приједлог је заснован на три главне компоненте IoT-PNP: (1) фајл описа који описује метаподатке и могућности уређаја, (2) комуникациони протокол између слоја мрежног пролаза и слоја уређаја за успостављање веза и (3) аутоматски поступак регистрације и за сензорске и за задаће способности [88].

Други модел развијен за складиштење модела градова је CityJSON. Ово је формат који кодира подскуп CityGML модела података користећи JavaScript нотацију објеката (JSON). Проблем са CityJSON форматом је ограничење неких објеката који нису обухваћени овим моделом (Табела 3). Биљеcki и др. [54] је рекао да већина доступних 3D модела градова садржи много геометријских и тополошких грешака. Најчешће грешке су непотпуне површине, дупли врхови, запремине које се само пресецају, итд.

Систем управљања GAMINESS нуди коришћење алгоритма за аутоматску поправку заснованог на тополошким правилима CityGML. Релациона база података користи структуру која омогућава идентификацију и приступ подацима у односу на други податак у бази података. У релационој бази података, подаци су организовани у табеле. У CityGML моделу, једна или више класа UML дијаграма се често мапирају у једну табелу, при чему име табеле одговара имену класе. Класе се комбинују у једну табелу према односима класа описаним у UML дијаграму класа модела. Apache Spark оквир нуди језгро за моделирање података и процесе засноване на алгоритмима машинског учења. Apache Spark омогућава да се искористи јединствена способност да се поједностави машинско учење кроз једну технологију. GAMINESS користи Apache Spark Core, који садржи основну функционалност Spark, укључујући компоненте за планирање задатака, управљање меморијом, опоравак од грешке, интеракцију са системима за складиштење и још много тога. Apache Spark SQL пружа подршку за интеракцију са Apache Spark преко SQL. Apache Spark Streaming омогућава обраду тока података уживо. MLib пружа више типова алгоритама за машинско учење, укључујући бинарну класификацију, регресију, груписање и сарадничко филтрирање, поред подршке функционалности као што су евалуација модела и увоз података.

3DCityDB је релациони систем за управљање базама података који користи PostgreSQL као систем за управљање базама података за складиштење и администрирање података. За потребе паметног града, постоји потреба да се испитују велики 3D скупови података. У предложеном моделу, да би подаци мјерења са сензора били учитани у GAMINESS окружење развијени су специфични кориснички дефинисани типови како би ти подаци били разумљиви системи те адекватно структурирани у форми модела. На овакав начин је извршен развој сасвим новог технолошког проширења Apache Spark програмског оквира у односу на основни модул. Модели градова приказани у 3D форми, доступни су на различитим нивоима детаља и пружају адекватне опције складиштења за колекције 3D модела градова. Модели треба да буду стандардизовани и структурирани са својим семантичким подацима. Узимајући у обзир сву анализирану литературу и презентовану дискусију, систем управљања GAMINESS има неколико предности:

- Модел Big Data управљања паметним градом (GAMINESS) је јединствено решење које пружа могућност управљања системом паметног града у складу са концептима великих података.
- За разлику од постојећих решења (нпр. 3DCityDB), модел проширује приказ унутрашњег простора са параметрима навигације интеграцијом IndoorGML стандарда у постојећи CityGML стандард у окружењу Apache Spark.
- Модел представља једно од првих студија које комбинује моделе који узимају у обзир геопросторну компоненту која је мапирана у платформу великих података (предложени модел даје ефикасније резултате обраде података у облаку тачака од предложеног модела у оквиру пројекта IQUMULUS) .
- GAMINESS омогућава интеграцију геопросторних и података стримованих са сензора на основу различитих стандарда и њихово превођење у Apache Spark структуру уз одговарајућу конверзију.
- Модел обезбеђује да се сви подаци чувају у једном систему паметног града, који омогућава комбиновање и анализу података на нивоу ДБМС који обезбеђује сложеније комбинације података. Продавница на концепту великих података обезбеђује максималне перформансе постојећег система и много бржи одговор на процесе доношења одлука. Ово није могуће ни у једном ДБМС заснованом на паметном граду у бази података основних релација.

Као резултат овог истраживања, физички модел и подаци сензора се чувају заједно у систему управљања GAMINESS. Систем GAMINESS пружа бољу платформу за анализу директно из системског складишта. Платформа једноставно може да пружи елементе визуелизације, будући да су сви подаци складиштени у оквиру једног система, при чему се скраћује вријеме одзива између различитих платформи.

Студија случаја је истраживала ефекте заједничког увоза података облака тачака/мјерења са сензора унутар оквира кластера у односу на класични увоз у релационе базе података. Увоз је направљен на фиксном броју чворова уз задржавање фиксног обима података. Коришћен је мали скуп података који се састоји од 10 фајлова који садрже 1.000.000 опажања. Студијом је вршено поређење увоза података, структурирања из различитих извора истовремено коришћењем GAMINESS Apache Spark система за управљање паметним градом. У оквиру Табеле 12 представљени су резултати поређења између два традиционална система управљања базама података (стандардног PostgreSQL и 3DCityDB на PostgreSQL платформи) и система управљања паметним градом GAMINESS кроз перформансе 5V проблема. У контексту проблема обима, GAMINESS је обезбедио боље опције увоза/процеса користећи алгоритаме великих података и обраду мањих делова за увоз кроз систем. У контексту проблема брзине, GAMINESS је показао боље перформансе. Завршио је исти задатак осам пута брже од 3DCityDB и четири пута брже од PostgreSQL. У контексту проблема разноврсности, 3DCityDB и PostgreSQL би могли да увозе податке из облака тачака, али за друге формате података као што су подаци сензора, било би неопходно усвојити нова правила стандардизације. У контексту проблема варијабилности, GAMINESS је проширен према препорукама стандарда Дигитални близанци и Inspire правилима животног вијека. Ово је осигурало да подаци о паметном граду имају валидне информације о времену, док су 3DCityDB и PostgreSQL задржали само статичке репрезентације физичког простора града. У контексту проблема вредности, GAMINESS би могао да тражи податке користећи JSON поруке директно на сензорима. У 3DCityDB и PostgreSQL, било је могуће упити само сачуване податке. Табела 12 показује да је систем управљања GAMINESS имао боље перформансе у обради података од традиционалних система управљања базама података. Један од разлога за боље резултате је тај што GAMINESS омогућава раздвајање фајлова у кластере и врши паралелну обраду података.

ТАБЕЛА 12. Перформансе учитавања

5 V Parameter	3DcityDB PostgreSQL		GAMINESS
Volume	1 фајл (21 милион)	1 фајл (21 милион)	10 фајлова по (21 милион)
Velocity	260 ms	128 ms	30 ms
Variety	Импорт структурираног облака тачака у .las формату		Импорт структурираног облака тачака у .las формату/импорт података са сензора кроз поруке и структурирање коришћењем трансформационог модела у sensor RDD
Variability	Статичка презентација физичког простора града		Модел дефинисан по концепту Дигиталног близанца (временска компонента дефинисана кроз lifespan у RDD)
Value	Сегментација слојева учитаних података		Сегментација слојева учитаних података. Екстракција информација са учитаних података који су коришћени у упитима на JSON порукама са сензора

5.9. ДИСКУСИЈА

У истраживању предложен је систем за управљање просторно – временским серијама података базираном на Big data парадигми. За разлику од класичних релационих база података у складу са својим концептом Big data складишта у основи представљају базу података која је другачија у односу на класичне релационе базе података са напредним проширењима. Док су стандардне базе података ефикасне за чување и процесирање структурираних података, предност Big data концепта се огледа у томе што укључује и полуструктуриране и неструктуриране типове података у своју структуру. Постоји више типова NoSQL база података који процесирају и чувају Big data.

На основу досадашњих истраживања може се установити да је ова тема и даље актуелна и недовољно истражена. Полазни концепти који су дефинисани у оквиру овог истраживања, а који се односе на успостављање паметног града у контексту представљања просторних карактеристика, верификовани су истраживањима у [129], [131]. Установљено је да једино библиотека Geomesa раду с просторно – временским типом података и то у ограниченом оквиру. Дио модела GAMINESS система за управљање подацима, верификован је у оквиру [117], [126], [127]. Паметни градови у стварном свијету су подржани мјешавином парадигми које користе мјешавину архитектура (централизоване, децентрализоване или комбинацију обје) и инфраструктуре попут middleware и IoT платформе за подршку креирању апликација за паметне градове. Ово истраживање је поставило IoT технологију у централни дио проширења геосензорског пакета. Да ова технологија мора бити саставни дио изградње паметног града верификовано је у [132], [133].

