

სამცხე-ჯავახეთის სახელმწიფო უნივერსიტეტი
ინჟინერიის, აგრარულ და საბუნებისმეტყველო მეცნიერებათა ფაკულტეტი



ირაკლი ხაჩიძე

ცოდნის მოდელირება გრაფული მანქანური სწავლების გამოყენებით

ინფორმატიკის სადოქტორო პროგრამა
სადისერტაციო ნაშრომი შესრულებულია ინფორმატიკის დოქტორის
აკადემიური ხარისხის მოსაპოვებლად

სამეცნიერო ხელმძღვანელები:
გულნარა ჯანელიძე, საინჟინრო მეცნიერებათა დოქტორი,
მოწვეული სპეციალისტი.
ლელა წითაშვილი, ინჟინერიის დოქტორი ინფორმატიკაში,
ასოცირებული პროფესორი.

ახალციხე
2025 წელი

განაცხადი

ირაკლი ხაჩიძე ვაცხადებ, რომ წარმოდგენილი სადისერტაციო ნაშრომი წარმოადგენს ჩემს ორიგინალურ ნამუშევარს და არ შეიცავს სხვა ავტორების მიერ აქამდე გამოქვეყნებულ, გამოსაქვეყნებლად მიღებულ ან დასაცავად წარმოდგენილ მასალას, რომელიც ნაშრომში არ არის მოხსენებული ან ციტირებული სათანადო წესების შესაბამისად.

ხელმოწერა:

თარიღი:

ანოტაცია

ცოდნის მოდელირება თანამედროვე სამყაროში წარმოადგენს კრიტიკულად მნიშვნელოვან კომპონენტს, რადგან მისი საშუალებით შეგვიძლია კომპლექსური ინფორმაციის სტრუქტურირება და ინტერპრეტაცია, რაც საჭიროა ინტელექტუალური სისტემებისთვის (მაგ. AI, ექსპერტული სისტემები, რეკომენდაციის ძრავები და სხვა). ასევე მსჯელობისა და სწავლის პროცესის ავტომატიზაციისათვის, რაც აძლიერებს სისტემების ადაპტაციას და გადაწყვეტილების მიღების უნარს, ინფორმაციის ხელახალ გამოყენებას და გაზიარებას, რაც თავისთავად განაპირობებს ცოდნის რესურსების სტანდარტიზაციას (მაგ. ცოდნის გრაფები, ონტოლოგიები); ასევე მნიშვნელოვანი ადგილი უჭირავს თანამედროვე ხელოვნური ინტელექტის მოდელებში ჰიბრიდული ინტეგრაციის შესაძლებლობის კუთხით, სადაც სიმბოლური და ქვესიმბოლური მეთოდები ერთმანეთს ერწყმის.

დღევანდელ სამყაროში, სადაც ინფორმაცია ერთ-ერთი ყველაზე ღირებული რესურსია, ცოდნის მოდელირება აუცილებელი წინაპირობაა ჭკვიანი სისტემების განვითარებისთვის, რომლებიც ავითარებენ ისეთ სფეროებს, როგორცაა მედიცინა, ინდუსტრიული ავტომატიზაცია, განათლება, მონაცემებზე დაფუძნებული გადაწყვეტილებების სისტემები და სხვა.

წარმოდგენილი დისერტაციის მიზანია გრაფული მანქანური სწავლების (Graph Machine Learning) მეთოდების გამოყენებით ცოდნის მოდელირება და მასთან დაკავშირებული კვლევითი პრობლემების ეფექტური გადაწყვეტა. კვლევის კონტექსტი დაკავშირებულია დიდი მოცულობის მონაცემების დამუშავებასთან და ცოდნის აგების პრობლემებთან, რომელიც წარმოიშვა თანამედროვე ინფორმაციულ სისტემებში.

კვლევის საგანს წარმოადგენს ისეთი საკითხები, როგორცაა განაწილებული ევოლუციური ალგორითმები, გენეტიკური პროგრამირება, აგრეთვე ცოდნის ინჟინერიის მიდგომები გრაფული მოდელირებისთვის, ნაკადური მონაცემების დამუშავება, ცოდნისადმი ბიოინჟინრული მიდგომები და სხვა. კვლევით ნაწილში გამოქვეყნდა სტატიები, რომლებიც მოიცავს შემდეგ საკითხებს: „განაწილებული გენეტიკური ალგორითმების გამოყენება ნაკადური დიდი მონაცემების

დამუშავებისთვის“, „ნაკადური მონაცემების განაწილებული დამუშავება გენეტიკური პროგრამირების გამოყენებით“, „ცოდნის ფილოსოფიის ახლებური გააზრება“, „ცოდნის ინჟინერიის ზოგიერთი ახალი ასპექტი“, „ცოდნის ამოღება დიდი ტექსტური მონაცემთა ნაკრებიდან“. თითოეულ სტატია წარმოადგენს კვლევის შედეგებს, რომლებიც დისერტაციას ამდიდრებს ახალი მიდგომებით.

ერთ-ერთი სიახლე არის გენეტიკური ალგორითმის განაწილებული ვერსია, ნაკადური მონაცემების ოპტიმიზაციისათვის. კვლევაში წარმოდგენილია ამ ალგორითმის სრული სამუშაო ციკლი. გამოყენებული ლამბდა არქიტექტურისა და განაწილებული გენეტიკური ალგორითმის კომბინაცია საშუალებას იძლევა დიდ მოცულობის ნაკადური მონაცემებზე ოპტიმიზაციის მასშტაბური ამოცანების გადაჭრის პროცესში მივაღწიოთ მაღალი წარმადობისა და დაბალი დაყოვნების პირობებს. მსგავსი პრინციპი იქნა გამოყენებული გენეტიკური პროგრამირებით ჩამოყალიბებულ ნაკადურ მონაცემთა განაწილებულ დამუშავებაში. შემუშავდა ალგორითმი, რომელიც ახორციელებს შემავალი ნაკადის ოპტიმალური ამონახსნის პოვნას. გენეტიკური პროგრამირების გამოყენების შედეგად აღწერილია სრულ პროცესი, რაც უზრუნველყოფს რეალურ დროში მონაცემთა ნაკადებზე დაბალი დაყოვნებით რეაგირებას.

დისერტაციაში ასევე განხილულია ცოდნის ინჟინერიის ახალი ასპექტები, სადაც აღწერილია ცოდნის წარმოდგენის ახლებური ხედვა. ჩვენი მიდგომის არსი მდგომარეობს იმაში, რომ ინფორმაცია უნდა განვიხილოთ არამატერიალურ ჭრილში, კვანტურ ფორმაში, უწყვეტი ჰოლოგრამის სახით. გარდა ამისა, წარმოდგენილია ახალი დოკინგისა და ასემბლინგის ალგორითმი ფრაგმენტული ცოდნის ინტეგრაციისათვის. აღნიშნული ალგორითმი უზრუნველყოფს სხვადასხვა წყაროებიდან მიღებული ცოდნის ფრაგმენტების გენერაციასა და აწყობას.

დისერტაციის მთავარ სიახლეს წარმოადგენს ცოდნის ფრაგმენტების დოკინგისა და ცოდნის ასემბლინგის ალგორითმი, სადაც ჩვენ განვიხილავთ ცოდნის აგების ჩვენეულ კონცეპციას, რომელიც პრინციპულად წარმოადგენს მოლეკულური გენერაციის სხვადასხვა პარადიგმების, კერძოდ, როგორცაა drug design, docking, jigsaw ერთგვარ გადმოტანას ცოდნის ინჟინერიის სფეროში. ეს განპირობებულია

მოლეკულური გეომეტრიული მსგავსებით ცოდნის გრაფულ წარმოდგენასთან. აქ, ფაქტობრივად, ცოდნის აგების ჩვენეული კონცეპცია გულისხმობს ფრაგმენტების შეერთებათა თითოეული კომბინაციის (გრაფის კონფიგურაციის) შეფასებას მისი ოპტიმიზაციის ანუ ენტროპიის მინიმიზაციის თვალსაზრისით. აღნიშნული ამოცანა მიეკუთვნება დიდი განზომილების, მრავალექტრემუმიან, არაწრფივი ოპტიმიზაციის კლასს, რომლის დეტერმინირებული მეთოდებით (მათ შორის, დინამიური პროგრამირების მეთოდის გამოყენების შემთხვევაშიც) დაკავშირებულია დიდ მოცულობის გამოთვლით სირთულეებთან.

დისერტაციაში შემუშავებული გრაფული მანქანური სწავლების მოდელები და ალგორითმები იძლევიან როგორც თეორიული, ისე პრაქტიკული ღირებულებას ინფორმაციულ სისტემებში, როგორც დიდი მონაცემების, ასევე რეალური დროში დასამუშავებელი მონაცემების ჭრილში. აღნიშნული კვლევების ერთობლიობა, განაწილებული გენეტიკური ალგორითმები, გენეტიკური პროგრამირების მოდელები და ცოდნის ფრაგმენტების გრაფული ინტეგრაცია შესაძლებელს ხდის შემოთავაზებული მეთოდების ფართოდ ადაპტირებას, მონაცემთა მართვის თანამედროვე სისტემებში და ინტელექტუალური ანალიზის სფეროებში.

ABSTRACT

I. Khachidze

Knowledge Modelling Using Graph Machine Learning

Knowledge modelling in the modern world is a critically important component, as it enables the structuring and interpretation of complex information required for intelligent systems (e.g., artificial intelligence, expert systems, recommendation engines, etc.). It also facilitates the automation of reasoning and learning processes, which enhances systems' adaptability and decision-making capabilities, as well as the reuse and sharing of information, necessitating the standardisation of knowledge resources (e.g., knowledge graphs, ontologies). Furthermore, it plays an important role in enabling hybrid integration in contemporary artificial intelligence models, where symbolic and subsymbolic methods converge.

In today's world, where information is one of the most valuable resources, knowledge modelling is a crucial prerequisite for the development of intelligent systems that advance fields such as medicine, industrial automation, education, and data-driven decision-making systems, among others.

The presented dissertation aims to apply graph machine learning methods to knowledge modelling and to effectively solve related research problems. The context of this research is connected with the processing of large volumes of data and the challenges of knowledge assembling that have emerged in modern information systems.

The research addresses issues such as distributed evolutionary algorithms, genetic programming, knowledge engineering approaches for graph modelling, stream data processing, and bio-inspired approaches to knowledge, among others. The research section includes published articles covering the following topics: Distributed genetic algorithms for processing large streaming data, Distributed processing of streaming data using genetic programming, A novel conceptualisation of the philosophy of knowledge, Some new aspects of knowledge engineering, and Knowledge extraction from large text data sets.

Each of the aforementioned articles represents a research result that enriches the dissertation with new approaches.

One of the novel contributions is a distributed version of the genetic algorithm for stream data optimisation. The study presents the full operational cycle of this algorithm. The

combination of a Lambda architecture and a distributed genetic algorithm enables high throughput and low latency in solving large-scale optimisation tasks on streaming data. A similar principle has been applied to distributed stream processing using genetic programming. An algorithm was developed to identify the optimal extraction of the incoming data stream. By using genetic programming, the entire process is described, ensuring a low-latency, real-time response to streaming data.

The dissertation also discusses new aspects of knowledge engineering, presenting a novel view of knowledge representation. The essence of this approach is that information should be considered in a non-material form, in a quantum form, as a continuous hologram. In addition, a new docking and assembling algorithm for the integration of fragmented knowledge is introduced. This algorithm ensures the generation and assembly of knowledge fragments obtained from different sources.

The main innovation of the dissertation is the algorithm for docking knowledge fragments and assembling knowledge. We present our concept of knowledge construction, which essentially transfers various molecular generation paradigms, specifically those used in drug design, docking, and jigsaw puzzles, into the field of knowledge engineering. This is motivated by the geometric similarity between molecular structures and knowledge graph representations. In practice, our concept of knowledge construction involves evaluating each combination of fragment connections (graph configurations) in terms of its optimisation, specifically, entropy minimisation. This task belongs to the class of high-dimensional, multi-extremum, nonlinear optimisation problems, for which deterministic methods are associated with very high computational complexity.

The graph machine learning models and algorithms developed in the dissertation provide both theoretical and practical value in information systems, in the context of both big data and real-time data processing. This combination of studies, distributed genetic algorithms, genetic programming models, and graph-based integration of knowledge fragments, makes it possible to widely adapt the proposed methods in contemporary data management systems and domains of intelligent analytics.

შინაარსი

შესავალი.....	10
თავი I. მონაცემთა მოპოვების სისტემები, დიდი მონაცემების დამუშავების პრობლემები (ლიტერატურის მიმოხილვა).....	15
1.1 მონაცემთა მოპოვების როლი გადაწყვეტილების მიღების პროცესში	15
1.2 მონაცემთა მოპოვების სისტემების მიმოხილვა და ანალიზი	21
1.2.1 მონაცემთა მოპოვების ტექნიკები	23
1.2.2 მონაცემთა წინასწარი მომზადება	27
1.3 დიდი მონაცემების დამუშავების პრობლემები და ამოცანები	32
1.3.1 დიდი მონაცემების დამუშავების პრობლემატიკის აღწერა.....	32
1.3.2 დიდი მონაცემების დამუშავების ამოცანები	35
თავი II. ცოდნის წარმოდგენა	37
2.1 ცოდნის წარმოდგენის მოდელები.....	37
2.1.1 სიმბოლური ცოდნის წარმოდგენის მიდგომები.....	37
2.1.2 ქვესიმბოლური ცოდნის წარმოდგენის მიდგომები	45
2.1.3 ჰიბრიდული და ნეიროსიმბოლური მიდგომები.....	55
2.1.4 ცოდნის წარმოდგენა თანამედროვე ხელოვნური ინტელექტის სისტემებში	60
2.2 ცოდნის გრაფული მოდელის აღწერა	64
თავი III - დიდი მონაცემების განაწილებული მოპოვება	69
3.1 მონაცემთა ლამბდა-არქიტექტურის დამუშავების კონცეფცია.....	69
3.1.1 ლამბდა არქიტექტურის გამოწვევები	74
3.2 მონაცემთა განაწილებული დამუშავების ტრაქტის აღწერა.....	75
3.2.1 განაწილებული მონაცემთა დამუშავების ძირითადი კომპონენტები	79
3.2.2 განაწილებული გენეტიკური ალგორითმების გამოყენება დიდი მონაცემების ნაკადური დამუშავებისთვის.....	84
3.2.3 ნაკადური მონაცემების განაწილებული დამუშავება გენეტიკური პროგრამირების გამოყენებით.....	92
3.3. ცოდნის ფრაგმენტის აგება.....	97
3.3.1 ცოდნის ფრაგმენტები ცოდნის მენეჯმენტში (Knowledge Fragments in Knowledge Management).....	97

3.3.2 ცოდნის ფრაგმენტები სემანტიკურ ქსელში (Knowledge Fragments in the Semantic Web).....	99
3.3.3 ცოდნის ფრაგმენტები კოგნიტურ მეცნიერებაში.....	102
3.3.4 ცოდნის ინჟინერიის ახალი ასპექტი.....	103
თავი IV - ცოდნის ასემბლინგისადმი ბიოინფორმატიკული მიდგომა	119
4.1 ცოდნის ფრაგმენტის წარმოდგენის მოლეკულური მოდელი.....	119
4.2 მოლეკულური დოკინგი და სტრუქტურაზე დაფუძნებული დიზაინის სტრატეგიების მიმოხილვა.....	131
4.2.1. Drug design მიდგომა ცოდნის აგების პროცესში.....	131
4.2.2. Docking მიდგომა ცოდნის აგების პროცესში.....	135
4.2.3. ცოდნის აგება როგორც Jigsaw Puzzle	139
4.3 გრაფული მანქანური სწავლების გამოყენება ცოდნის მოდელირებაში.....	151
4.4 ტენზორული აღრიცხვის გამოყენება ცოდნის აგების პროცესში	154
4.4.1 ტენზორები, როგორც ცოდნის კონტეინერები	155
4.4.2 ტენზორული დეკომპოზიცია და ცოდნის ინტეგრაცია	157
4.5 დიდი მონაცემების ინტელექტუალური ანალიზის ფრეიმვორკის არქიტექტურის აღწერა.	158
4.6 ფრაგმენტებისგან ცოდნის აგება გენეტიკური ალგორითმის გამოყენებით .	163
4.6.1. ცოდნის გრაფის ენტროპიის მინიმიზაციის ამოცანა	163
4.6.2. ევოლუციური ალგორითმების კლასიფიკაცია	165
4.6.3. გენეტიკური ალგორითმი (GA).....	166
4.6.4. ფრაგმენტებისგან ცოდნის აგება გენეტიკურ ალგორითმის გამოყენებით	175
დასკვნა	183
გამოყენებული ლიტერატურა	186
ცხრილების ნუსხა	190
სურათების /დიაგრამების/გრაფიკების ნუსხა.....	190
აბრევიატურები	192
Abbreviations.....	193

შესავალი

პრობლემის აქტუალობა

დღევანდელ მონაცემთა ეპოქაში ბიზნესი, სამეცნიერო კვლევები და ტექნოლოგიური სისტემები ყოველდღიურად გამოიმუშავენ უზარმაზარი მოცულობის ინფორმაციას, რომლის დამუშავება დიდი მოცულობის და სწრაფად ცვალებადობის გამო სტანდარტული მეთოდებით შეუძლებელია ითქვას, რომ შეუძლებელიცაა.

თანამედროვე კომპიუტერული მოდელები და შესაბამისი პროგრამული უზრუნველყოფა უკვე აღარ აკმაყოფილებს ხელოვნური ინტელექტის სისტემის სრულფასოვანი რეალიზაციის მოთხოვნებს. ინფორმაციის წარმოდგენის თანამედროვე ფორმებიც კი ჯერ კიდევ ვერ იძლევა კომპლექსური ტექსტუალური ინფორმაციის სემანტიკური მოდელირების საშუალებას.

კლასტერიზაციისა თუ კლასიფიკაციის ამოცანების გადაჭრისას, მანქანური სწავლების მოდელები მხოლოდ მონაცემთა ერთობლიობას აღიქვამენ ცოდნად. ამასთან, ჩვენ კვლავ ენაზე დაყრდნობით გამოვხატავთ ცოდნას - სიმბოლოების, ცალკეული სიტყვების ან ლიტერალების დონეზე, რაც განაპირობებს იმას, რომ მანქანური თარგმნის მეთოდები ჯერ კიდევ არასრულყოფილია. კომპიუტერულ სისტემებს ჯერ არ შეუძლიათ სრულყოფილად და ადეკვატურად თარგმნონ ერთი ენიდან მეორეზე, რაც იმას ნიშნავს, რომ ცოდნის წარმოდგენის არსებული მოდელები ჯერ კიდევ შორსაა სრულყოფილისგან. ამიტომ საჭიროა ახალი ალგორითმების და მიდგომების შემუშავება.

კვლევის სიახლე, მიზანი და მნიშვნელობა

დისერტაციის მთავარი სიახლე არის ცოდნის ფრაგმენტების დოკინგისა და ასემბლინგის ალგორითმი, რომელიც აღწერს ჩვენი კონცეპციით შექმნილ ცოდნის აგების მიდგომას. ეს მიდგომა წარმოადგენს მოლეკულური გენერაციის სხვადასხვა პარადიგმების, როგორცაა drug design, docking და jigsaw puzzle, გადატანას ცოდნის ინჟინერიის სფეროში. ეს ინოვაცია ემყარება მოლეკულური გეომეტრიის მსგავსებას ცოდნის გრაფულ წარმოდგენასთან. ჩვენი კონცეფცია გულისხმობს ფრაგმენტების

შეერთების კომბინაციების (გრაფის კონფიგურაციის) შეფასებას მისი ოპტიმიზაციის თვალსაზრისით, რაც გულისხმობს ენტროპიის მინიმიზაციას გრაფის რეკონფიგურაციის საშუალებით. ეს ამოცანა განეკუთვნება მაღალ განზომილებიან, მრავალ ექსტრემუმიან და არაწრფივ ოპტიმიზაციას, რომლის გადაწყვეტა დაკავშირებულია გამოთვლით სირთულეებთან, მათ შორის, დინამიკური პროგრამირების მეთოდის გამოყენებისას. დისერტაციაში შემუშავებული გრაფული მანქანური სწავლების მოდელები და ალგორითმები უზრუნველყოფენ როგორც თეორიულ, ისე პრაქტიკულ გატაწყვეტებს ინფორმაციული სისტემებისთვის, დიდი მონაცემების და რეალურ დროში დასამუშავებელი მონაცემებისთვის. შემოთავაზებული მეთოდები ადაპტირებადი იქნება თანამედროვე მონაცემთა მართვის სისტემებში და ინტელექტუალური ანალიზის მიმართულებებში.

გარდა ამისა სადისერტაციო ნაშრომში განხილულია მასთან დაკავშირებული კვლევები, რომლებიც გამოქვეყნებულია სტატიების სახით: **ცოდნის ინჟინერიის ზოგიერთი ახალი ასპექტი** - ეს ნაშრომი წარმოადგენს ცოდნის ინჟინერიის დარგის განვითარების მიმოხილვას. აქ ჩვენ განვიხილავთ ცოდნის ინჟინერიას თანამედროვე საინფორმაციო სისტემების, კერძოდ, ხელოვნური ინტელექტის, დიდი მონაცემების ანალიტიკის, ასევე ინფორმაციის ინტეგრაციისა და ცოდნის მართვის კონტექსტში. განვიხილავთ და აღვწერთ ახალ მიდგომებს, რომლებმაც მნიშვნელოვნად შეცვალეს ცოდნის წარმოდგენისა და დამუშავების შესახებ კვლევები. ცოდნის წარმოდგენის საილუსტრაციოდ, ჩვენ აღვწერთ ცოდნის წარმოდგენის გრაფულ მოდელს, რომელიც დაფუძნებულია ადამიანის ტვინის მოდელის მუშაობის პრინციპებზე, სადაც ცოდნის გრაფის (ჰიპერგრაფის) თითოეული წვერო (კვანძი) შეიცავს აქსონის დენდრიტული ტიპის ატრიბუტებს. მათი თავსებადობა, ანუ სინაფსის სინერგია, უზრუნველყოფს ცოდნის მოდელის მთლიანობის კრიტერიუმს. ასევე აღვწერთ ორ ნარატორს შორის ურთიერთქმედებას, სადაც შუალედურ ფუნქციას ასრულებს ადამიანის ენა - ლინგვისტური მთარგმნელი. ის ახორციელებს ურთიერთგარდაქმნას, როდესაც ცოდნის დაშლა იწყება უფრო მაღალი განზომილებიანი დონიდან ზემოდან ქვემოთ მეორე მხარეს გადასაცემად, სადაც შებრუნებული თანმიმდევრობით იწყება ცოდნის აგების პროცესი (მეფარიშვილი, ჯანელიძე and ხაჩიძე 2023, 28).

განაწილებული გენეტიკური ალგორითმების გამოყენება ნაკადური დიდი მონაცემების დამუშავებისთვის - ამ ნაწილში შემუშავებულია განაწილებული გენეტიკური ალგორითმი ლამბდა არქიტექტურის პირობებში. აღწერილია ალგორითმის მუშაობის სრული ციკლი, დაწყებული რეალურ დროის მოცემულ მონაკვეთში შემავალი მონაცემების ფრაგმენტებად დაყოფით, დამთავრებული ოპტიმალური ამონახსნის პოვნით. წარმოდგენილია შემუშავებული ალგორითმის ბლოკ-სქემა. ნაშრომში წარმოდგენილი მიდგომა იყენებს ლამბდა არქიტექტურისა და განაწილებული გენეტიკური ალგორითმების მიდგომების უპირატესობებს. ამდენად, მათი ერთობლივი გამოყენება მიზანშეწონილია დიდ მონაცემებში ოპტიმიზაციის ფართომასშტაბიანი ამოცანების გადასაწყვეტად რეალურ დროში (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 54).

ნაკადური მონაცემების განაწილებული დამუშავება გენეტიკური პროგრამირების გამოყენებით - ჩვენს ამოცანაში ნაკადი შემოედინება უწყვეტი სახით, კერძოდ, თუ ნაკადის ერთი ნაწილის შემოსვლა მოხდა $[t-1, t]$ დროის ინტერვალში, მაშინ ნაკადის შემდეგი ნაწილი შემოვა $[t, t+1]$ ინტერვალში. მონაცემთა ნაკადებში ოპტიმალური გადაწყვეტის პოვნის მიზნით, რადგან მონაცემების მოცულობა ძალიან დიდია, საჭიროა ნაკადის დაჩეხვა (splitting) დროში შემოსვლის შესაბამის ფრაგმენტებად და ცალკეული ფრაგმენტისთვის ხის სტრუქტურების გენერირება. გენერირებული ხეებიდან ცალკეულ დანაწევრებულ ნაწილში აირჩევა ხეების წყვილი, რომელიც ჩაითვლება მშობელ წყვილებად, შემდეგ ეტაპზე მშობელი ხეების წყვილები გაერთიანდება და ქმნიან მშობელთა პოპულაციას, რომლის ცალკეულ წევრს აქვს ხისებრი სტრუქტურა. მშობელთა პოპულაციაში შესრულდება გენეტიკური ოპერაციები და მათ შორის აირჩევა საუკეთესო ხე, რომელიც შეესაბამება საუკეთესო ამონახსნს. ეს პროცესი განმეორდება შემომავალი ნაკადის ყველა ფრაგმენტისთვის (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 99).

ცოდნის ამოღება დიდი ტექსტური მონაცემთა ნაკრებიდან - ამ ნაწილში ორიგინალურობა მდგომარეობს იმაში, რომ ცოდნის წარმოდგენა განხილულია, როგორც გრაფული მოდელი და ასევე მოიცავს ცოდნის ცალკეული გრაფული ფრაგმენტებიდან ერთიანი ცოდნის მოდელის აწყობას. ამ ორი ცნების გაერთიანება

ლამბდა არქიტექტურის კონტექსტში შესაძლებელს ხდის მონაცემთა აღმოჩენის ფართომასშტაბიანი და კომპლექსური ამოცანების ეფექტურ გადაწყვეტას დიდი მონაცემების პარალელური დამუშავებით.

ზემოაღნიშნულიდან გამომდინარე კვლევის მიზანია:

- შემუშავდეს ცოდნის ფრაგმენტების დოკინგისა და ასემბლინგის ალგორითმი - მოლეკულური გენერაციის სხვადასხვა პარადიგმების, როგორცაა drug design, docking და jigsaw puzzle, გადატანა ცოდნის ინჟინერიაში;
- შემუშავდეს განაწილებული გენეტიკური ალგორითმების გამოყენებით ნაკადური დიდი მონაცემების დამუშავების ალგორითმი - ლამბდა არქიტექტურისა და განაწილებული გენეტიკური ალგორითმების მიდგომების უპირატესობების ერთობლივი გამოყენება და ალგორითმის შემუშავება ნაკადური დიდი მონაცემების დამუშავებისთვის;
- შემუშავდეს გენეტიკური პროგრამირების გამოყენებით ნაკადური მონაცემების განაწილებული დამუშავების მიდგომა - გენეტიკური პროგრამირების გამოყენებით ნაკადური მონაცემების განაწილებული დამუშავება, ოპტიმალური გადაწყვეტის პოვნის მიზნით;
- შემუშავდეს ცოდნის ამოღების მექანიზმი დიდი ტექსტური მონაცემთა ნაკრებიდან - ცოდნის წარმოდგენა განხილულია, როგორც გრაფული მოდელი და ასევე მოიცავს ცოდნის ცალკეული გრაფული ფრაგმენტებიდან ერთიანი ცოდნის მოდელის აწყობას;
- შემუშავდეს ცოდნის ინჟინერიის ზოგიერთი ახალი ასპექტი, ცოდნის წარმოდგენის ახლებური ხედვა - ცოდნის განხილვა კვანტურ ჭრილში უწყვეტი ჰოლოგრამის ფორმაში;

დისერტაციაში წარმოდგენილი კვლევები ღირებულ შედეგებს შეიცავს, როგორც აკადემიური ისე ბიზნეს სექტორისთვის.

სადისერტაციო ნაშრომის მოკლე მიმოხილვა

სადისერტაციო ნაშრომი შედგება შესავალის, ოთხი ძირითადი თავისგან შესაბამისი დასკვნებით, რომელიც წარმოდგენილია კვლევის შედეგების სახით და ახლავს გამოყენებული ლიტერატურა

სადისერტაციო ნაშრომის შესავალში მოცემულია პრობლემის აქტუალობა და მისი მიმდინარე მდგომარეობა, ნაშრომის მიზანი, სიახლე და მისი მნიშვნელობა.

თავი I. მონაცემთა მოპოვების სისტემები, დიდი მონაცემების დამუშავების პრობლემები (ლიტერატურის მიმოხილვა) - პირველ თავში განიხილება მონაცემთა მოპოვების (Data Mining) როლი და ადგილი გადაწყვეტილების მიღების პროცესში. წარმოდგენილია მონაცემთა მოპოვების სისტემების მიმოხილვა და ანალიზი, ასევე განხილულია დიდი მონაცემების დამუშავებასთან დაკავშირებული პრობლემები და დისერტაციის ამოცანები.

თავი II. ცოდნის წარმოდგენა - მეორე თავი ეხება ცოდნის წარმოდგენის საკითხებს. გაწერილია არსებული მოდელების მიმოხილვა და განხილულია ცოდნის გრაფული მოდელის კონცეფცია, რომელიც წარმოადგენს დისერტაციის ძირითადი ალგორითმის მოდელის საფუძველს.

თავი III. დიდი მონაცემების განაწილებული მოპოვება - მესამე თავში წარმოდგენილია მონაცემთა ლამბდა-არქიტექტურის კონცეპტუალური სქემა, აღწერილია მონაცემთა განაწილებული დამუშავების ტრაქტი და ცოდნის ფრაგმენტების აგების მეთოდოლოგია, რაც საფუძველად დაედო შემდგომი კვლევის ალგორითმულ ნაწილს.

თავი IV. ცოდნის ასემბლინგისადმი ბიონფორმატიკული მიდგომა - მეოთხე თავი ფოკუსირებულია ცოდნის ასემბლინგის პროცესზე ბიონფორმატიკული მეთოდების გამოყენებით. განხილულია მოლეკულური მოდელების გამოყენება ცოდნის ფრაგმენტების წარმოდგენისთვის, გრაფული მანქანური სწავლების მოდელების ინტეგრაცია, ტენზორული აღრიცხვის მეთოდოლოგია და დიდი მონაცემების ინტელექტუალური ანალიზის ფრეიმვორკის არქიტექტურა. ამავე თავშია მოცემული ცოდნის ფრაგმენტების დოკინგისა და ცოდნის ასემბლინგის ალგორითმი, რომელიც დისერტაციის მთავარ კვლევას წარმოადგენს.

თავი I. მონაცემთა მოპოვების სისტემები, დიდი მონაცემების დამუშავების პრობლემები (ლიტერატურის მიმოხილვა)

1.1 მონაცემთა მოპოვების როლი გადაწყვეტილების მიღების პროცესში

მონაცემების უზარმაზარი მოცულობა, რომელიც ყოველდღიურად გენერირდება, წარმოშობს ახალი მიდგომების და მექანიზმების შემუშავების საჭიროებას. რადგან აღნიშნული მონაცემები პირველადი სახით გამოუსადეგარია და ის თავისთავად წარმოადგენს ქაოსურ და გაუგებარ მონაცემთა სიმრავლეს. ამიტომ საჭიროა მისი დამუშავება და მიღებული სასარგებლო ინფორმაციის გამოყოფა. ბოლო რამდენიმე წლის განმავლობაში ტენდენცია კიდევ უფრო გაიზარდა. უზარმაზარი რაოდენობით მონაცემები მუდმივად იქმნება და ინახება მონაცემთა საცავებში სხვადასხვა კვლევის სფეროში. ხშირად არა მხოლოდ დიდია მონაცემების რაოდენობა, არამედ ეს მონაცემები მუდმივად განახლებადია. გარდა ამისა, მონაცემთა ტიპებისა და წყაროების ძალიან ფართო დიაპაზონი, რაც საბოლოო ჯამში ქმნის დიდ მონაცემებს (Big Data). ეს ყველაფერი კი წარმოშობს მონაცემთა დამუშავების გამოწვევებს (რუზგასი, იაკუბელიენე and ბუვიტე 2016, 5).

დიდი მონაცემები: დიდი მონაცემების დამუშავება და ანალიზი გამოიყენება სხვადასხვა სფეროში, როგორცაა მედიცინა, ფინანსები, ეკონომიკა, ინჟინერია და ა.შ. აღნიშნული მონაცემების დამუშავებისთვის სხვადასხვა მეთოდებია შემუშავებული, როგორცაა კლასტერიზაცია, კლასიფიკაცია, სტატისტიკური და ვიზუალური ანალიზი და სხვა. დიდი მონაცემები ნიშნავს, რომ მათი მართვა საკმაოდ რთული ან შეუძლებელია მონაცემთა დამუშავების მარტივი პროგრამებისა და ხელსაწყოების გამოყენებით, მათი ზომისა და რთული სტრუქტურის გამო. ამიტომ მხოლოდ ერთი მეთოდის გამოყენება სასურველ შედეგს ვერ იძლევა, რასაც მიყვავართ მონაცემთა მოპოვების (Data Mining) მიდგომებთან. (მონაცემთა-მოპოვება-კლასიფიკაცია 2023)

მონაცემთა მოპოვება (DM). მონაცემთა მოპოვების ძირითადი ამოცანაა დიდი მოცულობის მონაცემებიდან შაბლონებისა და ცოდნის აღმოჩენის პროცესი. იგი მოიცავს სხვადასხვა ტექნიკის გამოყენებას, როგორცაა სტატისტიკური ანალიზი, მანქანური სწავლება და ხელოვნური ინტელექტი მონაცემებიდან ღირებული

ინფორმაციის მოსაპოვებლად. თუმცა ინფორმაციის ამოღება არ არის ერთადერთი პროცესი, რომელიც მონაცემთა მოპოვებას ახლავს თან. მონაცემთა მოპოვება მოიცავს სხვა პროცესებს, როგორცაა მონაცემთა გაწმენდა, მონაცემთა ინტეგრაცია, მონაცემთა ტრანსფორმაცია, მონაცემთა მოპოვება, ნიმუშის შეფასება და მონაცემთა პრეზენტაცია. მას შემდეგ, რაც ყველა ეს პროცესი დასრულდება, ჩვენ შეგვიძლია გამოვიყენოთ მიღებული ინფორმაცია სხვადასხვა მიზნებისთვის. მიღებული შედეგები ძალიან მნიშვნელოვანია გადაწყვეტილების მიღების პროცესში, ამიტომ მონაცემთა მოპოვება მნიშვნელოვან როლს ასრულებს გადაწყვეტილების მიღებაში, ორგანიზაციებს აწვდის შესაბამის ინფორმაციას, რომელიც მათ სჭირდებათ გადაწყვეტილებების მისაღებად. ის ეხმარება ბიზნესს ტენდენციების იდენტიფიცირებაში, მონაცემებში შაბლონების და კავშირების დადგენაში, რომლებიც შეიძლება გამოყენებულ იქნას ახალი სტრატეგიების შემუშავებისთვის, არსებული პროცესების ოპტიმიზაციისთვის.

ბიზნესი ითვისებს რისკებს და იღებს გადაწყვეტილებებს ფაქტებისა და შეხედულებების ფართო ნაკრების საფუძველზე. მომავლის შესახებ სანდო ცოდნა კრიტიკულად მნიშვნელოვანია მათთვის, რათა მიიღონ სწორი გადაწყვეტილებები რისკის დაბალი დონით. მოქმედების სიჩქარე ექსპონენტურად გაიზარდა ინტერნეტის ზრდასთან ერთად. კონკურენტულ სამყაროში გადაწყვეტილების მიღების სიჩქარე და შემდგომი მოქმედება შეიძლება იყოს მთავარი უპირატესობა.

არსებობს გადაწყვეტილებების მიღების ორი ძირითადი ტიპი: სტრატეგიული გადაწყვეტილებები და ოპერატიული გადაწყვეტილებები. სტრატეგიული გადაწყვეტილებები არის ის, რაც გავლენას ახდენს კომპანიის მიმართულებაზე. ახალ მომხმარებელთან დაკავშირების გადაწყვეტილება იქნება სტრატეგიული გადაწყვეტილება. ოპერატიული გადაწყვეტილებები უფრო რუტინული და ტაქტიკური გადაწყვეტილებებია, რომლებიც ორიენტირებულია უფრო მეტი ეფექტურობის განვითარებაზე.

სტრატეგიული გადაწყვეტილების მიღებისას თავად მიზანი შეიძლება არ იყოს მკაფიო და იგივე ეხება მიზნის მიღწევის გზასაც. გადაწყვეტილების შედეგები გარკვეული პერიოდის შემდეგ გამოჩნდება. DM-ს შეუძლია დაეხმაროს მრავალი შესაძლო სცენარის ანალიზში.

საოპერაციო გადაწყვეტილებები შეიძლება უფრო ეფექტური იყოს არსებული მონაცემების ანალიზის გამოყენებით. კლასიფიკაციის სისტემა შეიძლება შეიქმნას და მოდელირდეს არსებული ინსტანციების მონაცემების გამოყენებით დომენის კარგი მოდელის შესაქმნელად. ამ მოდელს შეუძლია მომავალში გააუმჯობესოს ოპერატიული გადაწყვეტილებების მიღების პროცესი.

მონაცემთა მოპოვების ყველაზე გავრცელებული გამოყენების სფეროები გადაწყვეტილების მიღებისას მოიცავს:

1. მარკეტინგი: მონაცემთა მოპოვება გამოიყენება მომხმარებლის ქცევისა და პრეფერენციების გასაანალიზებლად, ბაზრის ახალი შესაძლებლობების იდენტიფიცირებისთვის და მიზნობრივი მარკეტინგული კამპანიების შესაქმნელად.

2. რისკების მართვა: მონაცემთა მოპოვება გამოიყენება პოტენციური რისკებისა და თაღლითობის დასადგენად, როგორცაა საკრედიტო ბარათების თაღლითობა, სადაზღვევო თაღლითობა და ფულის გათეთრება.

3. ჯანდაცვა: მონაცემთა მოპოვება გამოიყენება პაციენტის მონაცემების გასაანალიზებლად, შაბლონების იდენტიფიცირებისთვის და შედეგების პროგნოზირებისთვის, რაც ექიმებს უფრო ზუსტი დიაგნოზის დასმაში და უფრო ეფექტური მკურნალობის გეგმების შემუშავებაში.

4. ფინანსები: მონაცემთა მოპოვება გამოიყენება ფინანსური მონაცემების გასაანალიზებლად, შაბლონების იდენტიფიცირებისთვის და ბაზრის ტენდენციების პროგნოზირებისთვის, რაც ინვესტორებს დაეხმარება უკეთესი საინვესტიციო გადაწყვეტილებების მიღებაში.

მონაცემთა მოპოვების გამოყენება გადაწყვეტილების მიღებისას მრავალრიცხოვანი და მრავალფეროვანია და ის შეიძლება გამოყენებულ იქნას ბიზნეს გადაწყვეტილებების ფართო სპექტრის ინფორმირებისთვის.

მონაცემთა მოპოვება შეიძლება გამოყენებულ იქნას გადაწყვეტილების მიღებისას შემდეგი გზების საშუალებით:

ტენდენციებისა და შაბლონების იდენტიფიცირება: მონაცემთა დიდი მოცულობის ანალიზით, მონაცემთა მოპოვებამ შეიძლება გამოავლინოს შაბლონები და ტენდენციები, რომლებიც შეიძლება ერთი შეხედვით არ იყოს აშკარა. ეს

შეხედულებები შეიძლება დაეხმაროს ბიზნესს ინფორმირებული გადაწყვეტილებების მიღებაში ყველაფრის შესახებ, მარკეტინგული სტრატეგიებიდან დაწყებული პროდუქტის განვითარებით დამთავრებული.

პროგნოზირებადი მოდელირება: მონაცემთა მოპოვება ასევე შეიძლება გამოყენებულ იქნას პროგნოზირებადი მოდელების შესაქმნელად, რომლებიც ეხმარება ბიზნესს მომავალი ტენდენციებისა და შედეგების პროგნოზირებაში. ეს შეიძლება იყოს განსაკუთრებით გამოსადეგი ინდუსტრიებში, სადაც პროგნოზირება კრიტიკულად მნიშვნელოვანია, მაგალითად ფინანსები, ჯანდაცვა და წარმოება.

მომხმარებელთა სეგმენტაცია: მონაცემთა მოპოვება შეიძლება დაეხმაროს ბიზნესს მომხმარებლების სხვადასხვა ჯგუფებად დაყოფაში, საერთო მახასიათებლების საფუძველზე, როგორცაა დემოგრაფიული მონაცემები, შესყიდვების და პრეფერენციები. შემდეგ ეს ინფორმაცია შეიძლება გამოყენებულ იქნას მიზნობრივი მარკეტინგული კამპანიებისა და მომხმარებელთა პერსონალიზებული გამოცდილების შესაქმნელად.

თაღლითობის გამოვლენა: მონაცემთა მოპოვება შეიძლება გამოყენებულ იქნას თაღლითური საქმიანობის ნიმუშების დასადგენად, როგორცაა საკრედიტო ბარათის თაღლითობა ან სადაზღვევო თაღლითობა. მონაცემთა დიდი მოცულობის ანალიზით, მონაცემთა მოპოვების ალგორითმებს შეუძლიათ საეჭვო ქცევის იდენტიფიცირება და შემდგომი გამოკვლევებისთვის მონიშვნა.

მიწოდების ჯაჭვის ოპტიმიზაცია: მონაცემთა მოპოვება შეიძლება გამოყენებულ იქნას მიწოდების ჯაჭვის მონაცემების გასაანალიზებლად და გაუმჯობესებისთვის საჭირო სფეროების იდენტიფიცირებისთვის, როგორცაა დაბრკოლებები ან არაეფექტურობა. ეს შეიძლება დაეხმაროს ბიზნესს მიწოდების ჯაჭვის ოპტიმიზაციაში და ხარჯების შემცირებაში.

მონაცემთა მოპოვების როლი ბიზნეს ანალიტიკასა და ცოდნის მენეჯმენტში

თანამედროვე ბიზნესში მონაცემები სწრაფად იქცევა უმნიშვნელოვანეს აქტივად, რომლის ეფექტური ანალიზი და ინტერპრეტაცია წარმოადგენს კონკურენტული

უპირატესობის მიღწევის ერთ-ერთ მთავარ გზას. მონაცემთა მოპოვება (Data Mining) სწორედ ამ პროცესში ასრულებს არსებით როლს. მიუხედავად იმისა, რომ ტრადიციული ბიზნეს ანალიტიკის სისტემები ყურადღებას ამახვილებენ ბიზნეს პროცესების შეფასებასა და ოპტიმიზაციაზე, მონაცემთა მოპოვება წარმოადგენს იმ ტექნოლოგიას, რომელიც მონაცემთა დიდ მასივებში დამალული კანონზომიერებების, კავშირებისა და შაბლონების აღმოჩენას უზრუნველყოფს.

ბიზნეს ანალიტიკის სისტემები, როგორც წესი, მუშაობენ სტრუქტურირებულ მონაცემებზე, რომლებიც მოიპოვება ოპერაციული სისტემებიდან, მონაცემთა საცავებიდან ან კონტენტის საცავებიდან. თუმცა, ამ მონაცემების სრული პოტენციალის გამოყენება შეუძლებელია მხოლოდ ანგარიშიანობის ან შედარებითი ანალიზის გზით. აქ ჩნდება მონაცემთა მოპოვების მნიშვნელობა, რადგან სწორედ ის იძლევა შესაძლებლობას, შეისწავლოს მონაცემებში დაფარული უცნობი, მაგრამ ბიზნესისთვის ღირებული ინფორმაცია.

მონაცემთა მოპოვების პროცესი მოიცავს სხვადასხვა მეთოდს და ალგორითმს, როგორცაა კლასიფიკაცია, კლასტერიზაცია, ასოციაციური ანალიზი, ანომალიების აღმოჩენა და რეგრესიული მოდელირება. მაგალითად, გაყიდვების მონაცემთა ანალიზისას მონაცემთა მოპოვება შეიძლება გამოავლინოს მომხმარებლის ქცევის უცნობი შაბლონები, რაც მენეჯმენტს საშუალებას მისცემს მიზანმიმართული კამპანიები დაგეგმოს, გააუმჯობესოს მარაგების მართვა და გაზარდოს მომგებიანობა.

ცოდნის მენეჯმენტის კონტექსტშიც მონაცემთა მოპოვებას უდიდესი მნიშვნელობა ენიჭება. ტრადიციული BI სისტემები მომხმარებლებს აწვდიან ინფორმაციას, ხოლო გადაწყვეტილების მიღებისა და მოქმედების პროცესი დიდწილად ადამიანის ცოდნაზე და გამოცდილებაზეა დამოკიდებული. მონაცემთა მოპოვება ამ პროცესს ავარტივებს და ავსებს, რადგან ის ავტომატურად პოულობს კანონზომიერებებს, რაც ბიზნეს ექსპერტებს აძლევს საშუალებას უფრო სწრაფად მიიღონ გადაწყვეტილებები და სწორად იმოქმედონ. (შმუელი, et al. 2018, 19)

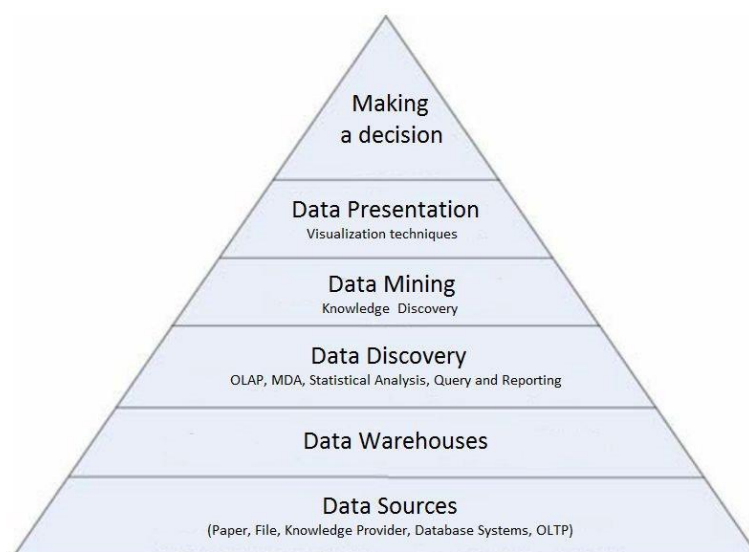
განსაკუთრებით მნიშვნელოვანია მონაცემთა მოპოვების და ბიზნეს პროცესებთან დაკავშირებული actionable ინფორმაციის კომბინაცია. ასე იქმნება საფუძველი ჭკვიანი ბიზნეს ანალიტიკის ჩარჩოსთვის, სადაც მონაცემები არა

მხოლოდ აღწერს წარსულს, არამედ წინასწარ განსაზღვრავს შესაძლო შედეგებს და რეკომენდაციებს იძლევა სამომავლო ქმედებებისთვის.

სამწუხაროდ, დღევანდელი BI აპლიკაციები და გადაწყვეტილებები ჯერ კიდევ მეტწილად მონაცემთა ცენტრირებული მიდგომით მუშაობენ და ნაკლებ ყურადღებას აქცევენ პროცესზე ორიენტირებულ მონაცემთა მოპოვების მოდელს. თუმცა, ეს ტენდენცია იცვლება და თანამედროვე ჭკვიანი BI ჩარჩოები უკვე ინტეგრირებას ახდენენ მონაცემთა მოპოვების მოდულს, რომლებიც თანამშრომლობენ ბიზნეს პროცესების მართვის, დაგეგმვის, კონტენტის მართვისა და თანამშრომლური პროგრამების სისტემებთან.

შედეგად, კომპანიები იღებენ არა მხოლოდ ინფორმაციას მიმდინარე საქმიანობაზე, არამედ შესაძლებლობას, გამოიყენონ პროგნოზული ანალიტიკა, შეამცირონ რისკები, დაიხვეწონ პროცესები და მიაღწიონ მაღალი ხარისხის მომხმარებელთა მომსახურებულ კომპანია აგროვებს გაყიდვების მონაცემებს 12 თვის განმავლობაში.

მაგალითი: მონაცემთა მოპოვება პოულობს, რომ მომხმარებელთა გარკვეული ჯგუფი მხოლოდ ფასდაკლებისას ყიდულობს გარკვეულ პროდუქტს. მიღებული ინფორმაციაზე დაყრდნობით მარკეტინგის განყოფილება გეგმავს სარეკლამო კამპანიას სპეციალურად ამ სეგმენტისთვის. შედეგად შესაძლოა კომპანიამ მიიღოს გაყიდვების 15%-იანი ზრდა 3 თვის განმავლობაში.

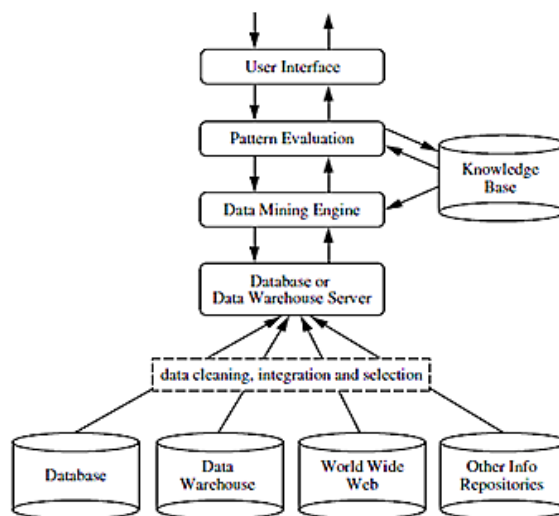


სურათი 1.1. მონაცემთა მოპოვების პროცესი ბიზნეს ანალიტიკის კონტექსტში. (წყარო: <https://shorturl.at/JVVBb>)

მონაცემთა მოპოვება არა მხოლოდ აძლიერებს ტრადიციული BI სისტემების ფუნქციონალს, არამედ ქმნის საფუძველს პროგნოზული ანალიტიკისა და პერსონალიზირებული გადაწყვეტილებების მისაღებად. მონაცემთა მოპოვება არის მძლავრი ინსტრუმენტი, რომელსაც შეუძლია მნიშვნელოვანი დახმარება გაუწიოს ორგანიზაციებს. ამიტომ ფართოდ გამოიყენება სხვადასხვა ინდუსტრიებსა და აპლიკაციებში ბიზნესის მუშაობის გასაუმჯობესებლად, რისკის შესამცირებლად და მომხმარებელთა გამოცდილების გასაუმჯობესებლად. მთლიანობაში, მონაცემთა მოპოვება ღირებული ინსტრუმენტია გადაწყვეტილების მიღებისთვის, რადგან მას შეუძლია დაეხმაროს ბიზნესს აღმოაჩინოს შეხედულებები და ტენდენციები, რომლებიც შეიძლება ერთი შეხედვით არ იყოს აშკარა. მონაცემთა ძალის გამოყენებით, ბიზნესს შეუძლია მიიღოს უფრო ინფორმირებული გადაწყვეტილებები და მოიპოვოს კონკურენტული უპირატესობა თავიანთ ინდუსტრიებში.

1.2 მონაცემთა მოპოვების სისტემების მიმოხილვა და ანალიზი

მონაცემთა მოპოვება წარმოადგენს ტექნოლოგიას, რომელიც აერთიანებს მონაცემთა ანალიზის მეთოდებს და რთულ ალგორითმებს დიდი მონაცემთა მასივების დასამუშავებლად. იგი ასევე არის აქტიური კვლევის სფერო, რომელიც მიზნად ისახავს მონაცემთა ანალიზის ახალი მეთოდების განვითარებას მონაცემთა ახალი ფორმებისათვის. ამ ხედვაზე დაფუძნებით, ტიპური მონაცემთა მოპოვების სისტემის არქიტექტურას შეიძლება ჰქონდეს შემდეგი ძირითადი კომპონენტები, რაც წარმოდგენილია სურათზე 1-ზე.



სურათი 1.2. მონაცემთა მოპოვების სისტემების არქიტექტურა (წყარო: <https://rb.gy/l2se9>)

მომხმარებლის ინტერფეისი: ეს მოდული უზრუნველყოფს მომხმარებლებსა და მონაცემთა მოპოვების სისტემას შორის კომუნიკაციას, რაც მომხმარებელს საშუალებას აძლევს დააყენოს მონაცემთა მოპოვების მოთხოვნა ან ამოცანა, მიაწოდოს ინფორმაცია ძიების კონცენტრირებისათვის და აწარმოოს შესწავლითი მონაცემთა მოპოვება შუალედური შედეგების საფუძველზე.

შაბლონების შეფასების მოდული: ეს კომპონენტი, როგორც წესი, იყენებს საინტერესობის (interestingness) საზომებს და მონაცემთა მოპოვების მოდულებთან ერთად მუშაობს, რათა ძიება კონცენტრირდეს საინტერესო შაბლონებზე.

მონაცემთა მოპოვების ძრავა: ეს წარმოადგენს მონაცემთა მოპოვების სისტემის ძირითად ნაწილს და, იდეალურად, შედგება ფუნქციური მოდულებისგან ისეთი ამოცანებისთვის, როგორცაა მონაცემთა აღწერა, ასოციაციებისა და კორელაციის ანალიზი, კლასიფიკაცია, პროგნოზირება, კლასტერული ანალიზი, ანომალიების ანალიზი და ევოლუციური ანალიზი.

ცოდნის ბაზა: ეს არის დარგობრივი ცოდნა, რომელიც გამოიყენება ძიების წარმართვისა ან მიღებული შაბლონების საინტერესობის შეფასებისათვის. ასეთი ცოდნა შეიძლება მოიცავდეს კონცეფტურ იერარქიებს, რომლებიც გამოიყენება ატრიბუტების ან მათი მნიშვნელობების სხვადასხვა აბსტრაქციის დონეზე დასაჯგუფებლად.