Креирани систем за трансформацију великих количина података у систем заснован на прикупљању, управљању и анализи великих количина података у Big data контексту, а

који има унапријеђено рјешење за проблем варијабилности у контексту просторно – временских типова података верификован је у [23], [24]. Као што је споменуто, једна од главних карактеристика модела паметног града је његова способност да генерише предузетничке активности. Да би успоставили паметни град неопходно је верификовати области које чине овај концепт и направити систем који ће их у том смислу и користити. Овај концепт је подржан у следећим истраживањима: [117], [133], [134]. У Табели 14 је приказана верификација предложеног модела. Комплетан систем је пројектован у оквиру више истраживања у својим појединим или цјелокупним дијеловима. У табели 14 је дат преглед истраживања која су користила овај концепт, па се стога може сматрати својеврсном верификацијом предложеног модела. Разлика у односу на доступна рјешења је у томе да је велики број приједлога концепата паметних градова у Big Data окружењу, остао на концептуалном нивоу. Ово рјешење осим предложеног концепта је понудило и верификацију имплементацијом система у Big Data Apache Spark окружењу. Када се посматрају постојећа рјешења у Big data парадигми, најчешће подразумијева рјешавање једног од 5V проблема проблема. У том смислу нарочит допринос се види у анализи и приступу рјешавања свих 5V проблема коришћењем предложене технологије.

ТАБЕЛА 13. Понуђена рјешења у односу на досадашња истраживања у области Big Data Smart City концепта према 5V

ПРОБЛЕМ	ИСТРАЖИВАЊЕ У КОНТЕКСТУ BIG DATA SMART CITY	ОГРАНИЧЕЊЕ У КОНТЕКСТУ BIG DATA SMART CITY	РЈЕШЕЊЕ У ОКВИРУ GAMINESS СИСТЕМА
Проблеми са количинама података (volume)	Zhou и други [76]	Нема могућност учитавања и рада са комплексним геометријама.	Могућност учитавања и рад са комплексним геометријама.
	Voehn и други [77]	Могућност учитавања и процесирања облака тачака само над основним класама.	Могућност учитавања и процесирања облака тачака под класама дефинисаним CityGML стандардом.
	Yu и други [78]	Користи Big data окружење само за дио аналитичких процеса. Нема могућност рада са комплексним геометријама.	Могућност учитавања и рад са комплексним геометријама.
	Bibri и други [12]	Нема могућност учитавања облака тачака и рада са тим скупом података.	Могућност учитавања и процесирања облака тачака под класама дефинисаним CityGML стандардом.
	Liu и други [79]	Нема могућност учитавања и рада са комплексним геометријама.	Могућност учитавања и рад са комплексним геометријама.
	Kang и други [80]	Нема могућност раслојавања облака тачака у структуре класа које су описане у складу	Могућност учитавања и процесирања облака тачака под класама дефинисаним CityGML стандардом.

		са CityGML стандардом.	
	Boehm и други [81]	Нема могућност учитавања и раслојавања облака тачака у структуре класа које су описане у складу са CityGML стандардом.	Могућност учитавања и процесирања облака тачака под класама дефинисаним CityGML стандардом.
	Alis и други [82]	Нема могућност учитавања и раслојавања облака тачака у структуре класа које су описане у складу са CityGML стандардом.	Могућност учитавања и процесирања облака тачака под класама дефинисаним CityGML стандардом.
	Dean и други [83]	Нема могућност учитавања и рада са комплексним геометријама.	Могућност учитавања и рад са комплексним геометријама.
	IQUMULUS [84]	Нема могућност учитавања и раслојавања облака тачака у структуре класа које су описане у складу са CityGML стандардом.	Могућност учитавања и процесирања облака тачака под класама дефинисаним CityGML стандардом.
Проблеми са разноликошћу података (variety)	Kim и други [24]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима.
	Claridades и други [85]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима.
	Hashem и други [6]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима.
	OGC Sensor Web Enablement Initiative [55]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима.
	Sundaram и други [86]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима.
	Chaturvedia и други [87]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима.
	Huang и други [88]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима.
	Gaur и други [89]	Коришћен као окружење само за обраду дијела података.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окружење.
	Saraswathi и други [90]	Коришћен као окружење само за обраду дијела података.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala

		Freedman и други [91]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	окужење. Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
Проблеми са брзином података (velocity)		Biļeckī и други [54]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
		Yao и други [92]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
		Hijazi и други [93]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
		Zhu и други [94]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
		52 ° North [95]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
		Ledoux и други [96]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
Проблеми са варијабилношћу података (variability)		Liu и други [77]	Нема развијену структуру за учитавање просторно – временских података.	Могућност учитавања и процесирања облака тачака под класамa дефинисаним CityGML стандардом и стриминг података са сензора под различитим стандардима формирања излазне поруке.
Проблеми са количином података (volume)		DABAMOS [97]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
		Engel и други [98]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
		Bibri и други [12]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
		InfluxDB [99]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.
	Druid [100]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.	
	NodeRed [101], [102]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.	
	Kodali и други [102]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.	
	Lin и други [103]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.	
	Ammar и други [68]	Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окужење.	
			Рјешење није базирано	Рјешење је у потпуности

на Big Data концептима, већ користи стандардне релационе системе.	базирано на Big Data концептима и користи Apache Spark Scala окружење.
Рјешење није базирано на Big Data концептима, већ користи стандардне релационе системе.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окружење.
Има само обезбијеђен дјелимичан сервис заснован на Big Data окружењу.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala Mlib Apache Spark Streaming окружење проширено са новим кориснички дефинисаним типовима и функцијама.
Има само обезбијеђен дјелимичан сервис заснован на Big Data окружењу. Нема подршку за просторне податке.	Могућност учитавања и рад са комплексним геометријама.
Има само обезбијеђен дјелимичан сервис заснован на Big Data окружењу. Нема подршку за просторне податке.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala Mlib Apache Spark Streaming окружење проширено са новим кориснички дефинисаним типовима и функцијама.
Има само обезбијеђен дјелимичан сервис заснован на Big Data окружењу. Нема подршку за просторне податке.	Рјешење је у потпуности базирано на Big Data концептима и користи Apache Spark Scala окружење и има могућност рада са комплексним геометријама.

У складу са верификацијом модела могуће је уочити да се посебан допринос огледа у оправданости опште хипотезе којом је успостављен нови модел управљања просторно-временским подацима у паметном граду у контексту Big data парадигме који омогућава геопроецесуирање великих количина просторно-временских података и њихову ефикаснију аналитичку обраду. У Табели 13 је на основу 5V параметара експлицитно се види предност ове поставке у односу на стандардне релационе системе. Предмет поређења је систем који користи сличне ресурсе у погледу типа и количине података. Ипак, треба напоменути да је тестирање извршено у ограниченим условима рачунарске лабораторије. Да је подешавање кластер система извршено у локалу и да је извршена оптимизација у погледу доступних ресурса. Предност оваквих система би се додатно

могла огледати у cloud процесирању. Приликом формирања тестног окружења коришћене су препоруке из истраживања Пајић [126].

Општа хипотеза је верификована кроз успостављање основног GAMINESS модела. Наиме, у концептуалном моделу је дефинисано проширење основног CityGML стандарда којим је омогућено формирање одговарајуће структуре која ће моћи адекватно прочитати варијабилне улазе у смислу различитог описа и шеме улазних података. На ниову имплементације је ријешено формирањем сложених кориснички дефинисаних типова који су омогућили прављење адекватног DataFrame у GAMINESS окружењу чиме су оправдана прва посебна хипотеза.

Дефинисање нових кориснички дефинисаних типова података као што је коришћени тип за Point или Sensor, омогућава превођење података из полуструктуриране форме у структурирану форму GAMINESS модела. На овакав начин су оправдане друга и трећа посебна хипотеза. Будући да је вршена оптимизација коришћења GAMINESS рјешења у односу на класичне системе, може се видјети да се водило рачуна и о количини записаних података у Big Data складишту у оквиру коришћеног продукционог окружења. Сви атрибути тачке су записани у једној колони коришћењем структуре кориснички дефинисаног типа. Због тога су развијане посебне кориснички дефинисане функције базиране за серијализацију / десеријализацију приликом сваког учитавања / ишчитавања. Такође за све врсте логичких упита на основу атрибута, неопходно је било дефинисати кориснички дефинисане функције. Међутим, такав приступ је омогућио боље перформансе у смислу временског одзива система у односу на класичне упите када је ријеч о геопросторним процесирањима и обради података. Овакав приступ је детаљено приказан у процесу коришћења основног концепта машинског учења проширењем основних mlib и Apache Spark Streaming библиотека на конкретном примјеру.

У оквиру овог истраживања извршено је проширење постојећих технологија за нове концепте који су уведени у Apache Spark Big Data окружење. Успостављен је нови систем, који напушта стандардне системе за чување података у Big Data контексту као што су Cassandra или Hive због њихових ограничења у погледу рада са просторно – временским типовима података, а са друге стране предложеним моделом су добијене предности коришћења Apache Spark технологије. Такође, до сада није постојало ниједно истраживање које је у обзир узимало CityGML стандард у његовом пуном обиму у контексту Big Data. Креиран је прелазни модел који је од имплементационе JSON шеме овог стандарда са дефинисаним проширењима креирао одговарајуће DataFrame оквире.

6. ЗАКЉУЧАК

У истраживању предложен је систем за управљања паметним градом, који се заснива на принципима парадигме великих података. На концептуалном нивоу извршено је унапређење постојећег CityGML модела, који је у складу са верзијом CityGML 3.3 и даљим проширењима. Побољшања су дата за Dynamizers, где је направљено проширење постојећег модела за IoT стандард у контексту паметног града. Систем управљања GAMINESS обезбеђује проширење основног модела LoD4 за параметре IndoorGML стандарда у контексту унутрашње навигације што се огледа кроз боље проналажење ћелија, хијерархијско представљање, дефиницију руте, анализу руте и анализу контекста.

На основу дефинисаног концептуалног модела система управљања GAMINESS, имплементација система је урађена у Apache Spark платформи за велике податке. Један од главних доприноса овог система је дефинисан модел за трансформацију система у Apache Spark архитектуру. Направљена је библиотека са процедурама за трансформацију концептуалног модела, који је експортиран као JSON шема у адекватне DataFrame. Да би се то обезбедило, развијени су адекватни сложени кориснички дефинисани типови за читање геопросторних података. Ово је омогућило читање геопросторних података у оквиру GAMINESS. Систем управљања GAMINESS је заснован на mapreduce алгоритмима, који обезбеђују бољих 5V параметара без обзира на класичне релационе системе управљања базама података. У класичном систему управљања базама података, постоји потреба за вертикалним убрзањем да би се обезбедиле боље перформансе на великим скуповима података. На нивоу имплементације, у систему управљања GAMINESS, наставак истраживања је описан у Пајић и остали, где је развијен модел за читање класификованих података облака тачака у Apache Spark окружење. Овај модел се користи за читање и уписивање података облака тачака у геопросторне оквиру података како би могли да рукују и формирају физичке моделе паметног града. Користећи сличан концепт у овом истраживању, развијен је метод за читање и структурирање података са различитих сензора на сензор DataFrame. Развијени алгоритам трансформације је коришћен за трансформацију порука са сензора кроз JSON поруке које се читају и чувају у сензорском RDD. У овом систему, сваки сензор је повезан на независном чвору, тако да постоји могућност да се на сваком чвору направе различити улазни услови кроз упит који утиче на JSON поруку која се трансформише у RDD.

Учитавање података линеарно даје боље перформансе са повећањем обима података. Резултати дају закључак да Apache Spark програмски оквир најбоље резултате даје на

максималној величини података. Резултати експеримента се такође могу тумачити као проток података, где се милиони тачака уносе у секунди. Општи закључак је да предложени систем добро ради са великим скуповима података и да ове концепте једино и има смисла користити када се у обзир узимају велике количине података. Једино тада су уочљиве боље перформансе система у односу на нека конвенционална рјешења.

Предложено решење се лако може проширити додатним операцијама на облацима тачака и сензорским подацима кроз имплементацију кориснички дефинисаних функција, различитих сегментација и екстракција карактеристика. Резултати студије случаја показују да Apache Spark ради боље од класичне релационе базе података са великим скуповима података облака тачака. Главни разлог за боље перформансе је паралелно извршавање упита. Због хоризонталне скалабилности Apache Spark, време извршења може се побољшати додавањем више рачунара у кластер. Наредна истраживања би требала да ставе фокус на проширење система за следеће области:

- креирање новог типа својства за различите типове сензора;
- побољшања модула чворова за независно повезивање сензора на систем;
- истраживање коришћења четвородимензионалног простора за динамичке облаке тачака;
- проширење модела како би се омогућила просторна спајања са оквирима података векторских геопросторних података;
- развој кориснички дефинисаних функција са графичким интерфејсом за креирање упита.

Предложени модел проширења је на адекватан начин документовао постојећа рјешења у области паметних градова. Изабран је CityGML као најсвеобухватнији стандард који описује систем паметног града. Предложена рјешења су дала резултате на нивоу верификације која су оправдала изградњу GAMINESS система на концепту управљања просторно – временским подацима у Big Data парадигми гдје је комплетна технолошка платформа развијена у овом оквиру што представља јединствено рјешење.

7. ЛИТЕРАТУРА

- [1] E. al Nuaimi, H. al Neyadi, N. Mohamed, and J. Al-Jaroodi, “Applications of big data to smart cities,” *Journal of Internet Services and Applications*, vol. 6, no. 1, p. 25, Aug. 2015, doi: 10.1186/s13174-015-0041-5.
- [2] Z. Zheng, P. Wang, J. Liu, and S. Sun, “Real-time big data processing framework: Challenges and solutions,” *Applied Mathematics and Information Sciences*, vol. 9, pp. 3169–3190, Jan. 2015, doi: 10.12785/amis/090646.
- [3] J. Espinosa, S. Kaisler, F. Armour, and W. Money, *Big Data Redux: New Issues and Challenges Moving Forward*. 2019. doi: 10.24251/HICSS.2019.131.
- [4] U. Sivarajah, M. Kamal, Z. Irani, and V. Weerakkody, “Critical analysis of Big Data challenges and analytical methods,” *J Bus Res*, vol. 70, Aug. 2016, doi: 10.1016/j.jbusres.2016.08.001.
- [5] L. Rodríguez, C. Rodriguez-Enriquez, J. Sánchez-Cervantes, J. Cervantes, J. García-Alcaraz, and G. Alor-Hernández, “A general perspective of Big Data: applications, tools, challenges and trends,” *J Supercomput*, vol. 72, Aug. 2015, doi: 10.1007/s11227-015-1501-1.
- [6] I. A. T. Hashem *et al.*, “The role of big data in smart city,” *Int J Inf Manage*, vol. 36, no. 5, pp. 748–758, 2016, doi: <https://doi.org/10.1016/j.ijinfomgt.2016.05.002>.
- [7] D. Jelonek, “Big Data Analytics in the Management of Business,” *MATEC Web of Conferences*, vol. 125, p. 04021, Jan. 2017, doi: 10.1051/mateconf/201712504021.
- [8] M. Kubina, M. Varmus, and I. Kubinova, “Use of Big Data for Competitive Advantage of Company,” *Procedia Economics and Finance*, vol. 26, pp. 561–565, Oct. 2015, doi: 10.1016/S2212-5671(15)00955-7.
- [9] “Apache Spark™ - Unified Engine for large-scale data analytics.” <https://spark.apache.org/> (accessed Feb. 09, 2023).
- [10] S. Nittel, A. Labrinidis, and A. Stefanidis, *GeoSensor Networks: Second International Conference, GSN 2006, Boston, MA, USA, October 1-3, 2006, Revised Selected and Invited Papers*. 2008. doi: 10.1007/978-3-540-79996-2.
- [11] M. Breunig *et al.*, “Geospatial Data Management Research: Progress and Future Directions,” *ISPRS Int J Geoinf*, vol. 9, p. 95, Feb. 2020, doi: 10.3390/ijgi9020095.
- [12] S. Bibri and J. Krogstie, *The Big Data Deluge for Transforming the Knowledge of Smart Sustainable Cities: A Data Mining Framework for Urban Analytics*. 2018. doi: 10.1145/3286606.3286788.
- [13] M. El-Hallaq, A. Alastal, and R. Salha, “Enhancing Sustainable Development through Web Based 3D Smart City Model Using GIS and BIM. Case Study: Sheikh Hamad City,” *Journal of Geographic Information System*, vol. 11, pp. 321–330, Jan. 2019, doi: 10.4236/jgis.2019.113019.