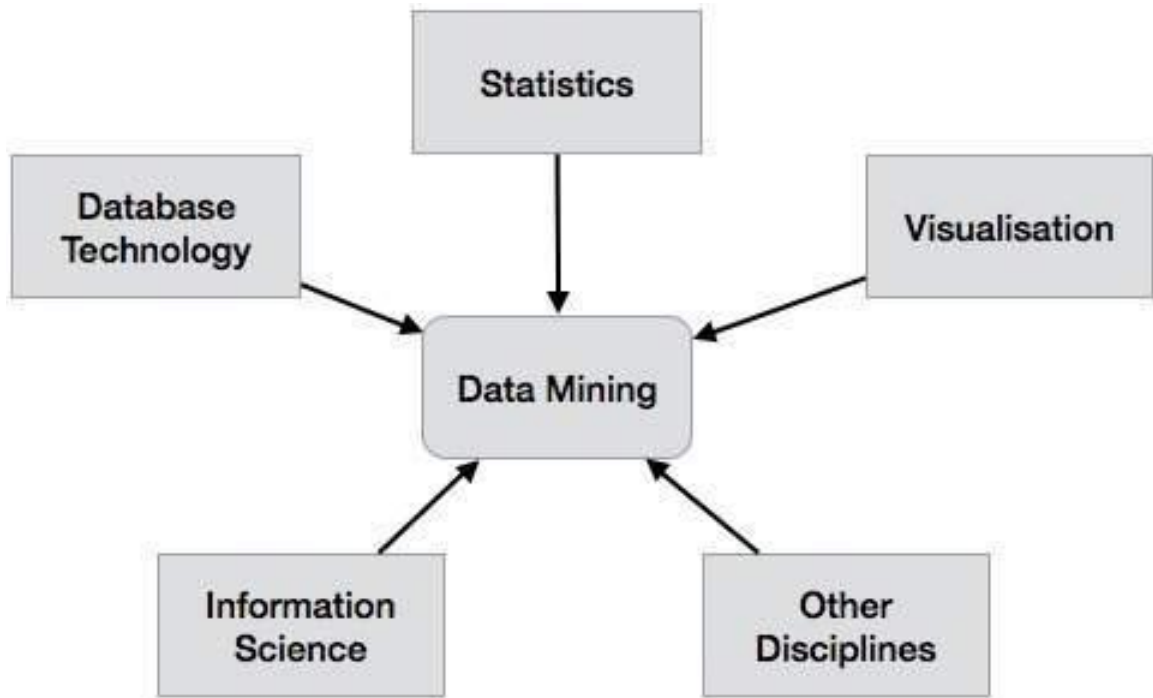
მონაცემთა ბაზის ან მონაცემთა საცავის სერვერი: მონაცემთა ბაზის ან მონაცემთა საცავის სერვერი პასუხისმგებელია შესაბამისი მონაცემების მოპოვებაზე, მომხმარებლის მონაცემთა მოპოვების მოთხოვნის საფუძველზე.

მონაცემთა ბაზა, მონაცემთა საცავი, ინტერნეტი (World Wide Web) ან სხვა ინფორმაციული საცავი: ეს შეიძლება იყოს ერთი ან ერთზე მეტი მონაცემთა ბაზა, მონაცემთა საცავი, ცხრილები ან სხვა ტიპის ინფორმაციული საცავები. მონაცემებზე შეიძლება განხორციელდეს მონაცემთა გასუფთავება და ინტეგრაცია.

მონაცემთა მოპოვების სისტემა შეიძლება კლასიფიცირდეს შემდეგი კრიტერიუმების მიხედვით:

- მონაცემთა ბაზის ტექნოლოგია,
- სტატისტიკა,

- მანქანური სწავლება,
- საინფორმაციო მეცნიერება,
- ვიზუალიზაცია,
- სხვა დისციპლინები.



სურათი 1.3. მონაცემთა მოპოვების კლასიფიკაციის კრიტერიუმები (წყარო: <https://rb.gy/l2se9>)

გარდა ამისა, მონაცემთა მოპოვების სისტემის კლასიფიკაცია ასევე შესაძლებელია შემდეგი ტიპის მიხედვით: (ა) მოპოვებული მონაცემთა ბაზები, მოპოვებული ცოდნა, გამოყენებული ტექნიკა და ადაპტირებული აპლიკაციები.

1.2.1 მონაცემთა მოპოვების ტექნიკები

მონაცემების მოპოვება შესაძლებელია გამოიყენებოდეს მომავალში უფრო ეფექტური გადაწყვეტილებების მისაღებად ან შეიძლება გამოყენებულ იქნას მონაცემების შესასწავლად საინტერესო ასოციაციური შაბლონების მოსაძებნად. სწორი ტექნიკა დამოკიდებულია უშუალოდ პრობლემის ტიპზე.

მონაცემთა მოპოვების მნიშვნელოვანი მეთოდები		
ზედამხედველობითი სწავლა: კლასიფიკაცია	მანქანური სწავლების მეთოდები	გადაწყვეტილებათა ხე ხელოვნური ნეირონული ქსელი
	სტატისტიკური მეთოდები	რეგრესია
უკონტროლო სწავლა: კვლევა	მანქანური სწავლების მეთოდები	კლასტერების ანალიზი ასოციაციის წესების მოპოვება

ცხრილი. 1.1: მონაცემთა მოპოვების მნიშვნელოვანი ტექნიკები(წყარო: ავტორი).

მონაცემთა მოპოვების გამოყენებით გადაჭრილი პრობლემების ყველაზე მნიშვნელოვანი კლასია კლასიფიკაციის პრობლემები. ეს არის პრობლემები, სადაც მონაცემები წარსული გადაწყვეტილებებიდან არის მოპოვებული იმ რამდენიმე წესისა და შაბლონის გამოსატანად, რაც მომავალში გააუმჯობესებს გადაწყვეტილების მიღების პროცესის სიზუსტეს. წარსული გადაწყვეტილებების მონაცემები ორგანიზებულია და მოპოვებულია გადაწყვეტილების წესების ან განტოლებისთვის, რომლებიც შემდეგ კოდიფიცირებულია უფრო ზუსტი გადაწყვეტილებების მისაღებად. კლასიფიკაციის ტექნიკას ეწოდება ზედამხედველობითი სწავლება, რადგან არსებობს გზა, რომ აკონტროლოთ მოდელის პროგნოზი სწორია თუ არასწორი.

გადაწყვეტილების ხეები (Decision Trees) არის იერარქიულად ორგანიზებული განშტოება, სტრუქტურირებული, რათა დაეხმაროს გადაწყვეტილების მიღებას მარტივად და ლოგიკურად. გადაწყვეტილების ხეები მონაცემთა მოპოვების ყველაზე პოპულარული ტექნიკაა, მრავალი მიზეზის გამო. მათ შორისაა:

1. გადაწყვეტილების ხეები ადვილად გასაგები და გამოსაყენებელია, როგორც ანალიტიკოსების, ასევე აღმასრულებლების მიერ. ისინი ასევე აჩვენებენ მაღალი პროგნოზირების სიზუსტეს.

2. ისინი ავტომატურად ირჩევენ ყველაზე შესაბამის ცვლადებს, როგორც ყველა ცვლადისგან ხელმისაწვდომი ცვლადები გადაწყვეტილების მიღებისთვის.

3. გადაწყვეტილების ხეები ტოლერანტულია მონაცემთა ხარისხის საკითხების მიმართ და არ საჭიროებს ბევრი მონაცემების მომზადებას მომხმარებლებისგან.

4. არაწრფივი ურთიერთდამოკიდებულებებიც კი კარგად შეიძლება დამუშავდეს გადაწყვეტილების ხეებით.

არსებობს მრავალი ალგორითმი გადაწყვეტილების ხეების განსახორციელებლად. მათ შორის ყველაზე ხშირად გამოყენებადს წარმოადგენს: C5, CART და CHAID.

რეგრესია (Regression) არის შედარებით მარტივი და ყველაზე პოპულარული სტატისტიკური მონაცემების მოპოვების ტექნიკა. მისი მიზანი არის გლუვი კარგად განსაზღვრული მრუდის მორგება მონაცემებზე. რეგრესიული ანალიზის ტექნიკა, მაგალითად, შეიძლება გამოყენებულ იქნას ენერჯის მოხმარების მოდელირებისთვის და პროგნოზირებისთვის, როგორც ყოველდღიური ტემპერატურის ფუნქცია. მონაცემების უბრალოდ გამოსახვა აჩვენებს არაწრფივ მრუდს. არაწრფივი რეგრესიის განტოლების გამოყენება ძალიან კარგად მოერგება მონაცემებს მაღალი სიზუსტით. ამრიგად, ენერჯის მოხმარება ნებისმიერ მომავალ დღეს შეიძლება განისაზღვროს ამ განტოლების გამოყენებით.

ხელოვნური ნეირონული ქსელი (Artificial neural network) (ANN) არის მონაცემთა მოპოვების დახვეწილი ტექნიკა ხელოვნური ინტელექტის ნაკადიდან კომპიუტერულ მეცნიერებაში. ის ბაძავს ადამიანის ნერვული სტრუქტურის ქცევას: ნეირონები იღებენ სტიმულს, ამუშავებენ მათ და აცნობენ მათ შედეგებს სხვა ნეირონებს თანმიმდევრულად, და საბოლოოდ ნეირონები იღებენ გადაწყვეტილებას. გადაწყვეტილების ამოცანის დამუშავება შესაძლებელია მხოლოდ ერთი ნეირონის მიერ და შედეგის გადაცემაც შესაძლოა საკმაოდ სწრაფდ მოხდეს. ალტერნატიულად, ნეირონების მრავალი ფენა შეიძლება იყოს ჩართული გადაწყვეტილების ამოცანაში, დომენის სირთულის მიხედვით. ნეირონული ქსელის „გაწვრთნა“ შესაძლებელია მრავალჯერადი გადაწყვეტილების მიღებით, მრავალი მონაცემთა წერტილით. ის გააგრძელებს სწავლას მისი შიდა გამოთვლებისა და კომუნიკაციის პარამეტრების კორექტირებით, წინა გადაწყვეტილებებზე მიღებული გამოხმაურების საფუძველზე. ნეირონების ფენებში გადაცემულ შუალედურ მნიშვნელობებს დამკვირვებლისთვის

შეიძლება არ ჰქონდეს ინტუიციური აზრი. ამრიგად, ნეირონული ქსელები განიხილება შავი ყუთის სისტემად.

გარკვეულ მომენტში, ნეირონული ქსელი საკმარისად ისწავლის და დაიწყებს ადამიანის ექსპერტის პროგნოზირების სიზუსტესთან ან კლასიფიკაციის ალტერნატიულ ტექნიკასთან შესაბამისობაში მოსვლას. ზოგიერთი ANN-ის პროგნოზები, რომლებიც დიდი ხნის განმავლობაში გაწვრთნილი იყო დიდი რაოდენობით მონაცემებით, გადამწყვეტად უფრო ზუსტი გახდა, ვიდრე ადამიანი ექსპერტები. ამ მომენტში, შეიძლება ANN-ების სერიოზულად განხილვა, რეალურ სიტუაციებში და რეალურ დროში გამოსაყენებლად.

ANN-ები პოპულარობა განპირობებულია მათი პოტენციურად მაღალი პროგნოზირების სიზუსტის გამო. ANN-ები ასევე შედარებით მარტივია დასანერგად და არ აქვთ რაიმე პრობლემა მონაცემთა ხარისხთან დაკავშირებით. თუმცა ANN-ები მოითხოვს უამრავ მონაცემს ვარჯიშისთვის, რათა განავითარონ კარგი პროგნოზირების უნარი.

კლასტერული ანალიზი (Cluster analysis) არის საძიებო სწავლის ტექნიკა, რომელიც ეხმარება მსგავსი ობიექტების ამოცნობას მონაცემებში. ეს არის ტექნიკა, რომელიც გამოიყენება ნივთების ბუნებრივი დაჯგუფების ავტომატური იდენტიფიკაციისთვის. მონაცემთა ინსტანციები, რომლებიც ერთმანეთის მსგავსია კატეგორიზირდება ერთ კლასტერში, ხოლო მონაცემთა ინსტანციები, რომლებიც ძალიან განსხვავდებიან ერთმანეთისგან, იყოფა ცალკეულ კლასტერებად. ამ შემთხვევაში შეიძლება იყოს ნებისმიერი რაოდენობის კლასტერები, რომლებიც შეიძლება წარმოიქმნას მონაცემებით. K ნიშნავს, მიდგომა პოპულარული ტექნიკაა და მომხმარებელს აძლევს მონაცემებიდან კლასტერების სწორი რაოდენობის (K) არჩევის შესაძლებლობას.

კლასტერული ანალიზი ასევე ცნობილია, როგორც სემენტაციის ტექნიკა. ტექნიკა აჩვენებს ნივთების ჯგუფს წარსული მონაცემებიდან. გამომავალი არის ცენტროიდები თითოეული კლასტერისთვის და მონაცემთა განაწილება მათ კლასტერზე. ცენტროიდის განმარტება გამოიყენება მონაცემთა ახალი ინსტანციების

მინიჭებისთვის, რომლებიც შეიძლება მიენიჭოს მათ კლასტერულ სახლებს. კლასტერირება ასევე არის ხელოვნური ინტელექტის ტექნიკის ოჯახის ნაწილი.

ასოციაციის წესები (Association rules) არის მონაცემთა მოპოვების პოპულარული მეთოდი ბიზნესში, განსაკუთრებით იქ, სადაც გაყიდვებია. ასევე ცნობილია, როგორც ბაზრის კალათის ანალიზი, ის ეხმარება პასუხის გაცემას კითხვებზე ჯვარედინი გაყიდვების შესაძლებლობების შესახებ. ეს არის პერსონალიზაციის ძრავის გული, რომელსაც იყენებენ ელექტრონული კომერციის საიტები, როგორცაა Amazon.com და ფილმების საიტები, როგორცაა Netflix.com. ტექნიკა გვეხმარება ცვლადებს (ერთეულებსა თუ მოვლენებს) შორის საინტერესო კავშირების (აფინიტების) პოვნაში. ისინი წარმოდგენილია $X \Rightarrow Y$ ფორმის წესებით, სადაც X და Y არის მონაცემთა ერთეულების ნაკრები. უკონტროლო სწავლის(Unsupervised learning) ფორმა, მას არ გააჩნია დამოკიდებული ცვლადი; და არ არსებობს სწორი ან არასწორი პასუხები. უბრალოდ არის უფრო ძლიერი და სუსტი კავშირები. ამრიგად, თითოეულ წესს აქვს მისთვის მინიჭებული ნდობის დონე. ასოციაციის წესები მანქანური სწავლების ოჯახის ნაწილის წარმოადგენს (შმუელი, et al. 2018, 16).

1.2.2 მონაცემთა წინასწარი მომზადება

მონაცემთა გაწმენდა და მომზადება

მონაცემთა ხარისხი გადამწყვეტია მონაცემთა მოპოვების პროექტის წარმატებისა და ღირებულებისთვის, წინააღმდეგ შემთხვევაში, სიტუაცია იქნება ისეთი, როგორც არის ნაგვის შეტანა და ნაგვის გამოტანა (GIGO – Garbage In, Garbage Out). შემომავალი მონაცემების ხარისხი განსხვავდება მონაცემთა წყაროსა და ბუნების მიხედვით. შიდა ოპერაციების მონაცემები, სავარაუდოდ, უფრო მაღალი ხარისხის იქნება, რადგან იქნება ზუსტი და თანმიმდევრული. სოციალური მედიიდან და სხვა საჯარო წყაროებიდან მიღებული მონაცემები ნაკლებად ექვემდებარება ბიზნესის კონტროლს და ნაკლებად სავარაუდოა, რომ იყოს სანდო.

მონაცემები თითქმის აუცილებლად უნდა გაიწმინდოს და გარდაიქმნას, სანამ მისი გამოყენება მოხდება მონაცემთა მოპოვებისთვის(DM). არსებობს მრავალი გზა, რისთვისაც შეიძლება საჭირო გახდეს მონაცემების გაწმენდა - გამოტოვებული

მნიშვნელობების შევსება, გამონაკლისების ეფექტების გაბატონება, ველების ტრანსფორმაცია, უწყვეტი ცვლადების დაკავშირება და მრავალი სხვა - სანამ ის იქნება მზად ანალიზისთვის. მონაცემთა გაწმენდა და მომზადება არის შრომატევადი, თუმცა შეიძლება იყოს ნახევრად ავტომატიზებული აქტივობა, რომელსაც შეიძლება დასჭირდეს მონაცემთა მოპოვების პროექტისთვის საჭირო დროის 60-დან 70 პროცენტამდე (შმუელი, et al. 2018, 26).

მონაცემთა გაწმენდის და მომზადების ეტაპი მოითხოვს შემდეგი პროცედურების შესრულებას:

1. უნდა წაიშალოს დუბლირებული მონაცემები. ერთი და იგივე მონაცემები შეიძლება მიღებულ იქნას სხვადასხვა წყაროდან. მონაცემთა ნაკრების შერწყმისას, დუბლირებული მონაცემები უნდა წაიშალოს.

2. გამოტოვებული მნიშვნელობები უნდა იყოს შევსებული, ან ეს სტრიქონები ამოღებული უნდა იქნას ანალიზიდან. გამოტოვებული მნიშვნელობები შეიძლება შეივსოს საშუალო ან მოდალური ან ნაგულისხმევი მნიშვნელობებით.

3. შეიძლება საჭირო გახდეს ელემენტების გარდაქმნა ერთი ერთეულიდან მეორეზე. მაგალითად, ჯანდაცვის ჯამური ხარჯები და პაციენტთა მთლიანი რაოდენობა შეიძლება შემცირდეს ხარჯი/პაციენტი, რათა მოხდეს ამ ღირებულების შედარება.

4. უწყვეტი მნიშვნელობებს შესაძლოა დასჭირდეთ რამდენიმე ჯგუფში მოთავსება. მაგალითად, სამუშაო გამოცდილება შეიძლება მოთავსდეს, როგორც დაბალი, საშუალო და მაღალი.

5. შეიძლება საჭირო გახდეს მონაცემთა ელემენტების კორექტირება, რათა დროთა განმავლობაში მონაცემები შედარებადი იყოს. მაგალითად, შესაძლოა საჭირო გახდეს ვალუტის ღირებულებების კორექტირება ინფლაციის მიმართ; ასევე შესაძლოა საჭირო გახდეს მათი კონვერტაცია საერთო ვალუტაში.

6. გარე მონაცემების ელემენტები უნდა მოიხსნას განხილვის შემდეგ, რათა თავიდან იქნას აცილებული შედეგების დამახინჯება. მაგალითად, ერთ დიდ დონორს შეუძლია შეცვალოს კურსდამთავრებულთა დონორების ანალიზი საგანმანათლებლო გარემოში.

7. მონაცემთა შერჩევას ნებისმიერი მიკერძოება უნდა გამოსწორდეს, რათა უზრუნველყოფილი იყოს მონაცემები ანალიზის ქვეშ მყოფი ფენომენების რეპრეზენტაციულობა. თუ მონაცემები მოიცავს ერთი სქესის უფრო მეტ წევრს, ვიდრე ტიპიურია ინტერესის პოპულაციისათვის, მაშინ საჭიროა კორექტირება მონაცემთა მიმართ.

8. შედარებადობის უზრუნველსაყოფად მონაცემები უნდა იყოს მიყვანილი ერთიდაიგივე მასშტაბამდე. გაყიდვების მონაცემები შეიძლება ხელმისაწვდომი იყოს ყოველდღიურად, მაგრამ გამყიდველის კომპენსაციის მონაცემები შეიძლება ხელმისაწვდომი იყოს მხოლოდ ყოველთვიურად. ამ ცვლადების დასაკავშირებლად, მონაცემები უნდა დავიყვანოთ ყველაზე დაბალ საერთო მნიშვნელამდე, ამ შემთხვევაში, ყოველთვიურამდე.

9. ინფორმაციის სიმკვრივის გასაზრდელად შეიძლება საჭირო გახდეს მონაცემების შერჩევა. ზოგიერთი მონაცემი შეიძლება არ აჩვენებდეს დიდ ცვალებადობას, რადგან ის არ იყო სათანადოდ ჩაწერილი ან სხვა მიზეზების გამო. ამ მონაცემებმა შეიძლება გააფუჭოს მონაცემების სხვა განსხვავებების ეფექტი და უნდა მოიხსნას მონაცემთა ინფორმაციის სიმკვრივის გასაუმჯობესებლად (ანილ 2015, 48).

მონაცემთა მოპოვების შედეგები

მონაცემთა მოპოვების ტექნიკა შეიძლება ემსახურებოდეს სხვადასხვა ტიპის მიზნებს. მონაცემთა მოპოვების შედეგები ასახავს მიზანს, რომელსაც ემსახურება იგი. მონაცემთა მოპოვების შედეგების მრავალი წარმოდგენა არსებობს.

მონაცემთა მოპოვების ერთ-ერთი პოპულარული ფორმა არის გადაწყვეტილების ხე. ეს არის იერარქიულად განშტოებული სტრუქტურა, რომელიც ეხმარება ვიზუალურად მიყვებს ნაბიჯებს მოდელზე დაფუძნებული გადაწყვეტილების მისაღებად. ხეს შეიძლება ჰქონდეს გარკვეული ატრიბუტები, მაგალითად, თითოეულ ტოტზე მინიჭებული ალბათობა. დაკავშირებული ფორმატი არის ბიზნეს წესების ერთობლიობა, რომელიც არის თუ-მაშინ პირობა, რომლებიც აჩვენებს მიზეზობრიობას. გადაწყვეტილების ხე შეიძლება შესაბამისობაში იყოს ბიზნესის წესებთან. თუ ობიექტური ფუნქცია პროგნოზირებაა, მაშინ გადაწყვეტილების ხე ან ბიზნეს წესები არის გამოსავლის წარმოდგენის ყველაზე შესაფერისი მეთოდი.

გამომავალი მონაცემი შეიძლება იყოს რეგრესიული განტოლების ან მათემატიკური ფუნქციის სახით, რომელიც წარმოადგენს მონაცემთა წარმოსადგენად საუკეთესო მრუდს. ეს განტოლება შეიძლება შეიცავდეს წრფივ და არა წრფივ ტერმინებს. რეგრესიის განტოლებები კარგი გზაა კლასიფიკაციის სავარჯიშოების წარმოსაჩენად.

ბიზნეს წესები არის საბაზრო კალათის ანალიზის სავარჯიშოების შედეგის სათანადო წარმოდგენა. ეს წესები არის if-then პირობები, რომლებიც დაკავშირებულია თითოეულ წესთან დაკავშირებული ალბათობის ზოგიერთი პარამეტრით. მაგალითად, ვინც რძეს და პურს ყიდულობს, კარაქსაც იყიდის (80 პროცენტის ალბათობით) (ანილ 2015, 49).

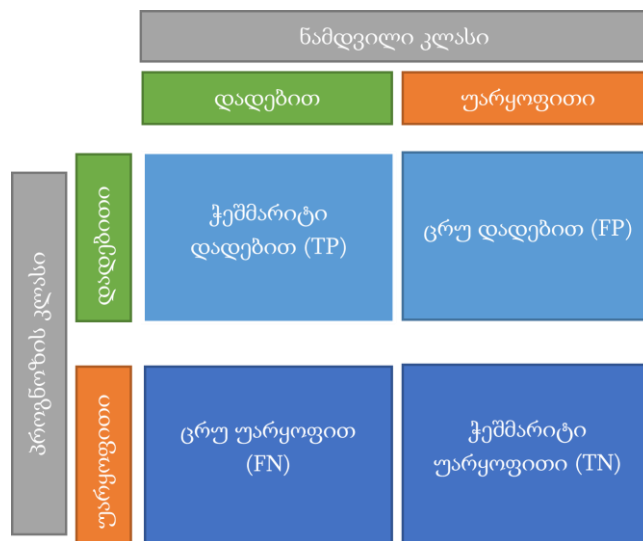
მონაცემთა მოპოვების შედეგების შეფასება

მონაცემთა მოპოვების პროცესების ორი ძირითადი ტიპი არსებობს: ზედამხედველობითი სწავლა და უკონტროლო სწავლა. ზედამხედველობით სწავლაში, გადაწყვეტილების მოდელი შეიძლება შეიქმნას წარსული მონაცემების გამოყენებით და მოდელი შეიძლება გამოყენებულ იქნას მომავალი მონაცემების ინსტანციებისთვის სწორი პასუხის პროგნოზირებისთვის. კლასიფიკაცია არის ზედამხედველობითი სასწავლო აქტივობის ძირითადი კატეგორია. კლასიფიკაციის მრავალი ტექნიკა არსებობს, მაგრამ გადაწყვეტილების ხეები ყველაზე პოპულარულია. თითოეული ეს ტექნიკა შეიძლება განხორციელდეს მრავალი ალგორითმით. საერთო პარამეტრი, ყველა კლასიფიკაციის ტექნიკისთვის არის პროგნოზირების სიზუსტე.

პროგნოზირებადი სიზუსტე = (სწორი პროგნოზები) / სულ პროგნოზები.

დავუშვათ, შემუშავდა მონაცემთა მოპოვების პროექტი, რათა შეიქმნას პროგნოზირებადი მოდელი კიბოთი დაავადებული პაციენტებისთვის, გადაწყვეტილების ხის გამოყენებით. ცვლადების და მონაცემთა ინსტანციების შესაბამისი ნაკრების გამოყენებით, შეიქმნა გადაწყვეტილების ხის მოდელი. მოდელი შემდეგ გამოიყენება მონაცემთა სხვა შემთხვევების პროგნოზირებისთვის. როდესაც ჭეშმარიტი პოზიტიური მონაცემების წერტილი დადებითია, ეს არის სწორი პროგნოზი, რომელსაც ეწოდება ნამდვილი დადებითი (TP). ანალოგიურად, როდესაც

ჭეშმარიტი უარყოფითი მონაცემების წერტილი კლასიფიცირდება როგორც უარყოფითი, ეს არის ნამდვილი უარყოფითი (TN). მეორეს მხრივ, როდესაც ჭეშმარიტ - დადებითი მონაცემების წერტილი კლასიფიცირდება მოდელის მიერ, როგორც უარყოფითი, ეს არის არასწორი პროგნოზი, რომელსაც ეწოდება ცრუ უარყოფითი (FN). ანალოგიურად, როდესაც ჭეშმარიტი-უარყოფითი მონაცემების წერტილი კლასიფიცირდება როგორც დადებითი, ის კლასიფიცირდება როგორც ცრუ დადებითი (FP). ამას ეწოდება დაბნეულობის მატრიცა, ცხრილი. 1.2.



ცხრილი. 1.2: დაბნეულობის მატრიცა (Confusion Matrix) (წყარო: ავტორი).

ამრიგად, პროგნოზირების სიზუსტე შეიძლება განისაზღვროს შემდეგი ფორმულით:

$$\text{პროგნოზის სიზუსტე} = (TP + TN) / (TP + TN + FP + FN).$$

კლასიფიკაციის ყველა ტექნიკას აქვს პროგნოზირებადი სიზუსტე, რომელიც დაკავშირებულია პროგნოზირებულ მოდელთან. უმაღლესი მნიშვნელობა შეიძლება იყოს 100 პროცენტი. პრაქტიკაში, 70 პროცენტზე მეტი სიზუსტის პროგნოზირებადი მოდელები შეიძლება ჩაითვალოს გამოსაყენებლად ბიზნეს დომენებში, ბიზნესის ბუნებიდან გამომდინარე.

არ არსებობს კარგი ობიექტური ზომები უკონტროლო სწავლის (unsupervised learning) ტექნიკის სიზუსტის შესაფასებლად, როგორცაა კლასტერული ანალიზი. ამ შემთხვევაში არ არსებობს მხოლოდ ერთი სწორი პასუხი მიღებული შედეგებიდან. სეგმენტაციის მოდელის ღირებულება დამოკიდებულია იმ მნიშვნელობაზე, რომელსაც გადაწყვეტილების მიმღები ხედავს ამ შედეგებში (ანილ 2015, 51).

1.3 დიდი მონაცემების დამუშავების პრობლემები და ამოცანები

1.3.1 დიდი მონაცემების დამუშავების პრობლემატიკის აღწერა

დიდი მონაცემები (Big Data) გულისხმობს მონაცემთა ისეთ მასშტაბურ მოცულობას, მრავალფეროვნებასა და სიჩქარეს, რომლის დამუშავება ტრადიციული მონაცემთა ანალიტიკური საშუალებებით შეუძლებელია. დიდი მონაცემები გამოირჩევა შემდეგი მახასიათებლებით: მოცულობა (Volume), მრავალფეროვნება (Variety), სიჩქარე (Velocity), სიზუსტე (Veracity), ღირებულება (Value).

დიდი მონაცემების დამუშავების ძირითადი გამოწვევები მოიცავს რამდენიმე მნიშვნელოვან კონცეფციას:

მონაცემთა ხარისხი და სისწორე: მონაცემთა დიდი ნაკადები ხშირად შეიცავს შეცდომებს, გაუმართავ ან დუბლირებულ ინფორმაციას, რაც ანალიზის სიზუსტეზე ახდენს გავლენას.

მონაცემთა სკალირება: უზარმაზარი მოცულობის მონაცემების დამუშავება საჭიროებს მასშტაბურ კომპიუტერულ რესურსებს და ინოვაციურ ტექნოლოგიურ პლატფორმებს (Hadoop, Spark, NoSQL).

განაწილებული მონაცემთა ბაზები: მონაცემები ხშირად არის განაწილებული სხვადასხვა გეოგრაფიულ და ჰეტეროგენულ სისტემებში, რაც ამცირებს პროცესის ეფექტიანობას.

რეალურ დროში ანალიზი: ბევრი აპლიკაცია (მაგალითად, სოც. ქსელების მონიტორინგი, ფინანსური ტრანზაქციები) საჭიროებს მონაცემთა ანალიზს რეალურ დროში, რაც დამატებით სირთულეებს ქმნის.

მონაცემთა უსაფრთხოება და კონფიდენციალურობა: GDPR, HIPAA და სხვა რეგულაციები მაღალი სტანდარტის დაცვას მოითხოვს.

გრაფული მონაცემების სპეციფიკური გამოწვევები: გრაფული მონაცემები (Graph Data) გამოირჩევა მაღალი კავშირიანობით, რაც არსებითად აძნელებს მათი დამუშავების ეფექტურობას.

დიდი მონაცემების დამუშავების ალგორითმები საჭიროებენ მაღალი კომპიუტერული რესურსების მოხმარებას, განსაკუთრებით მაშინ, როდესაც

ვექტორიზირებული და გრაფული ნეირონული ქსელები (Graph Neural Networks) გამოიყენება.

არც მონაცემთა მოპოვება არ არის გამონაკლისი, რადგან სწორედ დიდი მონაცემებისთვის არის განკუთვნილი. მონაცემთა მოპოვების ალგორითმები შეიძლება იყოს ძალიან რთული, ამავდროულად უშუალოდ მონაცემებიც არაა მარტივად ხელმისაწვდომი. მონაცემთა მოპოვება უნდა იყოს ინტეგრირებული სხვადასხვა ჰეტეროგენული მონაცემთა წყაროებითან. ეს ფაქტორები ასევე ქმნის გარკვეულ პრობლემებს. ძირითად პრობლემურ საკითხებს წარმოადგენს:

- მოპოვების მეთოდოლოგია და მომხმარებელთა ინტერაქცია,
- შესრულების საკითხები,
- მონაცემთა სხვადასხვა ტიპების საკითხები.



ნახ.1.4: მონაცემთა მოპოვების ძირითადი პრობლემური საკითხები(წყარო: ავტორი).

მოპოვების მეთოდოლოგია და მომხმარებელთა ინტერაქციის პრობლემები:

• **სხვადასხვა სახის ცოდნის მოპოვება მონაცემთა ბაზებში** - სხვადასხვა მომხმარებელი შეიძლება იყოს დაინტერესებული სხვადასხვა სახის ცოდნით. ამიტომ მონაცემთა მოპოვებისთვის აუცილებელია ცოდნის აღმოჩენის ამოცანების ფართო სპექტრი.

• **ცოდნის ინტერაქტიული მოპოვება აბსტრაქციის მრავალ დონეზე** - მონაცემთა მოპოვების პროცესი უნდა იყოს ინტერაქტიული, რადგან ის საშუალებას აძლევს მომხმარებლებს ფოკუსირება მოახდინონ შაბლონების ძიებაზე, უზრუნველყონ და

დახვეწონ მონაცემთა მოპოვების მოთხოვნები დაბრუნებული შედეგების საფუძველზე.

- **ფონური ცოდნის ინკორპორაცია** - აღმოჩენის პროცესის წარმართვისთვის და აღმოჩენილი შაბლონების წარმოსაჩენად, შეიძლება გამოყენებულ იქნას ფონური ცოდნა. ფონური ცოდნა შეიძლება გამოყენებულ იქნას აღმოჩენილი შაბლონების გამოსახატავად არა მხოლოდ მოკლე ტერმინებით, არამედ აბსტრაქციის მრავალ დონეზე.

- **მონაცემთა მოპოვების შეკითხვის ენები და მონაცემთა მოპოვების დროებითი მოპოვება** - მონაცემთა მოპოვების შეკითხვის ენა, რომელიც მომხმარებელს საშუალებას აძლევს აღწეროს ad hoc მოპოვების ამოცანები, უნდა იყოს ინტეგრირებული მონაცემთა საცავის შეკითხვის ენასთან და ოპტიმიზირებული უნდა იყოს მონაცემთა ეფექტური და მოქნილი მოპოვებისთვის.

- **მონაცემთა მოპოვების შედეგების პრეზენტაცია და ვიზუალიზაცია** - შაბლონების აღმოჩენის შემდეგ, ის უნდა იყოს მაღალი დონის ენებით გამოხატული და ვიზუალურად წარმოდგენილი. ეს წარმოდგენები უნდა იყოს ადვილად გასაგები.

- **ხმაურიანი ან არასრული მონაცემების დამუშავება** - მონაცემთა გაწმენდის მეთოდები საჭიროა ხმაურის და არასრული ობიექტების დასამუშავებლად მონაცემთა კანონზომიერებების მოპოვებისას. თუ მონაცემთა გაწმენდის მეთოდები არ არსებობს, აღმოჩენილი შაბლონების სიზუსტე დაბალი იქნება.

- **ნიმუშის შეფასება** - აღმოჩენილი შაბლონები საინტერესო უნდა იყოს, რადგან ისინი ან წარმოადგენენ საერთო ცოდნას, ან მოკლებულია სიახლეს.

შესრულების (Performance) პრობლემები:

შეიძლება წარმოიშვას შესრულებასთან დაკავშირებული პრობლემები, როგორცაა:

- **მონაცემთა მოპოვების ალგორითმების ეფექტურობა და მასშტაბურობა** - მონაცემთა ბაზებში არსებული უზარმაზარი მონაცემებიდან ინფორმაციის ეფექტურად ამოღების მიზნით, მონაცემთა მოპოვების ალგორითმი უნდა იყოს ეფექტური და მასშტაბური.

- პარალელური, განაწილებული და ინკრემენტული მოპოვების ალგორითმები - ისეთი ფაქტორები, როგორცაა მონაცემთა ბაზების უზარმაზარი ზომა, მონაცემთა ფართო განაწილება და მონაცემთა მოპოვების მეთოდების სირთულე, მოტივაციას უწევს მონაცემთა მოპოვების პარალელური და განაწილებული ალგორითმების შემუშავებას. ეს ალგორითმები ყოფს მონაცემებს დანაყოფებად, რომლებიც შემდგომში მუშავდება პარალელურად. ინკრემენტული ალგორითმები, მონაცემთა ბაზის განახლება მონაცემთა ამოღების თავიდან შესრულების გარეშე.

მონაცემთა ტიპების მრავალფეროვნების პრობლემები:

- რელაციური და რთული ტიპის მონაცემების მართვა - მონაცემთა ბაზა შეიძლება შეიცავდეს მონაცემთა კომპლექსურ ობიექტებს, მულტიმედიის მონაცემთა ობიექტებს, სივრცულ მონაცემებს, დროებით მონაცემებს და ა.შ. შეუძლებელია ერთი სისტემის მიერ ყველა ამ ტიპის მონაცემის მოპოვება.
- ინფორმაციის მოპოვება ჰეტეროგენული მონაცემთა ბაზებიდან და გლობალური საინფორმაციო სისტემებიდან - მონაცემები ხელმისაწვდომია სხვადასხვა მონაცემთა წყაროებიდან. ამიტომ ეს მონაცემთა წყარო შეიძლება იყოს სტრუქტურირებული, ნახევრად სტრუქტურირებული ან არასტრუქტურირებული. ამიტომ მათგან ცოდნის ამოღება გამოწვევებს მათებს მონაცემთა მოპოვებას (tutorialspoint.com 2022, 3).

1.3.2 დიდი მონაცემების დამუშავების ამოცანები

თანამედროვე ციფრულ გარემოში მონაცემების მასშტაბური ზრდა ორგანიზაციებისა და კვლევითი ინსტიტუტების წინაშე აყენებს რთულ ამოცანებს, რომლებიც დაკავშირებულია მონაცემთა ეფექტიან შეგროვებასთან, გაანალიზებასთან და მათგან ღირებული ცოდნის მოპოვებასთან. ტრადიციული მონაცემთა ანალიტიკური მეთოდები არაეფექტურია იმ მოცულობის, სიჩქარისა და მრავალფეროვნების მონაცემებთან, რასაც დიდი მონაცემები (Big Data) გულისხმობს.

დისერტაციის მიზანია სწორედ ამ პრობლემის გადაჭრის გზების შესწავლა და ინოვაციური სისტემის შემუშავება, რომელიც დაეფუძნება შემდეგ პრინციპებს:

1. **დიდი მოცულობის მონაცემთა შეგროვება და წინასწარი დამუშავება:** სისტემამ უნდა შეძლოს სხვადასხვა წყაროსგან მიღებული სტრუქტურირებული და არასტრუქტურირებული მონაცემების გაწმენდა, ნორმალიზაცია და ერთიან მონაცემთა მოდელში გარდაქმნა.

2. **ცოდნის ამოღება მონაცემთა მასივიდან:** გამოყენებულ იქნას მონაცემთა მოპოვების (Data Mining) და მანქანური სწავლების ტექნოლოგიები, რათა მოხდეს მონაცემებში დამალული კანონზომიერებების, შაბლონებისა და ურთიერთდამოკიდებულებების გამოვლენა.

3. **ცოდნის მოდელირება გრაფული მანქანური სწავლების გამოყენებით:** სისტემამ უნდა მოახდინოს მიღებული ცოდნის წარმოდგენა და ანალიზი გრაფული მონაცემთა მოდელების საშუალებით, სადაც კვანძები და წიბოები ასახავენ სხვადასხვა ობიექტებსა და მათ შორის არსებულ კავშირებს. ამისათვის დაგეგმილია ისეთი თანამედროვე გრაფული ალგორითმებისა და ნეირონული ქსელების გამოყენება, როგორცაა Graph Neural Networks (GNN), GraphSAGE, Graph Attention Networks (GAT) და სხვა.

4. **სისტემის ტესტირება და შეფასება:** შემუშავებული ალგორითმისა და სისტემის ეფექტიანობის დადასტურება.

ნაშრომის ძირითად ამოცანას წარმოადგენს ჰოლისტური მიდგომის შეთავაზება, რომელიც გააერთიანებს დიდი მონაცემების ანალიზისა და ცოდნის მოდელირების საუკეთესო პრაქტიკას გრაფზე დაფუძნებული ცოდნის მოდელირების ალგორითმის საშუალებით.

დისერტაცია შედგება ხუთი ძირითადი თავისგან, რომელთაგან თითოეული მიზნად ისახავს ცოდნის მოდელირების სისტემის შემუშავების ეტაპობრივი პროცესის გაშლას და ანალიზს დიდი მონაცემების დამუშავებისა და გრაფული მანქანური სწავლების კონტექსტში.

თავი II. ცოდნის წარმოდგენა

2.1 ცოდნის წარმოდგენის მოდელები

ცოდნის წარმოდგენა (Knowledge Representation, KR) ხელოვნური ინტელექტის სფეროს წარმოადგენს, რომელიც შეისწავლის, თუ როგორ შეიძლება სამყაროს შესახებ ცოდნის ფორმალურად სტრუქტურირება ისე, რომ კომპიუტერულმა სისტემამ შეძლოს მისი გამოყენება მსჯელობისა და გადაწყვეტილების მისაღებად (დელგრანდი, და სხვ. 2023, 4). ტრადიციულად, KR ფოკუსირებული იყო **სიმბოლურ წარმოდგენებზე** - ფაქტების, კონცეფციებისა და წესების (მაგ., ლოგიკური ფორმულები ან სემანტიკური გრაფები) ცხად, ადამიანის მიერ წასაკითხ სტილის კოდირებაზე. ბოლო ათწლეულების განმავლობაში, **ქვესიმბოლურმა წარმოდგენებმა** პოპულარობა მოიპოვა - ისინი ცოდნის კოდირებას ახდენენ რიცხვითი ან ნეირონული ფორმით განაწილებული ნიმუშების სახით (მაგ., ვექტორული ჩანერგვები ან ნეირონული ქსელის წონები) დისკრეტული სიმბოლოების ნაცვლად (კოლელოფი და რეგლი 2024, 7). თითოეულ პარადიგმას აქვს თავისი ძლიერი მხარეები, სიმბოლური მიდგომები გვთავაზობს ინტერპრეტაციულობას და ეთიკურ დასკვნას, ხოლო ქვესიმბოლური მეთოდები წარმატებით სწავლობენ მონაცემებიდან და ასევე მართავენ „გაურკვევლობას“. თანამედროვე ხელოვნური ინტელექტის სისტემები სულ უფრო მეტად ცდილობენ **ჰიბრიდული წარმოდგენების** საშუალებების შემუშავებას, რომლებიც აერთიანებენ ორივე წარმოდგენის მეთოდებს და იყენებენ სიმბოლური სტრუქტურისა და ნეირონული სწავლების დამატებით შესაძლებლობებს (კოლელოფი and რეგლი 2024, 2).

2.1.1 სიმბოლური ცოდნის წარმოდგენის მიდგომები

სიმბოლური მიდგომები ცოდნას წარმოადგენს დისკრეტული სიმბოლოების (როგორცაა სიტყვები, ლოგიკური პრედიკატები ან გრაფის კვანძები) და ფორმალური სტრუქტურების მეშვეობით. ისინი ხაზს უსვამენ ექსპლიციტურ, დეკლარაციულ ცოდნას, რომლის პირდაპირ ინტერპრეტაცია და მანიპულირება შესაძლებელია ლოგიკაზე დაფუძნებული დასკვნის წესების მეშვეობით (კოლელოფი and რეგლი 2024, 2). ჩვენ განვიხილავთ რამდენიმე ფუნდამენტურ სიმბოლურ KR ფორმალიზმს.

ლოგიკაზე დაფუძნებული წარმოდგენები. ლოგიკური ფორმალიზმები გამოიყენებენ მკაცრად განსაზღვრულ სინტაქსსა და სემანტიკას და სამყაროს აღწერენ ფაქტებისა და წესების საშუალებით. ყველაზე მარტივი ფორმაა პროპოზიციული ლოგიკა, რომელიც ცოდნას წარმოადგენს გამონათქვამების (პროპოზიციების) სახით, რომლებსაც აქვთ ჭეშმარიტი ან მცდარი მნიშვნელობა. პროპოზიციული ლოგიკა გვთავაზობს ლოგიკურ კავშირებს (AND, OR, NOT, IMPLIES), რომლითაც შესაძლებელია ფაქტების კომბინაცია და ახალი ჭეშმარიტებების გამოყვანა ფორმალური დასკვნის გზით. თუმცა, მას არ გააჩნია შიდა სტრუქტურა – ყოველი პროპოზიცია წარმოადგენს ატომურ ერთეულს, რომელსაც არ გააჩნია ობიექტების ან ურთიერთობების გაგება, რაც ზღუდავს მისი ექსპრესიულობის შესაძლებლობებს. შედარებით, პრედიკატული ლოგიკა (First-Order Logic, FOL) ამატებს ობიექტებს, პრედიკატებს (ურთიერთობებს) და რაოდენობრივ მაჩვენებლებს (quantifiers) (“for all”, “there exists”), რაც მნიშვნელოვნად ზრდის ექსპრესიულობის ძალას. მაგალითად, FOL-ის ბაზა საშუალებას იძლევა ზოგადი წესების გამოსახატად, როგორცაა $\forall x: \text{Cat}(x) \rightarrow \text{Animal}(x)$ („ყველა კატა არის ცხოველი“) და კონკრეტული ფაქტების გამოსახატად, როგორცაა $\text{Cat}(\text{Mittens})$. FOL საშუალებას აძლევს აგენტს ახალი ფაქტების გამოსახატად (მაგალითად $\text{Animal}(\text{Mittens})$) ლოგიკური დასკვნის წესების (მაგალითად modus ponens) მეშვეობით. თუმცა, ეს ექსპრესიულობის ძალა გარკვეულ ფასს მოითხოვს, სრული FOL-ში მსჯელობა ნახევრადგადაწყვეტადია (ინფერენციამ შესაძლოა არ დაასრულოს მუშაობა). პრაქტიკაში, რათა მივაღწიოთ რაციონალურ გამოთვლებს, ხელოვნური ინტელექტის სისტემები ხშირად ზღუდავენ ლოგიკას გადაჭრად ნაწილებად ან ამატებენ ალბათობებს გაურკვეველობის მოსაგვარებლად.

ერთ-ერთი მნიშვნელოვანი ლოგიკის ოჯახია აღწერითი ლოგიკები (Description Logics – DL), რომლებიც ქმნიან საფუძველს ვებ ონტოლოგიის ენისთვის (OWL). OWL არის DL-ზე დაფუძნებული ონტოლოგიის ენა, რომელიც საშუალებას იძლევა მდიდარი და კომპლექსური ცოდნის გამოსახატად კატეგორიებსა და მათ ურთიერთობებზე, ხოლო ამავე დროს უზრუნველყოფს ავტომატურ დასკვნას ლოგიკური თანმიმდევრულობის შესამოწმებლად ან იმპლიციტური ფაქტების

გამოსავლენად (www.w3.org 2012). პრაქტიკაში, ლოგიკაზე დაფუძნებული ცოდნის წარმოდგენა უზრუნველყოფს მტკიცე საფუძველს ცოდნაზე დაფუძნებული სისტემებისთვის, ის უზრუნველყოფს სანდობას (მხოლოდ ჭეშმარიტი დასკვნები გამომდინარეობს ჭეშმარიტი წინაპირობებიდან) და ახსნადობას (შესაძლებელია დასკვნის ლოგიკური ჯაჭვის გადამოწმება). თანამედროვე გამოყენებებს შორისაა წესებზე დაფუძნებული ექსპერტული სისტემები, ლოგიკური პროგრამირება (მაგალითად Prolog, Answer Set Programming), და ონტოლოგიებზე დაფუძნებული დასკვნის სისტემები. ლოგიკა კვლავ წარმოადგენს ცენტრალურ მიმართულებას აკადემიურ ცოდნის წარმოდგენის კვლევებში, თუმცა მისი ინტეგრაცია სწავლებაზე დაფუძნებულ მეთოდებთან დღეს აქტიური განვითარების სფეროა (დელგრანდი, და სხვ. 2023, 8-9).

სემანტიკური ქსელები - სემანტიკური ქსელი წარმოადგენს ცოდნას კვანძებისა და კავშირების გრაფის სახით, სადაც კვანძები შესაბამისია ობიექტებისა ან ცნებების, ხოლო მარკირებული წიბოები (კავშირები) გამოხატავენ მათ შორის ურთიერთობებს. ეს სტრუქტურა არსებითად წარმოადგენს ურთიერთობების გრაფულ კოდირებას – მაგალითად, მარტივ სემანტიკურ ქსელს შეიძლება ჰქონდეს კვანძები „კატა“ და „ცხოველი“ და მათ შორის წიბო წარწერით *is-a*, რომელიც მიუთითებს, რომ კატა ცხოველის ქვეკლასია. ასევე შესაძლებელია სხვა კავშირები, როგორიცაა *hasPart* (აქვს ნაწილი), *causes* (იწვევს) და სხვა. სემანტიკური ქსელები ხელოვნური ინტელექტის ერთ-ერთი ყველაზე ადრეული ცოდნის წარმოდგენის ფორმალიზმებია (1960-იანი წლებიდან) და გამოიყენებოდა იერარქიული ტაქსონომიების მოდელირებისთვის (მაგალითად, მემკვიდრეობითი იერარქიები), ნაწილი-მთელის ურთიერთობებისა და ასოცირებული ცოდნის გამოსახატად. კლასიკური მაგალითია *WordNet*-ის ქსელური წარმოდგენა – ეს არის დიდი ლექსიკური ონტოლოგია, სადაც სიტყვები ერთმანეთთან დაკავშირებულია სემანტიკური ურთიერთობებით (სინონიმია, ჰიპერნიმია და სხვა). სემანტიკური ქსელები ვიზუალურად ინტუიციურია და უჭერენ მხარს მარტივ დასკვნის პროცესებს წიბოების მემკვიდრეობით გადაადგილებით (მაგალითად, თვისებების მემკვიდრეობით გადაცემა *is-a* იერარქიაში). თანამედროვე ცოდნის გრაფები (Knowledge Graphs – KG) პირდაპირ გამომდინარეობენ ამ იდეიდან –

ფაქტობრივად, ცოდნის გრაფი შეიძლება განისაზღვროს როგორც „სემანტიკური ქსელი სტრუქტურირებული მონაცემების წარმოდგენისთვის“, რომელიც შესაძლებელს ხდის დასკვნას, ინფორმაციის ამოღებასა და კონტექსტური ცოდნის აღმოჩენას (დეჰალი, შარმა და რაჯაბი 2025, 13-14). განსხვავებით ფორმალური ლოგიკისგან, სემანტიკურ ქსელებში ხშირად არსებობს „ჯანსაღი აზრით“ ასოცირების შესაძლებლობა – მაგალითად, გააქტიურების გავრცელების ტექნიკა (*spreading activation*) საშუალებას აძლევს სისტემას მოძებნოს დაკავშირებული ცნებები გრაფში კავშირების მიყოლებით. თუმცა, ტრადიციულ სემანტიკურ ქსელებს აკლდათ სტანდარტიზებული სემანტიკა; თანამედროვე ცოდნის გრაფები ხშირად ამატებენ ლოგიკურ ფენას (მაგალითად, ონტოლოგიის სქემა ან RDF/SPARQL სემანტიკა) ერთმნიშვნელოვანი ინტერპრეტაციის უზრუნველსაყოფად.

საბოლოოდ, სემანტიკური ქსელები ასახავენ ასოცირებულ და ტაქსონომიურ ცოდნას სტრუქტურირებულ გრაფულ ფორმაში. ისინი ადვილად გასაგებია ადამიანისთვის და ბუნებრივად მისადაგებელი არიან ვიზუალიზაციისა და გადაადგილებაზე დაფუძნებული დასკვნისთვის, თუმცა მათი თანმიმდევრულობისა და სისრულის უზრუნველყოფა მოითხოვს ზუსტ დიზაინს ან ლოგიკურ საფუძველს.

კადრები (Frames) წარმოადგენს სტრუქტურირებული მონაცემების შაბლონებს, რომლებიც გამოსახავენ ტიპურ (სტერეოტიპულ) სიტუაციებს ან ობიექტებს. კადრი შეიძლება ჩაითვალოს ცოდნის ორგანიზებულ ჩანაწერად რომელიმე ცნების შესახებ – მას აქვს დასახელებული „სლოტები“ (ატრიბუტები) და შესაბამისი მნიშვნელობები, რომლებიც შეიძლება იყოს ნაგულისხმევი მნიშვნელობები, შეზღუდვები ან კავშირები სხვა კადრებთან. კადრების ცნება მარვინ მინსკიმ (Marvin Minsky) შემოიტანა 1974 წელს, როგორც გზა, რომლითაც შესაძლებელია იმ ჯანსაღი აზრის ცოდნის მოდელირება, რომელსაც ადამიანი იყენებს სცენებისა და კონტექსტების გასაგებად. მაგალითად, შეიძლება არსებობდეს „რესტორნის“ კადრი, რომელიც მოიცავს სლოტებს როლებისთვის, როგორცაა მომხმარებელი, ოფიციანტი, მზარეული, და ასევე ტიპური მოვლენებისთვის ან მოლოდინებისთვის (საკვების შეკვეთა, ჭამა, გადახდა). თითოეულ სლოტს შეიძლება ჰქონდეს ნაგულისხმევი ვარაუდები (მაგალითად, მომხმარებელი არის ადამიანი, საკვები უნდა შეიკვეთოს ჭამამდე) და

შესაძლოა უკავშირდებოდეს ქვეკადრებს (მაგალითად, მენიუ შეიძლება იყოს ქვეკადრი, რომელიც აღწერს საკვების ვარიანტებს). შესრულების დროს, კადრი წარმოადგენს ჩარჩოს მიღებული ინფორმაციის ინტერპრეტაციისთვის. სისტემა ავსებს სლოტებს დაკვირვებული დეტალებით და ნაგულისხმევი მნიშვნელობებით და გამოიტანს იმ ინფორმაციას, რომელიც პირდაპირ არ არის მოცემული.

რასელისა და ნორვიგის აღწერის მიხედვით, კადრებზე დაფუძნებული სისტემები ორგანიზებულად განალაგებენ ფაქტებს ობიექტებისა და მოვლენების შესახებ ტაქსონომიურ სტრუქტურაში, რაც შესაძლებელს ხდის თვისებების მემკვიდრეობით გადაცემას იერარქიაში. ეს ნიშნავს, რომ კადრები ბუნებრივად ქმნიან ქსელებს (მსგავსად სემანტიკური ქსელებისა), სადაც ზოგადი კადრები (მაგალითად, „საჯარო კვება“) გადასცემენ საერთო სლოტებს სპეციალიზებულ კადრებს (მაგალითად, „სწრაფი კვება“ ან „ელიტური რესტორანი“). უბრალო სემანტიკურ ქსელებთან შედარებით, კადრები გვთავაზობენ მეტ მოქნილობასა და კონტექსტს, ისინი მხარს უჭერენ გამონაკლისებს და პროცედურულ დანართებს – ანუ კოდს, რომელიც სრულდება გარკვეული პირობების დაკმაყოფილებისას. მაგალითად, სლოტს შეიძლება ჰქონდეს „IF-NEEDED“ პროცედურა, რომელიც საჭიროებისას ადგენს მნიშვნელობას, ან „IF-ADDED“ ტრიგერი, რომელიც განაახლებს დაკავშირებულ ცოდნას. კადრების უნარი, დეკლარატიული და პროცედურული ცოდნის ინტეგრირებისა, მნიშვნელოვანი უპირატესობა იყო ექსპერტულ სისტემებსა და ადრეულ ენობრივ პროცესორებში (NLP). კადრები ფართოდ გამოიყენებოდა 1980-იან წლებში ცოდნაზე დაფუძნებულ სისტემებში ამოცანების გადასაჭრელად, როგორცაა ვიზუალური სცენების ინტერპრეტაცია, დაგეგმვა და ბუნებრივი ენის გააზრება. მიუხედავად იმისა, რომ თანამედროვე გარემოში წმინდა კადრული სისტემები ნაკლებად გვხვდება, მათი ძირითადი იდეა შემონახულია ობიექტზე ორიენტირებულ მოდელირებასა და ონტოლოგიებში (კადრული სისტემა შეიძლება ჩაითვალოს როგორც ონტოლოგია ნაგულისხმევი მნიშვნელობებითა და მეთოდებით). მართლაც, OWL-ში შექმნილი თანამედროვე ონტოლოგიები ხშირად იმეორებენ კადრების თვისებებს (კლასები ატრიბუტებით, ნაგულისხმევი მნიშვნელობებით, სქემური შეზღუდვებით და ა.შ.).

2021 წლის ერთ-ერთი კვლევის მიხედვით, ონტოლოგიებმა დიდი ხნით გადაუსწრეს კადრებს, როგორც ძირითადი ცოდნის წარმოდგენის ფორმატმა, და კადრზე დაფუძნებული მოდელები დღეს ხშირად გამოიყენება ჰიბრიდულ ბაზებში ან სპეციფიკურ ნიშურ სისტემებში (ნაზარუკსი და ოსისი 2021, 48). კადრების ნაკლებ გავრცელებას ნაწილობრივ განაპირობებს მათი „შინაგანი შეზღუდვები“ (მაგალითად, ურთიერთგამომრიცხავი ნაგულისხმევი მნიშვნელობების ან ფართო მოცულობის ცოდნის მართვის სირთულე) და სტანდარტიზაციის ნაკლებობა, თუმცა კადრები რჩება ისტორიულად მნიშვნელოვან და კონცეპტუალურად მდიდარ მიდგომად ცოდნის წარმოდგენაში (ნაზარუკსი და ოსისი 2021).

სკრიპტები კადრების სპეციალიზებული სახეობაა, რომლებიც ორიენტირებულია მოვლენათა თანმიმდევრობაზე. როჯერ შენკმა (Roger Schank) და რობერტ ებელსონმა (Robert Abelson) ისინი 1977 წელს შემოიტანეს ცოდნის წარმოდგენის სფეროში. სკრიპტი წარმოადგენს სტერეოტიპულ ქმედებათა თანმიმდევრობის მოდელს ცნობილ კონტექსტში – ეს არსებითად ერთგვარი „მოთხრობის შაბლონია“, რომელსაც ხელოვნური ინტელექტი იყენებს მოვლენების გასაგებად და წინასწარ განსასაზღვრად. მაგალითად, კლასიკური „რესტორნის სკრიპტი“ შეიძლება მოიცავდეს შემდეგ თანმიმდევრულ სლოტებს ან „სცენებს“. რესტორანში შესვლა, ადგილის დაკავება, მენიუს წაკითხვა, საკვების შეკვეთა, ჭამა, ანგარიშის გადახდა და გასვლა. თითოეულ სცენას აქვს მოსალოდნელი მონაწილეები, ქმედებები და მიზეზ-შედეგობრივი კავშირები (მაგალითად, ჭამამდე საჭიროა შეკვეთა; ანგარიში იხდიან ჭამის შემდეგ). როდესაც AI-ს სისტემას გააჩნია „რესტორანში წასვლის“ სკრიპტი, მას შეუძლია შეავსოს ნარატივში გამოტოვებული დეტალები (მაგალითად, თუ ამბავი მხოლოდ აღნიშნავს, რომ პერსონაჟი რესტორნიდან გავიდა, სკრიპტის მიხედვით შეიძლება გამოიტანოს დასკვნა, რომ მან გადაიხადა ანგარიში) და განმარტოს მნიშვნელობები (მაგალითად, იცოდეს, რომ რესტორანში „ოფიციალის გამოძახება“ ნიშნავს მომსახურების მოთხოვნას და არა სატელეფონო ზარს). სკრიპტები შეიქმნა ბუნებრივი ენის გააზრების გასამარტივებლად. ტექსტის ან მოთხრობების დამუშავებისას სისტემა ადარებს წაკითხულ მოვლენებს ცნობილ სკრიპტს, რათა განსაზღვროს სიუჟეტის მნიშვნელობა.

შენკის Script Applier Mechanism (SAM) იყო ერთ-ერთი პირველი სისტემა, რომელიც იყენებდა სკრიპტებს გაზეთების და მოთხრობების გასაგებად სლოტების შევსების გზით (ვებერი, და სხვ. 2020, 7587).

სკრიპტები სისტემას ასევე აძლევენ მოლოდინების გენერირების საშუალებას. მაგალითად, თუ კონტექსტი ამბობს, რომ „ჯონი შევიდა რესტორანში და შეუკვთა საკვები“, სისტემამ გამოიტანოს პროგნოზი, რომ იგი სავარაუდოდ შეჭამს და გადაიხდის ანგარიშს. ეს ეხმარება ანომალიების გამოვლენაში (თუ რაიმე სკრიპტს არ შეესაბამება, ეს შესაძლებელია მნიშვნელოვანი იყოს) და გეგმების აგებაში.

მიუხედავად იმისა, რომ სკრიპტები დახვეწილად ასახავენ ჯანსაღ აზრზე დაფუძნებულ მოვლენით ცოდნას, ისინი იზიარებენ კადრების გამოწვევები, სკრიპტების ვრცელი ბიბლიოთეკების ხელით შექმნა შრომატევადია, ხოლო თავად სკრიპტები მოქნილობას კარგავენ არატიპურ ან მოულოდნელ სიტუაციებთან გამკლავებისას. თანამედროვე კვლევებში სკრიპტის კონცეფცია გადაიზარდა „მოვლენათა სქემებში“ და ყოველდღიური აქტივობების ცოდნაში, რომლებიც ხშირად ისწავლება დიდი ტექსტური კორპუსებიდან. 2020 წლის შემდეგ NLP მკვლევრებმა გამოიყენეს მანქანური სწავლება სკრიპტის მსგავსი ცოდნის – ე.წ. პროცედურული ცოდნის – გამოსავლენად ნარატივებიდან (ნაზარუკსი და ოსისი 2021). მიუხედავად ამისა, ორიგინალური სკრიპტების ფორმალიზმი რჩება ცოდნის წარმოდგენის მნიშვნელოვან ეტაპად. მან აჩვენა, რომ დროებითი და მიზეზობრივი ცოდნა რუტინული ქმედებების შესახებ შესაძლებელია სტრუქტურირებულ ფორმაში გადმოიცეს, რაც შესაძლებლობას აძლევს მანქანებს განახორციელონ პრიმიტიული ჯანსაღ აზრზე დაფუძნებული მსჯელობა.

ონტოლოგია წარმოადგენს დომენის ინტერესის სფეროს გაზიარებული კონცეპტუალიზაციის ფორმალურ, აშკარა სპეციფიკაციას. პრაქტიკაში, ონტოლოგიები სტრუქტურირებული ცოდნის ბაზებია, რომლებიც განსაზღვრავენ იმ ობიექტების ტიპებს (კლასებს ან ცნებებს), რომლებიც არსებობენ კონკრეტულ დომენში, ამ ობიექტებს შორის ურთიერთობებს (თვისებები, კავშირები), და ხშირად – წესებსა თუ შეზღუდვებს, რომლებიც მართავენ ამ დომენს. ონტოლოგიები დახვეწილი სემანტიკური ქსელების მსგავსია, მაგრამ მათ გააჩნიათ მტკიცე ლოგიკური

საფუძველი. ისინი, როგორც წესი, ორგანიზებულნი არიან ტაქსონომიურად (კლასისა და ქვეკლასის იერარქიებად) და განსაზღვრავენ კლასების თვისებებს (მათ შორის – ურთიერთობების დომენსა და დიაპაზონს, რაოდენობრივ შეზღუდვებს და ა.შ.). მაგალითად, ბიომედიცინაში ონტოლოგია შეიძლება განსაზღვრავდეს კლასებს, როგორცაა დაავადება, სიმპტომი, მკურნალობა და კავშირებს, როგორცაა იწვევს, მკურნალობს, აქვს სიმპტომი. ასეთ ონტოლოგიაში შეიძლება იყოს განცხადება, რომ *იწვევს* არის კავშირი დაავადებასა და სიმპტომს შორის, და რომ თუ დაავადება A იწვევს სიმპტომ B-ს, ხოლო დაავადება A არის დაავადება C-ის ქვეკლასი, მაშინ C-ც იწვევს B-ს (კავშირების მემკვიდრეობა). ონტოლოგიების ფორმალური ბუნება საშუალებას იძლევა გამოყენებული იყოს დასკვნის გამკეთებელი ალგორითმები (reasoners). მაგალითად, აღწერით ლოგიკაზე (Description Logic) დაფუძნებული დასკვნის სისტემის მეშვეობით შეიძლება შემოწმდეს ონტოლოგიის თანმიმდევრულობა (არაა თუ არა მასში წინააღმდეგობრივი განსაზღვრებები), ან გამოვლინდეს იმპლიციტური ქვეკლასური ურთიერთობები ან ობიექტის კუთვნილება.

W3C-ის ვებ ონტოლოგიის ენა (OWL), როგორც სტანდარტიზებული ონტოლოგიის ენა, დაფუძნებულია აღწერით ლოგიკაზე და ფართოდ გამოიყენება სემანტიკურ ვებსა და საწარმოს ცოდნის მართვაში. OWL ონტოლოგიები წარმოადგენენ ლოგიკაზე დაფუძნებულ ცოდნის წარმოდგენას, რომლითაც შესაძლებელია კომპლექსური ცოდნის კოდირება ობიექტებზე, ჯგუფებზე და ურთიერთობებზე, რაც კომპიუტერებს აძლევს საშუალებას ავტომატურად შეამოწმონ თანმიმდევრულობა ან გახადონ ფარული ცოდნა ცხადი (www.w3.org 2012). მაგალითად, თუ ონტოლოგია შეიცავს ინფორმაციას, რომ „ყველა ძუძუმწოვარი არის ცხოველი“ და „ყველა ძაღლი არის ძუძუმწოვარი“, სისტემა გამოიტანს დასკვნას, რომ „ყველა ძაღლი არის ცხოველი“, მიუხედავად იმისა, რომ ეს ფაქტი პირდაპირ არ იყო მითითებული. ხელოვნური ინტელექტის კონტექსტში, ონტოლოგიები წარმოადგენენ სიმბოლურ მსჯელობის სისტემების და ცოდნის გრაფების სიმბოლურ ხერხემალს. თანამედროვე ცოდნის გრაფებში ხშირად შედის ონტოლოგიური სქემა, რომელიც განსაზღვრავს მონაცემთა სიტყვებს და სქემურ შეზღუდვებს. მონაცემებზე

დაფუძნებული AI-ს ზრდასთან ერთად, ონტოლოგიებს ახალი როლი მიენიჭა. ისინი უზრუნველყოფენ საბაზისო ცოდნას (წინასწარ ცოდნას), რაც ეხმარება მანქანური სწავლების მოდელებს მსჯელობასა და ახსნადობაში (მაგალითად, განსაზღვრავენ იმ თვისებებსა და ურთიერთობებს, რომლებიც მოდელმა უნდა გაითვალისწინოს). უახლესი კვლევები (2020 წლიდან მოყოლებული) ნეიროსიმბოლურ AI-ში იკვლევენ ონტოლოგიების და ნეირონული ქსელების ერთობლივ გამოყენებას – მაგალითად, ისეთი დანაკარგის ფუნქციების (loss functions) შექმნით, რომლებიც აფიქსირებენ ონტოლოგიებიდან მიღებულ ლოგიკურ შეზღუდვებს, ან ცნებების ვექტორიზაციით (embedding), რათა ნეირონულმა მოდელმა სიმბოლური დომენის ცოდნა გამოიყენოს.

საბოლოოდ, ონტოლოგიები წარმოადგენენ სიმბოლური ცოდნის წარმოდგენის უახლეს მიდგომარეობას კომპლექსური დომენებისთვის, ისინი აერთიანებენ ლოგიკის გამოხატვის სიღრმეს სემანტიკური ქსელებისა და კადრების სტრუქტურულ ინტუიციასთან და მხარდაჭერილია სტანდარტიზებული დასკვნისა (reasoners) და შეკითხვის (SPARQL RDF/OWL მონაცემებისთვის) ხელსაწყოებით. როგორც 2021 წლის ერთ-ერთ კვლევაშია აღნიშნული, ონტოლოგიები ეფექტურად ჩაენაცვლნენ კადრებზე დაფუძნებულ ადრეულ მეთოდებს, როგორც ფორმალური ცოდნის წარმოდგენის ძირითად ფორმატს. ისინი მნიშვნელოვან როლს თამაშობენ იმ ხელოვნური ინტელექტის სისტემების შექმნაში, რომლებსაც სჭირდებათ დომენის ცნებების ღრმა გააზრება და მტკიცე მსჯელობის შესაძლებლობები (ნაზარუკსი და ოსისი 2021).

2.1.2 ქვესიმბოლური ცოდნის წარმოდგენის მიდგომები

ქვესიმბოლური მიდგომები წარმოადგენენ ცოდნის დამახსოვრების ისეთ ფორმებს, რომლებიც არ იყენებენ დისკრეტულ, ადამიანისთვის გასაგებ სიმბოლოებს. ამის ნაცვლად ცოდნა, როგორც წესი კოდირდება რიცხვობრივ პარამეტრებად, განაწილებულ შაბლონებად ან სწავლით მიღებულ ასოციაციებად კომპიუტერულ მოდელში. ქვესიმბოლური სისტემების პროტოტიპული მაგალითია ხელოვნური ნეირონული ქსელი, რომელშიც ცოდნა ინახება ნეირონებს შორის კავშირებზე მინიჭებული წონების სახით. ეს წონები იცვლება სასწავლო ალგორითმების მეშვეობით, რათა მოერგოს სასწავლო მონაცემების სტატისტიკურ კანონზომიერებებს.

ქვესიმბოლოური წარმოდგენები ხშირად განაწილებულია, განსხვავებით სიმბოლოური ცოდნის ბაზისგან, სადაც თითოეულ ერთეულს (სიმბოლოს) გააჩნია ცალსახა მნიშვნელობა, ნეირონულ ქსელში ცნება წარმოდგენილია ბევრი ერთეულის გააქტიურების შაბლონით, და თითოეული ერთეული მონაწილეობს მრავალი სხვადასხვა ცნების წარმოდგენაში. ეს განაწილებული ბუნება ქვესიმბოლოურ ცოდნას ანიჭებს მდგრადობას – იგი ტოლერანტულია ხმაურისა და არასრული ინფორმაციის მიმართ – და აქვს განზოგადების უნარი მრავალგანზომილებიან სივრცეში ინტერპოლაციის გზით. ქვემოთ განვიხილავთ ქვესიმბოლოური წარმოდგენების ძირითად პარადიგმებს.

განაწილებული ნეირონული წარმოდგენები. ქვესიმბოლოურ, განაწილებულ წარმოდგენებში ცოდნა იმპლიციტურად არის შენახული მოდელის პარამეტრების კონფიგურაციაში და არა აშკარად განსაზღვრულ სახით. ჯეფრი ჰინტონმა (Geoffrey Hinton) ეს პრინციპი ჩამოაყალიბა 1980-იან წლებში: ცნება შეიძლება წარმოდგენილი იყოს თვისებების (გააქტიურებების) ვექტორით, რომელიც გადანაწილებულია მრავალ ნეირონზე, ნაცვლად ერთი - ერთზე შესაბამისი „სიმბოლოს კვანძისა“. მაგალითად, ნეირონულ ქსელს, რომელიც სწავლობს ცხოველების შესახებ, შესაძლოა არ გააჩნდეს კონკრეტული „სპილო“-ს კვანძი; ამის ნაცვლად, ის გამოიმუშავებს შაბლონს ბევრ დამალულ ერთეულზე, რომლებიც კოლექტიურად კოდირებენ სპილოს თვისებებს (დიდი ზომა, ნაცრისფერი, აქვს ხორთუმი და ა.შ.). თითოეული ამ თვისების ერთეული ცალკე შესაძლოა გაუგებარი იყოს, მაგრამ ერთად ისინი აშკარად განასხვავებენ სპილოს სხვა ცხოველებისგან. ეს მიდგომა ხშირად კონტრასტში მოდის *წარმოდგენებთან* სადაც ერთი ერთეული თითქმის უდრის ერთ ცნებას (where one unit \approx one concept). თანამედროვე ღრმა სწავლების მოდელების უმეტესობა ყველა ფენაზე იყენებს განაწილებულ წარმოდგენებს. მაგალითად, გამოსახულების ამოცნობისთვის გაწვრთნილი ღრმა ნეირონული ქსელი ყოველი შეყვანილი გამოსახულების კოდირებას ახდენს გარკვეული ფარული თვისებების სივრცითი ვექტორით; მსგავსი გამოსახულებები (მაგალითად, ორი სპილოს სურათი) იქნება წარმოდგენილი ერთმანეთთან ახლომდებარე ვექტორებით, რაც ასახავს მოდელის მიერ შეძენილ ცოდნას ვიზუალური მსგავსების შესახებ. სწავლა ასეთ სისტემებში გულისხმობს

პარამეტრების დამუშავებას ისე, რომ ქსელის შიდა წარმოდგენის სივრცე ასახავდეს მონაცემთა დომენის სტრუქტურას. განაწილებული ქვესიმბოლური ცოდნის წარმოდგენის მახასიათებელია უნარი, დააფიქსიროს დელიკატური სტატისტიკური კორელაციები და შეასრულოს ინტერპოლაცია იმგვარად, რაც სიმბოლური სისტემებისთვის რთულად მიღწევადია. თუმცა, ასეთი ცოდნა ლატენტურია და პირდაპირ არ არის ინტერპრეტირებადი – შეუძლებელია „ამოიკითხო“ ლოგიკური ფაქტი ნეირონული ქსელის წონებიდან. ეს გაუმჭვირვალობა წარმოშობს ე.წ. ცოდნის ამოღების პრობლემას. როგორ გადავიტანოთ ქვესიმბოლური ცოდნა ადამიანისთვის გასაგებ ფორმატში – რაც წარმოადგენს აქტიურ კვლევის სფეროს ახსნად AI-ში (*Explainable AI*). მიუხედავად ამისა, განაწილებული წარმოდგენები ქმნიან იმ უდიდესი წარმატების საფუძველს, რაც ღრმა სწავლებას აქვს ისეთ დარგებში, როგორცაა კომპიუტერული ხედვა, მეტყველების ამოცნობა და ბუნებრივი ენის დამუშავება. ისინი იძლევიან, ნედლი მონაცემებიდან მახასიათებლებისა და კონცეფციების შესწავლის საშუალებას, ადამიანის მიერ საწყისი ცოდნის კოდირების გარეშე.

კოგნიტური მეცნიერების ტერმინებით, ქვესიმბოლური მოდელები შესაბამისობაშია ე.წ. *სისტემა 1*-თან (ინტუიცია) – სწრაფი, ასოციაციური და მონაცემებზე დაფუძნებული დამუშავების მექანიზმთან. ქვესიმბოლური ცოდნის ადრეული სისტემების კლასიკური მაგალითებია პერსეპტრონი (Perceptron) და ჰოპფილდის ქსელი (Hopfield Network, 1980-იანი წლები), რომლებმაც აჩვენეს, რომ მარტივ ნეირონულ ქსელებს შეეძლოთ შაბლონების სწავლა და დამახსოვრება აშკარა სიმბოლოების გარეშე (კოლელოფი და რეგლი 2024, 15). თანამედროვე მასშტაბური ნეირონული ქსელები (მაგალითად, ტრანსფორმერები მილიარდობით პარამეტრით) არსებითად წარმოადგენენ უკიდურესად მდიდარ განაწილებული წარმოდგენის სისტემებს. ისინი ლატენტურად ინახავენ სამყაროს შესახებ უზარმაზარ მოცულობის ცოდნას თავიანთ წონებში – რასაც ადასტურებენ დიდი ენობრივი მოდელები, რომლებსაც შეუძლიათ უპასუხონ ტრივიალურ შეკითხვებს ან განახორციელონ მსჯელობა იმ ცოდნაზე დაყრდნობით, რაც მათ სწავლის პროცესში მიიღეს (ალხამისი, et al. 2022, 3). ერთ-ერთი მნიშვნელოვანი გამოწვევა ასეთ სისტემებში ის არის, რომ

ისინი შეიძლება წარმოქმნიდნენ არასწორ ან შეუთავსებელ შედეგებს – ვინაიდან არ აქვთ ჭეშმარიტებისა და ლოგიკის აშკარა კონცეპტუალიზაცია. სწორედ ეს ახსნის ჰიბრიდული მიდგომების საჭიროებას – სისტემაში დამატებითი შეზღუდვების ან გარე ცოდნის ბაზების ჩართვით, რათა მოხდეს მოდელის მიმართულების განსაზღვრა.

ნეირონული ჩაშენება და წარმოდგენის სწავლა

ნეირონული ემბედინგები წარმოადგენს დისკრეტული ერთეულების (როგორცაა სიტყვები, ცნებები ან ობიექტები) ვექტორულ წარმოდგენებს, რომლებიც ისწავლება ნეირონული ქსელების მიერ. იდეა იმისა, რომ ცოდნა წარმოდგენილ იქნას უწყვეტ ვექტორულ სივრცეში, თანამედროვე ხელოვნური ინტელექტის ცენტრალური კომპონენტია. მაგალითად, სიტყვების ემბედინგები (მაგ. *Word2Vec* ან *GloVe*) ყოველს სიტყვას ლექსიკონში განსაზღვრავენ როგორც მჭიდრო ვექტორს (მაგალითად, 300 განზომილება), ისე, რომ სემანტიკურად მსგავსი სიტყვების ვექტორები კოსინუსური სიახლოვით ახლოსაა განლაგებული. ასეთი ემბედინგები თავის თავში ატარებს რეალური სამყაროს ცოდნის გასაოცარ ოდენობას მათი ურთიერთმდებარეობის გეომეტრიაში — ცნობილია, რომ ვექტორული არითმეტიკით შესაძლებელია ანალოგიების გამოძახება (მაგალითად: $\text{embedding}(\text{"king"}) - \text{embedding}(\text{"man"}) + \text{embedding}(\text{"woman"}) \approx \text{embedding}(\text{"queen"})$). ამრიგად, ემბედინგების სივრცე წარმოადგენს ტექსტური კორპუსებიდან შეძენილ სემანტიკურ ურთიერთობებს. ნეირონული ენის მოდელები, როგორცაა BERT და GPT, ქმნიან კონტექსტუალურ ემბედინგებს სიტყვებისა თუ წინადადებებისთვის — ეს ნიშნავს, რომ სიტყვის წარმოდგენა იცვლება მის გარშემო არსებული კონტექსტის შესაბამისად და ასახავს სემანტიკურ ნიუანსებს (მაგალითად, სიტყვა *“bank”* მიიღებს განსხვავებულ ემბედინგს ფინანსურ კონტექსტში და მდინარის კონტექსტში). ტექსტის ფარგლებს მიღმა, ემბედინგები გამოიყენება გამოსახულებებისთვის (მაგალითად, კონვოლუციური ნეირონული ქსელის ბოლო ფენის გააქტიურებებით), მუსიკისთვის, ცოდნის გრაფების ობიექტებისა და სხვა მონაცემებისთვის. წარმოდგენის სწავლა (*Representation Learning*) არის ფართო პარადიგმა, რომელიც გულისხმობს მონაცემებიდან ავტომატურად ფუნქციური წარმოდგენების სწავლას — ხოლო ემბედინგები ამის ნათელი მაგალითია. ცოდნის წარმოდგენის თვალსაზრისით,

ემბედიინგი შეიძლება ჩაითვალოს როგორც წერტილი ლატენტურ კონცეპტუალურ სივრცეში, რომელსაც სისტემა იყენებს სიმბოლოს ნაცვლად. ემბედიინგების ძალა მდგომარეობს იმაში, რომ ისინი აშკარად წარმოაჩენენ მსგავსებებს: ორი ფაქტი ან ობიექტი, რომელიც ერთმანეთთანაა დაკავშირებული, მიიღებს მსგავსი ვექტორებით წარმოდგენას — რაც შესაძლებელს ხდის განზოგადებას (სწორედ ამიტომ შეუძლია ღრმა NLP-ს პარაფრაზების ან არასრული შეყვანის დამუშავება). მნიშვნელოვანი ქვემიმართულებაა ცოდნის გრაფების ემბედიინგები (*Knowledge Graph Embeddings, KGE*), რომელთა მიზანია სიმბოლური ცოდნის გრაფების კვანძებისა (ობიექტების) და კავშირების (ურთიერთობების) უწყვეტ ვექტორულ სივრცეში წარმოდგენა. ბოლო ათწლეულში შემუშავდა მრავალი მოდელი, როგორცაა TransE, DistMult, ComplEx, RotatE და სხვები. ეს მოდელები სწავლობენ ემბედიინგებს ისე, რომ დაახლოებით შეინარჩუნონ გრაფის სტრუქტურული შეზღუდვები: მაგალითად, TransE (2013) განიხილავს ურთიერთობას როგორც გადატანის ვექტორს, ისე რომ: $\text{embedding}(\text{subject}) + \text{embedding}(\text{relation}) \approx \text{embedding}(\text{object})$ თითოეული ტრიპლეტისთვის (*subject, relation, object*) (ვანგი, ქიუ and ვანგი 2021, 5).

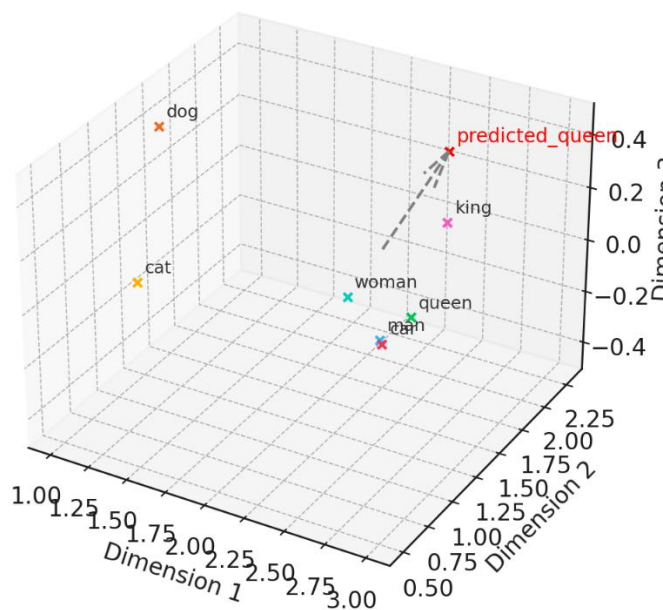
სხვა ოჯახი - სემანტიკური შესაბამისობის მოდელები (მაგ. DistMult, ComplEx) - იყენებს მსგავსების საზომებს (მაგალითად, სკალარული ნამრავლი ან კომპლექსური ვექტორული ურთიერთქმედებები) ტრიპლეტების სარწმუნოების განსასაზღვრად. უახლესი მიმოხილვები KGE მიდგომებს ორ დიდ კატეგორიად ყოფენ: დისტანციაზე დაფუძნებულებად და სემანტიკური შესაბამისობის მოდელებად. მიუხედავად კონკრეტული ტექნიკისა, KGE სისტემები აწარმოებენ დაბალი განზომილებების მჭიდრო ვექტორებს ყოველი ობიექტისა და ურთიერთობისთვის, რაც ფაქტობრივად სიმბოლური გრაფის შინაარსს გარდაქმნის ფორმაში, რომელიც შესაფერისია ნეირონული დამუშავებისთვის. მათ უპირატესობა ის არის, რომ ერთხელ, როცა ცოდნა წარმოდგენილია ვექტორულ სივრცეში, შესაძლებელია მასზე გამოყენებულ იქნას მანქანური სწავლების სრული აპარატი — კლასტერიზაცია, ნეირონული კლასიფიკაცია და სხვა. KGE მოდელებმა მნიშვნელოვნად გააუმჯობესეს შედეგები ამოცანებში, როგორცაა კავშირის პროგნოზირება (*link prediction* — ახალი ფაქტების ამოწევა ვექტორული არითმეტიკით) და ობიექტების კლასტერიზაცია. ისინი ასევე

ხელს უწყობენ ცოდნის გრაფების ინტეგრაციას სხვა ML მოდელებთან (მაგალითად, ობიექტის ემბედიנגების გამოყენება რეკომენდაციის სისტემაში). თუმცა, მინუსია ის, რომ გრაფის მდიდარი ლოგიკური სტრუქტურა შეიძლება ვექტორებით მხოლოდ დაახლოებით იყოს წარმოდგენილი; რთულია იმის უზრუნველყოფა, რომ ემბედინგები სრულად ითვალისწინებენ ონტოლოგიურ შეზღუდვებს ან რთულ ლოგიკურ წესებს. ამ პრობლემის გადასაჭრელად, თანამედროვე მიდგომები ემბედინგებში დამატებით ინფორმაციის ინტეგრაციას ცდილობენ: მაგალითად, ობიექტების ტექსტური აღწერილობები (literal-informed embeddings), დროითი მონაცემები დინამიკური ცოდნის გრაფებისთვის ან ლოგიკური წესები (ვანგი, ქიუ and ვანგი 2021, 6-8).

გრაფული ნეირონული ქსელები (*Graph Neural Networks, GNNs*) წარმოადგენენ კიდევ ერთ მიდგომას ცოდნის გრაფების წარმოდგენის სწავლებაში: GNN-ებზე დაფუძნებულ მოდელებს (მაგ. GraphSAGE, R-GCN) შეუძლია ინფორმაციის გავრცელება გრაფიკზე და ასევე კვანძების ემბედინგის შესწავლა ისე, რომ აღბეჭდოს მრავალი კვანძის გარშემო არსებული მეზობელი კვანძების სტრუქტურა.

მთლიანობაში, ნეირონული ემბედინგები ქმნიან ხიდს სიმბოლურ და ქვესიმბოლურ წარმოდგენებს შორის: ისინი დისკრეტულ ცოდნას აძლევენ რიცხვით ფორმას. ეს მიდგომა ყველგან გვხვდება თანამედროვე AI-ში — ყოველთვის, როცა ცოდნის წყარო (ტექსტი, გრაფი და სხვ.) გადაეცემა ნეირონულ მოდელს, პირველი ნაბიჯია მისი ჩაშენება. მიღებული წარმოდგენები ძლიერია, მაგრამ იმპლიციტური. მაგალითად, შეიძლება არ ვიცოდეთ, ემბედინგის რომელი განზომილება შეესაბამება მახასიათებელს „*არის ცხოველი*“, მაგრამ შეგვიძლია გავიგოთ მსგავსი ობიექტები მათი სიახლოვით: *კატის ემბედინგი უფრო ახლოსაა ძაღლთან, ვიდრე მანქანასთან*. 2021 წლის ერთ-ერთ მიმოხილვაში აღნიშნულია, რომ ცოდნის გრაფების ემბედინგებმა მიაღწიეს „ზუსტ, ეფექტიან და სტრუქტურულ წარმოდგენას“ მრავალ დარგში (ვანგი, ქიუ and ვანგი 2021, 12), თუმცა მაინც რჩება გამოწვევა - როგორ უზრუნველყოფდეს ემბედინგები საწყისი სიმბოლური ცოდნის მდიდარი სემანტიკის სრულყოფილ გადმოცემას.

3D Visualization of Word Embeddings



სურ 2.1. სიტყვების ემბედიგების 3D ვიზუალიზაცია (წყარო: ავტორი).

თითოეული სიტყვა წარმოდგენილია სამგანზომილებიან ვექტორულ სივრცეში, ხოლო წითელი წერტილი (*predicted_queen*) წარმოადგენს ვარაუდით მიღებულ პოზიციას, რომელიც მიღებულია შემდეგი ვექტორული ოპერაციით:

$$\mathbf{king} - \mathbf{man} + \mathbf{woman} \approx \mathbf{queen}$$

ეს სივრცული გამოსახულება აჩვენებს, როგორ შეიძლება სემანტიკური ურთიერთობები ნეირონულ მოდელებში კოდირდეს განზომილებებს შორის.

ვექტორულ-სიმბოლური არქიტექტურები (VSA) — რომლებიც ასევე ცნობილია როგორც ჰიპერგანზომილებიანი გამოთვლა (*hyperdimensional computing*) — წარმოადგენენ ქვესიმბოლური წარმოდგენის კლასს, რომლის მიზანია სიმბოლური სტრუქტურების ჩასმა მაღალგანზომილებიან ვექტორებში ისე, რომ შენარჩუნდეს მათემატიკური ოპერაციები, რომლებიც შეესაბამება სიმბოლურ მანიპულაციებს. SA-ში ყოველი ატომური სიმბოლო წარმოდგენილია მაღალგანზომილებიანი (მაგალითად, 1000 განზომილება) შემთხვევითი ვექტორით, რომლის კომპონენტები, როგორც წესი, ბინარულია ან რეალური რიცხვებია. რთული სტრუქტურები (როგორცაა თანმიმდევრული წყვილები, სიები ან თუნდაც გრაფები) კოდირდება ამ ვექტორების კომბინირებით, განსაზღვრული ოპერაციების მეშვეობით, როგორცაა:

- შეკვრა (*binding*) – ხშირად ელემენტების ნამრავლი ან წრიული კონვოლუცია ვექტორებს შორის;
- შეერთება (*bundling*) – ელემენტების შეკრება ან საშუალო არითმეტიკული.

ამ ოპერაციებით მიღებული ახალი ვექტორები კვლავ იმავე მაღალგანზომილებიან სივრცეში მდებარეობს, რაც შესაძლებელს ხდის რეკურსიულ კომპოზიციას. გამორჩეული თვისება ისაა, რომ შედეგი ვექტორები სიმბოლოების მსგავსად იქცევიან: მაგალითად, შესაძლებელია ავიღოთ ურთიერთობის ვექტორი $\$R\$$ და ობიექტის ვექტორი $\$A\$$ და მათი შეკვრით მივიღოთ ისეთი კომპოზიციური ვექტორი, რომელიც წარმოადგენს „ $\$R(A, _)\$$ “-ს. მოგვიანებით კი შესაძლებელია ამ კავშირის გაშლა (*unbind*) $\$R\$$ -ის ინვერსით და $\$A\$$ -ზე დაკავშირებული ინფორმაციის ამოღება.

VSA მოდელები სხვადასხვა ავტორებმა განავითარეს, მაგალითად:

- ჰინტონის შემცირებული წარმოდგენა,
- პლეიტის ჰოლოგრაფიული შემცირებული წარმოდგენები (HRR, 1995),
- კანერვას ჰიპერგანზომილებიანი გამოთვლა (2009),
- გეილერის MAP კოდირება და სხვ.

მიუხედავად მათ შორის არსებული განსხვავებებისა, ყველა ეს მოდელი იყენებს დიდ შემთხვევით ვექტორებს და ოპერაციებს, რომლებიც ქმნიან ფსევდო-ორთოგონალურ კოდირებებს იმავე ვექტორულ სივრცეში. მაღალი განზომილება კრიტიკულად მნიშვნელოვანია: დიდი განზომილებების (*curse of dimensionality*) გამო, შემთხვევითი ვექტორები 1000 განზომილებაში მაღალი ალბათობით თითქმის ორთოგონალურია, რაც ქმნის ერთგვარ „სუფთა სივრცეს“ ინფორმაციის საკოდირებლად. როდესაც ვექტორები ერთმანეთს ებმება ან ემატება, მიღებული შედეგი, როგორც წესი, მნიშვნელოვნად განსხვავდება ნებისმიერ შეყვანილ ვექტორისგან - თუ სპეციალურად არ არის ისე კონსტრუქციული — რაც შესაძლებელს ხდის მრავალ მონაცემზე დაყრდნობილი ინფორმაციის ერთობლივ (*superposed*) ფორმაში შენახვას, დაბნევის გარეშე. ეს მიდგომა ცნობილია როგორც გამოთვლა სუპერპოზიციის (*computing in superposition*) (დენის კლეიკო 2023, 9).

ინტუიციური მაგალითი: ვთქვათ, გვაქვს ვექტორები ცნებებისთვის John, Likes, Mary. შეგვიძლია პროპოზიციის Likes(John, Mary) კოდირება შემდეგი ფორმულით:

$V = \text{Likes} \circ (\text{John} + \text{Mary})$, სადაც \circ აღნიშნავს *შეკავშირების* ოპერაციას (binding), ხოლო $+$ აღნიშნავს *შერთებას* (bundling). (ეს გამარტივებული წარმოდგენაა.) ამ შემთხვევაში, ვექტორი V ინახავს ინფორმაციას, რომ ჯონი და მერი ერთმანეთთან დაკავშირებულნი არიან ურთიერთობით Likes. შესაძლებელია ინფორმაციის ამოღება, მაგალითად Mary-ს მიღება, ვექტორიდან V , *John* და *Likes* ვექტორების გაშლის გზით. პრაქტიკაში, VSA ოპერაციები იძლევა შესაძლებლობას, რომ შენახული იქნას წყვილების ერთობლიობა, თანმიმდევრობები (მაგალითად, პროპოზიციის ვექტორებთან თანმიმდევრული კავშირის გზით), და მათი ამოღება ხდებოდა მიახლოებითი შაბლონების დამთხვევით. მაღალგანზომილებიანი ვექტორების ბუნებრივი ცვლილებებისადმი მდგრადობა ნიშნავს, რომ თუნდაც კომპონენტების ნაწილი დაზიანდეს ან შეცდომით იყოს წარმოდგენილი, მსგავსების საზომებს მაინც შეუძლიათ სწორი შაბლონის აღმოჩენა. უახლესმა კვლევებმა განსაკუთრებით გამოკვეთეს VSA-ს მოქნილობა, როგორც ალტერნატიული გამოთვლის პარადიგმა. 2022 წლის მიმოხილვაში (*Proceedings of the IEEE*) აღნიშნულია, რომ VSA წარმოადგენს "ველური ტიპის ალგებრულ სტრუქტურას", რომელიც აწარმოებს მარტივ, მაგრამ ძლიერ ოპერაციებს მაღალგანზომილებიან ვექტორებზე და მხარს უჭერს ყველა იმ მონაცემურ სტრუქტურასა და მანიპულაციას, რომლებიც თანამედროვე გამოთვლისთვის არის აქტუალური. მნიშვნელოვნად აღინიშნება, რომ ვექტორულ-სიმბოლური არქიტექტურები (VSA) შესაძლებელს ხდიან ეფექტიან გადაწყვეტილებებს გარკვეული კომბინატორული ძიების ამოცანებისთვის - რაც მიიღწევა სუპერპროპოზიციისა და განაწილებული წარმოდგენის გამოყენებით. გარდა ამისა, VSA შეიძლება ძალიან ეფექტიანად განხორციელდეს ახალ ტექნოლოგიურ პლატფორმებზე — მათ შორის *ნეურომორფულ ჩიპებზე* (neuromorphic chips), რადგან მათი ძირითადი ოპერაციები ვექტორიზებულია და ხშირად კარგად ერგება სტოქასტიკურ (შემთხვევითობაზე დაფუძნებულ) გამოთვლით გარემოებებს (დენის კლეიკო 2023, 24).

ცოდნის წარმოდგენის კონტექსტში, ვექტორულ-სიმბოლური არქიტექტურები (VSA) განსაკუთრებით საინტერესოა, რადგან შლის ზღვარს სიმბოლურსა და ქვესიმბოლურს შორის: შესაძლებელია ისეთი ოპერაციების შესრულება, რომლებიც სიმბოლურ მანიპულაციებს ჰგავს (მაგალითად, როლებისა და შევსებების კომბინაცია ან ქვეშაბლონის არსებობის შემოწმება) — და ეს ყველაფერი ვექტორული არითმეტიკით. მაგალითად, VSA მეხსიერებაში მოთხოვნის (query) განხორციელება შეიძლება მოხდეს მოთხოვნის ვექტორის აგებით და ყველაზე ახლოს მდებარე შენახული ვექტორის მოძებნით — რაც ასოცირების საფუძველზე გახსენების (*associative recall*) მსგავსია. ეს მიდგომა გამოყენებული იქნა სხვადასხვა ამოცანის მოდელირებისთვის — როგორცაა ანალოგიების ამოცნობა, ცვლადების ბმა ლოგიკურ ფორმულებში და სხვა — განაწილებული წარმოდგენის პრინციპზე დაყრდნობით. მიუხედავად იმისა, რომ VSA ძირითადად კვლავ კვლევის თემად რჩება, მას უკვე მოეპოვება გამოყენებები კოგნიტური მოდელირებისა და ხელოვნური ინტელექტის სისტემებში: ბოლო წლების რამდენიმე ნამუშევარი კავშირს ავლენს ტრანსფორმერების ყურადღების მექანიზმებსა და VSA-ს მსგავს გამოთვლით პროცესებს შორის. (იუნგ, et al. 2024, 25)

საბოლოოდ, ვექტორულ-სიმბოლური არქიტექტურები (VSA) წარმოგვიდგენენ ცოდნის ჰიპერგანზომილებიან, განაწილებულ ფორმაში წარმოდგენის საშუალებას, რომელიც ამავე დროს შინარჩუნებს სიმბოლური წარმოდგენების ალგებრულ სტრუქტურას. VSA ადასტურებს, რომ შესაძლებელია ისეთი ოპერაციების განხორციელება, როგორცაა პერმუტაციები, შეკვრა და კომპოზიციური სტრუქტურების აგება, დისკრეტული სიმბოლოების გამოყენების გარეშე - რაც ხსნის ახალ გზას კოგნიტური მსჯელობისთვის, რომელიც იქნება მდგრადი და ნეირონულად რეალისტური. რადგან გამოთვლითი ტექნოლოგიები და AI მოდელები განაგრძობენ განვითარებას, VSA სულ უფრო მეტ ყურადღებას იპყრობს, როგორც მექანიზმი, რომელიც აერთიანებს ქვესიმბოლური წარმოდგენის მდგრადობასა და სიმბოლური წარმოდგენის კომპოზიციურობას (დენის კლეიკო 2023, 24).

2.1.3 ჰიბრიდული და ნეიროსიმბოლური მიდგომები

ცოდნის წარმოდგენის ჰიბრიდული მიდგომები მიზნად ისახავს სიმბოლური და ქვესიმბოლური მეთოდების ინტეგრაციას ერთიან სისტემაში. ასეთი შერწყმის მთავარი იდეაა მათი ძლიერი მხარეების კომბინირება:

- სიმბოლური წარმოდგენები უზრუნველყოფენ ინტერპრეტირებადობას, ადვილად იძლევიან წინასწარი ცოდნის ინტეგრაციას და ზუსტ ლოგიკურ მსჯელობას;
- ქვესიმბოლური წარმოდგენები კი იძლევიან საშუალებას ისწავლონ ნედლი მონაცემებიდან, გამოიჩინონ მდგრადობა ცვლილებების მიმართ და განახორციელონ მასშტაბური შაბლონების ამოცნობა.

ბოლო წლებში ეს შერწყმული მიდგომა ხშირად მოიხსენიება როგორც ნეიროსიმბოლური AI (*Neuro-Symbolic AI*), რომელიც მიჩნეულია მაღალი დონის ხელოვნური ინტელექტის შესაძლებლობების ერთ-ერთ მნიშვნელოვან მიმართულებად — მაგალითად, ჯანსაღ აზრზე დაფუძნებული მსჯელობის — მისაღწევად (კოლელოფი and რეგლი 2024, 2-3).

გარსესი და ლამბი (Garcez and Lamb, 2023) განსაზღვრავენ ნეიროსიმბოლურ ხელოვნურ ინტელექტს (*Neuro-Symbolic AI*) როგორც: „შერწყმული AI ჩარჩო, რომელიც ცდილობს სიმბოლური AI-სა და ნეირონული ქსელების დარგების გაერთიანებას, რათა შეიქმნას უფრო ძლიერი ჰიბრიდული AI მოდელი, რომელიც ფლობს მსჯელობის უნარს.“ პრაქტიკაში, ნეიროსიმბოლური სისტემები წარმოდგენილია მრავალფეროვანი ფორმით, თუმცა შესაძლებელია გამოვყოთ რამდენიმე საერთო შაბლონი ან არქიტექტურული მიდგომა:

–სიმბოლური ცოდნის ინტეგრაცია ნეირონულ ქსელებში: ამ მიდგომაში სიმბოლური ცოდნა (ლოგიკური წესები, ცოდნის გრაფები, ონტოლოგიები და სხვ.) ინტეგრირდება ნეირონულ მოდელებში იმისათვის, რომ მართოს სწავლა ან დასკვნის პროცესი. მაგალითად, შესაძლებელია ლოგიკური შეზღუდვების დამტკიცება რეგულარიზატორების ან დიფერენცირებადი დანაკარგის ფუნქციების სახით, რათა ნეირონული მოდელის გამოსავლები შეესაბამებოდეს ცნობილ წესებს. კონკრეტული შემთხვევაა ნეირონული კლასიფიკატორი, რომელსაც მართავს ონტოლოგია: თუ ონტოლოგია ამტკიცებს, რომ „ყველა კატა არის ცხოველი“,

მაშინ მოდელი დაჯარიმდება იმ შემთხვევაში, თუ ოდესმე კლასიფიცირებას გაუკეთებს კატას როგორც არაცხოველს - ამით მოდელის ქცევა თანხვედრაში მოდის სიმბოლურ მოლოდინებთან. სხვა ტექნიკაა ცოდნის გრაფის ემბედინგების (სიმბოლური KG-დან) გამოყენება დამატებითი მახასიათებლების სახით ან ნეირონული ქსელის ზოგიერთი ნაწილის ინიციალიზაცია ფაქტობრივი ცოდნით. ეს მიდგომა ფართოდ გამოიყენება რეკომენდაციის სისტემებში და კითხვაზე პასუხის სისტემებში, სადაც ნეირონული ქსელის პროგნოზები ეფუძნება ცოდნის გრაფიდან ამოღებულ ფაქტებს. უახლესი მუშაობა ცოდნით გამდიდრებულ ენობრივ მოდელებზე სწორედ ამ კატეგორიას მიეკუთვნება — მაგალითად, ConceptNet-ის (ჯანსაღი აზრის გრაფის) ინტეგრაცია ტრანსფორმერის არქიტექტურაში ან მის წინასწარ სწავლებაში, რათა მოდელმა ისწავლოს ცხადი რელაციური ცოდნის ჩაშენება.

– ნეირონული ქსელების გამოყენება სიმბოლურ ამოცანებზე (და პირიქით): ამ მიდგომაში, ნეირონული მეთოდები გამოიყენება იმ ამოცანების გადასაჭრელად, რომლებიც ტრადიციულად სიმბოლურად მიიჩნეოდა, ან პირიქით — სიმბოლური კომპონენტები გამოიყენება ნეირონული ამოცანების გასაუმჯობესებლად. მაგალითად, ნეირონული თეორემის დამტკიცება (*neural theorem proving*) — სადაც ნეირონული ქსელი სწავლობს, როგორ წარმართოს დასკვნის ძიების პროცესი ლოგიკურ სისტემაში იმით, რომ წინასწარ განსაზღვრავს, რომელი ფაქტები ან წესებია ცოდნის ბაზაში ამ კონკრეტული დასკვნისთვის ყველაზე მნიშვნელოვანი. ასეთ მიდგომებს განეკუთვნება DeepMind-ის *Differentiable Neural Logic Networks* და Rocktäschel & Riedel-ის *Neural Theorem Prover*. მეორე მხრივ, სიმბოლური ალგორითმები შეიძლება დაეხმარონ ნეირონულ ქსელებს — მაგალითად, სიმბოლური პლანერის (*symbolic planner*) გამოყენებით მაღალი დონის ქმედებების დაგეგმვისთვის, რომლებიც შემდეგ შესრულდება ნასწავლი ქვემოდულებით (*sub-modules*). ეს მიდგომა ხშირად გვხვდება რობოტიკის სფეროში. კვლევის კიდევ ერთი მიმართულება არის ნეირონული ქსელების გამოყენება სიმბოლური ცოდნის ამოსაღებად ნედლი მონაცემებიდან: მაგალითად:

- წესების ინდუქცია (ანუ ადამიანისთვის გასაგები წესების ამოღება ღრმა მოდელის ქცევიდან),

- ან ტექსტიდან რელაციური ფაქტების სწავლა ენობრივი მოდელების მეშვეობით.
- ეს სცენარები ქმნიან უკუკავშირის ციკლს, რომელშიც მიმდინარეობს ნეირონული და სიმბოლური წარმოდგენების ურთიერთგარდაქმნა (*neural <-> symbolic conversion*).

-გაერთიანებული მოდელები და დიფერენცირებადი სიმბოლური სტრუქტურები: ზოგიერთი ჰიბრიდული მიდგომა ცდილობს სიმბოლური და ქვესიმბოლური კომპონენტების მჭიდრო შერწყმას, ისე რომ სიმბოლური სტრუქტურები წარმოდგენილი იყოს დიფერენცირებად ფორმაში, რაც შესაძლებელს ხდის მათზე გრადიენტზე დაფუძნებული ოპტიმიზაციის გამოყენებას. ერთი მაგალითია ლოგიკური ტენზორული ქსელები (*Logic Tensor Networks, LTNs*), რომლებიც ლოგიკურ ფორმულებს ტენზორულ კალკულუსში აქცევენ ისე, რომ ჭეშმარიტების მნიშვნელობები დიფერენცირებადია და სისტემა შეიძლება ისწავლებოდეს მონაცემებზე დაყრდნობით ისე, რომ ფორმულები დაკმაყოფილდეს. სხვა მაგალითია ფარული ცვლადების (*latent variables*) გამოყენება ნეირონულ არქიტექტურებში, რომლებსაც გააჩნიათ სიმბოლური მნიშვნელობა — მაგალითად, ნეირონული მოდელი, რომელიც შეიცავს კომპონენტს, რომელიც აშკარად წარმოადგენს დისკრეტულ გადაწყვეტილებას (მაგ. წესის შერჩევა ან მარშრუტი ცოდნის გრაფში), კომბინაციაში უწყვეტ კომპონენტებთან. ასეთი სისტემების წვრთნა ხშირად მოითხოვს განმტკიცებითი სწავლის (*reinforcement learning*) გამოყენებას, განსაკუთრებით მაშინ, როდესაც დისკრეტულ კომპონენტებზე პირდაპირი გრადიენტი მიუწვდომელია. ერთ-ერთი მნიშვნელოვანი გამოყენების სფეროა ნეიროსიმბოლური მსჯელობა ცოდნის გრაფებზე, სადაც მოდელი სწავლობს, როგორ გადაადგილდეს სიმბოლურ ცოდნის გრაფში, ნეირონული მნიშვნელობების ფუნქციების ან პოლიტიკის ქსელების მეშვეობით — რაც ნიშნავს, რომ მოდელი სწავლობს გრაფზე ლოგიკური მსჯელობის უნარს (ამოცანებისთვის როგორცაა მრავალსაფეხურიანი კითხვებზე პასუხი ან ლოგიკური მოთხოვნების გადაწყვეტა). უახლესი კვლევები (2020–2024) ნეიროსიმბოლური მსჯელობის შესახებ ცოდნის გრაფებში (*Neural-Symbolic Graph Reasoning*) აჩვენებს, რომ გრაფულ ნეირონულ ქსელებთან სიმბოლური ლოგიკის შეზღუდვების (მაგალითად, იერარქიაში

ტრანზიტივობის დახურვა ან კავშირის ტიპის შემოწმება) შერწყმა აუმჯობესებს როგორც სისტემის შესრულებას, ისე მის ლოგიკურ თანმიმდევრულობას (დეჰალი, შარმა and რაჯაბი 2025, 28).

ჯანსაღი აზრი და კოგნიტური არქიტექტურები: ზოგიერთი ჰიბრიდული სისტემა შთაგონებულია ადამიანის აზროვნების კოგნიტური მოდელებით, სადაც შერწყმულია სწრაფი, ინტუიციური შეფასებები და ნელი, სიმბოლური განსჯა - ინსპირირებული დანიელ კაჰნემანის *სისტემა 1-ის* და *სისტემა 2-ის* ანალოგიით. ასეთ არქიტექტურებში, ქვესიმბოლური კომპონენტი (მაგალითად, ღრმა ნეირონული ქსელი ან ნასწავლი ჰურისტიკების ერთობლიობა) უზრუნველყოფს აღქმას და სთავაზობს შესაძლო გადაწყვეტილებებს, რომლებიც შემდეგ მოწმდება ან სრულდება სიმბოლური კომპონენტის მიერ (მაგ. ლოგიკაზე დაფუძნებული დასკვნის სისტემის ან პლანერის მიერ), რათა გადამოწმდეს მათი შესაბამისობა მაღალ დონის ცოდნასთან. მაგალითად, IBM-ის Neuro-Symbolic Concept Learner და MIT-ის Cogito მოდელი განასახიერებენ სისტემებს, რომლებშიც ვიზუალური ან ენობრივი გააზრება ხორციელდება ნეირონულად, მაგრამ მიღებული შედეგები ინტეგრირდება სიმბოლურ ცოდნის ბაზაში შემდგომი ლოგიკური მსჯელობისთვის. იდეა ერთი მაგალითია SAT ამომხსნელების (*SAT solvers*) ან შეზღუდვების ამომხსნელების (*constraint solvers*) გამოყენება სასწავლო ციკლში: ნეირონული ნაწილი სთავაზობს შესაძლო ამოცანების გადაჭრის ვარიანტებს, მაშინ როცა სიმბოლური ამომხსნელი უზრუნველყოფს გლობალურ თანმიმდევრულობას.

ცოდნის წარმოდგენის ჰიბრიდული (KR) მიდგომები ცდილობენ სიმბოლური და ქვესიმბოლური კომპონენტების შეთავსებადობის მიღწევას:

– ნეირონული ნაწილი უზრუნველყოფს სწავლას, განზოგადებას და ცვლილებების მიმართ მდგრადობას, მაშინ როცა სიმბოლური ნაწილი უზრუნველყოფს სტრუქტურას, ინტერპრეტირებადობას და წესებთან შესაბამისობას.

2024 წლის სისტემური მიმოხილვა ნეიროსიმბოლური AI-ს შესახებ ხაზს უსვამს, რომ პროგრესი მიღწეულია რამდენიმე მიმართულებით, მათ შორის: ინტეგრირებული ცოდნის წარმოდგენებში (მაგალითად, საერთო ემბედინგ-სივრცეები სიმბოლოებისა და ქვესიმბოლოებისთვის), შერწყმულ სასწავლო და დასკვნის ალგორითმებში და

ახსნადობის გაუმჯობესებაში, ნეირონული გადაწყვეტილებების სიმბოლურ ცნებებზე დაყრდნობით გამართვით (*grounding*) (კოლელოვი and რეგლი 2024, 30).