- [14] R. Zelenika, *Metodologija i tehnologija izrade znanstvenog i stručnog djela*, 4. izd. Rijeka: Ekonomski fakultet.
- [15] Ž. Adamović, Ж. Адамовић, and А. Ашоња, *Metodologija naučno-istraživačkog rada: naukologija, metodologija, tehnologija*. Novi Sad: Srpski akademski centar, 2014.
- [16] G. Lafortune, G. Fuller, G. Schmidt-Traub, and C. Kroll, "How Is Progress towards the Sustainable Development Goals Measured? Comparing Four Approaches for the EU," *Sustainability*, vol. 12, p. 7675, Sep. 2020, doi: 10.3390/su12187675.
- [17] "Cities - United Nations Sustainable Development Action 2015." <https://www.un.org/sustainabledevelopment/cities/> (accessed Feb. 09, 2023).
- [18] B. Silva, M. Khan, and K. Han, "Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities," *Sustain Cities Soc*, vol. 38, Feb. 2018, doi: 10.1016/j.scs.2018.01.053.
- [19] A. Salina and K. Y. R. Rao, "A Study on Tools of Big Data Analytics," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 4, pp. 18815–18826, Oct. 2016, doi: 10.15680/IJIRCCE.2016/0410149.
- [20] A. B. M. Moniruzzaman and S. Hossain, "NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison," *Int J Database Theor Appl*, vol. 6, Jun. 2013.
- [21] F. Prandi *et al.*, "Using citygml to deploy smart-city services for urban ecosystems," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. Volume XL-4/W1, May 2013, doi: 10.5194/isprsarchives-XL-4-W1-87-2013.
- [22] F. Biljecki, K. Kumar, and C. Nagel, "CityGML Application Domain Extension (ADE): overview of developments," *Open Geospatial Data, Software and Standards*, vol. 3, p. 13, Aug. 2018, doi: 10.1186/s40965-018-0055-6.
- [23] H. G. Ryoo, T. Kim, and K.-J. Li, *Comparison between two OGC standards for indoor space: CityGML and IndoorGML*. 2015. doi: 10.1145/2834812.2834813.
- [24] J.-S. Kim, S.-J. Yoo, and K.-J. Li, "Integrating IndoorGML and CityGML for Indoor Space," in *Web and Wireless Geographical Information Systems*, 2014, pp. 184–196.
- [25] S. Alawadhi *et al.*, "Building Understanding of Smart City Initiatives," in *Electronic Government*, 2012, pp. 40–53.
- [26] R. Carli, M. Dotoli, R. Pellegrino, and L. Ranieri, *Measuring and Managing the Smartness of Cities: A Framework for Classifying Performance Indicators*. 2013. doi: 10.1109/SMC.2013.223.
- [27] M. H. Moore, *Creating public value : strategic management in government / Mark H. Moore*. Cambridge, Mass: Harvard University Press, 1995.
- [28] A. Krylovskiy, M. Jahn, and E. Patti, *Designing a Smart City Internet of Things Platform with Microservice Architecture*. 2015. doi: 10.1109/FiCloud.2015.55.

- [29] L. Hernández-Callejo *et al.*, “A Survey on Electric Power Demand Forecasting: Future Trends in Smart Grids, Microgrids and Smart Buildings,” *Communications Surveys & Tutorials, IEEE*, vol. 16, pp. 1460–1495, Apr. 2014, doi: 10.1109/SURV.2014.032014.00094.
- [30] R. Kingston and J. Cauvain (nee Viitanen), “Smart Cities and Green Growth: Outsourcing Democratic and Environmental Resilience to the Global Technology Sector,” *Environ Plan A*, vol. 46, pp. 803 – 819, Apr. 2015, doi: 10.1068/a46242.
- [31] P. Chamoso, A. González-Briones, S. Rodríguez, and J. M. Corchado, “Tendencies of Technologies and Platforms in Smart Cities: A State-of-the-Art Review,” *Wirel Commun Mob Comput*, vol. 2018, p. 3086854, 2018, doi: 10.1155/2018/3086854.
- [32] I. Cabral, S. Costa, U. Weiland, and A. Bonn, “Urban Gardens as Multifunctional Nature-Based Solutions for Societal Goals in a Changing Climate,” in *Nature-Based Solutions to Climate Change Adaptation in Urban Areas: Linkages between Science, Policy and Practice*, N. Kabisch, H. Korn, J. Stadler, and A. Bonn, Eds. Cham: Springer International Publishing, 2017, pp. 237–253. doi: 10.1007/978-3-319-56091-5_14.
- [33] K. Nowicka, “Smart City Logistics on Cloud Computing Model,” *Procedia Soc Behav Sci*, vol. 151, pp. 266–281, Oct. 2014, doi: 10.1016/j.sbspro.2014.10.025.
- [34] T. Jeske, “Floating Car Data from Smartphones : What Google and Waze Know About You and How Hackers Can Control Traffic,” 2013.
- [35] Y. Liu, “Analysis of Government Public Management Information Service and Computer Model Construction Based on Smart City Construction,” *Math Probl Eng*, vol. 2022, pp. 1–9, Mar. 2022, doi: 10.1155/2022/4544439.
- [36] T. A. Nguyen and M. Aiello, “Energy intelligent buildings based on user activity: A survey,” *Energy Build*, vol. 56, pp. 244–257, 2013, doi: <https://doi.org/10.1016/j.enbuild.2012.09.005>.
- [37] C. Wilson, T. Hargreaves, and R. Hauxwell-Baldwin, “Smart homes and their users: A systematic analysis and key challenges,” *Pers Ubiquitous Comput*, vol. 19, pp. 463–476, Feb. 2014, doi: 10.1007/s00779-014-0813-0.
- [38] U. Gretzel, M. Sigala, Z. Xiang, and C. Koo, “Smart tourism: foundations and developments,” *Electronic Markets*, vol. 25, Aug. 2015, doi: 10.1007/s12525-015-0196-8.
- [39] C. Koo, F. Ricci, C. Cobanoglu, and F. Okumus, “Special issue on smart, connected hospitality and tourism,” *Information Systems Frontiers*, vol. 19, no. 4, pp. 699–703, 2017, doi: 10.1007/s10796-017-9776-9.
- [40] Z. Khan, Z. Pervez, and A. Abbasi, *Towards Cloud Based Smart Cities Data Security and Privacy Management*. 2014.
- [41] A. Solanas *et al.*, “Smart Health: A Context-Aware Health Paradigm within Smart Cities,” *IEEE Communications Magazine*, vol. 52, pp. 74–81, Aug. 2014, doi: 10.1109/MCOM.2014.6871673.

- [42] A. Hussain, R. Wenbi, A. L. da Silva, M. Nadher, and M. Mudhish, "Health and Emergency-Care Platform for the Elderly and Disabled People in the Smart City," *J. Syst. Softw.*, vol. 110, no. C, pp. 253–263, Dec. 2015, doi: 10.1016/j.jss.2015.08.041.
- [43] J. Borràs, A. Moreno, and A. Valls, "Intelligent tourism recommender systems: A survey," *Expert Syst. Appl.*, vol. 41, pp. 7370–7389, 2014.
- [44] T. Clohessy, T. Acton, and L. Morgan, "Smart City as a Service (SCaaS): A Future Roadmap for E-Government Smart City Cloud Computing Initiatives," *Proceedings - 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, UCC 2014*, pp. 836–841, Jan. 2015, doi: 10.1109/UCC.2014.136.
- [45] M. Janssen and J. van den hoven, "Big and Open Linked Data (BOLD) in Government: A Challenge to Transparency and Privacy?," *Gov Inf Q*, vol. 32, pp. 363–368, Oct. 2015, doi: 10.1016/j.giq.2015.11.007.
- [46] B. Ridel, P. Reuter, J. Laviolle, N. Mellado, N. Couture, and X. Granier, "The Revealing Flashlight: Interactive Spatial Augmented Reality for Detail Exploration of Cultural Heritage Artifacts," *J. Comput. Cult. Herit.*, vol. 7, no. 2, Jun. 2014, doi: 10.1145/2611376.
- [47] T. di Mascio, P. Vittorini, R. Gennari, A. Melonio, F. D. la Prieta, and M. Alrifai, "The Learners' User Classes in the TERENCE Adaptive Learning System," in *2012 IEEE 12th International Conference on Advanced Learning Technologies*, 2012, pp. 572–576. doi: 10.1109/ICALT.2012.68.
- [48] P. Chamoso, A. Rivas, S. Rodríguez, and J. Bajo, "Relationship recommender system in a business and employment-oriented social network," *Inf Sci (N Y)*, vol. 433–434, pp. 204–220, Apr. 2018, doi: 10.1016/j.ins.2017.12.050.
- [49] O. Said and M. Masud, "Towards Internet of Things: Survey and Future Vision," *International Journal of Computer Networks*, vol. 5, pp. 1–17, Feb. 2013.
- [50] G. Gröger and L. Plümer, "CityGML – Interoperable semantic 3D city models," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 71, pp. 12–33, 2012, doi: <https://doi.org/10.1016/j.isprsjprs.2012.04.004>.
- [51] T. H. Kolbe, "Representing and Exchanging 3D City Models with CityGML," in *3D Geo-Information Sciences*, J. Lee and S. Zlatanova, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 15–31. doi: 10.1007/978-3-540-87395-2_2.
- [52] P. Oosterom *et al.*, "Massive point cloud data management: Design, implementation and execution of a point cloud benchmark," *Comput Graph*, vol. 49, Feb. 2015, doi: 10.1016/j.cag.2015.01.007.
- [53] J.-M. Bahu, A. Koch, E. Kremers, and S. M. Murshed, "Towards a 3D Spatial Urban Energy Modelling Approach," *Int. J. 3 D Inf. Model.*, vol. 3, pp. 1–16, 2013.