აღსანიშნავია, რომ დიდი ცოდნის ბაზები, როგორცაა *ConceptNet* და *Cyc* (რომლებიც მთლიანად სიმბოლურია), ახალი ძალით განვითარდა, როდესაც ისინი ნეირონულ მოდელებთან ინტეგრირებულად გამოიყენეს — მაგალითად, მოდელს შეუძლია დაეყრდნოს ფაქტობრივ ცოდნას, რათა თავიდან აიცილოს ე.წ. "ჰალუცინაციები" (*hallucination*, ანუ არარსებული ინფორმაციის გენერირება). მეორე მხრივ, ნეირონული ქსელები ხელს უწყობენ სიმბოლური რესურსების შენარჩუნებას და გაფართოებას:

დიდი ენობრივი მოდელები შეიძლება გამოყენებულ იქნას ცოდნის გრაფების შესავსებად ან ახალი ლოგიკური წესების შეთავაზებისთვის მონაცემებზე დაყრდნობით. ეს ურთიერთქმედება ორმხრივია და ქმნის იმას, რასაც ბევრი მოიხსენიებს როგორც შერწყმული AI სისტემები (*Composite AI systems*) (დეჰალი, შარმა and რაჯაბი 2025, 28 - 32).

მიუხედავად პერსპექტიულობისა, ჰიბრიდულ მიდგომებს ასევე თან ახლავს ორივე მხარის გამოწვევები: როგორ უზრუნველყოს, რომ ნეირონულმა კომპონენტმა სიმბოლურ შეზღუდვებს რეალურად დაიცვას (და არ დაუშვას ნატიფი, შეუმჩნეველი დარღვევები); როგორ გაუმკლავდეს სიმბოლური კომპონენტის სიხისტესა და ცოდნის ხარვეზებს და როგორ მართოს ისეთი სისტემების სირთულე, რომლებიც აერთიანებენ სიმბოლურ და ქვესიმბოლურ ელემენტებს.

გარდა ამისა, არსებობს აქტიური დებატები: ზოგი მკვლევარი ამტკიცებს, რომ საკმარისად დიდი ნეირონული მოდელებმა შესაძლოა გააქრონ სიმბოლური სტრუქტურების საჭიროება - რადგან საკმარისი მონაცემების პირობებში, მოდელმა შესაძლოა წესები *იმპლიციტურად* ისწავლოს.

სხვა მკვლევრები კი ამ პოზიციას არ ეთანხმებიან და ამტკიცებენ, რომ გამოკვეთილი ცოდნის სტრუქტურების გარეშე, ხელოვნურ ინტელექტს არასდროს ექნება სანდო მსჯელობისა და გამჭვირვალობის უნარი - როგორც ერთ-ერთმა ნეიროსიმბოლურმა მხარდამჭერმა თქვა: „მთვარეზე ვერ ახვალ, თუ უბრალოდ უფრო და უფრო მაღალ ხეებზე აძვრები“.

უახლესი სამეცნიერო ლიტერატურის კონსენსუსი მიდრეკილია სინერჯისკენ: სიმბოლური და ქვესიმბოლური მეთოდების ინტეგრაცია განიხილება, როგორც გზა ერთმანეთის შეზღუდვების გადალახვისა და უფრო ზოგადი, სანდო AI-ის მისაღწევად. (კოლელოფი and რეგლი 2024, 35)

2.1.4 ცოდნის წარმოდგენა თანამედროვე ხელოვნური ინტელექტის სისტემებში

დიდმა ენობრივმა მოდელებმა, როგორცაა GPT-3, GPT-4, BERT და სხვები - მეცნიერების ყურადღება მიმართეს ცოდნის ახალი ტიპის წარმოდგენისკენ: ეს არის იმპლიციტური ცოდნა, რომელიც მოდელის პარამეტრებშია შენახული. ამ მოდელები სწავლობენ უზარმაზარ ტექსტურ კორპუსებზე და შედეგად შეუძლიათ კითხვებზე პასუხის გაცემა, წინადადებების დასრულება ან ამოცანების გადაწყვეტა - იმ ინფორმაციაზე დაყრდნობით, რომელიც მათ სწავლის პროცესში შეისწავლეს. კვლევებმა აჩვენა, რომ წინასწარ გაწვრთნილ დიდ ენობრივ მოდელებს “მნიშვნელოვნად დიდი ოდენობის ცოდნა აქვთ ფარულად დაშიფრული მათ პარამეტრებში” (ალხამისი, et al. 2022, 16). სხვა სიტყვებით რომ ვთქვათ, სწავლის პროცესში მოდელის წონები შთანთქავენ სტატისტიკურ ასოციაციებს, რომლებიც ასახავს რეალურ ფაქტებსა და ენის გამოყენების კანონზომიერებებს. მაგალითად, დიდი ენობრივი მოდელი (LLM) შეიძლება დაასრულოს ფრაზა: „The capital of France is _“ სიტყვით „Paris“, რადგან სასწავლო მონაცემებში ხშირად არის ასოცირებული „France“ და „capital“ პარიზთან. სიმბოლური ბაზისგან განსხვავებით, ეს ცოდნა არ ინახება როგორც ცხადად განსაზღვრული ტრიპლეტები ან ჩანაწერები - იგი გაბნეულია მილიარდობით წონის მნიშვნელობაში კომპლექსურ ნეირონულ ქსელში. მოდელი „აბრუნებს“ ამ ცოდნას იმით, რომ შეყვანილ ტექსტს ადარებს ნასწავლ წარმოდგენებს და ქმნის statistically ყველაზე სავარაუდო გაგრძელებას.

პარამეტრულ ცოდნას აქვს რამდენიმე აშკარა უპირატესობა. პირველ რიგში, მას არ სჭირდება ადამიანის მიერ ცალსახად გაწვრთნილი ზედამხედველობა ან ცოდნის ბაზის კურირება - მოდელი სწავლობს ნედლი ტექსტებიდან და ეფექტურად აწყობს თავის შინაგან წარმოდგენას ფაქტებსა და ენის წესებზე (ალხამისი, et al. 2022, 36). მეორე უპირატესობა ისაა, რომ ცოდნა კონტექსტუალურად მოქნილია: რადგან ის

ჩაშენებულია ნეირონულ ქსელში, რომელიც თანმიმდევრობის წინასწარმეტყველებას ახორციელებს, მოდელს შეუძლია ცოდნის სხვადასხვა ფრაგმენტების ერთმანეთთან კომბინაცია ისეთ ფორმებში, რომლებიც სწავლისას პირდაპირ არ უნახავს - რაც წარმოადგენს ზოგად აზროვნებას ან *ანალოგიით მსჯელობის* ფორმას. თუმცა, ამ მიდგომას სერიოზული ნაკლოვანებებიც აქვს:

- ცოდნა არაგამჭვირვალეა,
- მისი რედაქტირება რთულია,
- მოდელი მარტივად ცდება.

დიდ ენობრივ მოდელს შესაძლოა უზარმაზარი რაოდენობის ფაქტების ცოდნა ჰქონდეს, მაგრამ ის მაინც შეუძლია წარმოქმნას ე.წ. „ჰალუცინაციები“ - ანუ თავდაჯერებული განცხადებები, რომლებიც მცდარი ან უაზროა, რადგან მას არ გააჩნია მტკიცე მექანიზმი ჭეშმარიტების შესამოწმებლად ან მხოლოდ თანმიმდევრული ფაქტების გასახსენებლად. გარდა ამისა, მოდელის ცოდნის განახლება (მაგალითად, ახალი მოვლენის დამატება ან არასწორი ინფორმაციის კორექტირება) გართულებულია: თქვენ ვერ შეცვლით უბრალოდ ერთ ჩანაწერს, როგორც მონაცემთა ბაზაში — ამისთვის საჭიროა მოდელის ხელახალი წვრთნა, დაზუსტება (*fine-tuning*), ან სპეციალიზებული მოდელის რედაქტირების ტექნიკის გამოყენება.

2021–2023 წლებში მკვლევრები აქტიურად იკვლევდნენ, როგორ წარმოქმნიან და იწვევენ დიდენობრივი მოდელები (LLMs) ცოდნას. მაგალითად, LAMA benchmark (*L*anguage *M*odel *A*nalysis), რომელიც შემოთავაზებული იყო პეტრონისა და სხვების მიერ (Petroni et al., 2019), აჩვენა, რომ BERT-ს შეეძლო ფაქტობრივი ტრიპლეტების გახსენება გასაკვირი სიზუსტით, როცა ის ასრულებდა კლოზ-სტილის მოთხოვნებს (*cloze-style queries*) - ანუ წინადადებების შევსებას გამოტოვებული სიტყვით (ალხამისი, et al. 2022, 16).

შემდგომმა კვლევებმა და მიმოხილვებმა შეაფასეს, რომელი ტიპის ცოდნის სწავლაა ადვილი ან რთული ენობრივი მოდელებისთვის. მოდელები კარგად ართმევენ თავს პოპულარული და ხშირად ნახსენები ფაქტების დამახსოვრებას, მაგრამ ძნელად სწავლობენ ნაკლებად გავრცელებულ („long-tail“) ან კომპლექსურ

რელაციურ ცოდნას. შეინიშნება, რომ დიდ ენობრივ მოდელებში ნაწილობრივ იქმნება შიდა ცოდნის გრაფი ემბედიგ-სივრცეში, თუმცა ის არ არის მკაცრად სტრუქტურირებული - იგი უფრო ჰგავს პრობაბილისტურ ასოციაციურ ქსელს, ვიდრე ცხად ცოდნის ბაზას. LLM-ების ცოდნის გაუმჯობესების მცდელობებმა გამოიწვია ისეთი ტექნიკების განვითარება, როგორცაა: “ცოდნის ნეირონები” (*knowledge neurons*): აქ კონკრეტული ნეირონები ან ვექტორული მიმართულებები მოდელში იდენტიფიცირდება როგორც კონკრეტული ფაქტების მატარებლები. მაგალითად, შესაძლებელია აღმოჩნდეს ნეირონი, რომლის გამორთვისას მოდელი „ივიწყებს“, რომ საფრანგეთის დედაქალაქი პარიზია - რაც მიანიშნებს, რომ ეს ნეირონი მნიშვნელოვანი იყო იმ კონკრეტული ცოდნისთვის.

ასევე გაჩნდა მოდელის ცოდნის რედაქტირების მცდელობები: მიზნობრივი მონაცემებით მცირე მასშტაბის ხელახალი წვრთნის (*fine-tuning*) მეშვეობით, ან წონებზე რანგ-1 განახლებების (*rank-one updates*) გამოყენებით, რათა კონკრეტული მოთხოვნის (*query*) შედეგი შეიცვალოს - მაგალითად, შეცდომიანი ფაქტი გადაიქცეს ჭეშმარიტად - სრულად თავიდან გაწვრთნის გარეშე. ს მიმართულებები დღესაც აქტიური კვლევის სფეროს წარმოადგენს.

პარამეტრული ცოდნის შეზღუდვებიდან გამომდინარე, მნიშვნელოვანი ტენდენცია გახდა მოძიებაზე დაფუძნებული დიდი ენობრივი მოდელები (*retrieval-augmented LLMs*) — სისტემები, რომლებიც აერთიანებენ დიდ ენობრივ მოდელს გარე ცოდნის წყაროსთან. ამ მიდგომაში LLM გამოიყენება სიმბოლურ ცოდნის ბაზასთან ან დოკუმენტების კორპუსთან ერთად: როდესაც მოდელს მიეწოდება მოთხოვნა (*query*), სისტემა ჯერ პოულობს შესაბამის ფაქტებს ან ტექსტურ ნაწილებს (საძიებო სისტემების ან ვექტორული მსგავსების საფუძველზე ცოდნის გრაფიდან ან ტექსტურ ბაზიდან), და შემდეგ LLM ამ დამატებულ ინფორმაციას იყენებს თავის პასუხში.

ამ მიდგომის მიზანია ორი მიდგომის უპირატესობების შერწყმა: LLM-ის ენობრივი გამომეტყველება და მასშტაბურობა, გარე ცოდნის წყაროს სიზუსტე და ფაქტობრივ საფუძველზე დაფუძნებულობა (*grounding*). ერთ-ერთი ცნობილი ჩარჩოა RAG (*Retrieval-Augmented Generation*), სადაც მოდელს მიწოდებული აქვს მოძიებული

ტექსტური ფრაგმენტები დამატებითი კონტექსტის სახით, რის შედეგადაც ის უფრო დიდი ალბათობით აძლევს ფაქტობრივად სწორ პასუხს.

მაგალითად, LLM-ზე დაფუძნებული კითხვაზე პასუხის (QA) სისტემა შეიძლება ყოველ კითხვაზე პასუხის გაცემამდე: მოიძიოს რამდენიმე შესაბამისი აზრადი ვიკიპედიიდან, ან გააკეთოს მოთხოვნა ცოდნის გრაფში, რათა იპოვოს დაკავშირებული ობიექტები, და შემდეგ მოძიებს შეუძლია ამ მოძიებულ წყაროებზე დაყრდნობით შექმნას პასუხი. ეს მიდგომა მნიშვნელოვნად ამცირებს „ჰალუცინაციების“ რაოდენობას (ანუ მცდარი ან მოგონილი პასუხების გენერირებას) და საშუალებას აძლევს მოძიებს, მიაწოდოს წყაროს დასახელება თავის პასუხებთან ერთად - ანუ დაამატოს ფაქტების წყაროს წარმოშობა (*provenance*), რაც ამაღლებს პასუხების სანდოობას.

კიდევ ერთი ინტეგრაციის ფორმაა დიდი ენობრივი მოდელების (LLM) წინასწარი წვრთნა ან დაზუსტება ცოდნის გრაფის მონაცემებით. LLM-ებს ასევე შეუძლიათ იმოქმედონ როგორც ინტერფეისი ცოდნის გრაფებთან - მაგალითად: გადააქციონ ბუნებრივი ენის მოთხოვნა სტრუქტურირებულ მოთხოვნად (*SPARQL*), რომელსაც ცოდნის გრაფი შეასრულებს, ან განახორციელონ ნარატიული ტიპის მსჯელობა ცოდნის გრაფის მიერ მიწოდებულ ინფორმაციაზე დაყრდნობით.

შეჯამებით, დიდი ენობრივი მოდელები (LLM-ები) წარმოადგენენ ძალიან ძლიერ, მაგრამ იმპლიციტურ ცოდნის წარმოდგენის ფორმას. ისინი ადასტურებენ, რომ მასშტაბური, განაწილებული წარმოდგენები შეიძლება შეიცავდეს უზარმაზარ მოცულობას სამყაროს შესახებ ცოდნისა. თუმცა, ნდობისა და განახლებადობის პრობლემების გამო, ხელოვნური ინტელექტის თანამედროვე სისტემებში საუკეთესო პრაქტიკად მიიჩნევა LLM-ების კომბინირება ცხად (ექსპლიციტურ) ცოდნის წარმოდგენებთან. მიმართულება იმ არქიტექტურებისკენ მიდის, სადაც LLM-ს შეუძლია “მოეძიოს” ინფორმაცია ცოდნის გრაფიდან ან ბაზიდან - ისევე, როგორც ადამიანი ეძებს ფაქტებს წყაროებში - ხოლო ცოდნის გრაფი, თავის მხრივ, შეიძლება განვითარდეს ან გაფართოვდეს LLM-ის უნარების გამოყენებით (დეჰალი, შარმა and რაჯაბი 2025, 32).

2.2 ცოდნის გრაფული მოდელის აღწერა

ცოდნის გრაფები (Knowledge Graphs, KG) დღესდღეობით იქცნენ ცოდნის წარმოდგენის საფუძვლად, როგორც ინდუსტრიაში, ისე სამეცნიერო კვლევებში. ისინი უზრუნველყოფენ სტრუქტურირებულ და სიმბოლურ ცოდნის საცავს, რომელიც მოიცავს ფაქტობრივ და სემანტიკურ ინფორმაციას. ცოდნის გრაფი, როგორც წესი, შედგება: ობიექტების (კვანძების) ნაკრებისგან და ურთიერთობების (წიბოების) ნაკრებისგან, რაც ქმნის გრაფულ მონაცემთა მოდელს. გრაფს ასევე ხშირად ახლავს ონტოლოგიური სქემა (კლასები და თვისებები), რომელიც სწორიგებს ობიექტების ტიპებსა და ურთიერთობებს. ცოდნის გრაფის სიძლიერე იმაში მდგომარეობს, რომ ის ადამიანისთვის წაკითხვადია და შეუძლია დასაკავშირებლად გამოიყენოს განსხვავებული ტიპის ინფორმაცია ტიპიზებული კავშირების მეშვეობით - მაგალითად, პიროვნების დაკავშირება მის დაბადების ადგილთან, პროფესიასთან, ნაშრომებთან და ა.შ. თანამედროვე ცოდნის გრაფები წარმოდგენილია გენერალურ ენციკლოპედიური გრაფებიდან, როგორცაა Wikidata, DBpedia (Wikipedia-დან ამოღებული) და Freebase (Google-ის Knowledge Graph), და სპეციალიზებულ დომენურ გრაფებამდე, როგორცაა ბიომედიცინა, ფინანსები, ელექტრონული კომერცია და სხვა.

ცოდნის გრაფის სტრუქტურა, როგორც წესი, წარმოდგენილია დირექციული, მარკირებული გრაფის სახით, ხშირად განიხილება როგორც ტრიპლეტების კრებული (subject-predicate-object). მაგალითად: (*BarackObama* – *bornIn* – *Hawaii*) და (*Hawaii* – *isA* – *State*) არის ორი ტრიპლეტი ცოდნის გრაფში, რომლებიც ერთობლივად გულისხმობს მრავალსაფეხურიან ფაქტს. ცოდნის გრაფები ხშირად განასხვავებენ: ობიექტის კვანძებს და ცნებების კვანძებს რომლებიც დაკავშირებულია იერარქიით.

ეს სტრუქტურა იძლევა ონტოლოგიებთან ინტეგრაციის შესაძლებლობას, რაც თავის მხრივ შესაძლებელს ხდის თვისებების მემკვიდრეობით გადაცემას (inheritance) და კლასზე დაფუძნებული შეზღუდვების დაწესებას. ცოდნის გრაფის აგება შეიძლება განხორციელდეს სხვადასხვა გზით:

- ხელით, ექსპერტების მიერ (მაგ. ადრეული *Cyc* პროექტი);
- კრაუდსორსინგის გზით (*Wikidata* მხარდაჭერილია მომხმარებელთა მიერ);
- ავტომატური ინფორმაციის ამოღებით ტექსტებიდან და მონაცემთა ბაზებიდან.

თანამედროვე KG-ის აგებისას იყენებენ NLP ტექნიკებს, როგორცაა ობიექტების ამოცნობა (*entity recognition*) - ტექსტში სახელების ამოსაცნობად, ურთიერთობების ამოღება (*relation extraction*) - ობიექტებს შორის კავშირების დასადგენად და ნორმალიზაცია/დაკავშირება (*entity linking*) - ტექსტში ნახსენები სტრიქონების მიხედვით გრაფის სტანდარტულ კვანძებთან. უახლეს შემთხვევებში კი დიდი ენობრივი მოდელები (LLMs) აქტიურად გამოიყენება ცოდნის გრაფების აგების პროცესში, ისინი კითხულობენ არასტრუქტურირებულ ტექსტს და სთავაზობენ ახალ ტრიპლეტებს, ან ავსებენ უკვე ნაწილობრივ აგებულ გრაფებში არსებულ სიცარიელებებს (დეჰალი, შარმა and რაჯაბი 2025, 41).

მაგალითად, დიდ ენობრივ მოდელს (LLM) შეუძლია მიიღოს წინადადება, როგორცაა: „Paris is the capital of France“, და დააბრუნოს სტრუქტურირებული ტრიპლეტი: (Paris – capitalOf – France). დამატებით, აქტიური კვლევის სფეროა სქემის ინდუქცია (*schema induction*) - ანუ ონტოლოგიური სქემის ავტომატურად სწავლა მონაცემებიდან, სადაც მანქანური სწავლების ალგორითმები გვთავაზობენ, როგორ უნდა ორგანიზდეს ობიექტები იერარქიულად ან რა ტიპის ურთიერთობები არსებობს მათ შორის. ცოდნის გრაფის რეალურ დროში განახლება სერიოზულ გამოწვევას წარმოადგენს, რადგან ახალი ობიექტები და ფაქტები მუდმივად ჩნდება. ამ გამოწვევის საპასუხოდ მიმდინარეობს ტექნიკების შემუშავება, როგორცაა *zero-shot / few-shot updating*, სადაც LLM-ები ან სხვა პროგნოზირების მოდელები უშუალოდ ამატებენ ახალ კვანძებსა და წიბოებს გრაფში, რათა სისტემა მუდმივად განახლებადი იყოს. გრაფის აგების მეთოდის მიუხედავად, ცოდნის გრაფის ერთ-ერთი ძირითადი უპირატესობაა გამჭვირვალობა, ცოდნა წარმოდგენილია ცხადად და სტრუქტურირებულად, რის გამოც შესაძლებელია მოთხოვნების გაკეთება სტრუქტურირებულ ენაზე (მაგ. SPARQL-ით), ან AI სისტემების გადაწყვეტილებების ასახსნელად - მაგალითად „სისტემამ უპასუხა X, რადგან ცოდნის გრაფში არსებობს შემდეგი დაკავშირებული ფაქტები, რომლებიც ამყარებს X-ს“ (დეჰალი, შარმა and რაჯაბი 2025, 5).

ცოდნის გრაფების ემბედიგებზე (Knowledge Graph Embeddings, KGE) ვსაუბრობდით როგორც ქვესიმბოლურ წარმოდგენაზე. თანამედროვე ხელოვნური

ინტელექტის სისტემებში, ემბედიინგის ტექნიკები აუცილებელია ცოდნის გრაფების მასშტაბურად გამოსაყენებლად. როდესაც გვაქვს დიდი მოცულობის ცოდნის გრაფი, შესაძლებელია ასწავლო უწყვეტი ვექტორული წარმოდგენები თითოეული ობიექტისა და ურთიერთობისთვის ისე, რომ გრაფის სტრუქტურა შენარჩუნდეს ემბედიინგ-სივრცეში (ვანგი, ქიუ and ვანგი 2021, 19). ეს ემბედიინგები შესაძლებელს ხდის განზოგადებას - მაგალითად, ისინი შეიძლება წინასწარ განსაზღვრავდნენ ახალ კავშირს ორ ობიექტს შორის, თუ მათი ვექტორები „დასაბუთებულად ერგებიან“ რომელიმე ურთიერთობის ვექტორს. გარდა ამისა, ემბედიინგები შესაძლებელს ხდის ცოდნის გრაფის მონაცემების ინტეგრირებას ნეირონულ ქსელებთან, რადგან ნეირონული ქსელები მუშაობენ ვექტორულ წარმოდგენებზე. არსებობს ემბედიინგის მრავალი მოდელი, მათ შორის ტრანსლაციური მოდელები, როგორცაა TransE, RotatE (*RotatE იყენებს კომპლექსურ რიცხვებში როტაციას სიმეტრიის/ასიმეტრიის გამოსახატად*); ბილინეური მოდელები, როგორცაა DistMult და ComplEx; ღრმა სწავლებაზე დაფუძნებული მოდელები, როგორცაა ConvE (*რომელიც ემბედიინგებს გადააქვს კონვოლუციურ ნეირონულ ქსელზე მორგებულ ფორმატში*). ასევე არსებობს სპეციალიზებული ემბედიინგის მიდგომები, როგორცაა დროითი ცოდნის გრაფების ემბედიინგი (*temporal KG embeddings*) - დროით ასპექტებზე დაფუძნებული ურთიერთობებისთვის; მულტიმოდალური ემბედიინგები, რომლებიც ობიექტებთან დაკავშირებულ სურათებს ან ტექსტს აერთიანებს; ჰიპერბოლური ემბედიინგები, რომლებიც ობიექტებს მრუდოვან გეომეტრიაში ათავსებენ, რაც განსაკუთრებით კარგად ემსახურება იერარქიულ სტრუქტურებს (ვანგი, ქიუ and ვანგი 2021, 45-47).

ახალი და მზარდი იდეაა ვირტუალური ცოდნის გრაფები (**Virtual Knowledge Graphs, VKGs**) - სადაც ტრადიციული სიმბოლური გრაფის ნაცვლად, ფაქტები ემბედიინგდება და ინახება ვექტორულ ინდექსში. მოთხოვნაზე პასუხი მიიღება იმით, რომ მოთხოვნა (ან მისი კომპონენტები) გადადის ვექტორულ სივრცეში, და შემდეგ ხდება ყველაზე ახლოს მდებარე ვექტორების მოძიება - რაც პრაქტიკულად ნიშნავს მრავალსაფეხურიანი მსჯელობის განხორციელებას უწყვეტ სივრცეში (დეჰალი, შარმა and რაჯაბი 2025, 11).

მსჯელობის მექანიზმები (Reasoning Mechanisms): ცოდნის გრაფები (KGs) მხარს უჭერს მსჯელობის რამდენიმე ფორმას. ყველაზე პირდაპირი მეთოდია გრაფის ტრავერსირება ან ბილიკზე დაფუძნებული მსჯელობა: მაგალითად, იმის გასარკვევად, თუ როგორ უკავშირდებიან ერთმანეთს ორი ობიექტი, ალგორითმს შეუძლია მოძებნოს გრაფში ბილიკები (წიბოების თანმიმდევრობები), რომლებიც მათ აკავშირებს. ეს მეთოდი აქტიურად გამოიყენება მოთხოვნებზე პასუხისას (*query answering*) - მაგალითად: „მიპოვე ადამიანი, რომელიც დაიბადა ქალაქ X-ში და არის კომპანიის აღმასრულებელი დირექტორი ინდუსტრიაში Y“ - შეიძლება გადაწყდეს *bornIn* და *industry* წიბოების ტრავერსირებით.

მეორე ფორმაა ლოგიკური მსჯელობა, რომელიც ეფუძნება ონტოლოგიას/სქემას და განსაზღვრულ ლოგიკურ წესებს. მაგალითად: თუ ცოდნის გრაფის სქემაში ნათქვამია, რომ *partOf* ურთიერთობა ტრანზიტისა (ანუ თუ *A partOf B* და *B partOf C*, მაშინ *A partOf C*), მაშინ ლოგიკური დასკვნის მექანიზმს (*reasoner*) შეუძლია ახალი წიბოების განზოგადება, რომლებიც პირდაპირ მოცემული არ იყო. ონტოლოგიური მსჯელობის მექანიზმებს (*ontology reasoners*) შეუძლიათ: ობიექტების კლასიფიკაცია მათი ურთიერთობების საფუძველზე (მაგ. იმპლიციტური *isA* კავშირების მატერიალიზაცია), შეზღუდვების შემოწმება - მაგ., თუ ონტოლოგია განსაზღვრავს, რომ ერთ ადამიანს შეიძლება ჰყავდეს მაქსიმუმ 2 შვილი, მაგრამ კონკრეტულ კვანძს აქვს 3 *child* წიბო - ეს წარმოადგენს ინკონსისტენტობას ან რთულ ლოგიკურ მოთხოვნებზე პასუხი. ბევრ ცოდნის გრაფს (განსაკუთრებით სემანტიკური ვებ-დონეზე) ახლავს თან წესების ნაკრები ან მთლიანობის შეზღუდვები, მაგალითად OWL (Web Ontology Language) ან SPARQL-ის შიგნით განსაზღვრული წესები, რომლებიც ხელს უწყობენ ზემოხსენებულ ლოგიკურ მსჯელობას. დეტერმინისტული მსჯელობის მიღმა, შესაძლებელია სტატისტიკური მსჯელობაც, რაც ხდება ემბედიנגების მეშვეობით: მაგალითად, შეგვიძლია დავალაგოთ *tail*-ობიექტების კანდიდატები (ცნობილი *head* და ურთიერთობის მიხედვით), თუ რამდენად ახლოს არის შესაბამისი ვექტორული კომბინაცია ცნობილ მაგალითებთან - ეს ფაქტობრივად წარმოადგენს კავშირის პროგნოზირებას, როგორც მსჯელობის ფორმას. გრაფული ნეირონული ქსელები (GNNs) ასევე ასრულებენ სწავლად მსჯელობას, როდესაც ისინი

ავრცელებენ ინფორმაციას გრაფის სტრუქტურაში მრავალ რაუნდად (რაც ჰგავს multi-hop დასკვნას), და საბოლოოდ აკეთებენ პროგნოზს - მაგალითად კვანძის კლასიფიკაცია, ან კავშირის არსებობის პროგნოზირება. KG-ის ერთ-ერთი უპირატესობა ისაა, რომ ისინი შესაძლებელს ხდიან გამოხატულ (გასაგებ) სჯელობას (*explainable reasoning*): ცოდნის გრაფში, პასუხამდე მიმყვანი ბილიკი ან ქვეგრაფი შეიძლება გამოყენებულ იქნას, როგორც ახსნა. მაგალითად სისტემა პასუხობს „პერსონა დაიბადა საქართველოში“, რადგან არსებობს გზა: პერსონა → დაიბადა → თბილისში და თბილისი → მდებარეობს → საქართველოში - რაც წარმოადგენს ორ საფეხურიან (2-hop) დასკვნას ცოდნის გრაფში. ამგვარი მიდგომა ამსუბუქებს გამჭვირვალობის პრობლემას, რომელიც ხშირად ახასიათებს მხოლოდ ქვესიმბოლურ მოდელებს (ნეირონული ქსელები) (დეჰალი, შარმა and რაჯაბი 2025, 12 - 15).

მაღალი პასუხისმგებლობის სფეროებში (mission-critical domains), ცოდნის გრაფები ხშირად გამოიყენება AI-ის შედეგების გადამოწმებისთვის მაგალითად, მედიკამენტური ჩატბოტის რეკომენდაციები შეიძლება გადამოწმდეს სამედიცინო ცოდნის გრაფის საფუძველზე, რათა დარწმუნდნენ, რომ არ ხდება კრიტიკული წინააღმდეგობების დაშვება; ან იურიდიული დოკუმენტის შინაარსი შეიძლება გადამოწმდეს იურიდიულ ონტოლოგიასთან კავშირის საშუალებით, რათა უზრუნველყოს სამართლებრივი სიზუსტე და შესაბამისობა.

ცოდნის წარმოდგენის მიმდინარე მდგომარეობა მოიცავს სპექტრს ერთი მხრიდან - სიმბოლური ფორმალიზმები გვთავაზობს ცოდნის გააზრებისა და გადამოწმების ჩარჩოს, მეორეს მხრივ - ქვესიმბოლური წარმოდგენები უზრუნველყოფს სწავლისა და განზოგადების მძლავრ ინსტრუმენტს. მოსალოდნელია, რომ ხელოვნური ინტელექტი მომავალშიც ორივე მიმართულებას გამოიყენებს, ალბათ განვითარდება ისეთ წარმოდგენებში, რომლებიც აღარ იქნებიან მკაცრად სიმბოლური და არც ქვესიმბოლური, არამედ კომბინაცია - მაგ. ლატენტური სტრუქტურით გამდიდრებული ვექტორები, ნეირონული ანოტაციებით გაჯერებული გრაფიკები - რაც მანქანებს მისცემს საშუალებას არა მხოლოდ შეინახონ და მიიღონ ცოდნა, არამედ ნამდვილად გაუგონ და იმსჯელოს მასზე ადამიანისთვის დამახასიათებელი სიღრმით.

თავი III - დიდი მონაცემების განაწილებული მოპოვება

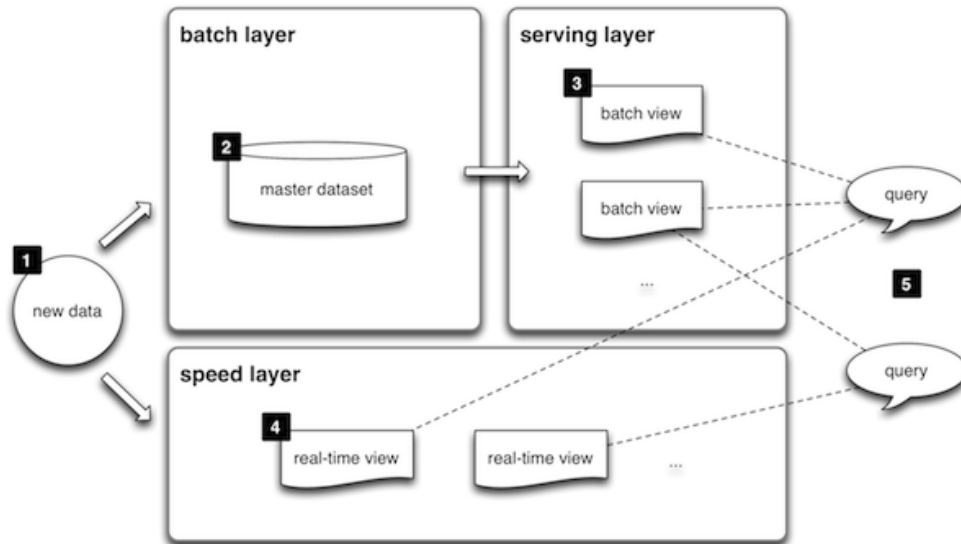
3.1 მონაცემთა ლამბდა-არქიტექტურის დამუშავების კონცეფცია

ლამბდა არქიტექტურა წარმოადგენს მონაცემთა დამუშავების დიზაინის ნიმუშს, რომელიც შექმნილია დიდი მოცულობის მონაცემების სამართავად და აერთიანებს როგორც პაკეტურ (batch), ისე რეალურ დროში, სტრიმინგზე დაფუძნებულ მონაცემთა დამუშავებას ერთიან სისტემაში. ეს კონცეფცია პირველად აღწერა ნეითან მარზმა 2011 წელს. ლამბდა არქიტექტურა შეიქმნა დიდი მონაცემების გავლენით, რათა დაბალანსებულიყო latency (დაყოვნება), throughput (გამტარუნარიანობა) და fault-tolerance (შეცდომებისადმი ამტანობა). ამ მიდგომის მიზანია უზრუნველყოს დროული ანალიტიკა სიზუსტის შემცირების გარეშე - ამისათვის იყენებს პაკეტურ (batch) დამუშავებას „ისტორიული“ მონაცემების სრულყოფილი სიზუსტით დამუშავებისთვის და სტრიმინგს ახალად მიღებული ინფორმაციის დაბალი დაყოვნებით განახლებისთვის.

ლამბდა არქიტექტურის ერთ-ერთი ძირითადი პრინციპია მხოლოდ დამატებაზე დაფუძნებული (append-only), უცვლელი (immutable) მონაცემთა საცავის გამოყენება, როგორც სარეგისტრაციო სისტემა (system of record). ყველა შემომავალი მონაცემი (ჩვეულებრივ, დროის შტამპით მონიშნული მოვლენები) ემატება ამ მთავარ dataset-ს, ნაცვლად იმისა, რომ წაიშალოს ან გადაიწეროს. ეს შესაძლებლობას იძლევა გამეორდეს გამოთვლები ზუსტად იგივე შედეგებით და საჭიროების შემთხვევაში შესაძლებელი გახდეს სისტემის სრულად გადათვლა შეცდომების გამოსასწორებლად. არქიტექტურის ეს მიდგომა განსაკუთრებით პოპულარული გახდა დიდი მონაცემების ჩარჩოების (frameworks) და რეალურ დროში ანალიტიკის მზარდი მოთხოვნების ფონზე. იგი გვთავაზობს, ტრადიციული სამუშაოების (მაგ. MapReduce) მაღალი დაყოვნების შემცირებას, რეალურ დროში დამუშავების ფენის დამატებით (speed layer). (ბერტრანი 2014).

ლამბდა-არქიტექტურაში შემავალი მონაცემების დამუშავება ორ პარალელურ ხაზში ხდება - ერთში პაკეტური (batch) და მეორეში რეალურ დროში (real-time), სადაც ორივე ხაზის შედეგები ერთიანდება მოთხოვნის (query) დასაკმაყოფილებლად. პრაქტიკაში, ეს გულისხმობს კომპლექსური ანალიტიკური

პროცესების დაყოფას მრავალ ფენად, სადაც თითოეულს საკუთარი, უნიკალური ფუნქცია აქვს. ლამბდა-არქიტექტურის ძირითადი შრეები მოიცავს: **პაკეტურ ფენას (Batch layer)**, **სიჩქარის ფენას (Speed layer)**, **სერვისის ფენას (Serving layer)**.



სურ 3.1. მაღალი დონის მონაცემთა ნაკადი ლამბდა არქიტექტურაში(წყარო: <https://shorturl.at/mYjBm>)

ლამბდას არქიტექტურა აერთიანებს ორ პარალელურ დამუშავების გზას - პაკეტურს და რეალურ დროში - რომლებიც ერთსა და იმავე პირველწყარო მონაცემებზე მუშაობენ, როგორც ეს სურ 3.1-ზე ჩანს. პაკეტური ფენა (Batch layer) პერიოდულად ამუშავებს დიდ პაკეტებად დაგროვილ მთავარ dataset-ს, რათა შექმნას ზუსტი პაკეტური ხედები (ანუ წინასწარ გამოთვლილი მოთხოვნის შედეგები). ამავე დროს, სიჩქარის ფენა (Speed layer) ამუშავებს შემომავალ მონაცემთა ნაკადებს რეალურ დროში და ქმნის ე.წ. „სიჩქარის ხედებს“ (speed views), რომლებიც ასახავენ უახლეს მონაცემებს. ორივე შრის შედეგები ინახება სერვისის ფენაში (Serving layer), რომელიც ამ ხედებს აინდექსირებს ან მის მატერიალიზაციას ახდენს, რაც უზრუნველყოფს მოთხოვნებისთვის დაბალი დაყოვნების წვდომას. მოთხოვნის (query) შესრულებისას, ლამბდა არქიტექტურის სისტემა, ძირითადად, იღებს მოთხოვნის პასუხის ძირითად ნაწილს პაკეტური ხედებიდან (რომლებიც ისტორიულ მონაცემებზეა დაფუძნებული) და მას დამატებით ურთავს უახლეს ინფორმაციას სიჩქარის ხედებიდან. ამ გზით მიიღწევა ბალანსი მონაცემთა სიზუსტესა და სისწრაფეს შორის. დამუშავების პროცესის გაყოფით პაკეტურ და სიჩქარის ფენებად, ლამბდა არქიტექტურა აღწევს სიზუსტისა და დაბალი დაყოვნების (low-latency) ბალანსს.

პაკეტური ფენა (Batch layer)

პაკეტური ფენა (Batch layer) პასუხისმგებელია მთავარი dataset-ის მართვაზე და მასშტაბური გამოთვლების შესრულებაზე პაკეტური ხედების (batch views) შესაქმნელად. ახალი მონაცემები მუდმივად ემატება მთავარ dataset-ს, ხოლო რეგულარული ინტერვალებით ხდება პაკეტური გამოთვლის დაწყება მთელი dataset-ის (ან მისი ძალიან დიდი სეგმენტის) ფარგლებში. ამ შრის მიზანია მაღალი სიზუსტე - შედეგების წარმოებამდე მიმდინარეობს ყველა არსებული მონაცემის დამუშავება. სხვა სიტყვებით რომ ვთქვათ, ნებისმიერი წარმოშობილი პაკეტური ხედი წარმოადგენს ფუნქციას, რომელიც გამოითვლება მონაცემთა სრული ისტორიის საფუძველზე გარკვეული მომენტამდე. იმ შემთხვევაში, თუ სისტემა აღმოაჩენს შეცდომას ან საჭიროა ლოგიკის რეტროსპექტული ცვლილება, პაკეტური ფენას შეუძლია ხელახლა გამოითვალოს „დაზარალებული“ ხედები საწყისი მონაცემების სრული არქივის საფუძველზე. ეს უზრუნველყოფს საბოლოო სიზუსტეს (eventual correctness) და ასწორებს ნებისმიერ შეუსაბამობას, რაც შესაძლოა სისწრაფის ფენის მიმდევრულ ან მიახლოებით გამოთვლილ შედეგებში წარმოიშვა. ამ დიზაინით ხდება შეცდომების ამტანობის (fault-tolerance) იმპლემენტაცია: ვინაიდან ნედლი მონაცემები უცვლელია და განუსაზღვრელი ვადით ინახება, სისტემას ყოველთვის შეუძლია შედეგების აღდგენა ან შევსება (backfilling) იმისთვის, რომ ხელახლა განახორციელოს გამოთვლები პირვანდელი წყაროს საფუძველზე (ლი 2019, 2).

ტექნოლოგიური თვალსაზრისით, პაკეტური ფენა ჩვეულებრივ ეყრდნობა განაწილებული დამუშავების ჩარჩოებს, რომლებიც ხელს უწყობს დიდი მოცულობის მონაცემების დამუშავებას. 2010-იანი წლების დასაწყისში ეს ძირითადად ხორციელდებოდა პაკეტური დამუშავების სისტემებით, როგორცაა **Hadoop MapReduce**, ხოლო მოგვიანებით უფრო სწრაფი მეხსიერებაში-მოქმედი გამოთვლებისთვის გამოიყენებოდა ისეთი ძრავები, როგორცაა **Apache Spark**. თანამედროვე ღრუბლოვანი მონაცემთა საწყობები და დიდი მონაცემების პლატფორმები (მაგალითად, **Snowflake**, **Amazon Redshift**, **Google BigQuery**, **Azure Synapse**) ასევე გამოიყენება პაკეტური ფენის ფუნქციის შესასრულებლად - ისინი ინახავენ გიგანტურ dataset-ებს და წინასწარ განსაზღვრული გრაფიკის მიხედვით

ამუშავებენ მათზე მატერიალიზებულ ხედებს. პაკეტური ფენის შედეგები - ე.წ. პაკეტური ხედები შეიძლება მოიცავდეს, მაგალითად, წინასწარ გამოთვლილ აგრეგირებულ ცხრილებს, მანქანური სწავლების მოდელებს მთლიანი dataset-ის საფუძველზე ან ნებისმიერ ანალიტიკურ მონაცემთა პროდუქტს, რომელიც სარგებლობს სრული პაკეტური გამოთვლით. ყოველთვის, როდესაც პაკეტური ფენა სრულდება, ის ან სრულიად ცვლის, ან აახლებს პაკეტურ ხედებს ახალი შედეგებით - შედეგებით, რომლებშიც უკვე ინტეგრირებულია უახლესი მონაცემები პაკეტური გამოთვლის ბოლო მომენტამდე. (ბერტრანი 2014, 2).

მნიშვნელოვანია აღინიშნოს, რომ რადგან პაკეტური ფენა თავიდან ამუშავებს უზარმაზარი მოცულობის მონაცემებს, მას ხშირად ახასიათებს მაღალი დაყოვნება (*latency*) - მაგალითად, სამუშაოები შეიძლება განხორციელდეს საათობრივად ან დღეში ერთხელ. თუმცა, ლამბდა არქიტექტურა მიზანმიმართულად ითვალისწინებს ამ მაღალ დაყოვნებას პაკეტურ ფენაში, იმ პირობით, რომ სისწრაფის ფენა უზრუნველყოფს მონაცემთა განახლების სიხშირესთან დაკავშირებული ახლად წარმოქმნილი სიცარიელების (*freshness gap*) შევსებას. პაკეტური ფენის სიძლიერე მდგომარეობს მის უნარში, შეასრულოს ღრმა და მასშტაბური გამოთვლები (მაგალითად, რთული აგრეგაციები ან რეტროსპექტული ანალიზი) სიზუსტისა და მთლიანობის გარანტიით, რაც ყოველთვის არ არის შესაძლებელი სტრიმინგის (მიმდინარე ნაკადის) კონტექსტში.

სიჩქარის ფენა (Real-Time Layer)

სისწრაფის ფენა (ასევე ცნობილია როგორც რეალურ დროში ან სტრიმინგზე დაფუძნებული ფენა) ავსებს პაკეტური ფენის ფუნქციებს იმით, რომ ამუშავებს მონაცემებს დაბალი დაყოვნებით და ფოკუსირდება ახლად შემოსულ მონაცემებზე. განსხვავებით პაკეტური ფენისგან, რომელიც ამუშავებს მთელ ისტორიულ dataset-ს, სისწრაფის ფენა თითოეულ ახალ ჩანაწერს (ან მცირე მიკროპაკეტს) ამუშავებს მაშინვე, როგორც კი ის სისტემაში შემოდის, რაც შესაძლებელს ხდის რეალურ დროში ან **თითქმის** რეალურ დროში შედეგების მიღებას. მისი ძირითადი როლია შეავსოს განახლების ინტერვალი (*refresh gap*), რომელიც წარმოიქმნება პაკეტური გამოთვლებს შორის ინტერვალში: იმ მომენტში, როცა პაკეტური ფენა ჯერ კიდევ არ მოიცავს

უახლეს მონაცემებს, სისწრაფის ფენა უზრუნველყოფს წამებში განახლებად ანალიზს ამ ახალი მოვლენების შესახებ. ამ პროცესში, სისწრაფის ფენა ცვლის სრულ სიზუსტეს ან მონაცემთა მთლიანობას დროული რეაგირების შესაძლებლობით; მას, როგორც წესი, არ შეუძლია ისეთი ღრმა და კომპლექსური გამოთვლების შესრულება, როგორებიც შესაძლებელია პაკეტურ ფენაში (დროის და არასრული მონაცემების გამო), თუმცა შეუძლია მოახდინოს ინკრემენტული განახლებები ან მიახლოებითი გამოთვლები, რათა სისტემის შედეგები მუდმივად იყოს მაქსიმალურად მიახლოებული რეალურ მდგომარეობასთან.

განსხვავებით პაკეტური ფენისგან, სისწრაფის ფენას უწევს ასინქრონული მონაცემებისა და მონაცემთა კორექციების დამუშავება რეალურ დროში, ვინაიდან მას არ აქვს შედეგების თავიდან გამოთვლის შესაძლებლობა. ხშირად, სისწრაფის ფენა იყენებს სტრიმინგის დამუშავების ჩარჩოებს, რომლებიც მუდმივად განახლებენ სისტემის მდგომარეობას. ამ ფენაში გავრცელებულ ტექნოლოგიებს მიეკუთვნება ისეთი სისტემები, როგორიცაა:

- **Apache Storm** (ერთ-ერთი პირველი სტრიმინგ პროცესორი, რომელიც გამოყენებული იყო მარზის თავდაპირველ რეალიზაციაში),
- **Apache Spark Streaming / Structured Streaming,**
- **Apache Flink,**
- **Apache Samza,**
- **Apache Kafka Streams,**
- და ღრუბლოვანი სერვისები, როგორიცაა **Amazon Kinesis** ან **Azure Stream Analytics.**

ეს ჩარჩოები უზრუნველყოფენ მოვლენის ნაკადების მიღებას (event ingestion) და საშუალებას იძლევიან შეიქმნას აგრეგაციები ან გამოთვლილი შედეგები მინიმალური დაყოვნებით (რამდენიმე წამამი). სისწრაფის ფენის შედეგი, ე.წ. რეალური დროის ხედი ხშირად ინახება სწრაფ, ჩაწერაზე ოპტიმიზებულ მონაცემთა საცავში, რომელიც უზრუნველყოფს ინკრემენტული განახლებებისა და სწრაფი კითხვების მხარდაჭერას. ასეთ საცავებად გამოიყენება NoSQL მონაცემთა ბაზები ან Key-Value ტიპის საცავები (ლი 2019).

სისწრაფის ფენა აგებულია იმ პრინციპზე, რომ მისი შედეგები საბოლოოდ ან **გამოუსადეგარი** გახდება, ან **დაიხვეწება** პაკეტური ფენის მიერ წარმოქმნილი მონაცემებით. სხვა სიტყვებით, **რეალურ დროში გამოთვლილი შედეგები** წარმოადგენს **დროებით პასუხებს**, რომლებიც ემყარება სწრაფად განახლებად ალგორითმებს და არა სრულყოფილ, ისტორიულ ანალიზზე დაფუძნებულ გათვლებს.

მომსახურების ფენა (Serving Layer)

მომსახურების ფენა წარმოადგენს ინტერფეისს, რომლის მეშვეობითაც საბოლოო მომხმარებლები ან აპლიკაციები ახორციელებენ მოთხოვნებს (queries) დამუშავებულ შედეგებზე. ეს ფენა ინახავს როგორც პაკეტური ფენის, ასევე სისწრაფის ფენის მიერ შექმნილ მონაცემებს და უზრუნველყოფს მათ ეფექტურ მოძიებასა და წვდომას. უმეტეს შემთხვევაში, მომსახურების ფენა ფუნქციონირებს როგორც **მოთხოვნის ფენა** - ეს შეიძლება იყოს მონაცემთა ბაზა ან ინდექსირების ტექნოლოგია, რომელიც ოპტიმიზებულია წინასწარ გამოთვლილი ხედების გამოყენებით დაბალი დაყოვნების მქონე კითხვებზე რეაგირებისთვის.

მომსახურების ფენის მთავარი გამოწვევაა შედეგების ორი წყაროს, პაკეტური და სისწრაფის ფენების შერწყმა ისე, რომ საბოლოო მომხმარებლისთვის პროცესი იყოს გამჭვირვალე და ერთგვაროვანი. მომხმარებლის მიერ გაგზავნილი მოთხოვნა არ უნდა, ასახავდეს კონკრეტული პასუხი მიღებულია რეალურ დროში დამუშავებული მონაცემიდან თუ ისტორიული პაკეტიდან, სისტემა თვითონ უნდა უზრუნველყოფდეს შერწყმის ლოგიკას და თანმიმდევრულობას.

3.1.1 ლამბდა არქიტექტურის გამოწვევები

მიუხედავად იმისა, რომ ლამბდა არქიტექტურა ძლიერ და მოქნილ მოდელად განიხილება, იგი ხშირად ექვემდებარება კრიტიკას თავისი შინაარსობრივი სირთულისა და პრაქტიკაში მომსახურების მაღალი ხარჯის გამო. სწორედ ის ფაქტი, რაც ლამბდა არქიტექტურას ანიჭებს მოქნილობას, **პაკეტური და სისწრაფის ფენების გამიჯვნა**, იმასაც ნიშნავს, რომ დამუშავების ერთი და იმავე ლოგიკისათვის საჭიროა ორი დამოუკიდებელი კოდის ბაზის ან დამუშავების პროცესის შენარჩუნება. მაგალითად, თუ მიზანია ისეთი მეტრიკის გამოთვლა, როგორცაა „**უნიკალური**

მომხმარებლების რაოდენობა დღეში“, ლამბდა არქიტექტურაში ეს შეიძლება განხორციელდეს ორჯერ:

- პირველად **პაკეტურ ფენაში** (მაგალითად Hive/Spark-ის გამოყენებით, მთელი dataset-ის აგრეგაციით),
- მეორედ **სისწრაფის ფენაში** (სტრიმინგ პროცესის სახით, რომელიც მუდმივად განახლებად მონაცემებზე მუშაობს და აწარმოებს მიმდინარე რაოდენობას).

იმის უზრუნველყოფა, რომ ეს ორი დამოუკიდებელი რეალიზაცია საბოლოოდ იძლევა იდენტურ შედეგს (რათა რეალურ დროში ხედი თანხვედრაში იყოს პაკეტურ ხედთან), ხშირ შემთხვევაში რთული და შეცდომებისადმი დაუცველი პროცესია. ფენებს შორის ბიზნესლოგიკის ნებისმიერი განსხვავებამ შეიძლება გამოიწვიოს მონაცემთა შეუსაბამობა. სხვა სიტყვებით რომ ვთქვათ, ლამბდა არქიტექტურის ორმაგი მონაცემთა ნაკადები (dual pipelines) ხშირად ორმაგად ზრდის როგორც განვითარების, ისე გამართვის ხარჯს. (ვაენერი 2022, 5).

3.2 მონაცემთა განაწილებული დამუშავების ტრაქტის აღწერა

ბოლო წლებში დიდი მონაცემების (Big Data) გავრცელებამ საფუძვლიანი ცვლილებები გამოიწვია იმაში, თუ როგორ მართავენ ორგანიზაციები მონაცემთა სამუშაო პროცესებს. პროგნოზების მიხედვით, გლობალური „მონაცემთა სფერო“ (datasphere) 2018 წლის 33 ზეტაბაიტიდან გაიზრდება 175 ზეტაბაიტამდე 2025 წლისთვის - ეს გახლავთ ექსპონენციალური ზრდა, რომელიც მოითხოვს მონაცემთა მოცულობის, სიჩქარისა და მრავალფეროვნების ახალი მიდგომებით მართვას. ტრადიციულ მონაცემთა მართვის სისტემებს, რომლებიც შექმნილი იყო სტრუქტურირებული, პაკეტზე ორიენტირებული დამუშავებისთვის, აღარ აქვთ საკმარისი შესაძლებლობები თანამედროვე მოთხოვნებისთვის. დღეს კომპანიები ცდილობენ მიიღონ რეალურ დროზე დაფუძნებული ანალიტიკა და ხელოვნური ინტელექტით წარმართული ანალიზი, რათა მოიპოვონ კონკურენტული უპირატესობა. ამ საჭიროებებზე პასუხად, ორგანიზაციები გადადიან განაწილებულ მონაცემთა დამუშავების ტრაქტებზე, რომლებიც იყენებენ კომპიუტერული კლასტერებისა და ღრუბლოვანი ინფრასტრუქტურის სიმძლავრეს მონაცემების

ფართომასშტაბიანი მიღების, დამუშავებისა და მიწოდებისთვის - როგორც მაღალი სიჩქარით, ისე მაღალი მოცულობით (ნაფაღე 2025, 43).

თანამედროვე განაწილებული მონაცემთა სისტემები დაფუძნებულია არქიტექტურულ პრინციპებზე, რომლებიც უზრუნველყოფს მასშტაბირებადობას, დაბალ დაყოვნებას და საიმედოობას. ერთ-ერთი მნიშვნელოვანი ევოლუცია მონაცემთა დამუშავებაში არის გადასვლა პაკეტურიდან რეალურ დროზე ან სტრიმინგზე ორიენტირებულ არქიტექტურებზე. ისტორიულად, მონაცემთა ინტეგრაცია ხორციელდებოდა პერიოდული პაკეტების სახით (მაგალითად, ETL სამუშაოები), რაც იწვევდა დიდ დაყოვნებას მონაცემის წარმოქმნასა და მის ანალიტიკურ გამოყენებას შორის. დღესდღეობით, განაწილებული გამოთვლების, საცავებში მოქმედი დამუშავების და ღრუბლოვანი პლატფორმების განვითარებამ შესაძლებელი გახადა სისტემის ფუნქციონირება ბევრად უფრო დაბალი დაყოვნებით, რეალურ დროში. კვლევები მიუთითებს, რომ ორგანიზაციებმა, რომლებმაც დანერგეს რეალურ დროზე დაფუძნებული მონაცემთა სისტემები, ოპერაციული ეფექტიანობა გაზარდეს 20%-ით, პაკეტურ მოდელებზე დამყარებულ სისტემებთან შედარებით. სტრიმინგზე დაფუძნებული სისტემები უწყვეტად ამუშავებენ შემომავალ მონაცემებს და აძლევენ ბიზნესებს საშუალებას იმოქმედონ ინფორმაციის მიღებისთანავე, დაყოვნების გარეშე. ბევრი ორგანიზაცია დღეს იყენებს ჰიბრიდულ არქიტექტურებს, რომლებიც მიზნობრივად მიმართავენ რეალურ დროში დამუშავებას დროის მიმართ მგრძობიარე მონაცემებისთვის, ხოლო პაკეტურ დამუშავებას დიდი მოცულობის ისტორიული მონაცემებისთვის ან რთული ტრანსფორმაციებისათვის. ეს დაბალანსებული მიდგომა (რომელსაც ხშირად ლამბდას არქიტექტურას ან მის ვარიანტებს უწოდებენ) აერთიანებს როგორც სისწრაფეს, ისე ღრმა ანალიზის შესაძლებლობას — გამოყენების შემთხვევის (use case) მიხედვით (ნაფაღე 2025, 43).

კიდევ ერთი ფუნდამენტური ცვლილება მონაცემთა დამუშავების არქიტექტურაში არის ტრადიციული **ETL სამუშაო პროცესებიდან** უფრო მოქნილ **ELT** და **მოვლენებზე დაფუძნებულ (event-driven)** პარადიგმებზე გადასვლა. კლასიკური **ETL** პროცესის ფარგლებში (ამოღება-გარდაქმნა-ჩატვირთვა), მონაცემები იძვრებოდა წყაროებიდან, ტრანსფორმირდებოდა ცენტრალურ სერვერზე ან ETL ინსტრუმენტში

და შემდეგ იტვირთებოდა მონაცემთა საწყობში. ეს მიდგომა ხშირად სჭირდებოდა წინასწარ დეტალურად განსაზღვრულ სქემას (schema), რაც მონაცემთა ზრდასთან ერთად ხდებოდა სირთულისა და სისტემური შეფერხებების გამომწვევი მიზეზი. ფაქტობრივად, მოძველებული ETL ტრაქტები თავიანთ რესურსს ძირითადად უთმობენ შენახვისა და ლოგიკის განახლების საქმიანობას (მაგალითად, ახალი მონაცემთა წყაროებისთვის გარდაქმნის წესების მოდიფიკაციას), ვიდრე ინოვაციებზე კონცენტრირებას. ასეთი სისტემები განსაკუთრებით რთულად უმკლავდებიან მონაცემთა მოცულობისა და ჰეტეროგენულობის ზრდას. ხშირად აწყდებიან წარმადობის ბარიერებს (performance bottlenecks), როდესაც სამუშაო დატვირთვა აღემატება ერთი სისტემის გამტარუნარიანობას პიკურ პერიოდებში (ნაფადე 2025, 45).

მეორე მხრივ, თანამედროვე მონაცემთა სისტემები ხშირად ეფუძნება ELT (Extract–Load–Transform) მოდელს, რომელიც იყენებს ღრუბლოვანი პლატფორმების მასშტაბურ პარალელურ გამოთვლით შესაძლებლობებს. ELT არქიტექტურის ფარგლებში, ნედლი მონაცემები ჯერ იტვირთება მასშტაბირებად საცავში ან მონაცემთა ტბაში (*data lake*), ხოლო ტრანსფორმაციის პროცესები ხორციელდება თავად ამ განაწილებულ საცავში ან გამოთვლით ძრავში. მაგალითად, ღრუბლოვან მონაცემთა საწყობებში ან lakehouse ტიპის სისტემებში, როგორებიცაა Snowflake, Amazon Redshift ან Google BigQuery. ამ ცვლილების ძირითადი უპირატესობაა ღრუბლოვანი რესურსების ელასტიურობის გამოყენება. კვლევების მიხედვით, ღრუბლოვან ELT არქიტექტურაზე დაფუძნებულ მიდგომებს შეუძლიათ განვითარების ციკლის ხანგრძლივობის შემცირება დაახლოებით 30%-ით, ტრადიციულ ETL მოდელთან შედარებით. ELT მოდელი მეტ მოქნილობას უზრუნველყოფს, ნედლი მონაცემები ხელმისაწვდომია ტრაქტის სხვადასხვა ეტაპზე მრავალგვარი გამოყენებისთვის. ამასთან, რთული სქემების განსაზღვრა ან ტრანსფორმაციის ლოგიკის შესრულება გადატანილია ტრაქტის შემდგომ ფაზაში, სადაც ის პარალელურად და მასშტაბურად შეიძლება განხორციელდეს. ეს მოდელი სრულად შეესაბამება მონაცემთა ტბებისა (*data lakes*) და lakehouse არქიტექტურების განვითარებას - ეს გახლავთ დიდი მოცულობის, სქემურად მოქნილი საცავები, რომლებიც ინახავენ ნედლ მონაცემებს და უზრუნველყოფენ როგორც ანალიტიკურ

მოთხოვნებზე რეაგირებას, ასევე მონაცემთა დამუშავებას ტრაქტის სხვადასხვა ეტაპზე (ნაფადე 2025, 46).

მასშტაბურობა და განაწილებული პარალელიზმი წარმოადგენს მონაცემთა ტრაქტის არქიტექტურის ქვაკუთხედებს. განაწილებული ტრაქტები შექმნილია ჰორიზონტალურად მასშტაბირებისთვის, რაც ნიშნავს ახალი მანქანებისა და კვანძების (nodes) დამატებას. ტრადიციული ვერტიკალური მასშტაბირების (მარტო ერთ მძლავრ სერვერზე გადართვის) ნაცვლად, ეს მიდგომა შესაძლებელს ხდის მუდმივად მზარდი მონაცემთა მოცულობისა და მომხმარებელთა დატვირთვის ეფექტიანად მართვას. თანამედროვე არქიტექტურაში დომინირებს ღრუბელზე დაფუძნებული (cloud-native) დიზაინი: ტრაქტები იგეგმება და მოქმედებენ ღრუბლოვანი ინფრასტრუქტურაზე, რომელიც უზრუნველყოფს გამოთვლითი და საცავის რესურსების მოთხოვნად პროვიზირებას (on-demand provisioning). ეს ნიშნავს, რომ ტრაქტს შეუძლია გაფართოვდეს ასეულობით ან ათასობით პროცესორზე დიდ სამუშაოზე მუშაობისას და კვლავ შემცირდეს, როცა დატვირთვა მცირდება. ასეთი მოქნილობა ხშირად მიუწვდომელია ფიქსირებული, ლოკალური ინფრასტრუქტურებისთვის. თანამედროვე ფრეიმვორკებმა, როგორებიცაა Apache Hadoop და Apache Spark, დანერგეს პრინციპი: გამოთვლის გადატანა მონაცემებთან, რაც გულისხმობს დიდი dataset-ების დაყოფას კლასტერზე და თითოეული ნაწილის პარალელურ დამუშავებას, საბოლოოდ კი შედეგების აგრეგაციას. ამ განაწილებულ დამუშავების მოდელს ემყარება მრავალი თანამედროვე სისტემის რეალიზაცია და ის უზრუნველყოფს როგორც მაღალ გამტარობას (throughput), ასევე შეცდომების ამტანობას. მაგალითად, თუ რომელიმე კვანძი მწყობრიდან გამოვა, სისტემა შეძლებს იმ კონკრეტული ამოცანის გადატანას და შესრულებას სხვა კვანძზე.

კიდევ ერთი მნიშვნელოვანი პრინციპია კომპონენტებისა და სერვისების განცალკევება (decoupling). ტრადიციული მონოლითური აპლიკაციის ნაცვლად, რომელიც მართავს ტრაქტის მთელ პროცესს, განაწილებული მონაცემთა ტრაქტები ემყარება მიკროსერვისებზე ორიენტირებულ მიდგომას, სადაც ტრაქტის თითოეულ ფაზას (მიღება, დამუშავება, შენახვა და სხვ.) მართავს სპეციალიზებული სისტემა, რომელიც ერთმანეთს უკავშირდება მკაფიოდ განსაზღვრული ინტერფეისებით.

მაგალითად, ტრაქტი შეიძლება იყენებდეს ერთ სერვისს ან კლასტერს რეალურ დროში მონაცემთა მიღებისთვის (მაგალითად, Kafka კლასტერი), სხვა დამუშავების ძრავას ტრანსფორმაციისთვის (მაგალითად, Spark-ის სამუშაოს ან Flink აპლიკაციის სახით), და განსხვავებულ საცავებს საბოლოო შედეგების შესანახად (მაგალითად, S3 მონაცემთა ტბა და Snowflake მონაცემთა საწყობი). ეს კომპონენტები ერთმანეთს უკავშირდებიან შეტყობინების რიგებით (message queues), API გამოძახებებით ან შუალედური საცავის მეშვეობით. კომპონენტების განცალკევება ზრდის მოქნილობასა და მასშტაბირებადობას. თითოეული ნაწილი დამოუკიდებლად შეიძლება მასშტაბირდეს ან განახლდეს საჭიროებისამებრ. ეს პრინციპი სრულად თავსებადია კონტეინერიზაციისა და ღრუბლოვანი გარემოს დანერგვის საუკეთესო პრაქტიკებისადმი (გოგარე 2024, 3).

3.2.1 განაწილებული მონაცემთა დამუშავების ძირითადი კომპონენტები

მიუხედავად იმისა, რომ განაწილებული მონაცემთა სისტემის არქიტექტურა შეიძლება განსხვავდებოდეს დიზაინის მიხედვით, მათ უმეტესობაში გვხვდება საერთო კომპონენტთა ან ფენათა ერთობლიობა, სადაც თითოეული ასრულებს კონკრეტულ ფუნქციას მონაცემთა ნაკადის მარშრუტში. ამ ძირითად კომპონენტებს მიეკუთვნება:

- მონაცემთა მიღება (data ingestion),
- მონაცემთა ტრანსფორმაცია/დამუშავება (data transformation/processing),
- მონაცემთა შენახვა (data storage),
- ტრაქტის ორკესტრაცია (pipeline orchestration)/სამუშაო ნაკადების მართვა.

თითოეული ამ კომპონენტი თავისთავად შეიძლება იყოს განაწილებული და მასშტაბირებადი. ერთობლივად, ისინი ქმნიან მონაცემთა მთლიანი პროცესის სამუშაო ნაკადს (end-to-end workflow), რომელიც ხშირად წარმოდგენილია როგორც დირექციული აციკლური გრაფი (DAG). სადაც, მაგალითად, მონაცემთა მიღება-გადაეცემა დამუშავება, შენახვა, მთელი ამ პროცესის კონტროლი ხდება ორკესტრაციის ფენის მეშვეობით.

მონაცემთა შეყვანა(Data Ingestion)

მონაცემთა მიღება (Data Ingestion) წარმოადგენს მონაცემთა ტრაქტის პირველ ეტაპს. ამ ფენაში ნედლი მონაცემები ერთ ან რამდენიმე წყაროდან იკრიბება და შედის ტრაქტის დამუშავების სისტემაში. განაწილებულ სისტემებში, მონაცემთა მიღების ფენა ხშირად უნდა მოიცავს მაღალი მოცულობის, უწყვეტი და მრავალფეროვანი წყაროებიდან შემომავალი მონაცემები. მაგალითად: სტრუქტურირებული მონაცემთა ბაზები, არასტრუქტურირებული ლოგები, IoT სენსორების ნაკადები, გარე API-ები და სხვა. მიღების ფენის ძირითადი ამოცანაა უზრუნველყოს, რომ მონაცემები საიმედოდ აღიწეროს და ხელმისაწვდომი გახდეს ტრაქტის მომდევნო ეტაპებისთვის, იქნება ეს რეალურ დროში დამუშავება თუ დაგეგმილი პაკეტური ინტერვალებით შესრულება. განაწილებულ ტრაქტებში, მონაცემთა მიღების არქიტექტურა აგებულია მასშტაბირებადობის პრინციპზე და მხარს უჭერს მონაცემთა ნაკადების სხვადასხვა რეჟიმს. მიღება შეიძლება განხორციელდეს მონაცემთა პერიოდული გამოთხოვით (pull model), მოვლენის ნაკადების პირდაპირი მიწოდებით (push/streaming), შეტყობინებების რიგების (message queues) ან ცვლილებების ჟურნალების (change logs) მეშვეობით (ბეჰერა, ვარდან და ჩილუკოროი 2024, 29).

მონაცემთა ტრანსფორმაცია და დამუშავება(Data Transformation and Processing)

მას შემდეგ, რაც მონაცემები მიღებულია, ისინი გადადიან ტრაქტის დამუშავებისა და ტრანსფორმაციის ფენაში. ეს კომპონენტი პასუხისმგებელია ნედლი მონაცემების გადაყვანაზე სტრუქტურირებულ, გაწმენდილ და გამდიდრებულ ფორმაში, რომელიც მზადაა ანალიზისთვის ან შენახვისთვის. განაწილებული სისტემების კონტექსტში, დამუშავების ფენა, როგორც წესი, ხორციელდება პარალელური მონაცემთა დამუშავების ჩარჩოებით, რომლებიც უზრუნველყოფენ ფართომასშტაბიანი გამოთვლების შესრულებას კლასტერულ გარემოში. დამუშავების ფენაში შესაძლოა მოექცეს სხვადასხვა სახის ოპერაციები, როგორცაა მონაცემების გაწმენდა, მრავალი ნაკადის გაერთიანება (stream join), მეტრიკების აგრეგაცია, სტატისტიკური ანალიზი ან მანქანური სწავლების ალგორითმების გამოყენება და სხვა ტრანსფორმაციები, რომლებიც ემსახურება კონკრეტულ ბიზნეს საჭიროებებს.

მნიშვნელოვანია აღინიშნოს, რომ დამუშავებისა და ტრანსფორმაციის ფენა განაწილებულ მონაცემთა ტრაქტში გაცილებით მეტია, ვიდრე მხოლოდ საბაზისო

ტრანსფორმაციები - ის ხშირად ახორციელებს კომპლექსურ მონაცემთა გამდიდრებასა და ხარისხის უზრუნველყოფის ზომებს. გავრცელებულ ნაბიჯებს მოიცავს:

- **მონაცემთა გაწმენდა** - არასწორი ან დაზიანებული ჩანაწერების ამოღება ან კორექტირება,
- **დუბლიკატების მოცილება** (*deduplication*),
- **ფორმატის ჰარმონიზაცია** - მონაცემთა სხვადასხვა წყაროდან შემოსული ფორმატების გასათანაბრებლად,
- **დამატებითი გამდიდრება** - წარმოებული ველების დამატება ან მონაცემთა დაკავშირება საკვანძო ცხრილებთან ან გარე ცნობებთან.

მაგალითად, სისტემამ შეიძლება გააანალიზოს ლოგები(log) და გარდაქმნას ისინი სტრუქტურირებულ ჩანაწერებად, ამოიღოს დაზიანებული სტრიქონები, მოახდინოს კონფიდენციალური ველების ანონიმიზაცია ან დაშიფვრა, და გამოთვალოს ახალი ფუნქციები (მაგალითად, მომხმარებლის სესიის ხანგრძლივობა ან მოძრავი საშუალო მაჩვენებლები), სანამ მონაცემები ტრაქტის მომდევნო ეტაპზე გადაინაცვლებს. ეს ამოცანებიც შეიძლება **განაწილებულად შესრულდეს კლასტერზე**, პარალელურად. თანამედროვე ჩარჩოები და ბიბლიოთეკები მხარს უჭერენ ასეთი ოპერაციების მასშტაბურ შესრულებას - მაგალითად, Apache Spark გთავაზობთ ბიბლიოთეკებს SQL-სტილის მონაცემთა მანიპულაციისთვის, მანქანური სწავლებისთვის და სხვა, რომლებიც პარალელურად სრულდება.

დამუშავების ფენის შედეგი, როგორც წესი, არის ანალიტიკისთვის მზა მონაცემი - ანუ სტრუქტურირებული და საჭირო დონეზე აგრეგირებული, ისე რომ შესაძლებელია მისი გამოყენება ანალიტიკური ინსტრუმენტების, მანქანური სწავლების მოდელების ან შენახვის სისტემების მიერ. ოკლედ რომ ვთქვათ, ტრანსფორმაციის ფენა გარდაქმნის ნედლ მონაცემებს ღირებულ ინფორმაციად. განაწილებული დამუშავების ჩარჩოების მეშვეობით, ამ ფენას შეუძლია ერთდროულად გაუმკლავდეს როგორც ისტორიულ პაკეტურ მონაცემებს, ისე მაღალი სიჩქარის რეალურ დროში ნაკადებს, რითაც ასრულებს კომპლექსურ გამოთვლებს იმ დროში, რაც ერთი მანქანისთვის შეუძლებელი იქნებოდა (ბეჰერა, ვარდან და ჩილუკოორი 2024, 31-32).

მონაცემთა საცავი (Data Storage)

მას შემდეგ, რაც მონაცემები დამუშავდება და ტრანსფორმაციას გაივლის, საჭიროა მათი შენახვა შესაბამის ადგილას, რათა ხელმისაწვდომი გახდეს ტრაქტის შემდეგი ეტაპებისთვის. განაწილებული ტრაქტის შენახვის ფენა უზრუნველყოფს მდგრად და მასშტაბირებად მონაცემთა საცავებს, როგორც შუალედური, ისე საბოლოო შედეგებისთვის. განაწილებული არქიტექტურების კონტექსტში, შენახვის ფენა, როგორც წესი, არ არის ერთიანი სისტემა, არამედ წარმოადგენს ფენებად დაყოფილ, სხვადასხვა ტიპის საცავების კომბინაციას, რომლებიც შერჩეულია მონაცემთა ტიპისა და წვდომის რეჟიმების შესაბამისად. შენახვის კომპონენტის ძირითადი ასპექტებია:

- **მასშტაბურობა** - მზარდი მოცულობის ეფექტიანი მართვა,
- **წარმადობა** - მონაცემების სწრაფი წაკითხვისა და მოპოვების მხარდაჭერა მომხმარებლების მხრიდან,
- **მონაცემთა ორგანიზება** - სქემების, დანაყოფების (partitioning) და ინდექსების სტრუქტურირებული მართვა,
- **ხარჯთეფექტიანობა** - მონაცემთა შენახვის ოპტიმიზაცია ღირებულების თვალსაზრისით.

თანამედროვე მონაცემთა სისტემებში მონაცემთა ტბები (data lakes) და მონაცემთა საწყობები (data warehouses) (ან მათი ანალოგიური სისტემები) ხშირად გამოიყენება ერთობლივად - როგორც დამატებითი და კომპლემენტარული გადაწყვეტილებები, რომლებიც ერთმანეთის შესაძლებლობებს ავსებენ.

შენახვის ფენის ძირითადი როლია მონაცემთა საიმედო შენარჩუნება და მათი ხელმისაწვდომობა ტრაქტის შემდგომი გამოყენებისთვის. იქნება ეს ანალიზი, მანქანური სწავლება, თუ აპლიკაციაში ინტეგრაცია. მონაცემთა ტბისა და საწყობის მოდელების კომბინაციით, ტრაქტი აღწევს როგორც მოქნილობას, ისე მაღალ წარმადობას. მონაცემთა ტბა (data lake) ინახავს ყველა ტიპის მონაცემს - რაც უზრუნველყოფს სისრულეს, მონაცემთა არდაკარგვას და მომავლისთვის განმეორებით დამუშავების შესაძლებლობას. მონაცემთა საწყობი ან მინი-საწყობი (warehouse ან marts) შეიცავს მნიშვნელოვნად გაფილტრულ და კურირებულ მონაცემებს, რომლებიც მზად არის სწრაფი ანალიზისთვის. განაწილებულ

სისტემებში, შენახვის სისტემებიც აგებულია განაწილებულად. ისინი მონაცემებს ან დაყოფენ (partition), ან მათ დუბლიკაციას (replication) ახდენენ მრავალ კვანძზე. მაგალითად HDFS (Hadoop Distributed File System) ან Cassandra ახორციელებენ მონაცემთა ბლოკების რეპლიკაციას რამდენიმე მონაცემთა კვანძზე, რაც უზრუნველყოფს შეცდომების ამტანობას და მონაცემთა ხელმისაწვდომობას სისტემური შეცდომების შემთხვევაშიც (ბეჰერა, ვარდან და ჩილუკოორი 2024, 33).

ორკესტრაცია და სამუშაო პროცესის მართვა

მთელი მონაცემთა ტრაქტის ზედამხედველობას ახორციელებს ორკესტრაციის ფენა, რომელიც პასუხისმგებელია სხვადასხვა ამოცანის დაგეგმვაზე, კოორდინაციაზე და ოპერაციულ მართვაზე. ორკესტრაცია უზრუნველყოფს, რომ ტრაქტის თითოეული ეტაპი შესრულდეს სწორ თანმიმდევრობით, სწორ დროს და შესაბამის რესურსებზე, ასევე მართავს დამოკიდებულებებს ამოცანებს შორის. მაგალითად, იმის გარანტიას, რომ მონაცემთა მიღების ეტაპი დასრულებულია, სანამ დაიწყება დამუშავების სამუშაო. განაწილებული ტრაქტების პირობებში, სადაც სხვადასხვა კომპონენტი შეიძლება სხვადასხვა კლასტერზე ან სერვისზე მუშაობდეს, ორკესტრაცია კრიტიკულად მნიშვნელოვანია პროცესის ავტომატიზაციისა და სისტემის საიმედოობის უზრუნველსაყოფად. ორკესტრაციის ფენა მოქმედებს როგორც ცენტრალური "ტვინი", რომელიც მართავს ტრაქტის შესრულების ლოგიკას (ხშირად წარმოდგენილი როგორც ამოცანების აციკლური მიმართული გრაფი – DAG), და უზრუნველყოფს ისეთ მახასიათებლებს, როგორცაა მონიტორინგი, შეტყობინებები (alerts) და შეცდომების მართვა ან აღდგენა (მუვა 2021, 318).

დაგეგმვის ფუნქციის მიღმა, ორკესტრაციის ხელსაწყოები უზრუნველყოფენ დამატებით მახასიათებლებს, როგორცაა გადატვირთვის პოლიტიკა (*retry policies*), შეტყობინებები და ალერტები, ლოგირება და აუდიტის კვალი (*audit trails*). თუ ტრაქტის რომელიმე ამოცანა ვერ სრულდება (მაგალითად, ტრანსფორმაციის სამუშაო გამოიტანს შეცდომას), **ორკესტრატორი** აფიქსირებს ამ შემთხვევას, არეგისტრირებს შეცდომას ლოგებში და ავტომატურად ახორციელებს ამოცანის ხელახალ მცდელობას ან მთელ ტრაქტს მონიშნავს როგორც წარუმატებელს და აგზავნის შეტყობინებას. ეს ხელს უშლის შეცდომების გავრცელებას და იძლევა სწრაფი რეაგირების

შესაძლებლობას. ორკესტრაციის სისტემები ხშირად ინტეგრირდება ვერსიის კონტროლის სისტემებთან (source control) და CI/CD პროცესებთან, რაც უზრუნველყოფს ტრაქტის დეფინიციების ვერსიის კონტროლს და ტესტირებას. თანამედროვე ორკესტრატორები სულ უფრო ხშირად უჭერენ მხარს დინამიური სამუშაო პროცესებისა და პირობითი ლოგიკის მართვას. მაგალითად, სხვადასხვა ბრუნვის გზის არჩევა მონაცემთა მნიშვნელობების ან გარე პირობების საფუძველზე.

განაწილებულ გარემოში, ორკესტრატორი ხშირად პასუხისმგებელია რესურსების პროვიზირებაზეც, ზოგიერთი მოწინავე სამუშაო პროცესი, მაგალითად, თვითონ ქმნის დროებით Spark კლასტერს ღრუბელში მძიმე პაკეტური სამუშაოს შესასრულებლად და მის დასრულების შემდეგ ავტომატურად აჩერებს მას ხარჯების დაზოგვის მიზნით. ამ ყველაფერს ახორციელებს ორკესტრაციის კონტროლის ქვეშ.

საბოლოოდ ორკესტრაცია წარმოადგენს განაწილებული მონაცემთა ტრაქტის საკონტროლო ფენას. სწორედ ის აერთიანებს ერთმანეთისგან განცალკევებულ კომპონენტებს ერთიან, თანმიმდევრულ სამუშაო ნაკადად. მძლავრი ორკესტრაციის ჩარჩოების გამოყენებით, ორგანიზაციები უზრუნველყოფენ, რომ მათი მონაცემთა ტრაქტები მუშაობენ შეუფერხებლად, სრულდებიან დაგეგმილი დროის მიხედვით და შეცდომების შემთხვევაში შეძლებენ აღდგენას. ეს განსაკუთრებით მნიშვნელოვანია მაშინ, როდესაც მონაცემთა ტრაქტი წარმოადგენს კრიტიკულ საბაზისო სისტემას წარმოებაში.

3.2.2 განაწილებული გენეტიკური ალგორითმების გამოყენება დიდი მონაცემების ნაკადური დამუშავებისთვის

დღითიდღე იზრდება დიდი მოცულობის მაღალი სიჩქარის მქონე ნაკადური მონაცემების დამუშავებისადმი ინტერესი. ასეთი მონაცემების დამუშავების სირთულე დაკავშირებულია იმ ფაქტთან, რომ ნაკადის შემოდინება უწყვეტია და სწრაფად ცვალებადი, რომლებიც გენერირდება სხვადასხვა წყაროდან და ხშირად ხასიათდება შემოდინების მაღალი სიჩქარით, დიდი მოცულობით და მოკლე სასიცოცხლო ციკლით. ნაკადური მონაცემების მნიშვნელოვნება მდგომარეობს მის უნარში წარმოადგინოს აქტუალური ინფორმაცია რეალური დროის რეჟიმში, რაც ხდომილებებზე დროულად რეაგირების, ცვალებად პირობებზე ადაპტირების

საშუალებას იძლევა. ამ თავში წარმოდგენილია ლამბდა არქიტექტურის და განაწილებული გენეტიკური ალგორითმების გამოყენების ანალიზი დიდ მონაცემთა დამუშავების და მაღალი წარმადობის გამოთვლების ამოცანებისთვის. დადასტურებულია მათი ერთობლივი გამოყენების უპირატესობა ოპტიმიზაციის ამოცანების გადასაწყვეტად (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 54).

თავში შემუშავებულია განაწილებული გენეტიკური ალგორითმი ლამბდა არქიტექტურის პირობებში. აღწერილია ალგორითმის მუშაობის სრული ციკლი, დაწყებული რეალურ დროის მოცემულ მონაკვეთში შემავალი მონაცემების ფრაგმენტებად დაყოფით, დამთავრებული ოპტიმალური ამონახსნის პოვნით. წარმოდგენილია შემუშავებული ალგორითმის ბლოკ-სქემა.

წარმოდგენილი მიდგომა იყენებს ლამბდა არქიტექტურისა და განაწილებული გენეტიკური ალგორითმების მიდგომების უპირატესობებს. ამდენად, მათი ერთობლივი გამოყენება მიზანშეწონილია დიდ მონაცემებში ოპტიმიზაციის ფართომასშტაბიანი ამოცანების გადასაწყვეტად რეალურ დროში (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 54).

პრობლემის მიმოხილვა

მიუხედავად იმის, რომ რეალურ დროში მონაცემთა დამუშავებისა და ანალიზის სისტემები უდიდეს უპირატესობას იძლევა გადაწყვეტილების მიღებისას, მითუმეტეს მაშინ თუ გადაწყვეტილება რეალურ დროშია მისაღები, გვერდს ვერ ავუვლით მათ რამდენიმე პრობლემურ ასპექტს. რეალური დროის სისტემების მიერ მიღებული ნაკადების სიჩქარე და მოცულობა, რომლებიც სხვადასხვა წყაროებიდან შეიძლება იყოს მიღებული, წარმოადგენს დიდ პრობლემას სისტემებისთვის, რადგან მონაცემთა ასეთი დიდი მოცულობის რეალურ დროში დამუშავება საკმაოდ რთულია, როგორც აპარატურის და ქსელისთვის, ასევე არსებული ალგორითმებისთვისაც. ასევე საყურადღებოა დაბალი დაყოვნებების მიღწევა, რათა არ მოხდეს შეფერხება გადაწყვეტილების მიღებისას (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 54).

აღნიშნული პრობლემები განსაკუთრებით მნიშვნელოვანია დიდ, განაწილებულ გარემოში, როგორცაა არამხოლოდ ბიზნესის მიმართულება, არამედ ასტრონომიული, მეტეო, ნივთების ინტერნეტი (IoT). მონაცემთა მოცულობისა და

სიჩქარის პრობლემების გადაჭრის გზა მოიცავს ტექნოლოგიური არჩევანის ერთობლიობას, მათ შორის ნაკადის დამუშავების ჩარჩოებს, განაწილებულ მესხიერებას და ოპტიმიზების ალგორითმებს. სწორედ ამიტომ ცხადია, რომ განაწილებული გენეტიკური ალგორითმების და ლამბდა არქიტექტურის ერთობლივი გამოყენება აღნიშნული პრობლემების მიმართ საუკეთესო არჩევანია (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 54).

ლამბდა არქიტექტურა და განაწილებული გენეტიკური ალგორითმი

კომპანიების მიზნების მისაღწევად „ისტორიული“ მონაცემების ანალიზი აღარ არის საკმარისი. დღეს, როდესაც ბაზრები უფრო დინამიურია, ვიდრე ოდესმე და მონაცემები გაცილებით სწრაფად, ზოგჯერ რამდენიმე წამში განიცდის ცვლილებას, რეალურ დროში მიღებული ინფორმაცია შეიძლება ბევრად უფრო მნიშვნელოვანი იყოს კომპანიისთვის, ვიდრე ინფორმაცია, რომელიც დროის გარკვეული მონაკვეთის განმავლობაში არის შეგროვებული და „დახარისხებული“. სწორედ დიდ და სწრაფად მზარდ, მათ შორის რეალურ დროში ნაკადურ მონაცემთა ანალიზისთვის არის მიზანშეწონილი ლამბდა არქიტექტურის გამოყენება. ამდენად, ამ მექანიზმს თავისი უნივერსალურობიდან გამომდინარე მაღალი სირთულის ამოცანების გადაწყვეტის უნარი აქვს, რასაც იგი უმკლავდება სამი ძირითადი ფენის საშუალებით (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 55):

Batch Layer: ფენა პასუხისმგებელია მონაცემთა დიდი მოცულობის დამუშავებასა და ანალიზზე პაკეტურ (Batch) რეჟიმში. ამ ფენაში მონაცემები აგრეგირდება, იფილტრება და ინახება პაკეტების სახით, რაც ანალიზისთვის მონაცემთა პაკეტური წარმოდგენის აგების საშუალებას იძლევა.

Speed layer: სიჩქარის ფენა ახდენს რეალურთან მიახლოებულ დროში მონაცემთა მიღებას ანალიზისთვის. მას მინიმალური დაყოვნება აქვს გამტარუნარიანობაში, რაც ბიზნესს სწრაფი გადაწყვეტის საშუალებას აძლევს (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 55).

Serving layer: სერვისის ფენა პასუხისმგებელია მომხმარებლისთვის ან კლიენტის აპლიკაციებისთვის მოთხოვნის შედეგების მიწოდებაზე. ის აერთიანებს პაკეტურ წარმოდგენას (Batch View) და რეალური დროის წარმოდგენებს (Real-time

View), რათა უზრუნველყოს მონაცემთა ერთიანი წარმოდგენა. მის მიმართ ძირითადი მოთხოვნაა, რომ დამუშავება შესრულდეს მაქსიმალურად პარალელური სახით, რათა მინიმიზდეს მონაცემთა ნაკრების ინდექსირების დრო. სანამ ინდექსირების დავალება შესრულების პროცესშია, ახლად შემოსული მონაცემები დგება რიგში შემდგომ დავალებაში ინდექსირებისთვის. (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 55)

როგორც ვხედავთ, Lambda არქიტექტურას აქვს ნებისმიერი მასშტაბის ინფორმაციულ ნაკადებთან ეფექტურად მუშაობის შესაძლებლობა, ამდენად მისი გამოყენება გამართლებულია დიდ მონაცემთა ნაკადების დამუშავების შემთხვევაში.

გენეტიკური ალგორითმის მიდგომა. გენეტიკურ ალგორითმებში (GA) კარგი გადაწყვეტილებები, როგორც წესი, შედგება კარგი სამშენებლო ბლოკებისგან, რომლებიც მიიღება ქრომოსომების მნიშვნელობების შესაბამისი კომბინაციებით (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 55).

გენეტიკური ალგორითმების მუშაობის პრინციპია: - საწყისი პოპულაციიდან (*current generation*) შუალედური პოპულაციის გენერირება (*intermediate generation*) შერჩევის (*selection*) გზით. ინდივიდების შეჯვარება (*recombination*) შუალედური პოპულაციიდან კროსოვერის (*crossover*) მეშვეობით, რასაც მივყავართ ახალი თაობის პოპულაციის მიღებასთან (*next generation*); ახალი თაობის მუტაცია.

სტანდარტულ გენეტიკურ ალგორითმში გამოიყენება შემდეგი გენეტიკური ოპერატორები: გადარჩევის ოპერატორი (სელექცია), შეჯვარების ოპერატორი (რეკომბინაცია), მუტაციის ოპერატორი, ინვერსიის ოპერატორი, შესაძლებელია ასევე სხვა გენეტიკური ოპერატორები როგორცა: ტრანსლოკაცია, სეგრეგაცია და სხვ.

კროსოვერი გენეტიკური ალგორითმის მნიშვნელოვანი ოპერაციაა, რომელიც გენების შეჯვარების საშუალებას იძლევა, რომლის შედეგად მიიღება შვილი პოპულაცია. ის გულისხმობს ორი მშობლის ნაწილების შერევას და შეჯვარებას ახალი პოპულაციის მისაღებად. (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 56)

ევოლუციურმა გამოთვლებმა დაადასტურეს მისი გამოყენების ეფექტურობა ოპტიმიზაციის რთული ამოცანების გადაწყვეტაში, მაგრამ ინფორმაციული ეპოქის სწრაფი განვითარებით და მის კვალდაკვალ დიდი მონაცემების გააქტიურებით გაიზარდა და გართულდა ევოლუციური მიდგომების მიმართ დასმული ამოცანები.

პრობლემა განსაკუთრებით აქტუალურია, როდესაც საძიებო სივრცე მოიცავს ლოკალური ოპტიმუმების უზარმაზარ რაოდენობას და ტრადიციული გენეტიკური ალგორითმი ვერ უმკლავდება მონაცემთა ნაკადების დამუშავებას რეალურ დროში (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 56).

პაკეტური დამუშავება ითვალისწინებს დიდი მოცულობის სტატიკური მონაცემების დამუშავებას, რომელიც უკვე ინახება სისტემაში. ამ პარადიგმის სიზუსტე ხშირად აღემატება ნაკადური დამუშავების სიზუსტეს, მაგრამ იგი არ ითვალისწინებს ახალ მონაცემებს დამუშავების დაწყების შემდეგ. ამდენად პაკეტური მოდელი ვერ იქნება ადაპტური რეალურ დროში მონაცემთა ნაკადების დამუშავებისთვის. მეორეს მხრივ ნაკადური დამუშავების მიზანია გაუმკლავდეს დიდი მონაცემების სისწრაფეს და დაამუშაოს იგი რეალურ დროში მისი შემოსვლისთანავე. ეს მიდგომა ამუშავებს ნაკადებს თანმიმდევრობით, ისე რომ ცალკეული ახალი მონაცემები შეიძლება ჩაირთოს მოდელის განახლებაში დაბალი დაყოვნებით. შესაბამისად, მოდელი ადაპტირდება ახალ შემომავალ მონაცემთა ნაკადზე (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 56).

განაწილებული გენეტიკური ალგორითმები (DGA) ინარჩუნებს კლასიკური გენეტიკური ალგორითმის (GA) ძირითად პრინციპებს, თუმცა იყენებს განაწილებულ გამოთვლით რესურსებს ეფექტურობისა და მასშტაბურობის გასაუმჯობესებლად. DGA-ს მთავარი იდეაა, რომ მისი ფუნქციონირება ხდება მრავალ მანქანაზე ან პროცესორზე, რათა უფრო ეფექტურად გადაჭრას ოპტიმიზაციის რთული პრობლემები. ალგორითმი ყოფს ინდივიდების პოპულაციას ქვეპოპულაციებად, რომლებიც დამოუკიდებლად მუშავდება სხვადასხვა გამოთვლით კვანძზე. ეს ქვეპოპულაციები ვითარდებიან და პერიოდულად ცვლიან ინფორმაციას გლობალური ძიების მისაღწევად. განაწილებული გენეტიკური ალგორითმების მუშაობის პრინციპი მოიცავს რამდენიმე ოპერაციას (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 56).

ინიციალიზაცია: DGA იწყება ინდივიდების პოპულაციის ინიციალიზაციის გზით ან რაიმე ევრისტიკის გამოყენებით. თითოეული ინდივიდი წარმოადგენს ოპტიმიზაციის პრობლემის პოტენციურ გადაწყვეტას.

შეფასება: თითოეული გამოთვლითი კვანძი აფასებს ინდივიდების შესაბამისობას მის ქვეპოპულაციაში. ფიტნესი წარმოადგენს იმას, თუ რამდენად კარგად წყვეტს ინდივიდი ოპტიმიზაციის პრობლემას, რომელიც ჩვეულებრივ იზომება ობიექტური ფუნქციით.

რეპროდუქცია: GA ოპერატორები, როგორცაა შერჩევა, კროსოვერი და მუტაცია, დამოუკიდებლად გამოიყენება თითოეულ სუბპოპულაციაზე ახალი შთამომავლების შესაქმნელად. ეს ოპერატორები მსგავსია შერჩევის, რეკომბინაციისა და მუტაციის ბუნებრივ გენეტიკური პროცესების, რათა ხელი შეუწყონ საძიებო სივრცის შესწავლასა და ექსპლუატაციას.

გაცვლა: რეგულარული ინტერვალებით ან გარკვეული რაოდენობის თაობების შემდეგ, კომუნიკაცია ხდება გამოთვლით კვანძებს შორის. ეს შეიძლება მოიცავდეს საუკეთესო ინდივიდების ქვეჯგუფის გაზიარებას ან გენეტიკური ინფორმაციის გაცვლას ქვეპოპულაციებს შორის. მიზანია მრავალფეროვნების დანერგვა და ინფორმაციის გაზიარების საშუალება საძიებო სივრცის სხვადასხვა ნაწილიდან (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 56).

კონვერგენცია: ევოლუციის პროცესი გრძელდება ახლად შექმნილ შთამომავლობით ინდივიდებთან. დროთა განმავლობაში, თითოეულ ქვეპოპულაციაში პოპულაციები პერსპექტიული გადაწყვეტილებებისკენ მიდის.

შეწყვეტა: ალგორითმი მთავრდება, როდესაც დაკმაყოფილებულია შეწყვეტის კრიტერიუმი, როგორცაა თაობების მაქსიმალური რაოდენობის მიღწევა, დამაკმაყოფილებელი გამოსავლის პოვნა ან წინასწარ განსაზღვრული გამოთვლითი დროის გადაჭარბება (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 56).

ფიტნეს ფუნქცია. ანაწილებულ გენეტიკური ალგორითმებში (DGA), ფიტნეს ფუნქცია არის გადაწყვეტი კომპონენტი, რომელიც განსაზღვრავს რამდენად კარგად მუშაობს ინდივიდი (ქრომოსომა) ოპტიმიზაციის პრობლემის გადაჭრისას. ფიტნეს ფუნქცია რაოდენობრივად განსაზღვრავს ინდივიდუალური გადაწყვეტის ხარისხს ან ვარგისიანობას პოპულაციაში. ეს არის ძირითადი შეფასების მეტრიკა, რომელიც გამოიყენება ევოლუციის პროცესის წარმართვისთვის, რეპროდუქციისთვის

ინდივიდების შერჩევით და მომავალი თაობის განსაზღვრით (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 57).

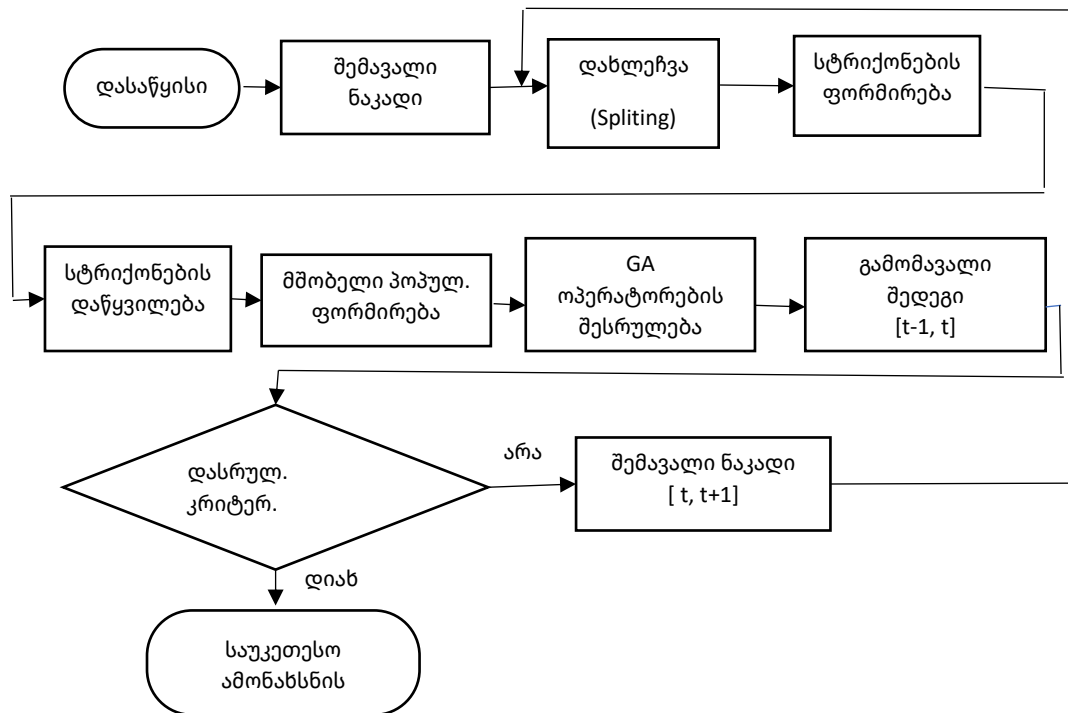
ფიტნეს ფუნქცია იღებს ინდივიდს (ქრომოსომა) და ანიჭებს ფიტნეს ქულას იმის მიხედვით, თუ რამდენად აკმაყოფილებს ინდივიდი ოპტიმიზაციის კრიტერიუმებს ან მიზნებს. რაც უფრო მაღალია ფიტნეს ქულა, მით უკეთესია ინდივიდის წარმადობა პოპულაციაში სხვებთან შედარებით.

განაწილებულ გარემოში, თითოეული გამოთვლითი კვანძი დამოუკიდებლად აფასებს ინდივიდების შესაბამისობას მის ქვეპოპულაციაში. სხვადასხვა კვანძიდან მიღებული ფიტნეს მნიშვნელობები გამოიყენება შერჩევისა და რეპროდუქციის პროცესის წარმართვისთვის, ასევე საუკეთესო ამონახსნის მისაღებად (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 57).

რეალურ დროში დიდი მოცულობის მონაცემთა ნაკადებში ოპტიმიზაციის ამოცანის გადაწყვეტის მიზნით ჩვენ მიერ შემუშავებულია განაწილებული გენეტიკური ალგორითმი ლამბდა არქიტექტურით, რომელიც შედგება შემდეგი ეტაპებისგან:

- შემავალი მონაცემების ერთი ნაკადის მიღება $[t-1, t]$ დროის ინტერვალში;
- შემავალი ნაკადის დახლეჩვა ფრაგმენტებად, Splitting ოპერაციის შესრულება;
- ცალკეულ ფრაგმენტში გენეტიკური ალგორითმის შესაბამისად სტრიქონების ანუ ქრომოსომების გენერირება;
- ქრომოსომების დაწყვილება, რომლითაც მიიღება მშობელი სტრიქონების წყვილი;
- ცალკეულ ფრაგმენტში განთავსებული მშობელი ქრომოსომების წყვილების გაერთიანება, რომელიც მოგვცემს მშობელ პოპულაციას;
- მშობელ პოპულაციაში ზემოთაღწერილი გენეტიკური ალგორითმის ოპერაციების შესრულება;
- $[t-1, t]$ დროის ინტერვალისთვის ოპტიმალური მნიშვნელობის მიღება;
- პროცესი მეორდება ახალი ნაკადისთვის, რომელიც უკვე მიიღება დროის $[t, t+1]$ ინტერვალში (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 58).

დიდ მონაცემთა დამუშავების ლამბდა არქიტექტურისთვის შემუშავებული განაწილებული გენეტიკური ალგორითმის ბლოკ-სქემა მოცემულია ნახ.3.1.-ზე:



ნახ.3.2. განაწილებული გენეტიკური ალგორითმი(წყარო: ავტორი).

შემუშავებული მიდგომა მოიცავს დიდი მონაცემების რეალურ დროში დამუშავებაში არსებული პრობლემების იდენტიფიცირების და გადაჭრის გზებს. განხილულია რეალურ დროში დამუშავების და განაწილებული გამოთვლების მიდგომები, როგორცაა ლამბდა არქიტექტურა და განაწილებული გენეტიკური ალგორითმები (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 57).

განაწილებული გენეტიკური ალგორითმები (DGA) და ლამბდა არქიტექტურა არის ორი განსხვავებული კონცეფცია, რომელიც გამოიყენება მონაცემთა დამუშავებისა და ოპტიმიზაციის სხვადასხვა ასპექტში. თუმცა, მათი ერთობლივი გამოყენება დიდ უპირატესობას იძლევა და აერთიანებს ორივე კონცეფციას: განაწილებული გენეტიკური ალგორითმების შერწყმა Lambda Architecture-სთან იყენებს როგორც განაწილებული გამოთვლის, ისე ლამბდა არქიტექტურის უპირატესობებს, რაც ხელს უწყობს ოპტიმიზაციის პრობლემის ეფექტურ გადაჭრას. ამავდროულად უნდა აღინიშნოს, რომ ლამბდა არქიტექტურა გამართლებულია მხოლოდ დიდი მოცულობის მონაცემთა დამუშავებისთვის, რომლის მთავარი მიზანის დაყოვნებების მინიმუმამდე შემცირება (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 58).

3.2.3 ნაკადური მონაცემების განაწილებული დამუშავება გენეტიკური პროგრამირების გამოყენებით

ამოცანის აქტუალობის თვალსაზრისით შეიძლება განვიხილოთ ქალაქის ტრანსპორტის მონიტორინგის სისტემა. მონაცემები ავტომობილის სიჩქარის, ადგილმდებარეობის, გზების, სატრანსპორტო ნაკადების მდგომარეობის შესახებ უწყვეტად გადაეცემა ნაკადური დამუშავების სისტემას. დამუშავების ეტაპზე სისტემა აანალიზებს მონაცემებს, განსაზღვრავს მომატებული ტრაფიკის უბნებს და ავლენს მოძრაობაში ანომალიებს. ასეთი მონაცემების საფუძველზე სისტემა ავტომატურად ითვლის ოპტიმალურ მარშრუტს, აფრთხილებს ავარიულ სიტუაციებზე და მძღოლებს აძლევს რეკომენდაციებს საცობებისგან თავის ასარიდებლად. ამდენად, მოცემულ სცენარში ნაკადური დამუშავება სატრანსპორტო სიტუაციაზე მომენტალურად რეაგირების, მოძრაობის ოპტიმიზების და საგზაო მოძრაობის უსაფრთხოების ამაღლების საშუალებას იძლევა (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 99).

რეალურ დროში მონაცემთა დამუშავება მოითხოვს დაყოვნებათა მინიმიზებას და სისტემის მაღალ მწარმოებლურობის უზრუნველყოფას, რისთვისაც გამოიყენება შესაბამისი მეთოდები და მიდგომები, რომელთაგან მნიშვნელოვანია:

პარალელიზმი და განაწილება: განაწილებული სისტემების და პარალელური გამოთვლების გამოყენება უზრუნველყოფს დიდი მოცულობის მონაცემების ეფექტურად დამუშავებას;

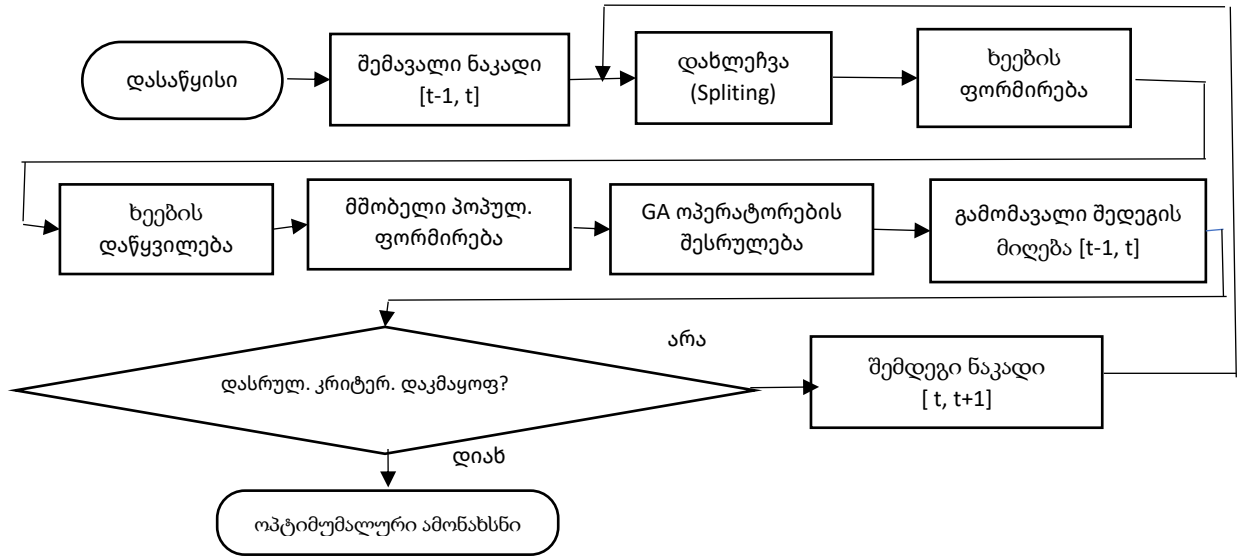
მონაცემთა კომპაქტურად წარმოდგენა: მონაცემთა წარმოდგენის ოპტიმიზება შეამცირებს მეხსიერებაზე და ქსელზე დატვირთვას, რაც თავის მხრივ შეამცირებს დაყოვნებებს (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 99).

ნაკადურ დამუშავებაში მონაცემთა დამუშავება და გარდაქმნა არის ბალანსის პოვნა აქტუალობას, სიზუსტესა და მწარმოებლურობას შორის. ეფექტური მეთოდების გამოყენებით გადაწყვეტა შეიძლება გახდეს გაცილებით რელევანტური და მნიშვნელოვანი მომხმარებლებისთვის. ამდენად, მიზანშეწონილად მიგვაჩნია რეალურ დროში მონაცემების ნაკადებში ოპტიმალური ამონახსნის მისაღებად განაწილებული გენეტიკური პროგრამირების გამოყენება (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 99).

შემთხვევითი ობიექტის შექმნა ასოცირდება შემთხვევითი ხის შექმნასთან. როგორც წესი, საკმარისია დაგენერირდეს სინტაქსურად კორექტული ხეები არსებული მონაცემებიდან გამომდინარე. ეს შეიძლება განხორციელდეს რეკურსიის გამოყენებით. იგი აგენერირებს ქვეხეებს, რომლებიც აბრუნებს განსაზღვრული ტიპის შედეგს. აქედან გამომდინარე პირობის გათვალისწინებით გენერირდება მემკვიდრეების მოთხოვნილი როდენობა. გენეტიკურ პროგრამებაში გადარჩევის იგივე მეთოდები გამოყენება, რაც გენეტიკურ ალგორითმებში. კროსოვერი გადარჩევის ხეებისთვის განისაზღვრება შემდეგი სახით - შემთხვევითი ტოტი პირველი არსებიდან ჩანაცვლება შემთხვევითი ტოტით მეორე არსებიდან. მუტაცია განისაზღვრება ხის შერჩეული ტოტის შემთხვევითი ჩანაცვლებით (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 100).

ჩვენს ამოცანაში ნაკადი შემოედინება უწყვეტი სახით, კერძოდ, თუ ნაკადის ერთი ნაწილის შემოსვლა მოხდა $[t-1, t]$ დროის ინტერვალში, მაშინ ნაკადის შემდეგი ნაწილი შემოვა $[t, t+1]$ ინტერვალში. მონაცემთა ნაკადებში ოპტიმალური გადაწყვეტის პოვნის მიზნით, რადგან მონაცემების მოცულობა ძალიან დიდია, საჭიროა ნაკადის დაჩეხვა (splitting) დროში შემოსვლის შესაბამის ფრაგმენტებად და ცალკეული ფრაგმენტისთვის ხის სტრუქტურების გენერირება. გენერირებული ხეებიდან ცალკეულ დანაწევრებულ ნაწილში აირჩევა ხეების წყვილი, რომელიც ჩაითვლება მშობელ წყვილებად, შემდეგ ეტაპზე მშობელი ხეების წყვილები გაერთიანდება და ქმნიან მშობელთა პოპულაციას, რომლის ცალკეულ წევრს აქვს ხისებრი სტრუქტურა. მშობელთა პოპულაციაში შესრულდება გენეტიკური ოპერაციები და მათ შორის აირჩევა საუკეთესო ხე, რომელიც შეესაბამება საუკეთესო ამონახსნს. ეს პროცესი განმეორდება შემომავალი ნაკადის ყველა ფრაგმენტისთვის. როგორც ზემოთ აღვნიშნეთ ნაკადი უწყვეტია, თუ საწყისი ფრაგმენტი შემოვიდა $[t-1, t]$ დროის ინტერვალში, შემდგომის შემოსვლის დრო იქნება, სავარაუდოდ, $[t, t+1]$ ინტერვალი. ალგორითმი მთავრდება, როდესაც დაკმაყოფილდება შეწყვეტის კრიტერიუმი, როგორც შეიძლება იყოს დამაკმაყოფილებელი გამოსავლის პოვნა ან წინასწარ განსაზღვრული გამოთვლითი დროის გადაჭარბება. ნახ.1-ზე წარმოდგენილია

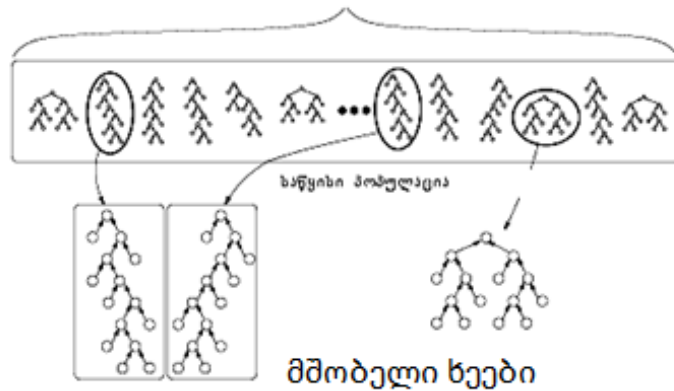
შემუშავებული ალგორითმის ბლოკ-სქემა (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 101).