- [54] F. Biljecki, J. Stoter, H. Ledoux, S. Zlatanova, and A. Çöltekin, “Applications of 3D City Models: State of the Art Review,” *ISPRS Int J Geoinf*, vol. 4, no. 4, pp. 2842–2889, 2015, doi: 10.3390/ijgi4042842.
- [55] “The Home of Location Technology Innovation and Collaboration | OGC.” <https://www.ogc.org/> (accessed Feb. 09, 2023).
- [56] “INSPIRE Directive | INSPIRE.” <https://inspire.ec.europa.eu/inspire-directive/2> (accessed Feb. 09, 2023).
- [57] Y. Lim, J. Edelenbos, and A. Gianoli, “Identifying the results of smart city development: Findings from systematic literature review,” *Cities*, vol. 95, p. 102397, Dec. 2019, doi: 10.1016/j.cities.2019.102397.
- [58] “Web Map Service | OGC.” <https://www.ogc.org/standards/wms> (accessed Feb. 09, 2023).
- [59] “Web Feature Service | OGC.” <https://www.ogc.org/standards/wfs> (accessed Feb. 09, 2023).
- [60] “Web Processing Service | OGC.” <https://www.ogc.org/standards/wps> (accessed Feb. 09, 2023).
- [61] “ISO - ISO 19101-1:2014 - Geographic information — Reference model — Part 1: Fundamentals.” <https://www.iso.org/standard/59164.html> (accessed Feb. 09, 2023).
- [62] “ISO - ISO 19103:2015 - Geographic information — Conceptual schema language.” <https://www.iso.org/standard/56734.html> (accessed Feb. 09, 2023).
- [63] “ISO - ISO 19107:2019 - Geographic information — Spatial schema.” <https://www.iso.org/standard/66175.html> (accessed Feb. 09, 2023).
- [64] “ISO - ISO 19108:2002 - Geographic information — Temporal schema.” <https://www.iso.org/standard/26013.html> (accessed Feb. 09, 2023).
- [65] “ISO - ISO 19109:2015 - Geographic information — Rules for application schema.” <https://www.iso.org/standard/59193.html> (accessed Feb. 09, 2023).
- [66] “ISO - ISO 19111:2019 - Geographic information — Referencing by coordinates.” <https://www.iso.org/standard/74039.html> (accessed Feb. 09, 2023).
- [67] “ISO - ISO 19115-2:2019 - Geographic information — Metadata — Part 2: Extensions for acquisition and processing.” <https://www.iso.org/standard/67039.html> (accessed Feb. 09, 2023).
- [68] M. Ammar, G. Russello, and B. Crispo, “Internet of Things: A survey on the security of IoT frameworks,” *Journal of Information Security and Applications*, vol. 38, pp. 8–27, Feb. 2018, doi: 10.1016/j.jisa.2017.11.002.
- [69] A. B. Chebudie, R. Minerva, and D. Rotondi, “Towards a definition of the Internet of Things (IoT),” 2014.

- [70] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context Aware Computing for The Internet of Things: A Survey," *IEEE Communications Surveys & Tutorials*, May 2013, doi: 10.1109/SURV.2013.042313.00197.
- [71] J. Kanter and K. Veeramachaneni, *Deep feature synthesis: Towards automating data science endeavors*. 2015. doi: 10.1109/DSAA.2015.7344858.
- [72] M. Vrtunski, "Model geosenzorske mreže za monitoring terena i objekata u realnom vremenu," *Универзитет у Новом Саду*, Oct. 2018, Accessed: Feb. 09, 2023. [Online]. Available: <https://nardus.mpn.gov.rs/handle/123456789/10397>
- [73] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Realtime Data Systems*, 1st ed. USA: Manning Publications Co., 2015.
- [74] D. Laney, "3D Data Management: Controlling Data Volume, Velocity, and Variety," Feb. 2001. [Online]. Available: <http://blogs.gartner.com/doug-laney/files/2012/01/ad949-3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>
- [75] J. S. Ward and A. Barker, "Undefined By Data: A Survey of Big Data Definitions," *ArXiv*, vol. abs/1309.5821, 2013.
- [76] Z. Huang, Y. Chen, L. Wan, and X. Peng, "GeoSpark SQL: An Effective Framework Enabling Spatial Queries on Spark," *ISPRS Int J Geoinf*, vol. 6, p. 285, Sep. 2017, doi: 10.3390/ijgi6090285.
- [77] K. Liu and J. Boehm, "CLASSIFICATION OF BIG POINT CLOUD DATA USING CLOUD COMPUTING," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3/W3, pp. 553–557, 2015, doi: 10.5194/isprsarchives-XL-3-W3-553-2015.
- [78] J. Yu, J. Wu, and M. Sarwat, *GeoSpark: a cluster computing framework for processing large-scale spatial data*. 2015. doi: 10.1145/2820783.2820860.
- [79] K. Liu, J. Boehm, and C. Alis, "Change detection of mobile LIDAR data using cloud computing," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B3, pp. 309–313, Jun. 2016, doi: 10.5194/isprs-archives-XLI-B3-309-2016.
- [80] K.-J. Li, T.-H. Kim, H.-G. Ryu, and H.-K. Kang, "Comparison of CityGML and IndoorGML -A Use-Case Study on Indoor Spatial Information Construction at Real Sites-," *Journal of Korea Spatial Information Society*, vol. 23, pp. 91–101, Aug. 2015, doi: 10.12672/ksis.2015.23.4.091.
- [81] J. Boehm, K. Liu, and C. Alis, "Sideloadng - Ingestion of large point clouds into the Apache Spark big data engine," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B2, pp. 343–348, Jun. 2016, doi: 10.5194/isprs-archives-XLI-B2-343-2016.
- [82] C. Alis, J. Boehm, and K. Liu, "Parallel processing of big point clouds using Z-Order-based partitioning," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and*