ნახ.3.2. მონაცემთა ნაკადში ოპტიმალური ამონახსნის პოვნა(წყარო ავტორი).

საწყისი პოპულაციის ფორმირებისა და მშობელ ხეებად დაწყვილების სქემა ნაჩვენებია ნახ.2.-ზე

მონაცემთა ნაკადის ერთი ფრაგმენტი



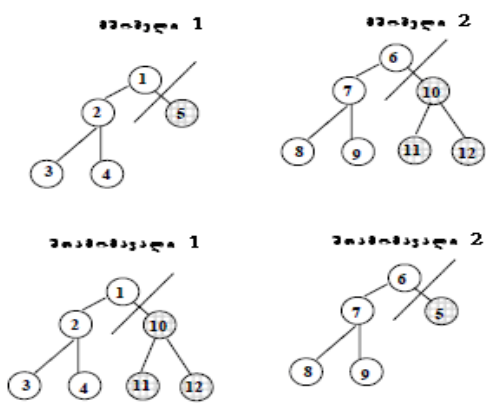
ნახ.3.3. მშობელი ხეების დაწყვილება (წყარო: ავტორი).

გენეტიკური პროგრამირების ოპერატორები, ისევე როგორც გენეტიკურ ალგორითმებში, არის შეჯვარება-კროსოვერი და მუტაცია. შემთხვევით-რანდომიზებულად ხდება კვეთის წერტილის არჩევა და ასევე რანდომიზებულად მუტაციის ან კროსოვერის ოპერაცია. ყოველი იტერაციის პროცესში, როგორც შეჯვარების, ისე მუტაციის ოპერაციის დროს დამუშავებულ ალგორითმში კომბინაციათა გადარჩევის მიზნით სრულდება შიდა ციკლური პროცესი, როდესაც ნაცვლად რანდომიზებული მეთოდისა, კვეთის წერტილი განისაზღვრება

ვარიანტების წინასწარი გადარჩევის შედეგად. აღნიშნული მიდგომა ერთი მხრივ, ზრდის გამოთვლების რაოდენობას მოცემულ იტერაციაში, მაგრამ მეორე მხრივ, ამცირებს იტერაციების რაოდენობას (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 102).

საწყისი პოპულაციიდან მიღებული მშობლების პოპულაცია წარმოადგენს ხეების სტრუქტურების სიმრავლეს. ხდება ყოველი ხის ღირებულებათა ანგარიში, შემდეგ ხეების რანჟირება. შემდეგ ეტაპზე უნდა განისაზღვროს მომავალი სტრატეგია: ან ამოირჩეს ორი საუკეთესო ხე, ე.წ. მშობლები, მათი შემდგომი შეჯვარების ოპერაციისათვის, ან შეირჩეს მხოლოდ ერთი საუკეთესო ხე მუტაციის ოპერაციისათვის (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 102).

ჩვენი ამოცანის შემთხვევაში გამოვიყენეთ კვეთის წერტილის ძებნისადმი განსხვავებული მიდგომა კერძოდ, ე.წ. „წონითი კოეფიციენტების“ გამოყენებით. შერჩეული საუკეთესო ხისათვის (ან ხეებისთვის) ხდება ცალკეული ოპერაციული ბლოკისთვის „წონითი კოეფიციენტების“ გამოთვლა და ყველაზე „მძიმე“ ბლოკის ანუ ქვე-ხის გამოვლინება, რომლის მიხედვითაც განისაზღვრება კვეთის წერტილი, რასაც მოჰყვება შეჯვარების ოპერაცია (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 102).

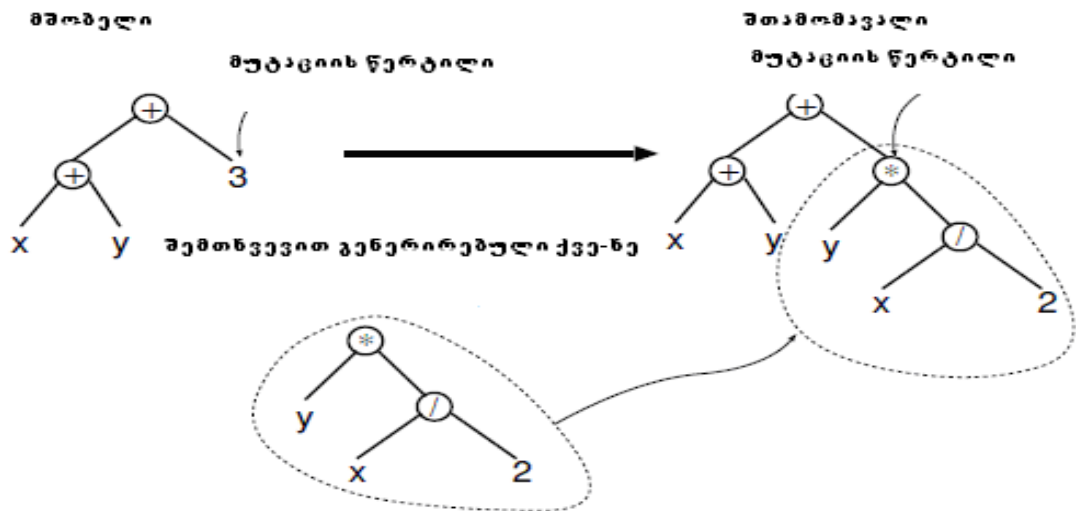


მშობელ ხეზე მუტაციის ოპერაციის შესრულების შემთხვევაში შერჩეული საუკეთესო ხისათვის ხდება „წონითი კოეფიციენტების“ გამოთვლა ანუ ყველაზე „მძიმე“ ტოტის გამოვლინება და საგანგებოდ რანდომიზებულად გენერირებული უფრო „მსუბუქი“ ტოტით ჩანაცვლება (წითაშვილი, ჯანელიძე და ხაჩიძე

2023, 103).

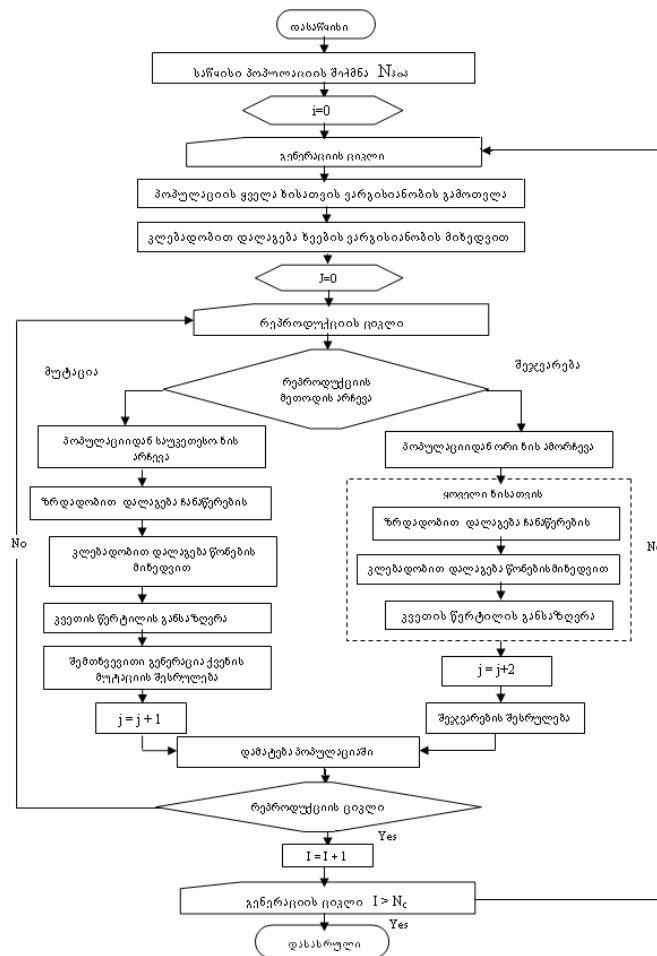
ნახ.3.4.-ზე მოცემულია ხეებში კვეთის წერტილის პოვნისა და შეჯვარების ოპერაციის სქემა (წყარო: ავტორი).

მუტაციის სტრატეგიის შემთხვევაში საუკეთესო ხისთვის რანდომიზებულად ხდება რაიმე ნაწილის მოკვეთა და მისი ჩანაცვლება სხვა რანდომიზებულად გენერირებული ფრაგმენტით. ციკლი განმეორდება, ვიდრე ძებნის პროცესი ჩაიცვლება ოპტიმუმის გარშემო (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 102).



ნახ.3.5. მუტაციის ოპერაცია

წარმოდგენილი მეთოდის მიხედვით, თითოეულ ხეში შეფასდება ცალკეული კვანძის წონითი კოეფიციენტი. ყოველი ხისთვის თითოეულ კვანძში გამოითვლება მონაცემების რაოდენობა. ნახ.5.-ზე მოცემულია ალგორითმი „წონითი კოეფიციენტების“ გამოყენებით (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 103).



ნახ.3.6. ალგორითმი „წონითი კოეფიციენტების“ გამოყენებით (წყარო: ავტორი)

წარმოდგენილი მიდგომა მნიშვნელოვნად შეამცირებს გამოთვლითი ოპერაციების რაოდენობას და შესაბამისად, გაზრდის სისტემის სწრაფქმედებას (წითაშვილი, ჯანელიძე და ხაჩიძე 2023, 103).

3.3. ცოდნის ფრაგმენტის აგება

თანამედროვე ხელოვნური ინტელექტის სისტემებში, ცოდნა ხშირად აღიქმება არა როგორც ერთიანი, მონოლითური ბლოკები, არამედ როგორც პატარ-პატარა ერთეულები, ცოდნის ფრაგმენტების ერთობლიობა. ცოდნის ფრაგმენტი გულისხმობს ცოდნის ცალკეულ ერთეულს ერთ ფაქტს, კონცეფციას, წესს ან ინტუიციურ იდეას, რომელსაც შეიძლება დაემატოს ან დაუკავშირდეს სხვა ფრაგმენტები და შექმნას ფართო გაგება. ტერმინი „ცოდნის ფრაგმენტის აგება“ ხაზს უსვამს იმას, თუ როგორ კონსტრუირდება და გამოიყენება ეს ერთეულები სხვადასხვა დისციპლინაში ცოდნის მენეჯმენტის სფეროში, ფრაგმენტები შეიძლება წარმოადგენდეს ორგანიზაციული გამოცდილების მცირე ნაწილაკებს ან გაკვეთილებს, რომლებიც იქნა მიღებული პრაქტიკიდან; სემანტიკური ვების (Semantic Web) კონტექსტში, ისინი ხშირად შეესაბამება ატომურ განცხადებებს მაგალითად, RDF ტრიპლებს ცოდნის გრაფში; კოგნიტურ მეცნიერებაში, ფრაგმენტები შეიძლება წარმოადგენდეს ინდივიდის გონებრივი მოდელის ელემენტარულ ნაწილებს.

3.3.1 ცოდნის ფრაგმენტები ცოდნის მენეჯმენტში (Knowledge Fragments in Knowledge Management)

ცოდნის მენეჯმენტის (Knowledge Management – KM) კონტექსტში, ცოდნის ფრაგმენტი წარმოადგენს მცირე, თვითკმარ ცოდნის ერთეულს, შესაძლებელია მისი განსზღვრა, დოკუმენტირება და განმეორებითი გამოყენება. ფრაგმენტს შეიძლება ჰქონდეს სხვადასხვა ფორმა, მაგალითად საუკეთესო პრაქტიკის მაგალითი, ბიზნეს წესი, ექსპერტის ინტუიციური ხედვა, ან წარსული გამოცდილებიდან მიღებული ცოდნა. თანამედროვე ორგანიზაციებში ხშირად ვხვდებით სიტუაციას, როდესაც კრიტიკული ცოდნა ფრაგმენტულადაა გადანაწილებული, გაფანტულია დოკუმენტებში, პლატფორმებზე ან კონკრეტული თანამშრომლების გონებაში. შესაბამისად, ცოდნის მენეჯმენტის ერთ-ერთი მნიშვნელოვანი ამოცანაა ამ ფრაგმენტების გამორჩევა სხვადასხვა წყაროდან, მათი ექსპლიცირება (ღიად

განსაზღვრა) და გაზიარების შესაძლებლობის შექმნა. მაგალითად სავსე მომსახურების ინჟინრის მიერ აღმოჩენილი ეფექტური ხერხი კონკრეტული გაუმართაობის აღმოსაფხვრელად ან პროექტის დასრულების შემდგომი ანალიზიდან მიღებული ძირითადი უკუკავშირი. ორივე წარმოადგენს **ცოდნის ფრაგმენტს**. ამგვარი ფრაგმენტების დაჭერა და დოკუმენტირება განსაკუთრებით მნიშვნელოვანია იმ პროცესში, როცა არაფორმალური ცოდნა გარდაიქმნება ფორმალურ და გაზიარებად ორგანიზაციულ ცოდნად. (ჩენი და იუან 2024, 548).

ცოდნის მენეჯმენტის არაერთი თეორიული ჩარჩო ცოდნის ფრაგმენტების აგებასა და გამოყენებას განიხილავს, როგორც ორგანიზაციული სწავლისა და ინოვაციის მნიშვნელოვან მექანიზმს. ერთ-ერთი ყველაზე გავლენიანი მოდელია ნონაკას (Nonaka 1995) ორგანიზაციული ცოდნის შექმნის თეორია ე.წ. SECI მოდელი, რომლის მიხედვითაც ცოდნა გარდაიქმნება ოთხი დინამიკური ფაზის გზით: Socialization, Externalization, Combination და Internalization. Externalization-ის ფაზაში, ინდივიდები არაფორმალურ ცოდნას გადმოსცემენ სიტყვიერად ან ვიზუალურად იდეების, მოდელების ან დიაგრამების სახით. სწორედ ამ პროცესში ხდება ცოდნის ფრაგმენტებად ჩამოყალიბება. ცალკეული იდეები, რომელთა გაგება და გაზიარება შესაძლებელია სხვა ადამიანებისთვის. სხვა მნიშვნელოვანი მიდგომაა ცოდნის ნიშნების (Patterns) გამოყენება მისი კოდიფიკაციისთვის. როგორც May და Taylor (2003) აღნიშნავენ, პრაქტიკოსებს შეუძლიათ არაფორმალური ცოდნის ფრაგმენტი გარდაქმნან სტრუქტურირებულ ნიმუშად, რომელიც მოიცავს **დასახელებას, კონტექსტს, პრობლემის აღწერას, ამოხსნის გზას და შედეგებს**.

როდესაც ცოდნის ფრაგმენტი გამოიხატება როგორც ნიმუში იგი გარდაიქმნება ტრანსფერულ ცოდნის ერთეულად, რომელსაც აქვს განსაზღვრული გამოყენების არეალი და მნიშვნელობა. ეს გაცილებით მეტია, ვიდრე უბრალო რჩევა ან წესი - იგი წარმოადგენს მინი-მოდელს ან მინი-თეორიას, რომელსაც სხვებიც მარტივად იყენებენ. ნიმუშზე დაფუძნებული მიდგომა ნათლად აჩვენებს, რომ ცოდნის ფრაგმენტის აგება მოიცავს მის სტრუქტურირებასა და მნიშვნელობის მინიჭებას, რათა იგი შესაძლებელი გახდეს სხვა ფრაგმენტებთან დაკავშირებისთვის და სისტემური ცოდნის აგებისთვის (ლინ, et al. 2003, 1724).

ცოდნის მენეჯმენტის (KM) სხვა თეორიული ჩარჩოები ყურადღებას ამახვილებენ ფრაგმენტების ერთმანეთთან დაკავშირებასა და ქსელების ფორმირებაზე. მაგალითად, ევოლუციური ქსელის მოდელი (Internal Knowledge Evolution Network – IKEN), სადაც თითოეული ფრაგმენტი იქნება ეს იდეა, მონახაზი თუ კომენტარი წარმოდგენილია კვანძად (node) მიმართულ ქსელში, რომელიც ასახავს, თუ როგორ განვითარდა ცოდნა პროექტის განმავლობაში (ლინ, და სხვ. 2003). ამ მოდელში, კვლევითი ჯგუფის მიერ მიღებული ყოველი შუალედური შედეგი ან აღმოჩენა, როგორც ცოდნის ფრაგმენტი. ინახება „ქეისის“ სახით და შემდგომ შესაძლებელია მისი მოძიება. ამავდროულად, თითოეულ ფრაგმენტს შორის ინახება სპეციფიკური კავშირები, რომლებიც ასახავს, როგორ გადაიზარდა ერთი ცოდნა მეორეში. IKEN ასახავს ქეისებზე დაფუძნებული დასკვნის (case-based reasoning) პერსპექტივას ცოდნის მენეჯმენტში. ცოდნის ფრაგმენტები აღიწერება კონტექსტუალური კავშირებით, რაც საშუალებას აძლევს მომხმარებლებს არამხოლოდ ცალკეული ფაქტები ამოიღონ, არამედ მათი მიღმა არსებული განზრახვისა და გამოცდილების ლოგიკური ჯაჭვის ანალიზიც შეუძლია. ასეთი ჩარჩოები ხელს უწყობს ცოდნის ფრაგმენტების აგებას იმით, რომ უზრუნველყოფს თითოეული ფრაგმენტის მკაფიოდ განსაზღვრულობას და პარალელურად მათ დაკავშირებას ფართო ცოდნის შექმნის ნარატივთან (ლინ, და სხვ. 2003, 1726).

3.3.2 ცოდნის ფრაგმენტები სემანტიკურ ქსელში (Knowledge Fragments in the Semantic Web)

სემანტიკური ქსელის (Semantic Web) კონტექსტში, ინფორმაცია თავიდანვე წარმოდგენილია ცოდნის ფრაგმენტებად, ეს არის მისი არქიტექტურული თავისებურება. სემანტიკური ქსელის საფუძველს წარმოადგენს მოდელი სახელწოდებით RDF (Resource Description Framework), რომელიც ცოდნას წარმოადგენს როგორც ტრიპლებს, სამი ელემენტისგან შემდგარ განაცხადს სუბიექტი-პრედიკატი-ობიექტი. თითოეული RDF ტრიპლი შეიძლება ჩაითვალოს მინიმალურ ცოდნის ფრაგმენტად - ის ქმნის კავშირს ორ კონცეპტს შორის. მაგალითად: *Paris – isCapitalOf – France* წარმოადგენს ატომურ ცოდნას, რომელიც აღწერს ფაქტობრივ კავშირს. ეს ატომური ფრაგმენტები წარმოადგენს ცოდნის

გრაფებისა და ონტოლოგიების საძირკველს. ფუნდამენტურად, სემანტიკური ვები არის გიგანტური, განაწილებული კოლექცია ისეთი ფრაგმენტებისა, რომელთაგან თითოეული გამოქვეყნებულია რომელიღაც URI-ზე და დაკავშირებულია სხვა ფრაგმენტებთან ლინკების მეშვეობით. ამ კონტექსტში, ტერმინი „ცოდნის ფრაგმენტი“ ხშირად აღნიშნავს პატარა ქვეგრაფს ან ინდივიდუალურ ფაქტს უფრო დიდი ცოდნის ბაზის ფარგლებში. მაგალითად, ერთი პიროვნების შესახებ აღწერილი RDF ტრიპლების ნაკრები შეიძლება ჩაითვალოს DBpedia-ს (სემანტიკური ცოდნის ბაზის) ფრაგმენტად. ცოდნის ფრაგმენტებად დაყოფით, სემანტიკური ვები უზრუნველყოფს ცოდნის ნაწილობრივი გადამუშავების შესაძლებლობას - აპლიკაციას შეუძლია მხოლოდ იმ ნაწილების გამოყენება, რაც კონკრეტული ამოცანისთვის სჭირდება, ყოველგვარი სრული ბაზის მოძიების გარეშე. ზოგიერთი მიდგომა პირდაპირ ემყარება ე.წ. Linked Data Fragments-ს. ეს არის მონაცემთა ნაკრებების ქვესეტები, მაგალითად ყველა ტრიპლი გარკვეული თემის შესახებ, ან ისეთი ტრიპლები, რომლებიც პასუხობს კონკრეტულ მოთხოვნის შაბლონს. ასეთ შემთხვევაში, მომხმარებელი არ მიმართავს მონოლითურ ბაზას, არამედ შეაგროვებს პასუხს ცალკეული ფრაგმენტების გაერთიანებით.

სემანტიკური ქსელი დაფუძნებულია ცოდნის წარმოდგენის (Knowledge Representation – KR) ისეთ ჩარჩოებზე, რომლებიც მხარს უჭერს როგორც ცოდნის ფრაგმენტების აგებას, ისე მათ აგრეგაციასა და კავშირს. RDF და მისი სქემა RDFS იძლევა შესაძლებლობას ნებისმიერმა შექმნას ახალი ტრიპლეთი, რაც ფაქტობრივად ნიშნავს ახალი ცოდნის ფრაგმენტების აგებას, რომლებიც ავსებენ მონაცემთა ქსელს. ონტოლოგიური ენები, როგორცაა OWL (Web Ontology Language), შესაძლებლობას აძლევს დიზაინერებს განსაზღვრონ ლოგიკური ურთიერთობები ფრაგმენტებს შორის. მაგალითად, კლასების იერარქიები, შეზღუდვები, რითაც შესაძლებელი ხდება კომბინირებულ ცოდნის საფუძველზე დასკვნის გამოტანა (*reasoning*). მნიშვნელოვან კონცეპტუალურ ჩარჩოს გვთავაზობს EDXML მოდელი. ტრადიციულ KR სისტემებში ჩვენ ვქმნით ერთიან ცოდნის ბაზას. მაგალითად, ჩარჩოების კოლექციას ან სემანტიკურ ქსელს, მონაცემებზე ან ექსპერტის ცოდნაზე დაყრდნობით. EDXML-ის მიდგომის შემთხვევაში, ნედლი მონაცემი წარმოადგენდეს მოვლენებს, რომლებიც

„წარმოქმნის ცოდნის ნაწილაკებს“. ეს ფრაგმენტები შემდგომში შეიძლება გაერთიანდეს და ჩამოყალიბდეს სრულფასოვან ცოდნის ბაზად. ეს მიდგომა შეესაბამება სემანტიკური ქსელის ძირითად პრინციპს, დაფიქსირებული მონაცემი ატარებს ფრაგმენტულ მნიშვნელობას, და მხოლოდ აგრეგაციის შედეგად იქმნება მდიდარი, სტრუქტურირებული ცოდნა.

კიდევ ერთი მნიშვნელოვანი ჩარჩოა **Linked Open Data** ინიციატივა, რომელიც ყოველ გამოქვეყნებულ მონაცემს, ტრიპლეტს ან ტრიპლეტების მცირე ჯგუფს განიხილავს ცოდნის ფრაგმენტად. ამ ფრაგმენტების ერთმანეთთან დაკავშირებით ვებსივრცეში იქმნება გლობალური ცოდნის გრაფი. გარდა ამისა, სემანტიკური ქსელის კვლევაში შემოღებულია ცნება **ონტოლოგიის მოდულები (ontology modules)** - ესენია ონტოლოგიის ქვეჯგუფები, რომლებიც დამოუკიდებლად სრულყოფილია კონკრეტული დომენის ან ამოცანისათვის. ონტოლოგიური მოდულების გამოყოფისა და გაერთიანების შესაძლებლობა მხარს უჭერს ფორმალური ჩარჩოები, როგორცაა კონსერვატიული გაფართოებები (conservative extensions) აღწერით ლოგიკაში (თეიკენ 2022).

ბოლო წლების განმავლობაში, სემანტიკური ქსელის კონცეფცია, ცოდნის ფრაგმენტები – მნიშვნელოვანი როლი შეიძინა თანამედროვე ტექნოლოგიებში. ერთ-ერთი მნიშვნელოვანი ტენდენციაა ცოდნის გრაფების ინტეგრაცია ნეირონულ მოდელებსა და დიდ ენობრივ მოდელებთან (LLMs). მაგალითად, RAG (Retrieval-Augmented Generation) სისტემები, რომლებიც 2020-იანი წლების ინოვაციას წარმოადგენს, იყენებენ გარე ცოდნის ფრაგმენტებს, რათა გააზრებულად ჩამოაყალიბონ დიდი ენობრივი მოდელის შედეგები. კიდევ ერთი მნიშვნელოვანი განვითარებაა სტრიმინგისა და რეალურ დროში განახლებადი ცოდნის გრაფების სფეროში. ამ შემთხვევაში, ცოდნის ფრაგმენტები მუდმივად ემატება სისტემას IoT მოწყობილობებიდან ან სოციალურ მედიიდან. ამ კონტექსტში გამოწვევაა ცოდნის ბაზის განახლება ფრაგმენტების დონეზე, სისტემის შეჩერების გარეშე. ამ პრობლემის გადასაჭრელად გამოიყენება ინკრემენტული დასკვნის ალგორითმები (incremental reasoning algorithms), რომლებიც უზრუნველყოფენ რეალურ დროში ცვლილებების ინტეგრირებას (შრამი, ვენერი and შმიდტი 2024, 5).

3.3.3 ცოდნის ფრაგმენტები კოგნიტურ მეცნიერებაში

კოგნიტურ მეცნიერებაში, ცოდნის ფრაგმენტი განისაზღვრება როგორც ინდივიდის ცოდნის მცირე ერთეული ან კომპონენტი, მაგალითად, პრიმიტიული კონცეფცია, განცალკევებული რწმენა, ინტუიციური წესი ან მენტალური წარმოდგენა იმის შესახებ, თუ როგორ მუშაობს სამყარო. ადამიანური ცოდნა, განსაკუთრებით დამწყებთათვის ან ყოველდღიურ აზროვნებაში ხშირად არ არის სრულად ინტეგრირებული. პირიქით, ადამიანები ფლობენ მრავალი ფრაგმენტულ იდეას, რასაც A. A. diSessa მოიხსენიებს როგორც „ცოდნა ნაწილებად“ (*knowledge-in-pieces*) (ა. დისესა., სმიტ III, ჯონ პ.,როშელი, ჯ. 1993, 116). მაგალითად, ბავშვის მიერ მოძრაობის გაგება შეიძლება შედგებოდეს რამდენიმე არაკავშირული ფრაგმენტისგან „*მოძრაობისთვის ძალაა საჭირო,*“ „*მძიმე საგნები უფრო სწრაფად ვარდებიან.*“ ეს იდეები შეიძლება გამოიყენებოდეს არათანმიმდევრულად და შეუსაბამოდ სხვადასხვა კონტექსტში, რადგან ისინი არ ქმნიან ერთიან ფიზიკურ თეორიას, არამედ არსებობს როგორც დამოუკიდებელი და ხშირად კონფლიქტური ნაწილები. კოგნიტური მეცნიერები სწავლობენ იმას, როგორ იძენს, ინახავს და აქტიურებს ადამიანი ასეთ ფრაგმენტებს. მიუხედავად იმისა, რომ ექსპერტებს გააჩნიათ უფრო ინტეგრირებული ცოდნის ჩარჩოები, ხშირად ისინიც კი ეყრდნობიან მრავალ მცირე ცოდნის ერთეულს, როგორცაა კონკრეტული ფაქტები, სქემები ან პროცედურები, რომლებიც შეიძლება ჩაითვალოს ფრაგმენტებად მათი ფართო კომპეტენციის შიგნით. ამგვარად, „ცოდნის ფრაგმენტის აგება“ კოგნიტურ გაგებაში აღნიშნავს იმ პროცესს, რომლის დროსაც ინდივიდი ქმნის ახალ კონცეპტს ან ცოდნის ერთეულს საკუთარ გონებაში, ხშირად მიღებული გამოცდილებიდან აბსტრაქციის გზით ან სწავლის მცირე ფრაგმენტებით (*bite-sized increments*).

ცოდნის ფრაგმენტებთან დაკავშირებით ერთ-ერთი ძირითადი თეორიული ჩარჩოა diSessa-ს "**Knowledge-in-Pieces**" (KiP) თეორია (ა. დისესა., სმიტ III, ჯონ პ.,როშელი, ჯ. 1993). აღნიშნული თეორია ამტკიცებს, რომ მოსწავლეების ინტუიციური ცოდნა შედგება არაერთი ნახევრად-დამოუკიდებელი ელემენტისგან, რომლებიც ცნობილია როგორც ფენომენოლოგიური პრიმიტივები (*phenomenological primitives*) ან *p-prims*. ეს ფრაგმენტები წარმოადგენს მარტივ ინტუიციურ იდეებს,

რომლებიც ადამიანმა ყოველდღიური გამოცდილებიდან აბსტრაქტულად განივითარა. მაგალითად, ერთ-ერთი p-prim შეიძლება იყოს ფრაზა: „უფრო მეტი მიზეზი იწვევს უფრო მეტ შედეგს“, რომელსაც მოსწავლე ფართო და ზოგად კონტექსტებში შეიძლება განავრცოს, დაწყებული ობიექტის ბიძგით, დამთავრებული სწავლაში ძალისხმევის მნიშვნელობით. KiP თეორიის მიხედვით, ეს ფრაგმენტები კონტექსტზე დამოკიდებულია და მხოლოდ შესაბამის სიტუაციებში აქტიურდება. თუმცა, ისინი არ არიან ავტომატურად კოორდინირებული ერთმანეთთან, მათ შორის არ არსებობს აუცილებელი ლოგიკური კავშირი. როგორც (ვოსნიადუ 2019, 2) აღნიშნავს, diSessa(ა.დისესა)-ს კვლევებმა ცხადყო, რომ მოსწავლეებს ხშირად არ გააჩნიათ ერთი თანმიმდევრული მცდარი წარმოდგენა, არამედ მათი გაგება შიდა წინააღმდეგობრივია და ფრაგმენტულად აგებული. მათი ახსნები სხვადასხვა კონტექსტში ეფუძნება სწორედ ამ ფრაგმენტულ ინტუიციებს (p-prim), რაც ქმნის მოზაიკურ, არათანმიმდევრულ ცოდნას. ეს ჩარჩო ამყარებს იმ იდეას, რომ ცოდნის აგება წარმოადგენს ფრაგმენტების შეკრებასა და ხელახალ ორგანიზაციას და არა მთლიანად ერთი თეორიის ჩანაცვლებას მეორით. შესაბამისად, სწავლის პროცესი გულისხმობს: არსებულ ფრაგმენტებზე მორგებას (tuning), ახლების დამატებას, და მათი თანდათანობით ჩამოყალიბებას მეტად სისტემურ, სტრუქტურირებულ ჩარჩოებად.

3.3.4 ცოდნის ინჟინერიის ახალი ასპექტი

თანამედროვე კომპიუტერული მოდელები და შესაბამისი პროგრამული უზრუნველყოფა უკვე აღარ აკმაყოფილებს ხელოვნური ინტელექტის სისტემის სრულფასოვანი რეალიზაციის მოთხოვნებს. ინფორმაციის წარმოდგენის თანამედროვე ფორმებიც კი ჯერ კიდევ ვერ იძლევა საშუალებას კომპლექსური ტექსტუალური ინფორმაციის სემანტიკური მოდელირებისათვის. კლასტერიზაციისა თუ კლასიფიკაციის ამოცანების გადაჭრისას, მანქანური სწავლების მოდელები მონაცემთა ერთობლიობას ცოდნად აღიქვამენ. ამასთან, ჩვენ კვლავ ვმუშაობთ ენაზე დაყრდნობით - სიმბოლოების, ცალკეული სიტყვების ან ლიტერალების (ტექსტუალური ელემენტების) დონეზე, რაც განაპირობებს იმას, რომ მანქანური თარგმნის მეთოდები ჯერ კიდევ არასრულყოფილია. კომპიუტერულ სისტემებს ჯერ

არ შეუძლიათ სრულყოფილად და ადეკვატურად თარგმნონ ერთი ენიდან მეორეზე. ცოდნის წარმოდგენის არსებული მოდელები მხოლოდ first-order მიდგომა ბუნებრივთან მიმართებით. ჩვენი მიდგომის არსი მდგომარეობს იმაში, რომ ინფორმაცია უნდა განვიხილოთ არამატერიალურ ჭრილში, კვანტურ ფორმაში, უწყვეტი ჰოლოგრამის სახით. ის, რასაც ჩვენ ვუწოდებთ „ინფორმაციას“, რეალურად წარმოადგენს მხოლოდ მისი მატერიალური ანარეკლს. ტექსტუალური ინფორმაციის დამუშავებისას ჩვენ ვმოქმედებთ სიმბოლოებით, სიტყვებით ან ლიტერალების სახით, მაშინ როდესაც ინფორმაციის მატარებელს თვითონ არ გააჩნია ცოდნა. სწორედ ამიტომ, ეს თავი ეძღვნება ცოდნის წარმოდგენისა და ათვისების ახალ გაგებას, რომელიც შეიძლება გახდეს არა მხოლოდ მარტივი ტექსტუალური ინფორმაციის, არამედ მაღალი დონის ცოდნისა და ცოდნის მოდელების დამუშავების საფუძველი და ამგვარად, გახდეს ხელოვნური ინტელექტის ახალი პარადიგმის დასაწყისი (ხაჩიძე, ჯანელიძე და მეფარიშვილი 2024).

პრობლემის აღწერა: ადამიანის არსებობა წარმოუდგენელია გარემოსთან ინფორმაციის გაცვლის პროცესის გარეშე. ჩვენ არ შეგვიძლია ინფორმაციის უშუალო აღქმა; ჩვენ აღვიქვამთ მხოლოდ მისი წარმოდგენის მატერიალურ ფორმას, რომელსაც ადამიანი ინტერპრეტაციას უკეთებს მხოლოდ მას შემდეგ, რაც იგი თავის ტვინის მიერ დამუშავდება და გააზრდება. ამავდროულად, უნდა ითქვას, რომ ინფორმაციის მატარებელს თვითონ არ გააჩნია ცოდნა. მაგალითად, მათემატიკის ნებისმიერი სახელმძღვანელო ვერ გადაწყვეტს ყველაზე ელემენტარულ ამოცანასაც კი. ან დავფიქრდეთ, როგორ შეიძლება აღვწეროთ ის ინფორმაცია, რასაც რეცეპტორები იღებენ სუნისა ან გემოს შესახებ? ეს, ცხადია, შესაძლებელია მხოლოდ გრაფის სახით, რომლის სტრუქტურაც შეესაბამება კონკრეტული ნივთიერების მოლეკულურ აგებულებას. აღსანიშნავია, რომ მანქანური თარგმანის არსებული მიდგომები კვლავ შორს არის სრულყოფილებისგან, კომპიუტერულ სისტემებს ჯერაც არ შეუძლიათ ერთ ენაზე არსებული ტექსტის ადეკვატურად თარგმნა მეორეზე. დღემდე, ინფორმაციის საძიებო სისტემებში გამოიყენება მხოლოდ შეზღუდული ბუნებრივი ენა, რაც ნაწილობრივ განპირობებულია იმით, რომ ჩვენ ვმუშაობთ სიმბოლოებით, ცალკეული სიტყვებით ან ლიტერალების სხვადასხვა ფორმით. მანქანური სწავლება,

კლასიფიკაციის ან კლასტერიზაციის ამოცანების გადაჭრისას, მხოლოდ მონაცემთა ერთობლიობას განიხილავს როგორც ცოდნას (ხაჩიძე, ჯანელიძე და მეფარიშვილი 2024, 62). ერთი მთავარი მიზეზი ამისა არის ის, რომ თანამედროვე კომპიუტერები არსებითად განსხვავდებიან ადამიანის ტვინისაგან. ცოდნის წარმოდგენის არსებული მოდელები მაგალითად, სემანტიკური ქსელები, კადრების სისტემები და სხვა, მხოლოდ პირველადი დონის მიახლოებაა ბუნებრივ ცოდნასთან.

ინფორმაციის ასპექტები: ინფორმაციის ცნება მოიცავს მნიშვნელობის რამდენიმე ტიპს, როგორცაა მონაცემები, ცოდნა (სხვადასხვა დონეზე), ინტელექტი, შეგნება და სხვ. შესაბამისად, აღნიშნულ კომპონენტებზე მსჯელობისას, თანამედროვე ინფორმაციული ტექნოლოგიები ინფორმაციის აღსაწერად იყენებენ ისეთ ცნებებს, როგორცაა: ეპისტემოლოგია, ონტოლოგია, ტაქსონომია, ენტროპია და სხვა. ცოდნის თეორიაში ტერმინი ონტოლოგია კონკრეტულად გამოიყენება ცოდნის აღწერის მოდელად. ანუ, როგორც ფორმალური სტრუქტურა, რომელიც ასახავს ცნებებსა და მათ შორის ურთიერთობებს კონკრეტულ დისციპლინაში ან საგნობრივ დომენში. რაც შეეხება ტაქსონომიას, იგი წარმოადგენს მეცნიერებას, რომელიც უკავშირდება ორგანიზმების დასახელებას, აღწერასა და კლასიფიკაციას. თუმცა ინფორმაციულ ტექნოლოგიებში ამ ცნების გამოყენება ხშირად ვრცელდება ცოდნის სტრუქტურირებასა და ორგანიზებაზე კლასების, ქვეკლასების და იერარქიების სახით.

ზოგადად, ინფორმაციის იერარქიული გრადაცია, როგორც ჩამოყალიბებული ჰიპერგრადის სისტემა (ანალოგიურად "nested dolls" პრინციპისა), დაკავშირებულია მის სირთულესთან, რაც განსაზღვრავს ინფორმაციული განზომილების ხარისხს და მოდელირებაში კომპონენტთა მრავალფეროვნებას. მეორეს მხრივ, გეომეტრიული თვალსაზრისით, ინფორმაციის განხილვა შესაძლებელია როგორც შინაარსის არსებობა სივრცულ ფორმაში (მველბერძნულიდან "in form" ნიშნავს - *რაც არის ფორმაში*). შესაბამისად, სემანტიკური ინფორმაციის ნებისმიერი დონე შეიძლება ასოცირდეს გეომეტრიულად რთულ ობიექტებთან ან ჰიპერგრადებთან, როგორცაა კომპლექსური ორგანული მოლეკულები, ნეირონული ქსელები ან მათი ჯგუფები (ენსამბლები) და სხვა. (ხაჩიძე, ჯანელიძე და მეფარიშვილი 2024, 62).

ინფორმაციის ცნება უშუალოდ დაკავშირებულია ენტროპიის ცნებასთან. ბუნებაში ნებისმიერი მოვლენა ხასიათდება ენტროპიის ცვლილებით, ზრდით ან კლებით, რაც საფუძვლად დაედო მრავალი თეორიული მოდელის განვითარებას, მათ შორის შენონის ინფორმაციის თეორიას, სინერგიას (synergy) და კომპლექსურობის (სირთულის) თეორიას. მეორეს მხრივ, ევოლუცია ინფორმაციის ზრდის პროცესია, რაც თერმოდინამიკის დონეზე განიხილება როგორც ენტროპიის საწინააღმდეგო ტენდენცია, რომელიც ასოცირდება სინერგიასთან, ხოლო ცნობიერების დონეზე იგი მოიცავს სინტაქტურ, სემანტიკურ და პრაგმატულ განზომილებებს.

ინფორმაციის კვანტური ხედვა: ჩვენი მიდგომის ძირითადი კვინტესენცია მდგომარეობს იმაში, რომ ინფორმაცია უნდა განვიხილოთ არამატერიალურ სფეროში, კვანტურ ფორმაში, უწყვეტი ჰოლოგრამის სახით. ის, რასაც ჩვენ ვუწოდებთ „ინფორმაციას“, წარმოადგენს მხოლოდ მისი მატერიალური ანარეკლს. კვანტურმა ფიზიკამ მეცნიერებაში შემოიტანა ინფორმაციის ახალი გაგება, კერძოდ კვანტური გადახლართულობა (quantum entanglement) კონცეპტი, რომელიც აღწერს არაკლასიკურ კორელაციებს კომპოზიტური კვანტური სისტემის ელემენტებს შორის. ევრინ შროდინგერმა შემოიღო ტერმინი „გადახლართვა“ და მას უწოდა „კვანტური მექანიკის დამახასიათებელი ნიშანი“. (ხაჩიძე, ჯანელიძე და მეფარიშვილი 2024, 63). ყველაზე მარტივი „გადახლართული“ მდგომარეობები, რომელთა განხილვაც შესაძლებელია, არის ორ-დონიანი კვანტური სისტემების (ქიუბიტების) წყვილთა გადახლართული მდგომარეობები, რომელთაგან თითოეულს გააჩნია საწყისი მდგომარეობები - $|0\rangle$ და $|1\rangle$. კვანტური სისტემის ევოლუცია აღწერილია შრედინგერის ცნობილ განტოლებით, რომელიც წარმოადგენს კვანტური მექანიკის ფუნდამენტურ დინამიკურ კანონზომიერებას.

დეკოჰერენციის (decoherence) არსი მდგომარეობს იმაში, რომ კვანტური სისტემა გადაყვანილ იქნას გარე დამკვირვებლისთვის კლასიკურ მდგომარეობაში, ანუ ისეთი ფორმით წარმოაჩინოს თავი, რომელიც კლასიკურ ლოგიკას შეესაბამება. სამყარო არსებითად დინამიკურია (ფენომენოლოგიური თვალსაზრისითაც) და ბიპოლარულია კვანტური დუალიზმის ან მისი ანტიპოდური ტანდემის გამო. ეს კვანტური ბიფურკაცია ორ ურთიერთდაპირისპირებული მხარე (ანტაგონისტურად)

არის თავად დინამიკის ძირითადი მიზეზი. აქ იგულისხმება ქაოსისა და კოსმოსის სუპერპოზიცია. თითოეული სუბიექტი, იქნება ეს მატერიალური თუ ფენომენური არსებითად პლურალისტურია, ანუ სინქრონულად არსებობს ნებისმიერი განზომილების სისტემაში, თუმცა ის შეიძლება მხოლოდ ეპისტემოლოგიურად იქნეს ცნობადი. გარდა ამისა, ყოველი სუბსტანცია (მათ შორის ინფორმაცია) და ფენომენი წარმოადგენს კვანტურად ჩახლართული გაურკვევლობის დისკრეტულ, დეკოჰერენტულ მატერიალიზაციას (ხაჩიძე, ჯანელიძე და მეფარიშვილი 2024, 63).

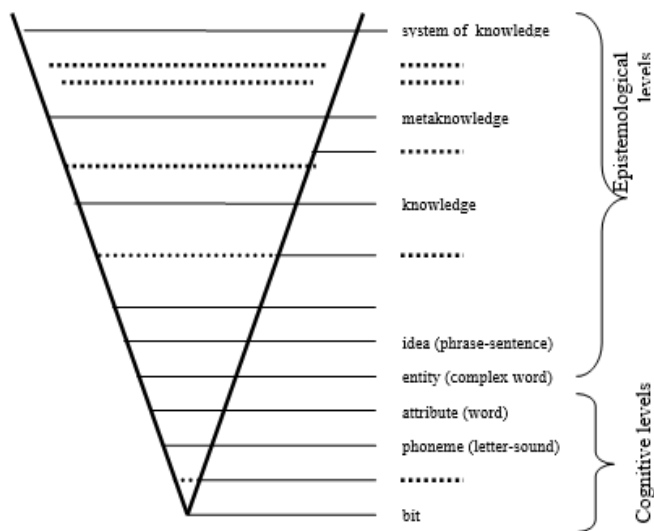
ცოდნის იერარქიული წარმოდგენა: ცოდნის წარმოდგენის რამდენიმე მოდელი გამოირჩევა, მათ შორისაა წინადადებითი ლოგიკა (პირველი წესის ლოგიკა), ტიპების თეორიები (მაღალი წესის ლოგიკა), სემანტიკური ქსელები, კადრები, სკრიპტები, წარმომქმნელი წესები, ონტოლოგიები და ცოდნის გრაფები. ცოდნის ფორმირება კი გრძელდება მთელი სიცოცხლის განმავლობაში, და ის მიმდინარეობს მრავალდონიან ad hoc ჰიპერგრაფულ ფრაქტალურ სისტემებსა და ეპისტემოლოგიურ ტოპოლოგიურ ტრანზაქციებში.

ამ თავში, კვლევითი ინტერესების შესაბამისად, ყურადღება გამახვილებულია ცოდნის გრაფულ მოდელზე. ადამიანის ცნობიერებაში ცოდნა და აზროვნება განუყოფელია - ანუ ინფორმაციის დამუშავებისა და ლოგიკური დასკვნის ხარისხი განისაზღვრება არსებულ ცოდნაზე დაყრდნობით. ქვეცნობიერ დონეზე, ცოდნა რეალიზდება ინსტინქტური სტერეოტიპების სისტემის მეშვეობით. ფრაქტალური სტრუქტუროგენეზის უნივერსალური პრინციპი შესატყვისად ვრცელდება ცოდნის სისტემის იერარქიულ სტრუქტურაზეც, რომლის ყველა დონე ქმნის მომდევნო განზომილების ხარისხობრივ საფუძველს. ცნობილია, რომ ინფორმაციის უმცირესი ერთეული ბიტია, რომელიც იერარქიულად აშენებს აღმავალ ფრაქტალურ სტრუქტურას, რაც შედგება შესაბამისი ეპისტემოლოგიური ფენებისგან (იხ. სურათი 3.1).

ცოდნის ფორმირების განხილვა შესაძლებელია მხოლოდ ცოდნის წარმოდგენის დაბალ დონეებზე, რადგან მაღალი დონის მეტაცნობიერების და აზროვნების მოდელირება სცილდება ადამიანის წარმოდგენის საზღვრებს. ცოდნის სისტემაში ინფორმაციის რაოდენობა იზრდება იერარქიული დონეების შესაბამისად,

რაც უკავშირდება განზომილებათა სიღრმესა და ხარისხს. დაბალ დონეებზე, ასობგერები ქმნის ათასობით სიტყვას, რომლებიც, თავის მხრივ, აგენერირებს იდეებსა და ცოდნას უფრო მაღალ, მაკრო დონეებზე. ტვინში არსებული ნებისმიერი ინფორმაცია აქსონ-დენდრიტული სინაფსების მეშვეობით რეალიზდება სემანტიკური გრაფების (ან ჰიპერგრაფების) სახით, რაც წარმოადგენს ნეირონულ დონეზე ცოდნის კავშირებისა და მნიშვნელობების მატერიალურ ანალოგს.

ცოდნის ევოლუციური პროცესი, ზოგადად, წარმოადგენს აღმავალ პროცესს იერარქიულ სისტემაში, სადაც მაღალ დონეზე გადასვლა შესაძლებელი ხდება მხოლოდ ქვედა დონის ფორმირების შემდეგ. ფრაქტალური სტრუქტურების თვითორგანიზების შედეგად, შესაძლებელია მიკროწვდომიდან მაკროწვდომამდე ცოდნის უფრო რთული და მრავალშრიანი ფორმირება. ეს პროცესი წარიმართება ენტროპიის მინიმიზაციის კრიტერიუმით, რაც გულისხმობს სისტემური უწყესრიგობის შემცირებას და სტრუქტურული ორგანიზების ზრდას. ცოდნის ყველა დონე შეიძლება ჩაითვალოს დასრულებულად მხოლოდ მაშინ, როდესაც მისი ენტროპია ხდება ნულის ტოლი. თუმცა, იმ ფონზე, როცა სწავლა გრძელდება და სისტემა იღებს ახალ ინფორმაციას, ცოდნა მაინც რჩება არასრული, ვინაიდან იგი განუწყვეტლივ გადადის ახალი სტრუქტურების ჩამოყალიბებაში.



სურ. 3.7: ცოდნის იერარქიული წარმოდგენა(წყარო: <https://shorturl.at/mztf5>).

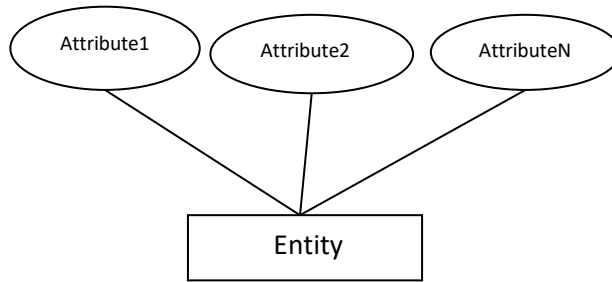
რეალური სამყაროს ყოველი ერთეული, რომელიც აღიქმება სენსორული არხების მეშვეობით, მიკრო დონეზე ასოცირდება იმ ატრიბუტებთან, რომლებიც

ცნობიერებაში აღიქმება და ეს ატრიბუტები ქმნის უნიკალურ ნეირონულ ქსელს, ანუ ტვინში წარმოქმნილი სემანტიკური ერთეულის ონტოლოგიური მოდელია. აღსანიშნავია, რომ ეს პროცესი დამოუკიდებელია ლინგვისტური ასპექტისგან, ენასთან დაკავშირებული ფორმა აქ არსებითად არ ფიგურირებს. ეს პრინციპი მეორდება ცოდნის სისტემის ყველა ეპისტემოლოგიურ დონეზე, რაც ქმნის ერთგვარ ფრაქტალურ სტრუქტურულ განმეორებადობას. ნეირონული გრაფები, როგორც ე.წ. ვირტუალური „მაკრონეირონები“, ცოდნის სისტემის მომდევნო დონეებზე აფორმირებენ სინერგიულ ნეირონულ ჯგუფებს განსხვავებული კონფიგურაციებით. ამგვარად, ფრაქტალური პრინციპების საფუძველზე, სხვადასხვა განზომილებით წარმოქმნილი პირველადი ცოდნის ეპისტემოლოგიური ფენები ქმნიან ცოდნის ერთიან სისტემას. კარგად ცნობილი ცოდნის მოდელი ადამიანის ტვინის ნეირონულ ქსელებში არ თავსდება ხელოვნური ინტელექტის თანამედროვე მოდელში, ისევე როგორც მრავალგანზომილებიანი სუპერგეომეტრია არ თავსდება ხაზობრივ ან ორგანზომილებიან გეომეტრიაში. დღევანდელი ყველა არსებული მიდგომა წარმოადგენს მხოლოდ ხელოვნური ინტელექტის პირველ მიახლოებას ბუნებრივთან. ამიტომ, მნიშვნელოვანია ახალი თაობის ხელოვნური ინტელექტის განვითარება, რომელიც დაფუძნებული იქნება ცოდნის წარმოდგენის ახალ პარადიგმაზე (ხაჩიძე, ჯანელიძე და მეფარიშვილი 2024, 63).

ადამიანის ტვინი და ცოდნა: ხელოვნური ინტელექტის განვითარების პროცესში, ზოგიერთი მეცნიერის აზრით, ხელოვნური ნეირონული ქსელები მიიჩნევა ადამიანის ტვინის ანალოგად, რაც ფუნდამენტურად მცდარი შეხედულებაა. მისი შემადგენელი კომპონენტებიც და ფუნქციონირების პრინციპებიც არანაირ კავშირში არ არიან ტვინის სტრუქტურასთან ან მისი მოქმედების პრინციპებთან. ტვინში არ არსებობს არც პროცესორი, არც მუდმივი ან ოპერატიული მეხსიერება (ROM ან RAM) და ა.შ. ტვინი ერთდროულად გადაჭრის ფუნქციურად სხვადასხვა ამოცანებს, რაც გამოირჩევა პარალელური და მრავალგანზომილებიანი დამუშავების უნარით. ტვინი მანიპულირებს რთული ვირტუალური გეომეტრიული ობიექტებით ნეირონული ერთობლიობების (ensemble) მეშვეობით, და არა ფორმალური ალგორითმებით (ხაჩიძე, ჯანელიძე და მეფარიშვილი 2024, 64). კომპიუტერისგან განსხვავებით, ტვინში

არ არსებობს ციკლები, შემთხვევითი შერჩევა ან მათემატიკური გამოთვლები ტრადიციული გაგებით. ტვინი წარმოადგენს მაღალი განზომილებისა და ექსტრემალური სირთულის ლოგიკურ მანქანას, რომელიც ფუნქციონირებს ანალოგურ ანუ უწყვეტ რეჟიმში. კიდევ ერთი განსხვავება იმაში მდგომარეობს, რომ ემბრიონალურ ეტაპზე ტვინი იძენს უზარმაზარი რაოდენობის ნეირონებს, რომლებიც ფუნქციონირებენ მთელი ცხოვრების განმავლობაში. მნიშვნელოვანი სხვაობა იმაშია, რომ ტვინი იყენებს ტოპოლოგიურად რთულ, მუდმივად რეკონფიგურირებად ანუ დინამიურ ნეირონულ (სინაფსურ) ქსელს, რომელიც ვერ შეესაბამება კლასიკური ციფრული კომპიუტერის სტრუქტურას.

ცოდნის წარმოდგენის მოდელები იერარქიაში განსხვავდებიან დონეების მიხედვით. ამასთანავე, უნდა განვასხვაოთ ცოდნის გარე რეფლექსია და შიდა (რეალური) მოდელები. რეალური მოდელი წარმოდგენილია ტვინის ნეირონული ერთობლიობების იერარქიულ სტრუქტურაში და ჩვენთვის დამალულია, მაშინ როდესაც გარე წარმოდგენა არსებობს გამჭვირვალე და თვალსაჩინო ფორმით, რომელიც შეიძლება დაიყოს ორ ძირითად კატეგორიად: ლინგვისტურ და ონტოლოგიურ. თუ განვიხილავთ ნებისმიერ სიტყვას ან ცნებას, როგორც ფონემების გარკვეული თანმიმდევრობას (სტრიქონებს), მაშინ ენის შესწავლის პროცესში იგი ქმნის ონტოლოგიურ „ატომურ“ მოდელს, რომელიც შედგება შესაბამისი ატრიბუტებისგან (იხ. სურათი 3.2). შემდეგ, ამ კონკრეტული თანმიმდევრობის წაკითხვის ან მოსმენისას ცნობიერება აქტიურდება ამ მოდელთან მიმართებაში. მაგალითად, სიტყვები „table“, „стол“, „მაგიდა“ განსხვავებულ ენებში დაკავშირებულია ერთი და იმავე ონტოლოგიურ მოდელთან, რომელიც უნივერსალურია, მაგრამ მათი აღქმა შესაძლებელია მხოლოდ კონკრეტული ენის პირობებში. აღსანიშნავია, რომ ეს პროცესი გართულებულია ლინგვისტიკაში არსებული სინონიმებისა და ჰომონიმების არსებობით, რაც მრავალჯერადობას და მულტი-რეფერენციულობას იწვევს ერთსა და იმავე მოდელზე.

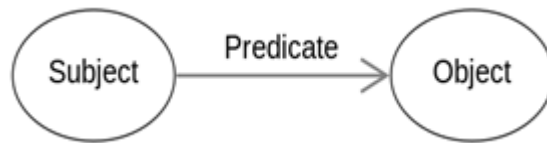


სურ. 3.8: ცნების ონტოლოგიური „ატომური“ მოდელი (წყარო: <https://shorturl.at/mztf5>).

არსებობს კიდევ ერთი მნიშვნელოვანი ნიუანსი, რომელიც მნიშვნელოვნად ცვლის ონტოლოგიურ მოდელს. მაგალითად, ორი სიტყვა, რომლებიც შედგება ერთი და იმავე ასოებისაგან, მაგრამ განსხვავებული თანმიმდევრობით – „მთა“ და „თმა“ შეიცავს სრულიად განსხვავებულ ონტოლოგიურ მოდელებს შესაბამისი ატრიბუტებით. ამით დასტურდება, რომ მითითებულ ენაში არა მხოლოდ ასო-ბგერების შემადგენლობა, არამედ მათი თანმიმდევრობაც არის არსებითად მნიშვნელოვანი, რადგან სწორედ ეს თანმიმდევრობა განსაზღვრავს ცნების მნიშვნელობასა და მის დაკავშირებას შესაბამის მოდელთან. როდესაც ჩვენ ვხედავთ ან ვისმენთ კონკრეტულ სიტყვას მის შესაბამის ენაში, ტვინი აქტივაციას ახდენს შესაბამის ნეირონულ ცენტრში, რომელიც შეიცავს ამ სიტყვის შიდა (რეალურ) ონტოლოგიურ მოდელს. აღსანიშნავია, რომ ეს შიდა მოდელი არის უნივერსალური, რაც ნიშნავს, რომ იგი არ არის დამოკიდებული კონკრეტულ ენაზე და საერთოა ყველა ადამიანისთვის.

ამრიგად, მიკრო დონეზე, რეალური სამყაროს ნებისმიერი ერთეული, იქნება ეს სუბიექტი, მოვლენა თუ ობიექტი შეესაბამება ცნობიერებაში აღქმულ ატრიბუტებს, რომელთა ერთობლიობაც ქმნის უნიკალურ ონტოლოგიურ მოდელს. აღნიშნული მოდელი წარმოდგენილია ვარსკვლავისებრი ნეირონული სტრუქტურის სახით, რომელშიც ცენტრალურ ელემენტს უკავშირდება მისი ატრიბუტები, და ეს კავშირები განსაზღვრავენ ცნებას. ამავე დროს, ეს სტრუქტურა დამოუკიდებელია ენის ლინგვისტური ასპექტებისგან, ანუ ენის ფორმა არ განსაზღვრავს შიდა ონტოლოგიურ მოდელს, რომელიც უნივერსალურია და წარმოდგენილია ტვინის ნეირონულ დონეზე, როგორც ყველა ადამიანის საერთო კოგნიტური მექანიზმი.

შემდეგ, ფრაზისა და წინადადების დონეზე, ყალიბდება სტრუქტურა, რომელიც შედგება სიტყვათა თანმიმდევრობისგან. თითოეული იდეა (ფრაზა-წინადადება) აღწერილია ფორმით: ((სუბიექტი, პრედიკატი, ობიექტი), დრო), რაც ცნობილია, როგორც სემანტიკური ტრიპლეტები. სუბიექტი წარმოადგენს მოვლენის უნიკალურ იდენტიფიკატორს, ხოლო პრედიკატი და ობიექტი აღწერენ მოვლენის თვისებებსა და შინაარსს. ამ სტრუქტურას ხშირად ემატება დროის შტამპი (timestamp), რომელიც საშუალებას იძლევა აღიწეროს მოვლენის დინამიკა და დროითი განვითარება.

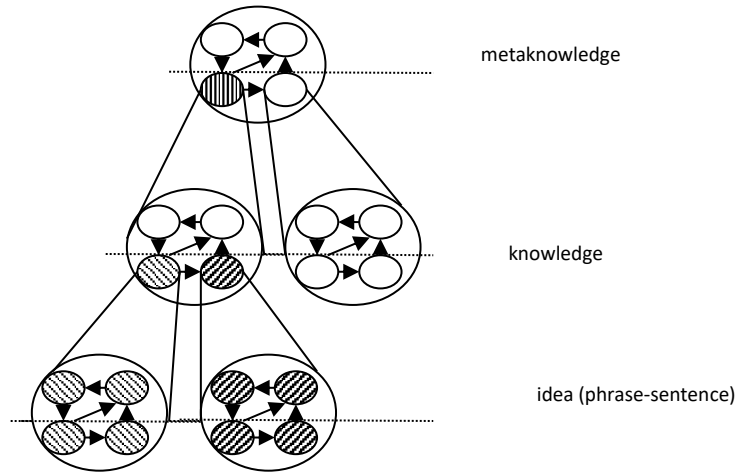


სურ. 3.9: სუბიექტი, პრედიკატი, ობიექტი, (წყარო: <https://shorturl.at/mztf5>).

ონტოლოგია წარმოადგენს ფორმალიზებულ სისტემას ცოდნის წარმოსადგენად, სადაც თითოეული კომპონენტი დაკავშირებულია შემდეგ კომპონენტთან ატრიბუტების მეშვეობით. ერთეულებს შორის კავშირის ხარისხი შეიძლება შეფასდეს მათი ატრიბუტების შესაბამისობის მიხედვით, რაც აღწერილია როგორც „სემანტიკური სინაპსი“, რომელიც, თავის მხრივ, შეიძლება სტატისტიკურად, ალბათობით განისაზღვროს. თითოეული იდეა (ფრაზა-წინადადება) ან კონცეპტი ხასიათდება ე.წ. ინტეგრირებულობის კრიტერიუმით – A, რომელიც განისაზღვრება მისი კომპონენტების სემანტიკური შესაბამისობით ან სინერგიულობით. ზოგადად, ცოდნის წარმოდგენასა და განსაკუთრებით ცოდნის კონსტრუქციის პროცესში, სინერგია განიხილება როგორც ენტროპიის ანტიპოდი – ანუ სისტემა მიიღტვის შინაგან თანხვედრობასა და სტრუქტურულ წესრიგისკენ იმგვარად, რომ უზრუნველყოს ცოდნის კომპონენტებს შორის მაღალი ხარისხის კავშირი და მთლიანობა (მეფარიშვილი, ჯანელიძე და ხაჩიძე 2023, 30).

შემდეგ დონეზე, უკვე წინადადება განიხილება როგორც ერთეული (კვანძი), რომლის გრაფიც წარმოადგენს მიკრო დონეზე ცოდნის პირველადი მოდელს. შესაბამისად, მაკრო დონეზე წარმოდგენილი ცოდნის მოდელი განიხილება როგორც სემანტიკური მაკრო ერთეული, რომელიც ქმნის მეტაცნობიერების ჰიპერგრაფს, და

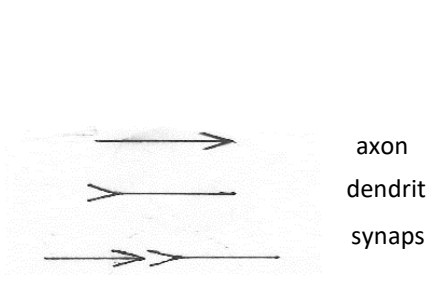
ასე გრძელდება იერარქიის ზედა ფენებში (ხაჩიძე, ჯანელიძე და მეფარიშვილი 2024, 64).



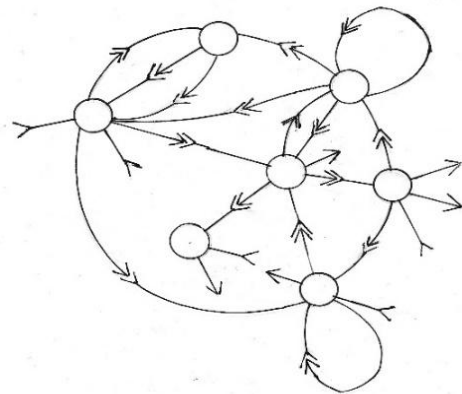
სურ. 3.10: მეტაცნობიერების ჰიპერგრაფი (წყარო: <https://shorturl.at/mztf5>).

სინერგიაზე დაფუძნებული მოდელირების მიდგომა: თანამედროვე სისტემური მოდელები უფრო მეტად ეფუძნება არათანაბარწონიან მოდელებს, რომელთა თეორიული საფუძველი წარმოადგენს ენტროპიის კონცეფციას. აღნიშნულმა მიდგომამ გამოიწვია მრავალმხრივი მოდელების განვითარება, რომლებიც ეფუძნება ენტროპიას, სინერგიასა და სირთულის თეორიას. რაც შეეხება სინერგიას, იგი გულისხმობს იმას, რომ სისტემური მთლიანობა ფლობს ისეთ თვისებებს (ფუნქციურ ეფექტებს), რომლებიც არ გააჩნია ცალკეულ კომპონენტებს. სინერგიის გარეშე მთლიანობაც არ არსებობს. თითოეული სისტემა ხასიათდება სტრუქტურით, შემადგენლობითა და მდგომარეობით. სისტემის მდგომარეობა კი შეიძლება აღწერილ იქნას შეუთავსებლობის სხვადასხვა დონით. ზოგადად, ნებისმიერი მასშტაბური სისტემა და მისი კომპონენტები პირობითად შეიძლება განიხილოს როგორც ნეირონული მოდელი (იხ. სურათი 3.5ბ). სწორედ ამ მიზნით შეიქმნა გრაფისა ან ჰიპერგრაფის მსგავსი სტრუქტურები, რათა აღწერილიყო ნებისმიერი დონის კომპლექსური სისტემა, მათ შორის ტვინის ნეირონული მოდელი, როგორც სამყაროს ფრაქტალური ანარეკლი (ხაჩიძე, ჯანელიძე და მეფარიშვილი 2024, 65).

ა) ნეირონული მოდელის ელემენტები.



ბ) ნეირონული მოდელის მაგალითი.



სურ. 3.11: ა, ბ (წყარო: <https://shorturl.at/mztf5>).

სინაფსური კავშირის შედეგად ნეირონებს შორის ხდება მათი შეზრდა და იქმნება ახალი ენსემბლი, რომელიც წარმოადგენს სინერგიულ-ენტროპიული კავშირის ერთიანობას. თითოეული სინაფსი ან ორი ნეირონის ურთიერთქმედება განმეორებით ქმნის ახალ ერთეულს, ერთიან ნეირონულ კლასტერს, რომელსაც გააჩნია სინერგია-ენტროპიის გადანაწილების ახალი სტრუქტურა, ცვალებადი ბალანსითა და ადაპტაციური (ფიტნეს) ფუნქციით. შექმნის პროცესი მიმდინარეობს მაშინ, როდესაც ენტროპია გარდაიქმნება სინერგიად, ხოლო დაშლისას სინერგია გადადის ენტროპიაში, ანუ მათ შორის ხდება ურთიერთ გარდაქმნა (მეფარიშვილი, ჯანელიძე და ხაჩიძე 2023, 32).

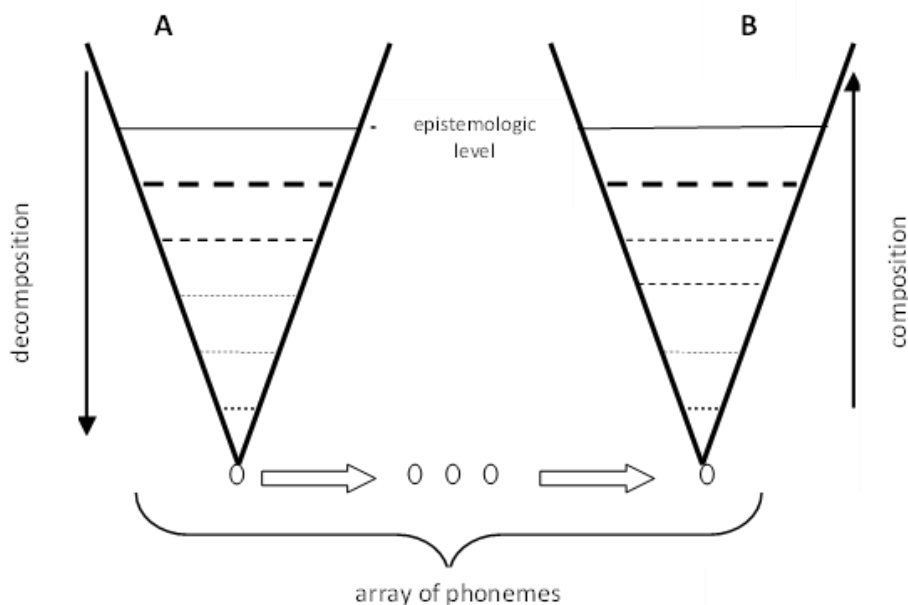
ინფორმაციის მიმოცვლა: ინფორმაციის პრიმიტიული გაცვლა მაგალითად, ტელეპატია დროთა განმავლობაში შეიცვალა სიტყვიერი მეტყველებით, ხოლო მოგვიანებით წერილობითი კომუნიკაციით. მოცემულ კონტექსტში განვიხილავთ გარემოსთან ინფორმაციის გაცვლის ინტერაქტიულ ასპექტს. მაგალითად, ხშირად ხდება ისე, რომ ორ ადამიანს, რომლებიც ერთი და იმავე ენით საუბრობენ, მეტყველებას მორფოლოგიურად და სინტაქტურად სწორად აღიქვამენ, მაგრამ მათ შორის შინაარსობრივი გაგება არ ყალიბდება, ანუ მცირდება ურთიერთგაგება კონკრეტულ თემაზე. ზოგადად, გაგების პროცესი გულისხმობს სემანტიკური ჰიპერგრაფების გავლის უნარს, რაც ხორციელდება საუბრის მონაწილეთა ცოდნის სისტემის ეპისტემოლოგიურ დონეზე. რაც უფრო მაღალია ამ ჰიპერგრაფების გადაკვეთა, მით უფრო მაღალია ურთიერთგაგების ხარისხი. ინტერაქტიულობის

დისკუსიისას ყურადღება გამახვილებულია შემდეგ სქემაზე (იხ. სურათი 3.12), სადაც ენობრივი შუამავლის ფუნქციას ადამიანის ენა ასრულებს, ლინგვისტური "თარჯიმანი" ახორციელებს ორმხრივ გარდაქმნას.

მეტაცოდნის დონეზე ხდება დეფრაგმენტაცია და აქტივაცია მომხსენებლის (ან ინფორმაციის გადამცემი სუბიექტის) ცოდნის სისტემაში. იმისათვის, რომ მოხდეს ინფორმაციის გადაცემის დამშლელი ტრაქტის ქვევით მოძრაობა, მიმდინარეობს შემდეგი თანმიმდევრული ეტაპები:

ასო-ხმა (ფონემების) → სიტყვა → ფრაზა → მეტყველება შესაბამის ენაზე → იდეის ცოდნის ფრაგმენტი → ცოდნა → მეტაცოდნა.

ინფორმაციის გადაცემა ნიშნავს ასო-ხმის ჯაჭვის ან მასივის ფორმით კოდირებულ შინაარსს, რომელიც შეგრძნებით აღიქმება მიმღები მხარის (B-ს) მიერ. მიღებისას მიმდინარეობს საპირისპირო პროცესი, ამაღლებული კომპოზიციის ტრაქტი. ამ შემთხვევაში, ქვეცნობიერის დონეზე მიმდინარეობს: მორფოლოგიური ანალიზი, ასო-ხმის თანმიმდევრობიდან სიტყვების ამოცნობა; სინტაქსური ანალიზი, წარმოთქმული წინადადებების აგება; სემანტიკური ანალიზი, შინაარსობრივი მნიშვნელობის გააზრება.



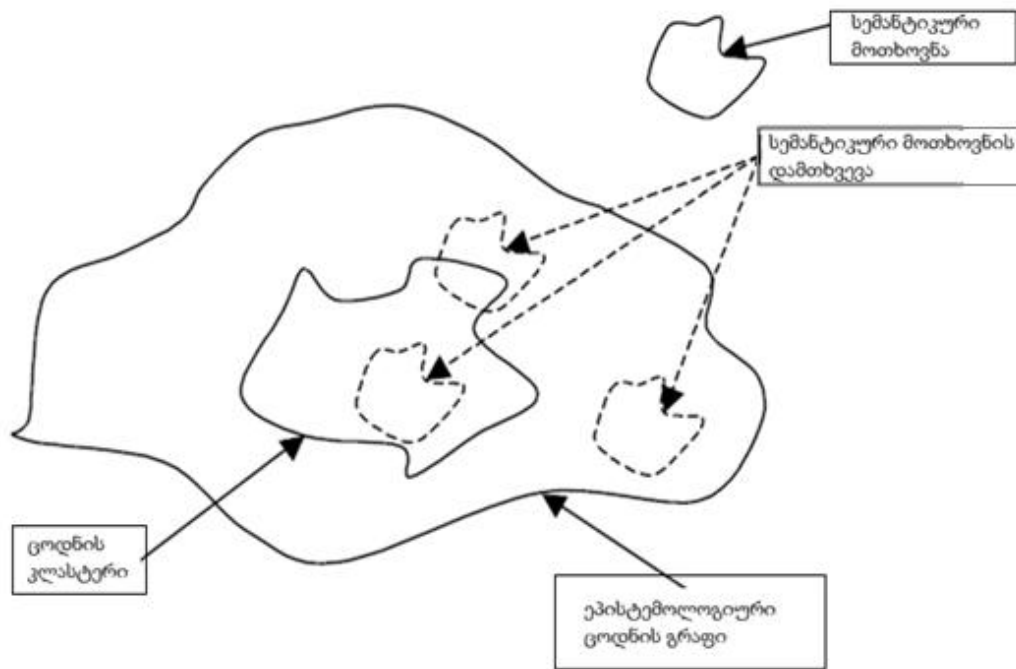
სურ. 3.12: ორმხრივი ინფორმაციის მიმოცვლა (წყარო: ავტორი).

ინტერაქტიულ პროცესში, მაღალი განზომილების დონიდან ცოდნა იწყებს ქვევით მოძრაობას იმისათვის, რომ სემანტიკური "ქსოვილის" დეაგრეგაცია მოხდეს

და იგი გადაეცეს მეორე მხარეს "მაფის" სახით, რაც სიმბოლურად ასახავს ლათინური სიტყვის *textus* („ქსოვილი“) ეტიმოლოგიას. მიღების მხარეს, ეს "ქსოვის პროცესი" იწყება სწორედ ამ გადმოცემული "მაფისგან". გაგება დამოკიდებულია იმაზე, რამდენად შეესაბამება მიღებული პროდუქტი ორიგინალს, ანუ შეძლო თუ არა მიმღებმა ზუსტად აღედგინა იმ ცოდნის სტრუქტურა, რაც გადამცემმა გააგზავნა.

სწავლა როგორც ცოდნის აგების პროცესი: ჩვენი იდეა განიხილავს ცოდნის წარმოდგენის ჰიპერგრაფზე დაფუძნებულ მიდგომას, რომლის მიხედვითაც ეპისტემოლოგიური ცოდნის მოდელი შედგება კვანძებისაგან, რომლებიც წარმოადგენს ერთეულებს (დაბალი დონის ცოდნის გრაფი) და ამ ერთეულებს შორის არსებული კავშირებისაგან. ცოდნის კონსტრუქციის რეკურსიული პროცესი წარმოადგენს იერარქიულად აღმავალ პროცესს, რომელიც მიზნად ისახავს ცოდნის არასრულობის, ანუ ენტროპიის მინიმიზაციას. შესაბამისად, ყველა დონეზე ეპისტემური ცოდნა ითვლება სრულად მხოლოდ მაშინ, როდესაც მისი ენტროპია ნულს გაუტოლდება.

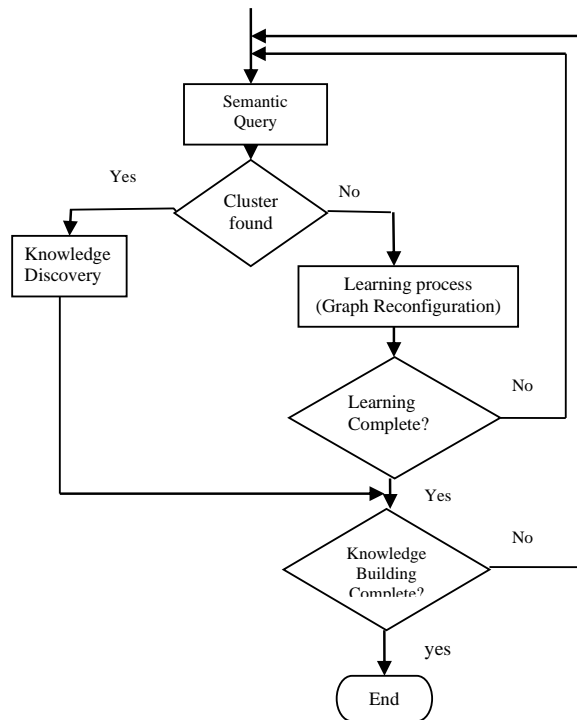
ცოდნის აღმოჩენა შეიძლება განვიხილოთ, როგორც პროცესი, რომლის დროსაც სემანტიკური მოთხოვნა ემთხვევა (ან ეძებს შესაბამისობას) ცოდნის გრაფში წარმოდგენილ ეპისტემურ ცოდნის კლასტერს (იხილე სურათი 3.7). მოთხოვნის შესაბამისობა გულისხმობს იმას, რომ ცოდნის ლანდშაფტზე მოიძებნა მიზნობრივი სემანტიკური კლასტერი. წინააღმდეგ შემთხვევაში, აუცილებელი ხდება ცოდნის მოდელის სირთულის დაძლევა თვითსწავლის მეშვეობით. ეს ხორციელდება სემანტიკური გრაფის რეკონფიგურაციით ან თვითორგანიზებული „კერვის“ (შეკავშირების) პროცესით, რაც ეფუძნება ენტროპიის მინიმიზაციის კრიტერიუმს. ეპისტემოლოგიური ცოდნის აგების პროცესი გრძელდება მანამ, სანამ ცოდნის მოდელი სრულყოფილად არ იქცევა.



სურ. 3.13: ცოდნის აღმოჩენის პროცესი (წყარო: <https://shorturl.at/mztf5>).

როდესაც სემანტიკური მოთხოვნა შეყვანილია, მის შესაბამისობაზე შესაძლებელია განსაზღვრა მოთხოვნისა და ცოდნის ფრაგმენტების გრაფის გადაკვეთის საფუძველზე. თითოეულ შემთხვევაში, შესაძლებელია ენტროპიის მინიმიზაციის კრიტერიუმით შემდეგი ოპტიმიზაციის ვარიანტების განხორციელება. თუ ეს გადაკვეთა სრულია, ეს ნიშნავს ცოდნის აღმოჩენის შემთხვევას. წინააღმდეგ შემთხვევაში, ვითარება იქნება გაურკვეველი, რაც დამოკიდებულია შესაბამისი ენტროპიის ან გრაფების განსხვავების მნიშვნელობაზე. ზოგადად, ცოდნის მენეჯმენტის პროცესი შეიძლება წარმოდგენილ იქნას როგორც ქვემოთ მოცემული ალგორითმი. მისი განხორციელების ორი შესაძლო გზა არსებობს (იხილე სურათი 3.14).

თვითსწავლებადი ცოდნის აგების თვითორგანიზებული პროცესი განავითარებს გრაფზე დაფუძნებულ ცოდნის წარმოდგენის მოდელს (Graph-based Knowledge Representation – GBKR), ენტროპიის შემცირებით. ყოველი მომდევნო შემომავალი ინფორმაციის მიღების შემდეგ, ცოდნის მოდელების კონფიგურაცია გარდაიქმნება ენტროპიული გრადიენტის მიხედვით.



სურ. 3.14: ცოდნის მენეჯმენტის პროცესი (წყარო: <https://shorturl.at/mztf5>).

ოგორც ზემოთ აღნიშნა, თანამედროვე კომპიუტერული მოდელები და შესაბამისი პროგრამული უზრუნველყოფა ხელოვნური ინტელექტის სისტემის რეალიზაციისთვის უკვე აღარ არის დამაკმაყოფილებელი. ცოდნის წარმოდგენის ახალი მიდგომის საშუალებით შესაძლებელი იქნება მუშაობა არა მხოლოდ ტექსტის მარტივ მოდელებზე, არამედ მაღალი დონის ცოდნაზე და ცოდნის მოდელებზე, რაც განსაკუთრებით აქტუალურია ინფორმაციულ ეპოქაში. ეს, თავის მხრივ, გახდება ხელოვნური ინტელექტის ახალი პარადიგმის დასაწყისი. ამ პარადიგმის ნიშნები უკვე შეიმჩნევა აღნიშნულ ნაშრომში (ხაჩიძე, ჯანელიძე და მეფარიშვილი 2024, 66).

თავი IV - ცოდნის ასემბლინგისადმი ბიონფორმატიკული მიდგომა

4.1 ცოდნის ფრაგმენტის წარმოდგენის მოლეკულური მოდელი

ფრაზა „ცოდნის ფრაგმენტის წარმოდგენის მოლეკულური მოდელი“ ქმნის ანალოგიას ცოდნასა და ქიმიას შორის. ისევე, როგორც მოლეკულები შედგება ატომებისგან, რთული ცოდნის წარმოქმნა შესაძლებელია მცირე ფრაგმენტებიდან, ანუ ინფორმაციის ატომური ერთეულებიდან. მოცემულ მოდელში ცოდნის ფრაგმენტები წარმოადგენს ძირეულ სამშენებლო ერთეულებს დამოუკიდებელ ცოდნის ნაწილებს, როგორცაა ფაქტები, კონცეფციები ან მარტივი წესები, რომელთა შეკავშირებით ან გაერთიანებით იქმნება სტრუქტურირებული ერთობლიობები („მოლეკულები“), რაც ასახავს უფრო რთულ იდეებსა თუ სფეროებს. ცოდნის ფრაგმენტებად წარმოდგენა გულისხმობს ინფორმაციის დაშლას მცირე, თვითკმარ ნაწილებად, რომელთა შემდგომი დაკავშირება ან კომბინირება სისტემატური წესების მიხედვით ხორციელდება უფრო დიდი ცოდნის სტრუქტურების ან შემეცნებითი სქემების ჩამოსაყალიბებლად. ეს მიდგომა თანხვედრაშია კოგნიტური მეცნიერებისა და ინფორმაციის თეორიის ტრადიციულ იდეებთან, რომელთა თანახმადაც ცოდნის მოდულარიზაცია შესაძლებელია. თითოეული ფრაგმენტის ცალკეულ ერთეულად განსახილველად, მათი რეკომბინაცია ხდება შესაძლებელი, ანალოგიურად ატომებიდან მოლეკულის აწყობის პროცესისა, რაც ცოდნაში არსებულ ურთიერთობებსა და შემადგენლობებს ზუსტად ასახავს.

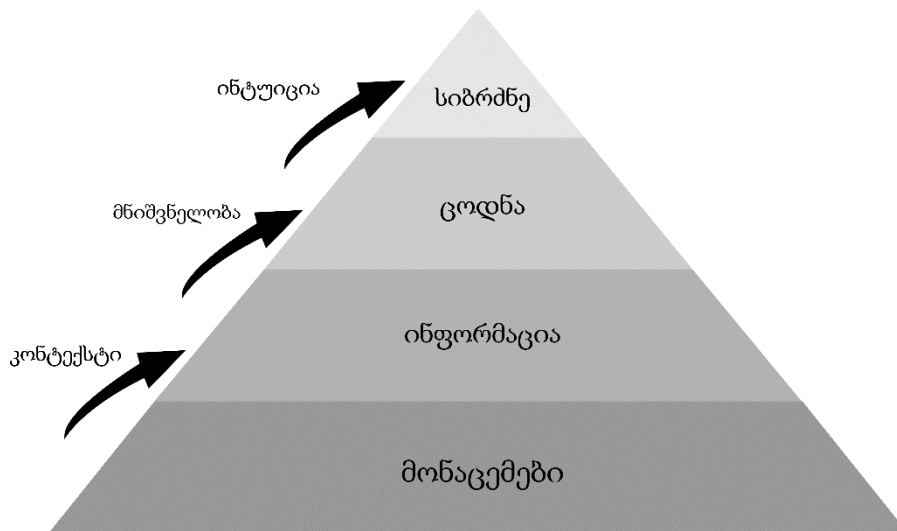
ზოგადად, მონაცემთა მოპოვება (Data Mining) მიზნად ისახავს უზარმაზარი მონაცემთა წყაროებიდან შესაბამისი მონაცემების აღმოჩენას, შესწავლასა და ანალიზს სხვადასხვა ტექნიკებისა და ალგორითმების გამოყენებით. თუმცა, დიდი მოცულობის მონაცემების (Big Data) შემთხვევაში, ტრადიციული მიდგომებითა და ალგორითმებით მათი დამუშავება და მათგან ცოდნის მოპოვება სერიოზულ სირთულეებთან არის დაკავშირებული. დიდი მოცულობის მონაცემებიდან ცოდნის მოპოვება აღნიშნავს პროცესს, რომელიც მოიცავს უზარმაზარი მონაცემთა ნაკრებებიდან ცოდნისა და კანონზომიერებების ამოღებას, ასევე ლამბდა არქიტექტურისა და დიდი მონაცემების

პარალელური დამუშავების პარადიგმის ერთობლივ გამოყენებას. ამ ნაშრომის ორიგინალურობა ეფუძნება ცოდნის გრაფიკული მოდელის სახით წარმოდგენას, ასევე ინდივიდუალური ცოდნის გრაფული ფრაგმენტებიდან ერთიანი ცოდნის მოდელის აგებას. ლამბდა არქიტექტურის კონტექსტში დიდი მოცულობის მონაცემების პარალელური დამუშავებისას, ამ ორი კონცეფციის ერთობლივი გამოყენება შესაძლებელს ხდის ცოდნის მოპოვების ფართომასშტაბიანი ამოცანების შესრულებას (ხაჩიძე 2024, 91).

მონაცემთა მოპოვება ასევე განისაზღვრება, როგორც უზარმაზარი მოცულობის მონაცემებში შაბლონების აღმოჩენისა და ახალი მონაცემების პროგნოზირების პროცესი. მონაცემთა მეცნიერება, როგორც წესი, მიჰყვება გარკვეულ მეთოდოლოგიას, რომელიც განსაზღვრავს, თუ როგორ უნდა გადაიჭრას ამოცანა და როგორ უნდა იმუშაოს მონაცემებთან, ალგორითმებთან ან მოდელებთან. ეს მეთოდოლოგია წარმოადგენს ცოდნის აღმოჩენის პროცესის აღწერას სტრუქტურირებული მონაცემებიდან. გასული საუკუნის ბოლოდან მონაცემთა მეცნიერების კვლევებმა დაიწყო ცოდნის აღმოჩენის (Knowledge Discovery) ან მონაცემთა ბაზებში ცოდნის აღმოჩენის (Knowledge Discovery in Databases – KDD) განსაზღვრა, როგორც ცოდნის აღმოჩენის პროცესის (Knowledge Discovery Process – KDP) სინონიმი. აღნიშნული პროცესი მონაცემთა მოპოვებას განიხილავს, როგორც ერთ-ერთ საფეხურს ცოდნის მიღების ძალისხმევაში (ხაჩიძე 2024, 91).

მიუხედავად იმისა, რომ KDD (ან KDP) და მონაცემთა მოპოვება ხშირად გათანაბრებულად გამოიყენება, სინამდვილეში მონაცემთა მოპოვება წარმოადგენს მთლიან პროცესის მხოლოდ ერთ საფეხურს, რომელიც ეხება იმ ალგორითმების გამოყენებას, რომლებიც მონაცემებიდან შაბლონების მოპოვებას ახორციელებს.

თანამედროვე საინფორმაციო ტექნოლოგია ინფორმაციას აღწერს შემდეგი კონცეფციებით: **DIKW პირამიდა** (DIKW Pyramid – მონაცემი, ინფორმაცია, ცოდნა და სიბრძნე) (ხაჩიძე 2024, 92).



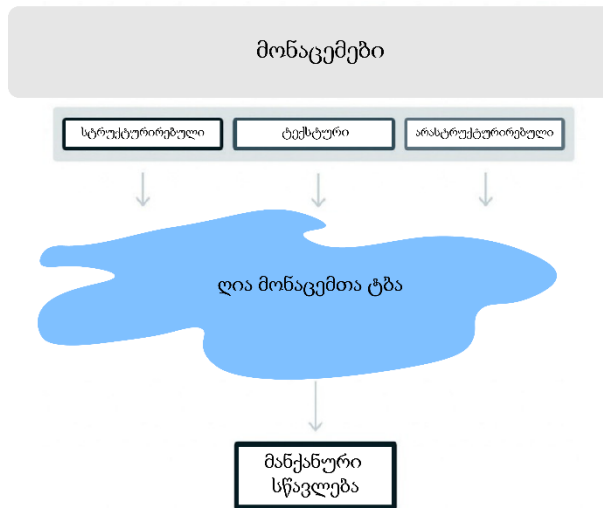
სურ. 4.1. DIKW პირამიდა (DIKW Pyramid – მონაცემი, ინფორმაცია, ცოდნა და სიბრძნე).

დიდი მოცულობის მონაცემების იქნება ის სტრუქტურირებული, ნახევრადსტრუქტურირებული თუ არასტრუქტურირებული, ყველა ტიპის შესანახად ხშირად გამოიყენება მონაცემთა ტბა (Data Lake). მოცემულ თავში ცოდნის ინჟინერია განიხილება მონაცემთა ტბების, დიდი მონაცემების ანალიტიკის, ასევე ინფორმაციის ინტეგრაციისა და ცოდნის მენეჯმენტის კონტექსტში.

როგორც ცნობილია, ცოდნის ინჟინერია ხელოვნური ინტელექტის დარგია და მოიცავს ცოდნის დიდ მოცულობას, როგორცაა მეტამონაცემები და მონაცემთა ობიექტის შესახებ ინფორმაცია, რომელიც აღწერს მის შინაარსს, სტრუქტურასა და დამუშავების პროცესებს. ვინაიდან ცოდნა ძირითადად ტექსტურ მონაცემებზეა დაფუძნებული, მნიშვნელოვანია გავითვალისწინოთ დიდი მოცულობის ტექსტების საცავები - მონაცემთა ტბები, რომლებიც ჯერ კიდევ შეიცავენ გარკვეულ შეზღუდვებს, მათ შორის: ტრანზაქციების მხარდაჭერის ნაკლებობა (რაც ართულებს მათი განახლებას) და ACID პრინციპებთან შეუთავსებლობა (რაც ზღუდავს ერთდროულ წვდომასა და ჩაწერას) (ხაჩიძე 2024, 92).

მონაცემთა საწყობებთან შედარებით, მონაცემთა ტბები შეიცავს დაუმუშავებელ მონაცემებს მრავალფეროვან ფორმატში, რაც განაპირობებს როგორც მიმდინარე, ისე მომავალი მიზნებისთვის მათ გამოყენებას.

ცოდნის მენეჯმენტის პროცესში ჩართული ურთიერთქმედების მრავალფეროვნება ასახავს იმ ტექსტუალურ ფრაგმენტთა მრავალფეროვნებასაც, რომლებიც მონაცემთა ტბებთან არის დაკავშირებული (ხაჩიძე 2024, 93).



სურ.4.2. მონაცემთა ტიპები ტბებში (წყარო: ავტორი).

გარდა ამისა, მომდევნო წლებში დიდი მონაცემების, განსაკუთრებით კი მონაცემთა ტბების პარადიგმის განვითარებამ გააჩინა ცოდნის მოპოვების ახალი ძალისხმევის საჭიროება, რაც გამოწვეული იყო სრულიად განსხვავებული მიდგომების შემუშავების აუცილებლობით.

მნიშვნელოვანია აღინიშნოს, რომ მანქანური სწავლება ძირითადად შეზღუდულია კლასიფიკაციისა და კლასტერული ანალიზის ამოცანების გადაწყვეტით. თუმცა მოცემულ კონტექსტში კლასტერი არ წარმოადგენს ცოდნას — ის მხოლოდ მონაცემთა ერთობლიობას ასახავს, მაშინ როცა ცოდნა უნდა განიხილებოდეს უფრო გრავის სახით, რაც შესაძლებლობას იძლევა უკეთესად აღწეროთ ობიექტებს შორის კავშირები და სტრუქტურა.

მოცემული მიდგომა ეფუძნება ცოდნის ინჟინერიის ახალ გააზრებას, რომელიც ეფუძნება ცოდნის იერარქიულ წარმოდგენაზე დამყარებულ სინერგიულ მიდგომას. ცოდნის წარმოდგენის მოდელები განსხვავდებიან დონეების მიხედვით. გარდა ამისა, გამოირჩევა ცოდნის გარე ასახვა და შიდა (რეალური) მოდელები. გარე ასახვა აღწერილია და ხილულია. თავის მხრივ, გარე აღქმა იყოფა ლინგვისტურ და ონტოლოგიურ ტიპებად (ხაჩიძე 2024, 94).

თუ სიტყვას ან ცნებას განვიხილავთ, როგორც ფონემების (სტრიქონების) გარკვეულ თანმიმდევრობას, რომელიც ენის შესწავლის პროცესში ქმნის შესაბამის ატრიბუტებზე დაფუძნებულ ონტოლოგიურ მოდელს, მაშინ ამ თანმიმდევრობის წაკითხვის ან მოსმენისას ის ცნობიერებაში ასოცირდება აღნიშნულ მოდელთან.

თუმცა არსებობს ერთი მნიშვნელოვანი ნიუანსი — ფონემების თანმიმდევრობა არსებითად ცვლის ონტოლოგიურ მოდელს. მაგალითად, ორი სიტყვა, რომელიც შედგება იგივე ასოებისგან, მაგრამ განსხვავებულ თანმიმდევრობაში, ქმნის სრულიად განსხვავებულ ონტოლოგიურ მოდელებს შესაბამისი ატრიბუტებით. შესაბამისად, მნიშვნელოვანია არა მხოლოდ ფონემათა შემადგენლობა, არამედ მათი თანმიმდევრობაც.

მიკრო დონეზე, ნებისმიერი რეალური ობიექტი (საგანი, მოვლენა ან სხვა ერთეული) ასოცირდება ცნობიერებაში აღქმულ ატრიბუტებთან, რომლებიც გადმოიცემა სენსორული არხების მეშვეობით. ეს ატრიბუტები ქმნის უნიკალურ ნეირონულ ქსელს, ანუ წარმოდგენილია როგორც ონტოლოგიური სემანტიკური ერთეულის მოდელი. მათ არ გააჩნიათ ლინგვისტურ ასპექტებთან დამოკიდებულება.

რაც შეეხება ფრაზისა და წინადადების დონეს — ის ქმნის სტრუქტურას სიტყვათა თანმიმდევრობის საფუძველზე. როგორც ცნობილია, ონტოლოგია წარმოადგენს ფორმალიზმს ცოდნის წარმოდგენისთვის (ხაჩიძე 2024, 95).

ზოგადად, თითოეული იდეის (ფრაზის/წინადადების) აღწერა შესაძლებელია შემდეგნაირად: **(სუბიექტი, პრედიკატი, ობიექტი), დრო**. სუბიექტი წარმოადგენს მოვლენის უნიკალურ იდენტიფიკატორს, ხოლო პრედიკატი და ობიექტი აღწერს მოვლენის მახასიათებლებს. მოვლენის დინამიკის აღწერისთვის შესაძლებელია დაემატოს დროის შტამპი. თითოეული იდეა (ფრაზა/წინადადება) ან ცნება ხასიათდება ე.წ. **„A მთლიანობის კრიტერიუმით“**, რომელიც განისაზღვრება მისი კომპონენტების სემანტიკური შესაბამისობითა და სინერგიულობით. **მთლიანობა** წარმოადგენს ერთ-ერთ ყველაზე მნიშვნელოვან კრიტერიუმს ცოდნის ინჟინერიაში, განსაკუთრებით ცოდნის აგების პროცესში, რაც უშუალოდ უკავშირდება სინერგიის ცნებას.

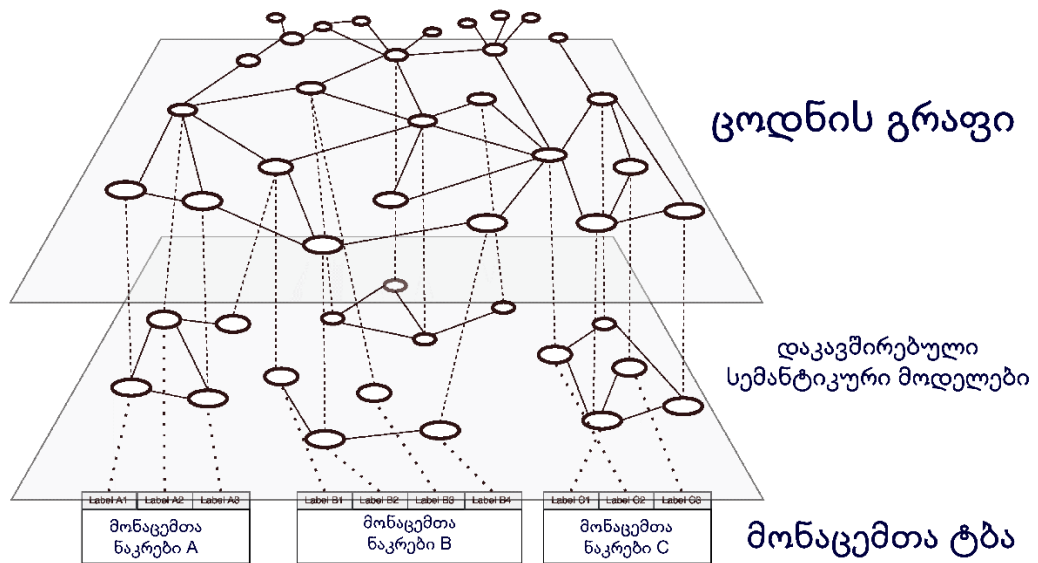
შემდეგ დონეზე, თავად წინადადება განიხილება როგორც **ერთეული (კვანძი)**, რომლის გრაფი წარმოადგენს ცოდნის პირველდონიან მოდელს. მოცემული ცოდნის მოდელი მაკრო დონეზე უკვე აღიქმება როგორც **სემანტიკური მაკრო ერთეული**, რომელიც ქმნის მეტაცნობიერების ჰიპერგრაფს და ა.შ. იერარქიულ დონეებზე (ხაჩიძე 2024, 95).

არსებობს აუცილებელი პირობა, რომელიც ვრცელდება ცოდნის სისტემის ყველა ეპისტემოლოგიურ დონეზე: გარკვეული ცოდნის გრაფები სხვადასხვა კონფიგურაციით შეიძლება ჩაითვალოს როგორც მომდევნო დონის ცოდნის გრაფის კვანძები. შესაბამისად, **ფრაქტალური პრინციპების შესაბამისად**, არსებობს პირველადი ცოდნის ეპისტემოლოგიური ფენები სხვადასხვა განზომილებით, რომლებიც ერთობლიობაში ქმნიან ცოდნის ერთიან სისტემას.

დღემდე არსებული ყველა მიდგომა წარმოადგენს ხელოვნური ინტელექტის პირველად და შეზღუდული მოდელირების მცდელობებს. აქედან გამომდინარე, მნიშვნელოვანია ახალი თაობის ხელოვნური ინტელექტის განვითარება, რომელიც დაეფუძნება ცოდნის წარმოდგენის ახალ პარადიგმას (ხაჩიძე 2024, 96).

მანქანური სწავლების როლი ცოდნის მენეჯმენტში - რაც შეეხება მანქანურ თარგმნას, არსებული მიდგომები ჯერ კიდევ შორს არის სრულყოფისგან, რადგან კომპიუტერს არ შეუძლია ტექსტუალური ინფორმაციის ადეკვატურად გადათარგმნა. **მონაცემთა მოპოვების პროცესში კლასტერული ან კლასიფიკაციის ამოცანების გადაჭრისას, მანქანური სწავლება ცოდნას განიხილავს მხოლოდ როგორც მონაცემთა ერთობლიობას.** ერთ-ერთი მიზეზი ამგვარი შეზღუდული ხედვისა ის არის, რომ თანამედროვე კომპიუტერები იშვიათად იყენებენ ცოდნის წარმოდგენის მოდელებს, როგორცაა სემანტიკური ქსელები, ჩარჩო სისტემები და სხვ., რომლებიც მხოლოდ პირველადი მიმსგავსებაა ბუნებრივი მოდელებისადმი.

მანქანური სწავლება ძირითადად კონცენტრირებულია კლასტერულ ან კლასიფიკაციის ამოცანებზე. თუმცა მონაცემთა კლასტერი არ წარმოადგენს ცოდნას - იგი მხოლოდ მონაცემთა ერთობლიობას ასახავს. **ცოდნა კი უფრო განსახილველია როგორც გრაფი ან ჰიპერგრაფი,** რაც იძლევა მეტი სემანტიკური კავშირის და სტრუქტურის აღბეჭდვის საშუალებას (ხაჩიძე 2024, 96).



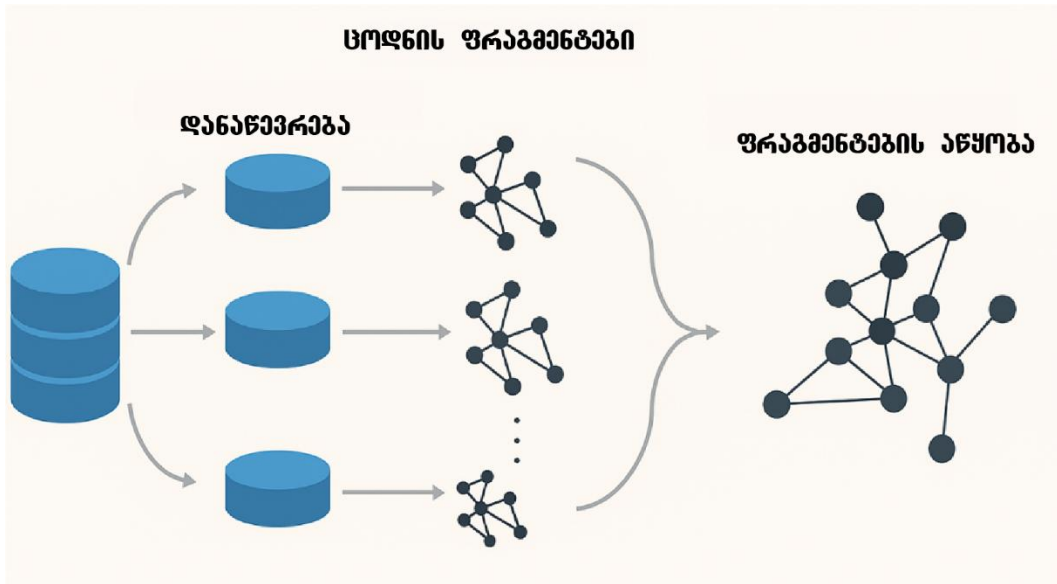
სურ. 4.3. ცოდნის ჰიპერგრაფი (წყარო: ავტორი).

მოცემულ სტატიაში ცოდნის ინჟინერია განიხილება თანამედროვე საინფორმაციო სისტემების კონტექსტში, კერძოდ ხელოვნური ინტელექტის, დიდი მონაცემების ანალიტიკის, ინფორმაციის ინტეგრაციისა და ცოდნის მენეჯმენტის ჭრილში. სტატია ეფუძნება ცოდნის წარმოდგენის ახალ გააზრებას (ხაჩიძე 2024, 96).

მანქანური თარგმნის არსებული მიდგომები შორს არის სრულყოფისგან, ვინაიდან კომპიუტერს არ შეუძლია ტექსტუალური ინფორმაციის ადეკვატური თარგმნა. მონაცემთა მოპოვებისას კლასტერული ან კლასიფიკაციის ამოცანების გადაჭრისას, მანქანური სწავლება ცოდნას განიხილავს მხოლოდ როგორც მონაცემთა ერთობლიობას. ამის ერთ-ერთი მიზეზია ის, რომ თანამედროვე კომპიუტერული სისტემები იშვიათად იყენებენ ცოდნის წარმოდგენის მოდელებს, როგორცაა სემანტიკური ქსელები, ფრეიმ სისტემები და სხვა - მაშინ როცა ეს მოდელები მხოლოდ ბუნებრივი ცოდნის პირველდონიან და შეზღუდულ მიმსგავსებას წარმოადგენს.

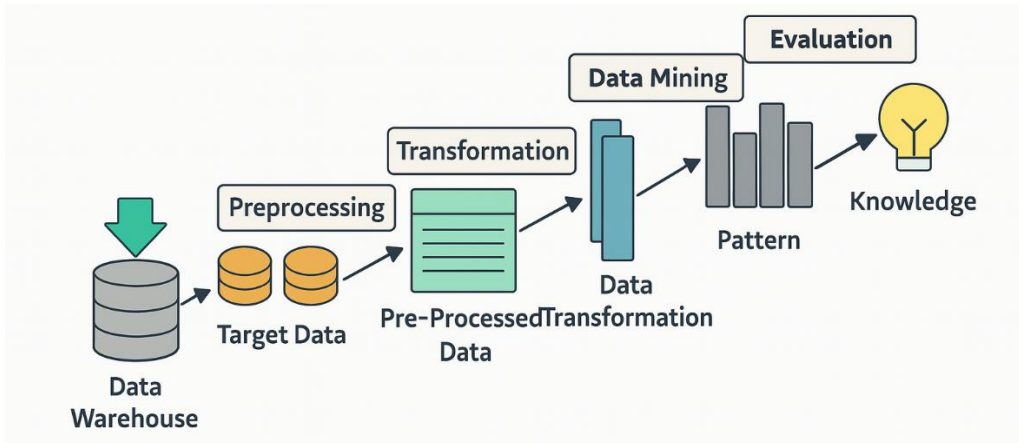
დიდი მოცულობის ტექსტური მონაცემებიდან ცოდნის მოპოვების ფაზები (ჩვენი მიდგომა)

განვიხილოთ ცოდნის მოპოვების პროცესი დიდი ტექსტური მონაცემთა ნაკრებებიდან ლამბდა არქიტექტურის გამოყენებით. ჩვენი მიდგომის მიხედვით, ტექსტური მონაცემთა ნაკრებიდან ცოდნის მოპოვების პროცესი შედგება ოთხი ფაზისგან, როგორც ეს მოცემულია სურათ 4.4-ზე:



სურ. 4.4. ცოდნის მოპოვების პროცესის ოთხი ფაზა (წყარო: ავტორი).

- გაყოფის ფაზა – ამ ეტაპზე დიდი მოცულობის ტექსტური მონაცემთა ნაკრები იშლება ფრაგმენტებად.
- მეორე ფაზა წარმოადგენს KDD პროცესს, როდესაც ტექსტუალური ფრაგმენტებიდან ფორმირდება ცოდნის გრაფული ფრაგმენტები (იხ. სურათი 4.4).
- მესამე ფაზაში სრულდება ცოდნის მოდელის აწყობის პროცესი ცოდნის გრაფული ფრაგმენტებიდან, KDD პროცესის შესაბამისად (იხ. სურათი 4.5). KDD პროცესის მოდელი მოიცავს შემდეგ ეტაპებს, რომელთა თითოეული შემავალი მონაცემი წარმოადგენს წინა ეტაპის გამომავალს. პროცესი ხორციელდება იტერაციულად (ანალიტიკოსის მხრიდან საჭიროების შემთხვევაში უკუკავშირის ჩართვით) და ინტერაქტიულად (ხაჩიძე 2024, 97).



სურ. 4.5. KDD პროცესი (წყარო: ავტორი).

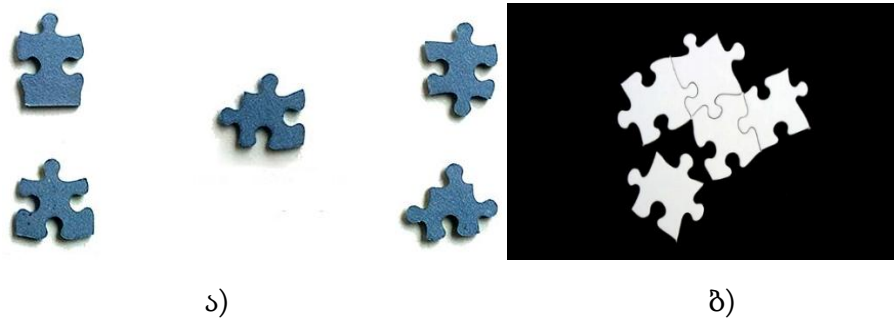
– მეოთხე ფაზა – როდესაც ცოდნის გრაფული მოდელის ოპტიმიზაცია ხორციელდება სინერგიული და დამატებითი (დოკინგის) მიდგომების მეშვეობით.

დიდი მოცულობის მონაცემთა მოპოვების პროცესის დასაჩქარებლად, ეფექტური გზაა სასწავლო პროცესის პარალელური განაწილება სხვადასხვა კომპიუტერულ სისტემაზე. მოცემული ჩარჩო სისტემა შექმნილია ისე, რომ ძალიან დიდი მოცულობის ტექსტური მონაცემები პარალელურ რეჟიმში დაამუშავოს, სამუშაოს დაყოფით დამოუკიდებელ ამოცანებად (ხაჩიძე 2024, 97).

ცოდნის მოპოვებისას, დიდი მოცულობის ტექსტური მონაცემებისგან, გამოიყენება პარალელური გამოთვლის პრინციპებზე დაფუძნებული ალგორითმები, როდესაც მონაცემთა დიდი ნაკრები იყოფა რამდენიმე ქვეჯგუფად და მათზე დამოუკიდებლად ხორციელდება მოპოვების ალგორითმების გამოყენება. თითოეული ქვეჯგუფის დამუშავების შედეგად მიიღება ცოდნის გარკვეული ფრაგმენტი. აღნიშნული ფაზა წარმოადგენს ცოდნის საერთო მოდელის აგების პროცესს ამ ფრაგმენტების საფუძველზე, პარალელური გამოთვლების გამოყენებით.