- Spatial Information Sciences*, vol. XLI-B2, pp. 71–77, Jun. 2016, doi: 10.5194/isprs-archives-XLI-B2-71-2016.
- [83] J. Dean and S. Ghemawat, *MapReduce: Simplified Data Processing on Large Clusters*, vol. 51. 2004. doi: 10.1145/1327452.1327492.
- [84] “IQumulus® --> Industrial Machine Learning | The Data Science Experts.” <https://iqumulus.com/> (accessed Feb. 09, 2023).
- [85] A. R. Claridades, I. Park, and J. Lee, “Integrating IndoorGML and Indoor POI Data for Navigation Applications in Indoor Space,” *Journal of the Korean Society of Surveying Geodesy Photogrammetry and Cartography*, vol. 2019, pp. 359–366, Nov. 2019, doi: 10.7848/ksgpc.2019.37.5.359.
- [86] S. Sundaram, “Improvement of HADOOP Ecosystem and Their Pros and Cons in Big Data,” *International Journal Of Engineering And Computer Science*, May 2016, doi: 10.18535/ijecs/v5i5.57.
- [87] K. Chaturvedi and T. H. Kolbe, “INTEGRATING DYNAMIC DATA AND SENSORS WITH SEMANTIC 3D CITY MODELS IN THE CONTEXT OF SMART CITIES,” *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-2/W1, pp. 31–38, 2016, doi: 10.5194/isprs-annals-IV-2-W1-31-2016.
- [88] C.-Y. Huang and H.-H. Chen, “An Automatic Embedded Device Registration Procedure Based on the OGC SensorThings API,” *Sensors*, vol. 19, p. 495, Jan. 2019, doi: 10.3390/s19030495.
- [89] S. Gaur and G. Gupta, “Framework for Monitoring and Recognition of the Activities for Elderly People from Accelerometer Sensor Data Using Apache Spark,” 2020, pp. 734–744. doi: 10.1007/978-981-15-1420-3_79.
- [90] A. Saraswathi, M. A, A. R, and K. Porkodi, *Real-Time Traffic Monitoring System Using Spark*. 2019. doi: 10.1109/ICESE46178.2019.9194613.
- [91] “Spark Streaming and IoT by Mike Freedman.” <https://www.slideshare.net/SparkSummit/spark-streaming-and-iot-by-mike-freedman> (accessed Feb. 09, 2023).
- [92] Z. Yao *et al.*, “3DCityDB - a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML,” *Open Geospatial Data, Software and Standards*, vol. 3, May 2018, doi: 10.1186/s40965-018-0046-7.
- [93] I. Hijazi, T. Krauth, A. Donaubaue, and T. Kolbe, “3DCITYDB4BIM: A SYSTEM ARCHITECTURE FOR LINKING BIM SERVER AND 3D CITYDB FOR BIM-GIS-INTEGRATION,” *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-4–2020, pp. 195–202, Aug. 2020, doi: 10.5194/isprs-annals-V-4-2020-195-2020.
- [94] W. Zhu, A. Simons, S. Wursthorn, and A. Nichersu, *Integration of CityGML and Air Quality Spatio-Temporal Data Series via OGC SOS*. 2016.

- [95] “Home - 52°North Spatial Information Research GmbH.” <https://52north.org/> (accessed Feb. 09, 2023).
- [96] H. Ledoux, K. Otori, K. Kumar, B. Dukai, A. Labetski, and S. Vitalis, “CityJSON: a compact and easy-to-use encoding of the CityGML data model,” *Open Geospatial Data Software and Standards*, vol. 4, p. 4, Jun. 2019, doi: 10.1186/s40965-019-0064-0.
- [97] “Home | DABAMOS.de.” <https://www.dabamos.de/> (accessed Feb. 09, 2023).
- [98] P. Engel and B. Schweimler, “Development of an Open-Source Automatic Deformation Monitoring System for Geodetical and Geotechnical Measurements,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. XL-5/W8, pp. 25–30, Apr. 2016, doi: 10.5194/isprs-archives-XL-5-W8-25-2016.
- [99] “InfluxDB Times Series Data Platform | InfluxData.” <https://www.influxdata.com/> (accessed Feb. 09, 2023).
- [100] “Druid | Database for modern analytics applications.” <https://druid.apache.org/> (accessed Feb. 09, 2023).
- [101] “Node-RED.” <https://nodered.org/> (accessed Feb. 09, 2023).
- [102] R. Kodali and A. Anjum, *IoT Based HOME AUTOMATION Using Node-RED*. 2018. doi: 10.1109/ICGCIoT.2018.8753085.
- [103] C.-Y. Lin, E. Chu, L.-W. Ku, and J. Liu, “Active Disaster Response System for a Smart Building,” *Sensors (Basel)*, vol. 14, pp. 17451–17470, Sep. 2014, doi: 10.3390/s140917451.
- [104] “Spark SQL & DataFrames | Apache Spark.” <https://spark.apache.org/sql/> (accessed Feb. 09, 2023).
- [105] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark*. Beijing: O’Reilly, 2015. [Online]. Available: <https://www.safaribooksonline.com/library/view/learning-spark/9781449359034/>
- [106] H. Karau, A. Konwinski, P. Wendell, and M. Zaharia, *Learning Spark: Lightning-Fast Big Data Analytics*, 1st ed. O’Reilly Media, Inc., 2015.
- [107] M. Isard and Y. Yu, “Distributed data-parallel computing using a high-level programming language,” *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009.
- [108] C. Chambers *et al.*, “FlumeJava: easy, efficient data-parallel pipelines,” in *ACM-SIGPLAN Symposium on Programming Language Design and Implementation*, 2010.
- [109] M. Amović, M. Govedarica, V. Pajić, and S. Vasiljević, “Spatio-temporal types and data analysis in Big Data paradigm,” *AGG+ Journal for Architecture, Civil Engineering, Geodesy and Related Scientific Fields*, vol. 3, no. 1, pp. 066–075, Dec. 2015, doi: 10.7251/AGGPLUS1503066A.
- [110] “Apache Hadoop.” <https://hadoop.apache.org/> (accessed Feb. 09, 2023).

- [111] “Apache Cassandra | Apache Cassandra Documentation.”
https://cassandra.apache.org/_/index.html (accessed Feb. 09, 2023).
- [112] “Apache Hive.” <https://hive.apache.org/> (accessed Feb. 09, 2023).
- [113] “DataFrame - org.apache.spark.sql.DataFrame.”
<https://spark.apache.org/docs/1.3.1/api/scala/index.html#org.apache.spark.sql.DataFrame>
(accessed Feb. 09, 2023).
- [114] M.-O. Löwner, B. Joachim, G. Gröger, and K.-H. Häfele, *New Concepts for Structuring 3D City Models - an Extended Level of Detail Concept for CityGML Buildings*, vol. 7973. 2013. doi: 10.1007/978-3-642-39646-5_34.
- [115] L. Brink, J. Stoter, and S. Zlatanova, “Modeling an application domain extension of CityGML in UML,” *OGC Best Practice 12-066*, Dec. 2012, doi: 10.5194/isprsarchives-XXXVIII-4-C26-11-2012.
- [116] T. Kutzner, K. Chaturvedi, and T. Kolbe, “CityGML 3.0: New Functions Open Up New Applications,” *PFG – Journal of Photogrammetry Remote Sensing and Geoinformation Science*, vol. 88, Feb. 2020, doi: 10.1007/s41064-020-00095-z.
- [117] V. Pajić, “Model upravljanja velikim serijama geoprostornih podataka,” *Универзитет у Новом Саду*, Oct. 2021, Accessed: Feb. 09, 2023. [Online]. Available: <https://nardus.mpn.gov.rs/handle/123456789/18916>
- [118] R. Richter and J. Döllner, “Concepts and techniques for integration, analysis and visualization of massive 3D point clouds,” *Comput Environ Urban Syst*, vol. 45, pp. 114–124, 2014, doi: <https://doi.org/10.1016/j.compenvurbsys.2013.07.004>.
- [119] “ISO - ISO/TC 211 - Geographic information/Geomatics.”
<https://www.iso.org/committee/54904.html> (accessed Feb. 09, 2023).
- [120] “The Scala Programming Language.” <https://www.scala-lang.org/> (accessed Feb. 09, 2023).
- [121] “GeoSpark Notebook - Databricks.” <https://www.databricks.com/notebooks/geospark-notebook.html> (accessed Feb. 09, 2023).
- [122] “GitHub - harsha2010/magellan: Geo Spatial Data Analytics on Spark.”
<https://github.com/harsha2010/magellan> (accessed Feb. 09, 2023).
- [123] “GitHub - locationtech/geomesa: GeoMesa is a suite of tools for working with big geo-spatial data in a distributed fashion.” <https://github.com/locationtech/geomesa> (accessed Feb. 09, 2023).
- [124] “GeoWave Overview.” <https://locationtech.github.io/geowave/overview.html> (accessed Feb. 09, 2023).
- [125] “GeoTrellis - Home.” <https://geotrellis.io/> (accessed Feb. 09, 2023).

- [126] V. Pajić, M. Govedarica, and M. Amović, "Model of Point Cloud Data Management System in Big Data Paradigm," *ISPRS Int J Geoinf*, vol. 7, no. 7, 2018, doi: 10.3390/ijgi7070265.
- [127] M. Amović, M. Govedarica, A. Radulović, and I. Janković, "Big Data in Smart City: Management Challenges," *Applied Sciences*, vol. 11, no. 10, 2021, doi: 10.3390/app11104557.
- [128] A. Donaubauber, F. Straub, N. Panchaud, and C. Vessaz, "A 3D Indoor Routing Service with 2D Visualization Based on the Multi-Layered Space-Event Model," in *Lecture Notes in Geoinformation and Cartography*, 2013, pp. 453–469. doi: 10.1007/978-3-642-34203-5_25.
- [129] D. Popovic, J. Radovic, D. Jovanović, and V. Pajić, *3D Models of Objects in Process of Reconstruction*. 2018.
- [130] D. Jovanović *et al.*, "Building Virtual 3D City Model for Smart Cities Applications: A Case Study on Campus Area of the University of Novi Sad," *ISPRS Int J Geoinf*, vol. 9, no. 8, 2020, doi: 10.3390/ijgi9080476.
- [131] J. Radovic, "3D MODELS OF OBJECTS IN PROCESS OF RECONSTRUCTION," *САВРЕМЕНА ТЕОРИЈА И ПРАКСА У ГРАДИТЕЉСТВУ*, vol. 13, Jun. 2018, doi: 10.7251/STP1813283R.
- [132] M. L. Hamzah, A. Purwati, S. Sutoyo, A. Marsal, S. Sarbani, and N. Nazaruddin, "Implementation of the internet of things on smart posters using near field communication technology in the tourism sector," *Computer Science and Information Technologies*, vol. 3, pp. 194–202, Nov. 2022, doi: 10.11591/csit.v3i3.p194-202.
- [133] I. Gorelova, D. Dmitrieva, M. Dedova, and M. Savastano, "Antecedents and Consequences of Digital Entrepreneurial Ecosystems in the Interaction Process with Smart City Development," *Adm Sci*, vol. 11, no. 3, 2021, doi: 10.3390/admsci11030094.
- [134] A. Glass and J. R. Noennig, "Synthetic Pedestrian Routes Generation: Exploring Mobility Behavior of Citizens through Multi-Agent Reinforcement Learning," *Procedia Comput Sci*, vol. 207, pp. 3367–3375, 2022, doi: <https://doi.org/10.1016/j.procs.2022.09.395>.
- [135] G. Agugiaro, S. Hauer, and F. Nadler, "Coupling of CityGML-based Semantic City Models with Energy Simulation Tools: some Experiences," 2015.

8. ПРИЛОЗИ

ПРИЛОГ 1. Извод из модела конверзије JSON шеме у DataFrames

```
package doktorat.schema

import org.apache.spark.sql.types._
import play.api.libs.json._

import scala.annotation.tailrec
import scala.io.Source

case class SchemaType(typeName: String, nullable: Boolean)
private case class NullableDataType(dataType: DataType, nullable: Boolean)

object SchemaConverter {

  val SchemaFieldName = "name"
  val SchemaFieldType = "type"
  val SchemaFieldId = "id"
  val SchemaStructContents = "properties"
  val SchemaArrayContents = "items"
  val SchemaRoot = "/"
  val Definitions = "definitions"
  val Reference = "$ref"
  val TypeMap = Map(
    "string" -> StringType,
    "number" -> DoubleType,
    "float" -> FloatType,
    "integer" -> LongType,
    "boolean" -> BooleanType,
    "object" -> StructType,
    "array" -> ArrayType,
    "pointcloud" -> PointCloudType
  )
  /*From articеле Pajic et all*/
  )
  var definitions: JsObject = JsObject(Seq.empty)
  private var isStrictTypingEnabled: Boolean = true

  def disableStrictTyping(): SchemaConverter.type = {
    setStrictTyping(false)
  }

  def enableStrictTyping(): SchemaConverter.type = {
    setStrictTyping(true)
  }

  private def setStrictTyping(b: Boolean) = {
    isStrictTypingEnabled = b
    this
  }

  def convertContent(schemaContent: String): StructType =
    convert(parseSchemaJson(schemaContent))
}
```

```
def convert(inputPath: String): StructType =
convert(loadSchemaJson(inputPath))
def convert(inputSchema: JsObject): StructType = {
  definitions = (inputSchema \
Definitions).asOpt[JsObject].getOrElse(definitions)
  val name = getJsonName(inputSchema).getOrElse(SchemaRoot)
  val typeName = getJsonType(inputSchema, name).typeName
  if (name == SchemaRoot && typeName == "object") {
    val properties = (inputSchema \
SchemaStructContents).asOpt[JsObject].getOrElse(
      throw new NoSuchElementException(
        s"Root level of schema needs to have a
[$SchemaStructContents]-field"
      )
    )
    convertJsonStruct(new StructType, properties,
properties.keys.toList)
  } else {
    throw new IllegalArgumentException(
      s"schema needs root level called <$SchemaRoot> and root type
<object>. " +
      s"Current root is <$name> and type is <$typeName>"
    )
  }
}

def getJsonName(json: JsValue): Option[String] = (json \
SchemaFieldName).asOpt[String]

def getJsonId(json: JsValue): Option[String] = (json \
SchemaFieldId).asOpt[String]

def getJsonType(json: JsObject, name: String): SchemaType = {
  val id = getJsonId(json).getOrElse(name)

  (json \ SchemaFieldType).getOrElse(JsonNull) match {
    case JsString(s) => SchemaType(s, nullable = false)
    case JsArray(array) =>
      val nullable = array.contains(JsonString("null"))
      array.size match {
        case 1 if nullable =>
          throw new IllegalArgumentException("Null type only is
not supported")
        case 1 =>
          SchemaType(array.apply(0).as[String], nullable =
nullable)
        case 2 if nullable =>
          array.find(_ != JsonString("null"))
            .map(i => SchemaType(i.as[String], nullable =
nullable))
            .getOrElse {
              throw new IllegalArgumentException(
                s"Incorrect definition of a nullable parameter at
<$id>"
              )
            }
      }
  }
}
```



```
    }
    case _ if isStrictTypingEnabled =>
      throw new IllegalArgumentException(
        s"Unsupported type definition <${array.toString}> in
schema at <$id>"
      )
    case _ => // Default to string as it is the "safest" type
      SchemaType("string", nullable = nullable)
  }
  case JsNull =>
    throw new IllegalArgumentException(s"No <$SchemaFieldType>-
field in schema at <$id>")
  case t => throw new IllegalArgumentException(
    s"Unsupported type <${t.toString}> in schema at <$id>"
  )
}
}
private def parseSchemaJson(schemaContent: String) =
  Json.parse(schemaContent).as[JsonObject]

def loadSchemaJson(filePath: String): JsonObject = {
  Option(getClass.getResource(filePath)) match {
    case Some(relPath) =>
      parseSchemaJson(Source.fromURL(relPath).mkString)
    case None => throw new IllegalArgumentException(s"Path can not
be reached: $filePath")
  }
}
@tailrec
private def convertJsonStruct(schema: StructType, json: JsonObject,
jsonKeys: List[String]): StructType = {
  jsonKeys match {
    case Nil => schema
    case head :: tail =>
      val enrichedSchema = addJsonField(schema, (json \
head).as[JsonObject], head)
      convertJsonStruct(enrichedSchema, json, tail)
  }
}
def traversePath(loc: List[String], path: JsPath): JsPath = {
  loc match {
    case head :: tail => traversePath(tail, path \ head)
    case Nil => path
  }
}
private def checkRefs(inputJson: JsonObject): JsonObject = {
  val schemaRef = (inputJson \ Reference).asOpt[JsString]
  schemaRef match {
    case Some(loc) =>
      val searchDefinitions = Definitions + "/"
      val defIndex = loc.value.indexOf(searchDefinitions) match {
        case -1 => throw new NoSuchElementException(
          s"Field with name [$Reference] requires path with
[$searchDefinitions]"
        )
        case i: Int => i + searchDefinitions.length
      }
    }
  }
}
```

```
    }
    val pathNodes = loc.value.drop(defIndex).split("/").toList
    traversePath(pathNodes, JsPath)
      .asSingleJson(definitions) match {
        case JsDefined(v) => v.as[JsonObject]
        case _: JsUndefined =>
          throw new NoSuchElementException(s"Path [$loc] not
found in $Definitions")
      }
    case None => inputJson
  }
}
private def addJsonField(schema: StructType, inputJson: JsonObject,
name: String): StructType = {

  val json = checkRefs(inputJson)
  val fieldType = getFieldJsonType(json, name)

  schema.add(getJsonName(json).getOrElse(name),
fieldType.dataType, nullable = fieldType.nullable)
}
private def getFieldJsonType(json: JsonObject, name: String):
NullableDataType = {
  val fieldType = getJsonType(json, name)
  TypeMap(fieldType.typeName) match {

    case dataType: DataType =>
      NullableDataType(dataType, fieldType.nullable)

    case ArrayType =>
      val innerJson = checkRefs((json \
SchemaArrayContents).as[JsonObject])
      val innerJsonType = getFieldJsonType(innerJson, "")
      val dataType = ArrayType(innerJsonType.dataType,
innerJsonType.nullable)
      NullableDataType(dataType, fieldType.nullable)

    case StructType =>
      val dataType = getDataJsonType(json, JsPath \
SchemaStructContents)
      NullableDataType(dataType, fieldType.nullable)
  }
}
private def getDataJsonType(inputJson: JsonObject, contentPath: JsPath):
DataType = {
  val json = checkRefs(inputJson)

  val content = contentPath.asSingleJson(json) match {
    case JsDefined(v) => v.as[JsonObject]
    case _: JsUndefined => JsonObject(Seq.empty)
  }

  convertJsonStruct(new StructType, content, content.keys.toList)
}
}
```

ПРИЛОГ 2. Генерисање температурног DataFrame у предикцији

```
import numpy as np
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()
def generate_cycle_randomly(unit_id, cycle, model_variant, label):
    temp = 50+skewnorm.rvs(-8, size=1).item() + np.random.normal(0,
5)
    return (unit_id, cycle, model_variant, round(temp, 2), label)
def generate_cycles(unit_id, model_variant, init_rul):
    res = []
    rul = init_rul
    for cycle, vals in enumerate(range(rul)):
        res.append(generate_cycle_randomly(unit_id=unit_id,
cycle=cycle, model_variant=model_variant, label=rul))
        rul -= 1
    return res
data = generate_cycles(unit_id=0, model_variant='A', init_rul=45) +
generate_cycles(unit_id=1, model_variant='B', init_rul=40)
df = spark.createDataFrame(data, schema=["unit_id", "cycle",
"model_variant", "temp", "label"])
```

Овај Образац чини саставни део докторске дисертације, односно докторског уметничког пројекта који се брани на Универзитету у Новом Саду. Попуњен Образац укоричити иза текста докторске дисертације, односно докторског уметничког пројекта.

План третмана података

Назив пројекта/истраживања
Модел управљања просторно – временским подацима у паметном граду
Назив институције/институција у оквиру којих се спроводи истраживање
а) Универзитет у Новом Саду, Факултет техничких наука, Департман за рачунарство и аутоматику
Назив програма у оквиру ког се реализује истраживање
1. Опис података
<i>1.1 Врста студије</i> <i>Укратко описати тип студије у оквиру које се подаци прикупљају</i> Докторска дисертација _____
<i>1.2 Врсте података</i> а) квантитативни ✓ б) квалитативни ✓
<i>1.3. Начин прикупљања података</i> а) анкете, упитници, тестови б) клиничке процене, медицински записи, електронски здравствени записи в) генотипови: навести врсту _____ г) административни подаци: навести врсту _____ д) узорци ткива: навести врсту _____ ђ) снимци, фотографије: навести врсту _____

е) текст, навести врсту __ Литературни извори _____

ж) мапа, навести врсту _____

з) остало: описати __ подаци прикупљени виртуелним сензорима, подаци прикупљени ласерским скенирањем _____

1.3 Формат података, употребљене скале, количина података

1.3.1 Употребљени софтвер и формат датотеке:

а) Excel фајл, датотека _____

б) SPSS фајл, датотека _____

в) PDF фајл, датотека _____

г) Текст фајл, датотека _____

д) JPG фајл, датотека _____

е) Остало, датотека __.las, .geojson, .json__

1.3.2. Број записа (код квантитативних података)

а) број варијабли ____ Велики број _____

б) број мерења (испитаника, процена, снимака и сл.) ____ Велики број _____

1.3.3. Поновљена мерења

а) да

б) **не** ✓

Уколико је одговор да, одговорити на следећа питања:

а) временски размак измедју поновљених мера је _____

б) варијабле које се више пута мере односе се на _____

в) нове верзије фајлова који садрже поновљена мерења су именоване као _____

Напомене: _____

Да ли формати и софтвер омогућавају дељење и дугорочну валидност података?

а) **Да** ✓

б) Не

Ако је одговор не, образложити _____

2. Прикупљање података

2.1 Методологија за прикупљање/генерисање података

2.1.1. У оквиру ког истраживачког нацрта су подаци прикупљени?

а) експеримент, навести тип _____ поређење перформанси између два система, стандардног базираног на класичним системима управљања базама података и новог модела базираног на Big data парадигми _____

б) корелационо истраживање, навести тип _____

ц) анализа текста, навести тип _____

д) остало, навести шта _____

2.1.2 Навести врсте мерних инструмената или стандарде података специфичних за одређену научну дисциплину (ако постоје).

2.2 Квалитет података и стандарди

2.2.1. Третман недостајућих података

а) Да ли матрица садржи недостајуће податке? Да **Не** ✓

Ако је одговор да, одговорити на следећа питања:

а) Колики је број недостајућих података? _____

б) Да ли се кориснику матрице препоручује замена недостајућих података? Да Не

в) Ако је одговор да, навести сугестије за третман замене недостајућих података

2.2.2. На који начин је контролисан квалитет података? Описати

__ Квалитет података контролисан је провјером учитаних података , да ли подаци одговарају структури читавања у стандардним системима, те да ли је дошло до губитака у подацима. _____

2.2.3. На који начин је извршена контрола уноса података у матрицу?

_ Контрола уноса података у матрицу изведена је на бази експертског знања. _____

3. Третман података и пратећа документација

3.1. Третман и чување података

3.1.1. Подаци ће бити депоновани у _ Репозиторијуму докторских дисертација на _____
Универзитету у Новом Саду. _____

3.1.2. URL адреса __ <https://cris.uns.ac.rs/searchDissertations.jsf> _____

3.1.3. DOI _____

3.1.4. Да ли ће подаци бити у отвореном приступу?

а) Да ✓

б) Да, али после ембарга који ће трајати до _____

в) Не

Ако је одговор не, навести разлог _____

3.1.5. Подаци неће бити депоновани у репозиторијум, али ће бити чувани.

Образложење

3.2 Метаподаци и документација података

3.2.1. Који стандард за метаподатке ће бити примењен? __ Стандард који примењује _____

Репозиторијум докторских дисертација Универзитета у Новом Саду. _____

3.2.1. Навести метаподатке на основу којих су подаци депоновани у репозиторијум.

Ако је потребно, навести методе које се користе за преузимање података, аналитичке и процедуралне информације, њихово кодирање, детаљне описе варијабли, записа итд.

3.3 Стратегија и стандарди за чување података

3.3.1. До ког периода ће подаци бити чувани у репозиторијуму? _____

3.3.2. Да ли ће подаци бити депоновани под шифром? Да Не

3.3.3. Да ли ће шифра бити доступна одређеном кругу истраживача? Да Не

3.3.4. Да ли се подаци морају уклонити из отвореног приступа после извесног времена?

Да Не

Образложити

4. Безбедност података и заштита поверљивих информација

Овај одељак МОРА бити попуњен ако ваши подаци укључују личне податке који се односе на учеснике у истраживању. За друга истраживања треба такође размотрити заштиту и сигурност података.

4.1 Формални стандарди за сигурност информација/података

Истраживачи који спроводе испитивања с људима морају да се придржавају Закона о заштити података о личности (https://www.paragraf.rs/propisi/zakon_o_zastiti_podataka_o_licnosti.html) и одговарајућег институционалног кодекса о академском интегритету.

4.1.2. Да ли је истраживање одобрено од стране етичке комисије? Да **Не** ✓

Ако је одговор Да, навести датум и назив етичке комисије која је одобрила истраживање

4.1.2. Да ли подаци укључују личне податке учесника у истраживању? Да **Не** ✓

Ако је одговор да, наведите на који начин сте осигурали поверљивост и сигурност информација везаних за испитанике:

- a) Подаци нису у отвореном приступу
- б) Подаци су анонимизирани
- ц) Остало, навести шта

5. Доступност података

5.1. Подаци ће бити

a) јавно доступни ✓

б) доступни само уском кругу истраживача у одређеној научној области

ц) затворени

Ако су подаци доступни само уском кругу истраживача, навести под којим условима могу да их користе:

Ако су подаци доступни само уском кругу истраживача, навести на који начин могу приступити подацима:

5.4. Навести лиценцу под којом ће прикупљени подаци бити архивирани.

6. Улоге и одговорност

6.1. Навести име и презиме и мејл адресу власника (аутора) података

___ Младен Амовић, mladen.amovic@aggf.unibl.org _____

6.2. Навести име и презиме и мејл адресу особе која одржава матрицу с подацима

___ Младен Амовић, mladen.amovic@aggf.unibl.org _____

6.3. Навести име и презиме и мејл адресу особе која омогућује приступ подацима другим истраживачима

___ Младен Амовић, mladen.amovic@aggf.unibl.org _____