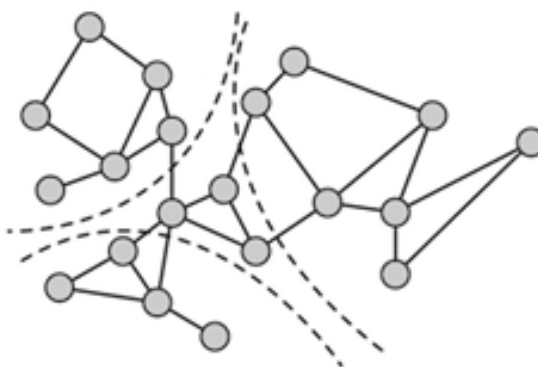
და ბოლოს, ტექსტური მონაცემებიდან მოპოვებული ცოდნის გრაფული მოდელის ოპტიმიზაცია ხორციელდება ენტროპიის მინიმიზაციის, და არა მისი ელიმინაციის მიზნით. მოპოვებული ცოდნის მოდელის ოპტიმიზაცია წარმოადგენს ცალკე სამეცნიერო ამოცანას, რომელიც სცდება ამ სტატიის ფარგლებს. შესაბამისად, შევიფარგლებით ცოდნის ფრაგმენტებიდან შეკრების ამოცანით, რომელსაც გარკვეული ანალოგიები გააჩნია როგორც მოზაიკური თავსატეხების აგებაში (*იხ. სურათი 4.6 ა, ბ*), ისე მოლეკულური ან მედიკამენტური მოდელების პროცესებში (ხაჩიძე 2024, 97).

თავსატეხების ფორმებისგან განსხვავებით (*სურათი 4.6 ა*), ცოდნის ფრაგმენტების რაოდენობა უკიდურესად დიდია, თუმცა მათი გაგება შესაძლებელია როგორც აქსონურ-დენდრიტული კავშირის მოდელის სახითაც. ამავდროულად, მთავარი კრიტერიუმი შეკრების პროცესში არის ფრაგმენტთა შესაბამისობისა და სინერგიის მაქსიმიზაცია.



სურ. 4.6. jigsaw puzzles (წყარო: ავტორი).

რაც შეეხება მეორე ანალოგიას, აქ წარმოდგენილია გაცილებით მრავალფეროვანი ფორმები მათი აქსონურ-დენდრიტული მოდელების მოლეკულების სახით, რაც მნიშვნელოვნად ზრდის ფრაგმენტთა შეერთების ვარიანტების რაოდენობრივ ცვალებადობას. რა თქმა უნდა, უცვლელად რჩება შეკრების მთავარი კრიტერიუმი - ფრაგმენტების შეთავსებადობისა და სინერგიის მაქსიმიზაცია (ხაჩიძე 2024, 98).



სურ. 4.7. (წყარო: ავტორი).

ამ თავში ვცადეთ ტექსტური მონაცემების ფრაგმენტის წარმოდგენა „ინფორმაციული მოლეკულის“ ფორმით. პირველ რიგში, გამოვკვეთეთ დესკრიპტორების პრობლემა, რომლებიც წარმოადგენს ტექსტური მონაცემების ერთობლიობას. სწორედ მათი მეშვეობით უნდა აგებულიყო ცოდნის გრაფზე დაფუძნებული მოდელი. ამ ამოცანაში ძირითადი სირთულე წარმოადგენს ფრაგმენტების შეთავსების პრობლემას. თავდაპირველად ვგეგმავდით კონვოლუციური ნეირონული ქსელის (CNN) გამოყენებას, სადაც თავიდანვე ჩავრთეთ რამდენიმე კონვოლუციური ფენა (ხაჩიძე 2024, 98).

ზოგადად, CNN ეფექტურად მუშაობს მაღალგანზომილებიან ტენზორებთან, თუმცა ჩვენს შემთხვევაში, მონაცემთა ფრაგმენტები წარმოადგენენ მაღალგანზომილებიან გრაფებს, რაც წარმოაჩენს CNN-ის მსგავსი რთული მოდელების არაეფექტიანობას მოცემულ კონტექსტში. ამ ნაწილში გამოვკვეთეთ ცოდნის ორი განსხვავებული ტიპი: სინერგიული ცოდნა და კომპლემენტარული (დოკინგის) ცოდნა, რომლებიც ცალ-ცალკე იწვევს განსხვავებულ შედეგებს ცოდნის შექმნის თვალსაზრისით (ხაჩიძე 2024, 98).

ენტროპია თერმოდინამიკაში და ინფორმაციის თეორიაში - თერმოდინამიკური ენტროპიის სტატისტიკური განსაზღვრება ემთხვევა კლავიუსის კლასიკურ ენტროპიას, რომელიც ფიზიკური სისტემებისთვის, ჩვეულებრივ, აღნიშნულია სიმბოლოთი S . მოცემული ნაწილი იკვლევს, რა კავშირები არსებობს ამ ორ ცნებას შორის და რამდენად შესაძლებელია მათი ერთობლივად განხილვა (ხაჩიძე 2024, 98).

სადაც:

$$dE = -pdV + TdS$$

$$S = -k \sum_i p_i \log p_i,$$

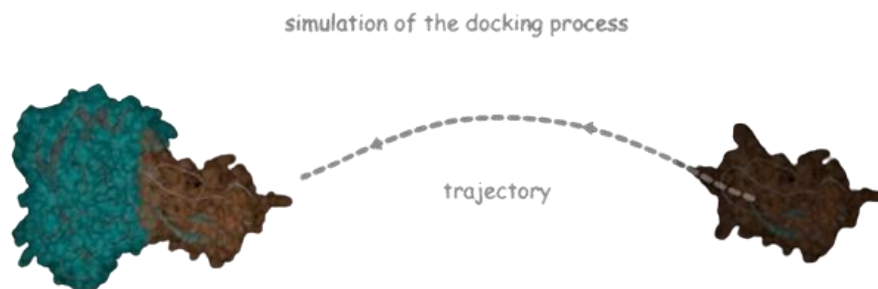
სადაც p_i აღნიშნავს მიკრომდგომარეობას i არსებობის ალბათობას, რომელიც აღებულია წონასწორობის ენსამბლიდან. ინფორმაციის თეორიაში ენტროპიის განსაზღვრის გამოსახულება, რომელიც ჩამოყალიბებულია კლაუდ შენონის (E. Shannon) მიერ, შემდეგი ფორმისაა:

$$H = - \sum_i p_i \log p_i,$$

სადაც p_i წარმოადგენს შეტყობინების m_i -ის არსებობის ალბათობას, რომელიც აღებულია შეტყობინებათა სივრციდან M . თერმოდინამიკური ენტროპიის სტატისტიკური განსაზღვრება მოლეკულურ მოდელირებაში ემთხვევა კლავიუსის კლასიკურ ენტროპიას. რაც შეეხება ცოდნის შეკრების ამ ორი ძირითადი სტრატეგიის ახალ მიდგომას, იგი ითვალისწინებს მათ ერთობლივ ან კომპლექსურ გამოყენებას. მოლეკულური დოკინგისგან განსხვავებით, სადაც მოლეკულების შეერთებისას ძირითადად გამოიყენება ენერჯის მინიმუმი, როგორც კრიტერიუმი

თერმოდინამიკის პირველ კანონის შესაბამისად, ჩვენს შემთხვევაში — ანუ ტექსტური ფრაგმენტების შეერთებისას - შევთავაზეთ შენონის ენტროპიის ინფორმაციული ვერსია.

მოცემული სტატია გვთავაზობს განსხვავებულ და უფრო დინამიკურ მიდგომას, თუ რას წარმოადგენს სტრატეგიული ალიანსების საბოლოო მიზანია სინერგია. ცოდნის სინერგიის მნიშვნელობა ნათლად გამოიხატება ფრაგმენტების ინტეგრაციისას; თუმცა ცოდნის ინჟინერია ძირითადად ორიენტირებულია ცოდნის შეთავსებადობასა და ცოდნის დამატებადობაზე (დოკინგი).



სურ. 4.8. დოკინგის პროცესის სიმულაცია (წყარო: ავტორი).

ცოდნის ფრაგმენტებიდან ცოდნის მოდელის აწყობის მიზანია ცოდნის მართვაში (KM) თავსებადობისა და ურთიერთშემავსებლობის განხილვა. მიუხედავად ამისა, თუ ისინი კარგად არიან ინტეგრირებულნი, ისინი ორი ურთიერთშემავსებელი პროცესია.

სტატიაში განხილულია დიდი მოცულობის მონაცემების მოპოვების ახალი მიდგომა, კერძოდ კი ცოდნის მოპოვების პროცესი დიდი ტექსტური მონაცემთა ნაკრებიდან ლამბდა არქიტექტურის გამოყენებით. ჩვენი მიდგომის შესაბამისად, ტექსტური მონაცემებისგან ცოდნის მოპოვების პროცესი მოიცავს შემდეგ ფაზებს: დიდი ტექსტური მონაცემების გაყოფა; ტექსტუალური ფრაგმენტებიდან ცოდნის გრაფული ფრაგმენტების ფორმირება; ცოდნის გრაფული ფრაგმენტებიდან ცოდნის მოდელის აწყობა. ამ მიდგომის ძირითადი სპეციფიკა მდგომარეობს იმაში, რომ ცოდნა წარმოადგენილია გრაფული მოდელის სახით, ასევე ცალკეული გრაფული ფრაგმენტებიდან ერთიანი ცოდნის მოდელის აგებაში. ამ ორი კონცეფციის ერთობლივი გამოყენება, დიდი მონაცემების პარალელურ დამუშავებაში ლამბდა არქიტექტურის კონტექსტში, შესაძლებელს ხდის ფართომასშტაბიანი მოპოვებითი ამოცანების შესრულებას (ხაჩიძე 2024, 98).

4.2 მოლეკულური დოკინგი და სტრუქტურაზე დაფუძნებული დიზაინის სტრატეგიების მიმოხილვა

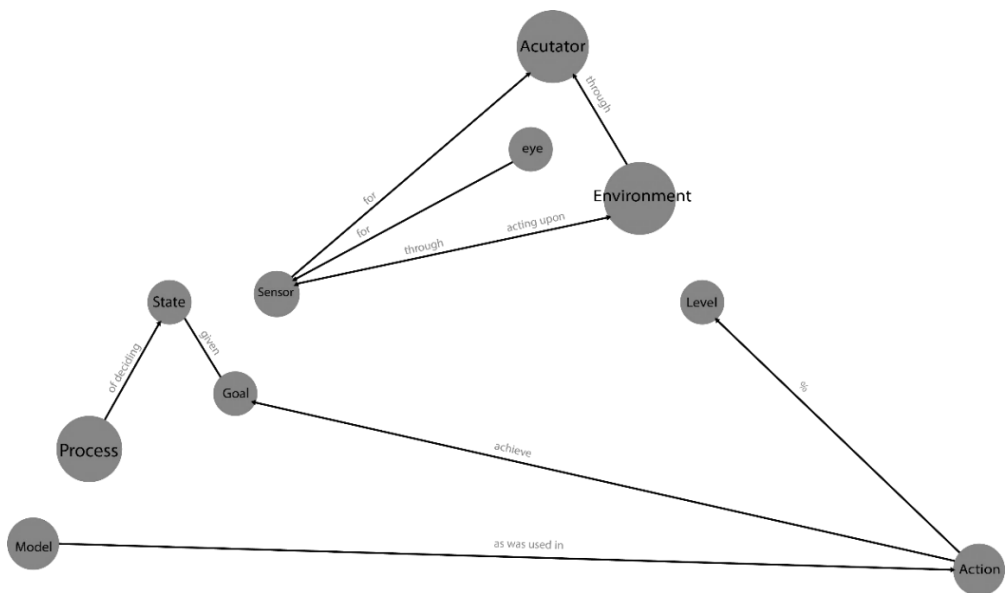
4.2.1. Drug design მიდგომა ცოდნის აგების პროცესში

ხელოვნურ ინტელექტში, საყოველთაო ცოდნა განისაზღვრება, როგორც სამყაროს შესახებ იმ საბაზისო ფაქტებისა და გაგებების ფართო ერთობლიობა, რაც ინდივიდს ჩვეულებრივ უნდა ჰქონდეს. მაგალითად, იმის ცოდნა, რომ წყალი იყინება 0°C ტემპერატურაზე ან რომ ჩიტებს ფრენა შეუძლიათ, წარმოადგენს საყოველთაო ცოდნის მაგალითებს. ასეთი ტიპის ცოდნა ხშირად წარმოდგენილია ცოდნის გრაფებში, როგორც დისკრეტული ფაქტების კოლექცია. თითოეული ფაქტი, ჩვეულებრივ, წარმოდგენილია ტრიპლეტის სახით, ეს არის სამნაწილიანი წარმოდგენა, რომელიც მოიცავს სუბიექტს, პრედიკატს და ობიექტს. ტრიპლეტი კოდირებას უკეთებს ერთ-ერთ ურთიერთობას (მაგ.: (წყალი, აქვს თვისება, დუდილის ტემპერატურა – 100°C)), რაც ფუნქციურად ატომურ ფაქტს წარმოადგენს. ეს ტრიპლეტები ქმნის ცოდნის გრაფების ფუნდამენტურ სტრუქტურებს, ანალოგიურად, როგორც ატომები ან მარტივი მოლეკულები ქიმიამში. როდესაც საყოველთაო ცოდნა სტრუქტურირდება მრავალ პატარა და გადამოწმებად ტრიპლეტად, იქმნება სემანტიკური გრაფი, რომელზეც მანქანებს შეუძლიათ გონივრულად მუშაობა. თითოეული ტრიპლეტი წარმოადგენს ცოდნის ფრაგმენტს, და დიდი მოცულობის ცოდნის გრაფში, ასეთი მრავალი ფრაგმენტი ურთიერთდაკავშირებულია საერთო სუბიექტებითა ან ცნებებით. ეს ურთიერთდაკავშირებული ფაქტების ქსელი წარმოადგენს საყოველთაო ცოდნის მოდელს, რომელიც ერთდროულად არის როგორც ადამიანისთვის წაკითხვადი, ისე მანქანისთვის დამუშავებადი. ტრიპლეტების დეტალებში ჩაშლილი სტრუქტურა უზრუნველყოფს ცოდნის მოდულურობას: როგორც რთული მოლეკულები მარტივი ატომებისგან შედგება, ასევე რთული ცოდნა წარმოიქმნება მარტივი ფაქტების ერთმანეთთან დაკავშირებით.

სტრუქტურაზე დაფუძნებული დიზაინი

ცოდნის ინჟინერიაში, განსაკუთრებით კი დიდი ტექსტური მასივებიდან ცოდნის გრაფების აგების პროცესში, შეგვიძლია ცოდნა წარმოვადგინოთ მცირე ცოდნის ფრაგმენტებით, ესენი წარმოადგენენ ტექსტიდან გამოყოფილ ატომურ

ფაქტებს ან ტრიპლეტებს. დიდი არასტრუქტურირებული ტექსტი (მაგალითად, სახელმძღვანელო ან დოკუმენტების კრებული) შეიძლება დაიყოს უფრო მარტივად შესასწავლ ფრაგმენტებად (წინადადებები ან წინადადების ნაწყვეტები). თითოეული ასეთი ტექსტური ფრაგმენტიდან შესაძლებელია ერთი ან რამდენიმე ტრიპლეტის გამოყოფა (სუბიექტი–პრედიკატი–ობიექტი), რომლებიც ასახავენ ამ ნაწყვეტის ფაქტობრივ შინაარსს. თითოეული ასეთი ტრიპლეტი წარმოადგენს მცირე მოლეკულურ ფრაგმენტს ცოდნის გრაფში, რომელიც ასახავს კონკრეტულ ურთიერთობას ან ფაქტს ტექსტიდან. მაგალითად, წინადადებიდან - „თბილისი არის საქართველოს დედაქალაქი და მისი მოსახლეობაა 1 მილიონი“, შეიძლება მივიღოთ შემდეგი ტრიპლეტები: (თბილისი, დედაქალაქია, საქართველო), (თბილისი, მოსახლეობა, 1 მილიონი). ეს არის ცოდნის ფრაგმენტები — მცირე, ცალკეულად მარტივი ფაქტები, თუმცა ერთად ისინი ქმნიან დიდი მასშტაბის ცოდნის სტრუქტურას. სურათი 4.9 ასახავს ცოდნის გრაფს, რომელიც განკუთვნილია ტრიპლეტების ვიზუალიზაციისთვის, მათ შორის ერთი პარაგრაფის ერთეულებისა და ურთიერთობების წარმოსაჩენად.



სურ. 4.9. ცოდნის გრაფი ერთი აბზაცისთვის (წყარო: ავტორი).

Drug Design-ის პროცესში, ფრაგმენტები ერთმანეთთან ერთიანდება იმის მიხედვით, თუ როგორ ჯდება ისინი მთლიან სტრუქტურაში; ხოლო ცოდნის ინჟინერიაში, ტრიპლეტები დაკავშირებულია საერთო ერთეულების ან ცნებების საფუძველზე, რაც ქმნის უფრო დიდ კონტექსტს (ონტოლოგიას ან ცოდნის სქემას).

მაგალითად, თბილისის შესახებ ორი ტრიპლექტი შეიცავს საერთო ერთეულს, „თბილისი“, რაც იძლევა მათ შორის, გრაფში კავშირის შესაძლებლობას: (თბილისი → capitalOf → საქართველო) და (თბილისი → population → 1 მილიონი). მიერთების წერტილი განისაზღვრება საერთო ელემენტით (ამ შემთხვევაში, საერთო ერთეულით). მრავალი ასეთი ტრიპლექტის გაერთიანებით იწყობა ერთიანი ცოდნის გრაფი, სადაც კვანძები (ერთეულები) მოქმედებენ, როგორც საყრდენი სტრუქტურები, ხოლო პრედიკატები, როგორც დამაკავშირებელი წერტილები, რომლებიც ფრაგმენტებს ერთმანეთთან აკავშირებენ. ცოდნის გრაფის საერთო „დიზაინი“ ხშირად ონტოლოგიის მიხედვით ფორმირდება, რაც მედიკამენტების დიზაინში ცილის სტრუქტურის ანალოგიურ როლს ასრულებს, ის წარმოადგენს მიდგომას, თუ როგორ უნდა შეერთდეს ელემენტები ერთმანეთთან. მაგალითად, თუ ონტოლოგია ადგენს, რომ ქალაქს შეიძლება ჰქონდეს ისეთი ატრიბუტები, როგორცაა მოსახლეობა და ქვეყანა, მაშინ ეს სქემა წარმართავს ფაქტების სწორად შეკრების პროცესს რათა „დედაქალაქი“ დაკავშირდეს სწორად ქვეყანასთან, ხოლო „მოსახლეობა“ შესაბამის რიცხვთან და ა.შ (კუმარი and დინაკრან 2020, 4).

აღსანიშნავია, რომ ცოდნის ფრაგმენტების მიერთების პროცესი იტერაციულია და ხშირად საჭიროებს ოპტიმიზაციას. საწყისმა ამოღებამ შეიძლება გამოიწვიოს ინფორმაციის არაერთგვაროვანი ან თუნდაც ზედმეტი ნაწილები, რომელთა შეჯერებაა საჭირო, რომლებიც შემდგომში საჭიროებს შერწყმასა და გადამუშავებას. ამ პროცესს ეწოდება ცოდნის შერწყმა (knowledge fusion), როდესაც ერთმანეთთან გადამკვეთი ან დუბლირებული ტრიპლექტები ერთიანდება, ხოლო კონფლიქტური ტრიპლექტების გადაჭრა ხდება, რათა გაუმჯობესდეს მთლიანობის დონე. ეს პროცესი საჭიროა, რათა ფრაგმენტების შეერთება ოპტიმიზირდეს და გაუმჯობესდეს საერთო შეთავსება და აღმოიფხვრას არასასურველი ურთიერთქმედებები. ცოდნის შერწყმის პროცესში შესაძლებელია ციკლური კორექტირება მაგალითად, შესაძლოა ჩაემატოს გამოტოვებული კონტექსტი, ამოვარდეს შეცდომით აგებული ტრიპლექტების რაოდენობა, ან განიმარტოს არაერთმნიშვნელოვანი ელემენტები (მაგალითად, ერთი კონტექსტის „თბილისი“ იგივეა, რაც მეორე კონტექსტის შემთხვევაში მოცემული „თბილისი“, და არა ერთი და იმავე სახელის მქონე სხვადასხვა ერთეული).

იტერაციული პროცესის თითოეულ ციკლში, ერთი ეტაპის შედეგი (მაგალითად, ნაწილობრივი გრაფი) ხდება შემდეგი ეტაპის შემავალი მონაცემი, ხოლო ალგორითმები იყენებენ უკუკავშირს, ანალიტიკისთვის შედეგებს რათა გადაამოწმონ შედეგები, შეცდომების გამოსასწორებლად ან გამოტოვებული ელემენტების დასამატებლად. ამგვარი იტერაციული შერწყმა უზრუნველყოფს, რომ საბოლოო ცოდნის გრაფი არ იყოს უბრალოდ მოპოვებული ფაქტების მექანიკური გაერთიანება, არამედ იყოს კომპლექსური, ინტეგრირებული ცოდნის სტრუქტურა (კუმარი and დინაკრან 2020, 5).

ცოდნის გრაფების ტექსტიდან აგება მექანიკურ რეჟიმში საკმაოდ შრომატევადია, თუმცა თანამედროვე ტექნოლოგიები შესაძლებელს ხდის მთელი ამ პროცესის ავტომატიზაციას. ისევე როგორც კომპიუტერული ინსტრუმენტები (მაგალითად, მოლეკულური დოკინგი და ვირტუალური სკრინინგი) ავტომატიზაციას ახდენენ Drug design პროცესში, ასევე ბუნებრივი ენის დამუშავებისა (NLP) და მანქანური სწავლების ხელსაწყოები იძლევა საშუალებას ცოდნის მოპოვების ავტომატიზაციისათვის. ავტომატიზებული ცოდნის გრაფის აგების ტიპური სისტემა მოიცავს რამდენიმე თანმიმდევრულ ეტაპს, რომელთა მეშვეობითაც სრულდება მოპოვება, სტრუქტურირება და შეკავშირება:

- *ტექსტის სეგმენტაცია* – დიდი დოკუმენტები იშლება პატარა ერთეულებად (პარაგრაფებად, წინადადებებად ან წინადადების ნაწილებად). თითოეული სეგმენტი დამოუკიდებლად მუშავდება ფაქტების გამოსაყოფად.
- *ინფორმაციის მოპოვება* – თითოეული ტექსტური სეგმენტიდან გამოითხოვება შესაბამისი ერთეულები და მათ შორის არსებული ურთიერთობები ბუნებრივი ენის დამუშავების (NLP) მეთოდების გამოყენებით. გამოიყენება ისეთი ტექნოლოგიები, როგორცაა: დასახელებული ერთეულების ამოცნობა (Named Entity Recognition – NER); ურთიერთობების ამოცნობა (Relation Extraction), იმის დასადგენად, თუ როგორ უკავშირდებიან ერთეულები ერთმანეთს ერთსა და იმავე წინადადებაში. ამ ეტაპის შედეგია დაუმუშავებელი ტრიპლეტების კოლექცია (*სუბიექტი, პრედიკატი, ობიექტი*).

- *ცოდნის შერწყმა და დაკავშირება* – მოპოვებული ტრიპლეტები ხშირად ეხება ერთსა და იმავე რეალურ ერთეულებს ან ცნებებს, თუმცა სხვადასხვა ტერმინით (მაგალითად, „CPU“ და „Central processing Unit“). ამ ფაზაში სისტემა ტრიპლეტებში არსებულ ერთეულებს აცნობიერებს და აკავშირებს ცოდნის ბაზაში არსებულ კანონიკურ იდენტიფიკატორებთან (მაგალითად, ორივე ტერმინის, „CPU“ და „Central processing Unit“ დაკავშირდება ცოდნის გრაფში ერთსა და იმავე კვანძთან) [pmc.ncbi.nlm.nih.gov]. ასევე ხორციელდება ტრიპლეტების შერწყმა, რომლებიც სემანტიკურად ეკვივალენტურია ან ერთსა და იმავე ერთეულს უკავშირდება. ეს შერწყმის ეტაპი აერთიანებს ინდივიდუალურ ფაქტებს ერთიან გრაფულ სტრუქტურაში.
 - *ვალიდაცია და დახვეწა* – აგებული ცოდნის გრაფი შემოწმდება შედგენილობისა და სიზუსტის თვალსაზრისით. მაღალი დონის სისტემებში ეს ფაზა შეიძლება მოიცავდეს ფაქტების გადამოწმებას სხვა წყაროებთან შედარებით. გრაფის დახვეწა შესაძლებელია იტერაციულად, დაკარგული ფაქტების დამატებით, არასწორ ფაქტთა ამოღებით.
- ეს პროცესი ეფუძნება Drug Discovery მიდგომას, სადაც ახალი „ფრაგმენტები“ (ფაქტები) დროთა განმავლობაში ემატება ცოდნის გრაფს, ხოლო თვითონ ეს ცოდნის გრაფი მუდმივად ოპტიმიზირდება.

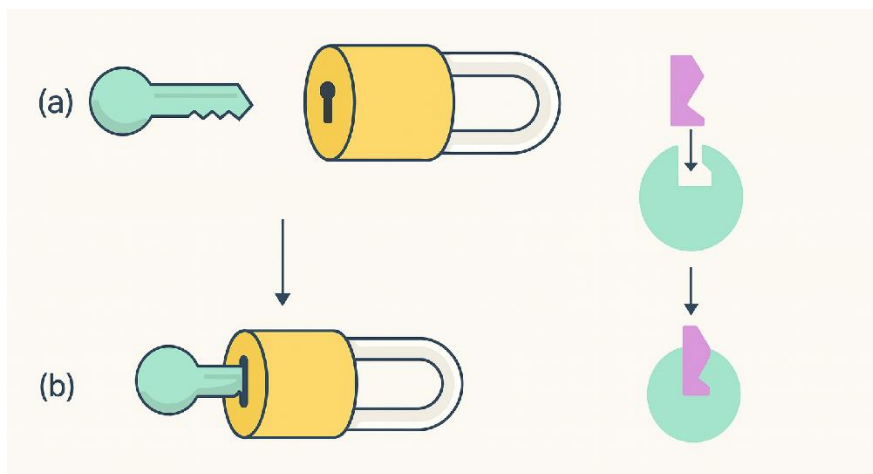
4.2.2. Docking მიდგომა ცოდნის აგების პროცესში

მოლეკულური დოკინგი, ანუ ლიგანდის (პატარა მოლეკულის) და რეცეპტორის (როგორც წესი, ცილის) შეერთების კომპიუტერული სიმულაცია შეიძლება აიხსნას კავშირის და ინტეგრაციის ანალოგიებით. როგორც ხომალდი ფრთხილად მიადგება ნავსადგურს, რათა სათანადო "დოკზე დადგეს", ისე ლიგანდი ახდენს ნავიგაციას ცილის მოლეკულურ ზედაპირზე, რათა იპოვოს მისთვის შესაფერისი მიზმის ადგილი. იგივე პრინციპი გვხვდება ნეირომეცნიერებაშიც, როცა ერთი ნეირონის აქსონის დაბოლოება "დოკირდება" მეორე ნეირონის დენდრიტულ ხერხემალზე სინაფსის პროცესში, რაც ინფორმაციის გადაცემას უზრუნველყოფს. ყველა ამ შემთხვევაში, წარმატებული დოკინგი მოითხოვს მონაწილე მხარეებს შორის

თავსებადობას: გემი პორტს უნდა შეესაბამებოდეს ფორმით და ზომით, ლიგანდმა უნდა შეავსოს შემაკავშირებელი ჯიბე და ნეირონის აქსონი უნდა დაუკავშირდეს რეცეპტორულ დენდრიტს.

მოცემული ანალოგია გამოვიყენოთ, რათა დავახასიათოთ მოლეკულური დოკინგის ძირითადი ცნებები, როგორცაა: სკორინგის ფუნქციები, კონფორმაციული მოქნილობა, ოპტიმიზაციის ალგორითმები, ანალოგიურად, ცოდნის ფრაგმენტული ნაწილების ერთ თანმიმდევრულ მთლიანობად გაერთიანებისა. ამ შემთხვევაში შეთავსებადობა (compatibility) არის გადამწყვეტი ფაქტორი, როდესაც ფრაგმენტები კარგად ერგება ერთმანეთს, ისინი ქმნიან სტაბილურ კომპლექსს (ხომალდი-ნავსადგურის დაკავშირება, ცილისა და ლიგანდის კომპლექსი, თუ ნეირონული წრე). ხოლო შეთავსებადობის არარსებობა წარმოშობს არეულობას (*მაღალ ენტროპიას*), რაც ხელს უშლის ინტეგრაციას.

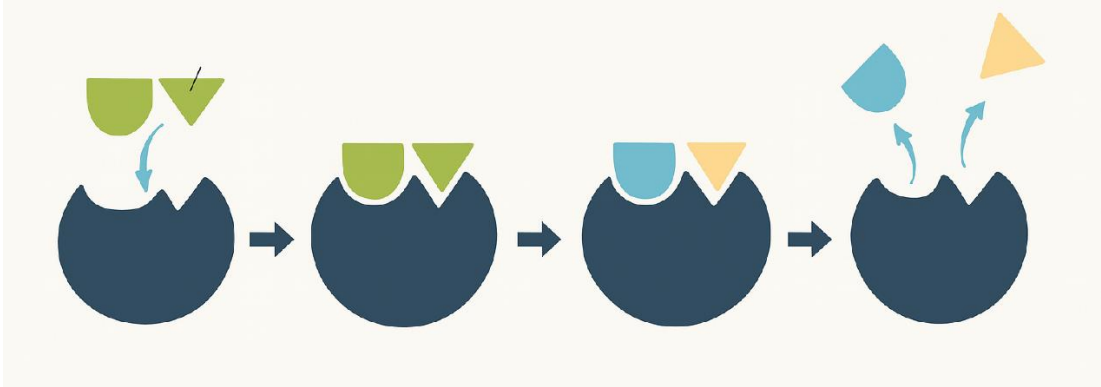
დოკინგის პროცესში, ორი ან მეტი ფრაგმენტი ერთმანეთს უერთდება და ქმნის სტაბილურ კომპლექსს. კლასიკური ხედვის მიხედვით, რომელიც უკავშირდება ემილ ფიშერის „საკეტი და გასაღების“ მოდელს, ფრაგმენტი წარმოადგენს საკეტს, ხოლო ცოდნა გასაღებს, რომელიც ზუსტად უნდა მოერგოს მოცემულ სტრუქტურას, რათა შეიქმნას შეერთება.



სურ. 4.10. „საკეტი და გასაღების“ მოდელი (წყარო: ავტორი).

უფრო დახვეწილ ვარიანტს გვთავაზობს კომლანდის „ინდუცირებული მორგება“ (induced-fit) მოდელი, სადაც როგორც საკეტი (ფრაგმენტი), ისე გასაღები (ცოდნა) შესაძლოა შეიცვალოს ან მოერგოს უკეთესი კავშირის მისაღწევად (რავალი და

განათრა 2022, 13 - 14). ეს ორი მოდელი ასახავს იმას, თუ როგორ ებმის ცოდნის ელემენტები ან იდეები არსებულ სისტემებს: ზოგჯერ ახალი იდეა სრულად ჯდება არსებულ ჩარჩოში (*lock-and-key*), ხოლო სხვა შემთხვევებში ორივე ჩარჩოც და იდეაც გარდაიქმნება ურთიერთშეთავსების მისაღწევად (*induced-fit*).



სურ. 4.11. „ინდუცირებული მორგება“ (*induced-fit*) (წყარო: ავტორი).

იგივე პრინციპს ემყარება გემისა და ნავსადგურის ანალოგია: ნავსადგურს (*ფრაგმენტს*) გააჩნია კონკრეტული ფორმა, ხოლო ცოდნას (*knowledge*) უწევს მანევრირება ან საკუთარი სტრუქტურის კორექტირება, რათა დადგეს შესაბამის დოკზე. დოკინგის არსი მდგომარეობს სწორ შეთავსებასა და სტაბილურ შეერთებაში, რაც შემდგომში შესაძლებელს ხდის ინფორმაციის გადაცემას ან ქმედების განხორციელებას.

სკორინგის ფუნქციები: შეთავსებადობის შეფასება. მას შემდეგ, რაც ცოდნის ფრაგმენტი „მიაღწევს“ კონკრეტულ ფრაგმენტულ ადგილს, როგორ განვსაზღვროთ, წარმატებული იყო თუ არა ეს დოკინგი? ამისათვის შეგვიძლია გამოვიყენოთ სკორინგის ფუნქციები, როგორც შეთავსებადობის ტესტი. დოკინგის ალგორითმები ქმნიან შეერთების მრავალი სავარაუდო პოზიციას (ე.წ. poses) და შემდეგ თითოეულს აფასებენ შეთავსებადობის ხარისხის მიხედვით, სკორინგის ფუნქციების მეშვეობით. სკორინგის ფუნქცია არსებითად წარმოადგენს შეთავსებადობის მათემატიკურ მაჩვენებელს, რაც უმეტესად გამოხატავს ცოდნისა და ფრაგმენტის კომპლექსის შეკავშირების ენტროპიის შეფასებას. საერთო ჰიპოთეზის თანახმად, ყველაზე დაბალი ენერჯიის პოზა შეესაბამება ოპტიმალურ ბმის მოდელს. სკორინგის ფუნქციები რაოდენობრივად აფასებს ძირითად ინტერაქციებს მაგალითად, როგორიცაა:

ენტროპიის დაკარგვა შეერთებისას. ყველა ამ ფაქტორის გათვალისწინებით, სკორინგის ფუნქცია თითოეული პოზისთვის გამოითვლის ერთიან „შეთავსებადობის“ მნიშვნელობას ანუ fitness score-ს. მაღალი შეთავსებადობა (დაბალი ენტროპია) ნიშნავს, რომ ცოდნა ზუსტად შეესაბამება ფრაგმენტის ფორმასა და შინაარსს.

სკორინგის ფუნქციას შეგვიძლია ვუწოდოთ შეთავსებადობის ინდექსი ცოდნის ინტეგრაციის კონტექსტში. როდესაც ვცდილობთ ინფორმაციის ინტეგრირებას სხვადასხვა წყაროდან ან დისციპლინიდან, ჩვენ ქვეცნობიერად ვაფასებთ, რამდენად კარგად ერწყმის კონცეფციები ერთმანეთს — მაღალი შეფასება ნიშნავს, რომ იდეები ერთმანეთს აძლიერებენ მინიმალური წინააღმდეგობებით (დაბალი „ენტროპიით“), ხოლო დაბალი შეთავსებადობა ქმნის განცდას, რომ იდეები ძალითაა შერწყმული ან არასტაბილურია (*მაღალი ენტროპია, მაღალი შეუთავსებლობა*). ადამიანის ტვინში არსებული ნეირონების ანალოგიურად, სინაფსური კავშირის სიმტკიცე შეიძლება ჩაითვალოს ბიოლოგიურ სკორინგის ფუნქციად მხოლოდ მაშინ, როდესაც აქსონის ნეირომედიატორები სრულად ერგება დენდრიტის რეცეპტორებს, და სინაფსს გააჩნია სტრუქტურული მხარდაჭერა, სიგნალი გადაეცემა ძლიერად, რაც „დოკინგის“ წარმატებულობის მაჩვენებელია სიგნალის შეთავსებადობის თვალსაზრისით.

ცოდნის ინტეგრაციის პროცესში, ენტროპია შეიძლება ჩაითვალოს უწყესრიგობისა და შეუთავსებლობის საზომი, რომელიც საჭიროა მაქსიმალურად შემცირდეს სტაბილური შეერთების მისაღწევად. თერმოდინამიკაში ენტროპია აღნიშნავს მიკრონაკვეთების რაოდენობას ანუ კომპონენტების მოძრაობის თავისუფლების ხარისხს. დოკინგამდე, ცოდნის ფრაგმენტს შეიძლება ჰქონდეს მრავალი შესაძლო კონფორმაცია და ორიენტაცია (*მაღალი ენტროპია*), ხოლო თვითონ ფრაგმენტს შეიძლება ჰქონდეს მოქნილი რეგიონები, რომლებიც შესაძლოა იმყოფებოდეს მრავალ მდგომარეობაში. როდესაც ცოდნის ფრაგმენტი “დოკირდება” და ქმნის სტაბილურ კომპლექსს, ის ჩვეულებრივ კარგავს ამ თავისუფლებას, რაც ნიშნავს ენტროპიის შემცირებას.

რომ წარმოვიდგინოთ სცენარი, სადაც უამრავი დაშლილი ფაქტი ან იდეაა წარმოდგენილი, რაც გულისხმობს ცოდნის მაღალი ენტროპიის მდგომარეობას, სადაც ფრაგმენტები მწყობრად არ არის განლაგებული და არ არსებობს თანმიმდევრული სტრუქტურა. ამ ფრაგმენტების ინტეგრირება ერთიან ცოდნად გულისხმობს კავშირების ჩამოყალიბებას (*ანუ ფრაგმენტების „დოკინგს“*) ისე, რომ მთლიან სისტემაში არსებული ენტროპია შემცირდეს. ცოდნის ფრაგმენტებს შორის შეუთავსებლობა (*მაგალითად, წინააღმდეგობრივი მონაცემები*) გვევლინება როგორც ერთგვარი „ენტროპიული ბარიერი“, რომელიც აფერხებს ინტეგრაციას. მხოლოდ მაშინ, როცა მოიძებნება შეთავსებადი ინტერფეისი (*შეიძლება ეს იყოს ახალი იდეა ან ალტერნატიული ინტერპრეტაცია*), შესაძლებელია ფრაგმენტების შეკავშირება სტაბილურ სტრუქტურად, რაც ამცირებს საერთო ენტროპიას.

4.2.3. ცოდნის აგება როგორც Jigsaw Puzzle

Jigsaw Puzzle-ში და ხელოვნურ ინტელექტის სისტემებშიც კომპლექსური მთლიანობა მრავალი ფრაგმენტებისგან იგება. თავსატეხის (Puzzle) ამოხსნა კლასიკური NP-hard (გამოთვლითი სირთულის თეორიაში, გამოთვლით პრობლემას H ეწოდება NP-hard, თუ ყველა L ამოცანისთვის, რომლის ამოხსნაც შესაძლებელია არადეტერმინისტულ პოლინომურ დროში, არსებობს პოლინომური-დროში შემცირება L-დან H-მდე.) ამოცანა, რომელიც მოითხოვს ნაწილის მორგებას და სივრცულ პირობებში მათი შესაბამისი განლაგების პოვნას. ანალოგიურად, ხელოვნურ ინტელექტში ცოდნის აგება გულისხმობს მრავალ მონაცემთა ელემენტის (მონაცემთა წერტილები, ფაქტები, სემანტიკური ერთეულები, ცოდნის გრაფის კვანძები) გაერთიანებას ერთიან, თანმიმდევრულ სტრუქტურაში, ანუ „მთლიანი სურათი.“ თითოეული ცოდნის ფრაგმენტი ჰგავს თავსატეხის (Puzzle) ნაწილს: ის შეიცავს ნაწილობრივ ინფორმაციას და სრულ მნიშვნელობას მხოლოდ მაშინ იძენს, როდესაც სხვა ნაწილებთან სწორად არის დაკავშირებული. როგორც თავსატეხებში (Puzzle), ასევე ცოდნის აგებისას თავსატეხის განცალკევებული ნაწილები ბუნდოვანი შეიძლება იყოს (მაგალითად, „ცისფერი ცა“, ფრაგმენტი არ გვაწვდის საკმარის მინიშნებას), ამიტომ სწორი განლაგების განსაზღვრისთვის აუცილებელია

გლობალური კონტექსტი. თავსატეხის (Puzzle) ამოხსნის მიდგომა ერთგვარი ინტუიციური ჩარჩოა: ხელოვნურ ინტელექტს უწევს ცოდნის ინტეგრაციის ამოცანის „ამოხსნა“ იმით, რომ დაადგენს, რომელ ნაწილებს შორის არის შესაბამისობა, სწორად განალაგებს მათ და თანდათან ააწყობს სრულ ცოდნის სტრუქტურას („მოლიან სურათს“). ამ თავში განვიხილავთ თავსატეხის (Jigsaw Puzzle) კონცეფტუალურ მოდელს, რომელშიც ინტეგრირებულია პრინციპები გენეტიკური ალგორითმებიდან, ნეირონული კომბინატორული ოპტიმიზაციიდან, ენტროპიაზე დაფუძნებული სირთულის მეტრიკებიდან და ვექტორებზე დაფუძნებული სემანტიკური წარმოდგენებიდან. გვინდა წარმოვადგინოთ, თუ როგორ შეესაბამება კლასიკური თავსატეხის ამოხსნის სტრატეგიები მაგალითად როგორცაა, პირველ რიგში კიდების აწყობა, შუალედური სეგმენტების შექმნა, „ყურადღების“ გამოყენება შესატყვისობების მოსაძებნად და გლობალური შესაბამისობის უზრუნველყოფა, ცოდნის აგების ეფექტურ მიდგომებს ხელოვნურ ინტელექტში.

თავსატეხის ნაწილები, როგორც ცოდნის ფრაგმენტები

ზუსტად ისე, როგორც თავსატეხის ფრაგმენტები ერთობლივად ქმნის ვიზუალურ გამოსახულებას, ხელოვნური ინტელექტის ცოდნის ბაზებიც შედგება მზგავსი ფრაგმენტებისგან, რომლებიც ცოდნის “ფრაგმენტებს წარმოადგენენ”. ეს ცოდნის „ნაწილები“ შეიძლება სხვადასხვა ფორმით არსებობდეს:

- **დაუმუშავებელი მონაცემები ან ფაქტები** – ინდივიდუალური მონაცემთა წერტილები, რომლებიც დამოუკიდებლად არასრულფასოვანია;
- **სემანტიკური ერთეულები** – ცნებები, ტოკენები ან წინადადებები, რომლებიც შეიცავს მნიშვნელობას, მაგრამ საჭიროებენ კონტექსტს (თავსატეხის ფრაგმენტის მსგავსად, რომელიც მხოლოდ სურათის ნაწილს ასახავს);
- **ცოდნის ვექტორები (ემბედინგები)** – ცოდნის ფრაგმენტების განაწილებული წარმოდგენები ვექტორულ სივრცეში;
- **გრაფის კვანძები ან ქვეგრაფები** – ცოდნის გრაფის ან ონტოლოგიის ელემენტები, რომლებიც დაკავშირებულია სხვა ელემენტებთან, მაგრამ ჯერ კიდევ არ წარმოადგენს სრულად ინტეგრირებულ გრაფს.

თითოეული ფრაგმენტი წარმოადგენს მინიშნებას (ანალოგიურად ფრაგმენტის ფორმის, ფერის ან შტრიხების მსგავსად) იმის შესახებ, თუ როგორ შეიძლება მისი დაკავშირება სხვებთან. პრინციპულად მნიშვნელოვანია, რომ არც ერთი ცალკეული ნაწილი არ არის საკმარისი, სრული ცოდნა მხოლოდ მაშინ ჩნდება, როდესაც ფრაგმენტები სწორად შეესაბამება, სწორად განლაგდება და ერთიან სტრუქტურად შეკრული იქნება. თუ თავსატეხებში ნაწილები ერთმანეთს ერთმნიშვნელოვანი ზღვრის მიხედვით ებმება, ცოდნის შეკრებისას ფრაგმენტები ერთმანეთს უკავშირდება ლოგიკური ან სემანტიკური კავშირებით. თავსატეხის ორი ნაწილის თავსებადობა (ფერის თანმიმდევრულობა, ფორმის შესაბამისობა და სხვ.) პარალელურია ორი ცოდნის ფრაგმენტის სემანტიკური თავსებადობისა (მაგალითად, ორი ფაქტი, რომელიც ერთსა და იმავე ობიექტს ეხება, ან ორი გრაფის კვანძი, რომელსაც ერთმანეთთან შინაარსობრივი კავშირი აქვს). შესაბამისად, ცენტრალურ ამოცანას წარმოადგენს შესაბამისობის ფუნქციის განსაზღვრა, რომელიც შეაფასებს, რამდენად „ესადაგება“ ერთმანეთს ორი ფრაგმენტი. თავსატეხებზე ჩატარებული კვლევები ყურადღებას ამახვილებს ისეთ ფუნქციებზე, რომლებიც ნაწილის საზღვრებს ადარებს (ფერის გარდამავალი ტონები, ფორმის თანხვედრა); ანალოგიურად, ხელოვნურმა ინტელექტმა უნდა შეაფასოს სემანტიკური შესაბამისობა, მაგალითად, ვექტორული სიახლოვის ან რელაციური შეზღუდვების მიხედვით. საბოლოოდ, ცოდნის აგება გულისხმობს ისეთი ფრაგმენტების აღმოჩენას, რომლებიც მაღალი შესაბამისობით ხასიათდებიან, და მათი დაკავშირება ჰგავს თავსატეხის ნაწილების შეხამებას. მაგრამ, ეს პროცესი მრავალი მიზეზის გამო საკმაოდ რთულ ამოცანას წარმოადგენს: თუნდაც ზომიერი სიდიდის თავსატეხს გააჩნია ასტრონომიული რაოდენობის შესაძლო კომბინაციები, და მსგავსად, დიდი რაოდენობის ცოდნის ფრაგმენტების აგება შეიძლება მრავალი განსხვავებული გზით მოხდეს.

გენეტიკური ალგორითმების მიდგომა

ერთი-ერთი მძლავრი პარადიგმა დიდი ზომის თავსატეხების ამოხსნისთვის არის გენეტიკური ალგორითმების (GA) გამოყენება. GA-ზე დაფუძნებულ თავსატეხის ამოხსნის სისტემაში თითოეული სავარაუდო გადაწყვეტა კოდირდება ყველა ნაწილის

გარკვეული განლაგების სახით (მაგალითად, ნაწილების პერმუტაცია ან გრიდული განთავსება). შესაბამისობის შეფასების (fitness) ფუნქცია აფასებს, რამდენად სწორად არის აწყობილი სავარაუდო გადაწყვეტა. როგორც წესი, იგი აჯამებს მეზობელი ნაწილების წყვილთა შესაბამისობის ხარისხს. შემდგომ GA ავითარებს გადაწყვეტათა პოპულაციას შერჩევის, კროსოვერის (crossover) და მუტაციის გზით. მნიშვნელოვანია აღინიშნოს Sholomon და თანაავტორების მიერ შემუშავებული ინოვაციური კროსოვერის ოპერატორი, რომელიც აერთიანებს ორი „მშობელი“ გადაწყვეტიდან სწორად აწყობილ სეგმენტებს. სხვა სიტყვებით რომ ვთქვათ, თუ თავსატეხის გარკვეული ნაწილები სწორად არის აწყობილი სხვადასხვა მშობელ ნაწილებში, კროსოვერი საშუალებას იძლევა ამოიციოს და გააერთიანოს ეს სეგმენტები ერთ „შვილ“ გადაწყვეტაში ისე, რომ შეინარჩუნოს მათი სტრუქტურული მთლიანობა (შოლომონი, დავიდი and ნეითანიოუ 2013, 1770). ეს ევოლუციური სტრატეგია აღმოჩნდა ძალზე ეფექტური. მისი მეშვეობით ათიათასობით ნაწილისგან შემდგარი თავსატეხების ამოხსნა მაღალი სიზუსტით გახდა შესაძლებელი.

ცოდნის აგების ფარგლებში, გენეტიკური ალგორითმის მიდგომის შემთხვევაში შეიძლება ცოდნის სტრუქტურები (მაგალითად, ფაქტების შესაძლო განლაგებები ან კავშირები ცოდნის გრაფში) განვიხილოთ, როგორც ინდივიდები პოპულაციაში. შესაბამისობის შეფასების (fitness) ფუნქცია შეიძლება დაფუძნებული იყოს შეკრებილი ცოდნის თანმიმდევრულობასა და სისრულეზე. მაგალითად, შეიძლება უარყოფილი იქნას ურთიერთსაწინააღმდეგო კავშირები და მიღებული იქნას ლოგიკურად თანმიმდევრული ფრაგმენტების დაკავშირება. მრავალგვარი ჰიპოთეზის გენერირებისა და მათი კროსოვერის გზით, სისტემა შეძლებს გადაწყვეტილების სივრცის ეფექტიანად გამოკვლევას ისე, რომ არ „ჩარჩეს“ ადგილობრივ ოპტიმუმში (რაც ერთ-ერთი ცნობილი საფრთხეა მიტაცებული შეკრების სტრატეგიებისას). კროსოვერის ოპერაცია შეესაბამება ნაწილობრივ შეკრებილი ცოდნის მოდულების შერწყმას: თუ ერთი ჰიპოთეზა სწორად აერთიანებს ფაქტების ერთ ქვეჯგუფს (ცოდნის პაზლის „სეგმენტს“), ხოლო მეორე ჰიპოთეზა სხვა ქვეჯგუფს, კროსოვერი ამ ქვეწყვეტილების ინტეგრაციის შესაძლებლობას იძლევა. განსაკუთრებით მნიშვნელოვანია, როგორც Sholomon და თანაავტორები აღნიშნავენ,

რომ GA უზრუნველყოფდეს სეგმენტების პოზიციის დამოუკიდებლობას, სწორი ნაწილობრივი სტრუქტურა შესაძლოა მშობლის გადაწყვეტაში არასწორად იყოს განთავსებული, მაგრამ ალგორითმმა უნდა შეძლოს მისი ამოცნობა, როგორც ვალიდური ქვეგადაწყვეტისა და მისი სწორად პოზიციონირება შვილის სტრუქტურაში. ეს ანალოგიურია იმ შემთხვევისა, როდესაც აღმოჩენილია თანმიმდევრული ცოდნის კლასტერი (მაგალითად, ურთიერთდაკავშირებული ფაქტების ნაკრები კონკრეტულ ქვეთემაზე), რომელიც თავდაპირველად არასწორ კატეგორიაში იყო განსაზღვრული, ხოლო შემდეგ სწორ კონტექსტში გადანაწილდა ფართო ცოდნის ბაზის ფარგლებში.

თაობების განმავლობაში ევოლუციური შეკრების პროცესი ხელს უწყობს შედარებით სრულყოფილი ცოდნის სეგმენტების გამრავლებას, მათი თანდათანობითი გაფართოებისა და გაერთიანების გზით. ზუსტად ისე, როგორც GA-ზე დაფუძნებული თავსატეხის ამომხსნელი სისტემები ზრდიან დაკავშირებული სეგმენტების სიდიდეს. დროდადრო განხორციელებული მუტაციები (შემთხვევითი მცირე ცვლილებები) უზრუნველყოფს ახალი კომბინაციების წარმოშობას ან ალტერნატიული ინტეგრაციების გამოვლენას, რაც განსაკუთრებით მნიშვნელოვანია უზარმაზარი საძიებო სივრცის ათვისებისთვის.

ნეირონული კომბინატორიული ასამბლეა: ოპტიმიზაცია:

განსხვავებულ მიდგომას გვთავაზობს ნეირონული კომბინატორული ოპტიმიზაცია, რომლის მიხედვითაც ღრმა ნეირონული ქსელები სწავლობენ თავსატეხის ნაწილების აგებას მართული ძიების (guided search) გზით. ტრადიციულ ამომხსნელ სისტემებს აუცილებლად უნდა მოეძებნათ $N!$ პერმუტაცია $N!$ ფრაგმენტებზე, თუმცა Pointer Network-მა (Vinyals და სხვები.) შემოიტანა მექანიზმი, რომელიც თავს არიდებს ამ კომბინატორულ სირთულეს, ყურადღების (attention mechanism) მექანიზმის მეშვეობით. ქსელი თანმიმდევრულად უთითებს პერმუტაციაში ჩასამატებელ მომდევნო ელემენტზე. Dery და თანაავტორები ამ მიდგომას იყენებენ თავსატეხის ამომხსნისთვის, სადაც ისინი აერთიანებენ კონვოლუციურ ნეირონულ ქსელს (CNN) ნაწილების ვიზუალური მახასიათებლების ამოსარებად და Pointer Network-ს, რომელიც განსაზღვრავს ნაწილების განლაგების

თანმიმდევრობას. მათ მოდელში, თავსატეხის თითოეული ნაწილი c_i თავდაპირველად კოდირდება როგორც მახასიათებლების ვექტორი (ემბედიინგი), რაც მიიღება წინასწარ გაწვრთნილი CNN-ის მეშვეობით. მიღებული ემბედიინგები $\{\lambda_i\}$ წარმოადგენს ნაწილებს ვექტორულ სივრცეში, რომელიც ასახავს მათ ვიზუალურ მსგავსებებს (მაგალითად, ტექსტურის უწყვეტი სეგმენტები მიიღებენ მაღალი შესაბამისობის მქონე ემბედიინგებს) (დერი, მენგისტუ და ავე 2017). როდესაც მოცემულია ფრაგმენტების ემბედიინგთა ჯგუფი, Pointer Network-ის რეკურენტული დეკოდერი, რომელსაც მართავს ყურადღების მექანიზმი, ირჩევს თითო ნაწილს გადასაწყვეტ თანმიმდევრობაში ჩასასმელად. ყოველი ნაბიჯისას ქსელი გამოთვლის ყურადღების ქულებს (ანუ შესაბამისობის მაჩვენებლებს) მიმდინარე ნაწილობრივ აგებულ მდგომარეობასა და დარჩენილ ნაწილებს შორის და ირჩევს იმ ნაწილს, რომელიც ყველაზე მეტად შეესაბამება შემდეგ პოზიციას. ამ პროცედურის გამეორების გზით, ქსელი აგებს პერმუტაციას, რომელმაც იდეალურად უნდა აღადგინოს თავდაპირველი გამოსახულება.

ცოდნის აგების კონტექსტში, ნეირონული ასამბლერი ანალოგიურად ანაწილებს ცოდნის ფრაგმენტებს ვექტორულ სივრცეში და ყურადღებაზე დაფუძნებულ (attention mechanism) მოდელს იყენებს მათი სტრუქტურულ შედეგად გადაქცევისთვის (მაგალითად, ფაქტებიდან თანმიმდევრული ნარატივის აგება ან არგუმენტაციის ნაბიჯების სწორად განლაგება).

ყველა შესაძლო კომბინაციის ძიების ნაცვლად, ნეირონული ქსელი მონაცემებზე დაყრდნობით სწავლობს, როგორ უკავშირდებიან ერთმანეთს ფრაგმენტები ტიპურად, რაც საშუალებას აძლევს იწინასწარმეტყველოს, რომელი ნაწილი უნდა განთავსდეს შემდეგ. ეს ჰგავს ადამიანის ინტუიციურ მიდგომას - როგორ პოულობს ის სავარაუდო დამთხვევებს ფერის ან ფორმის მიხედვით; ნეირონული მოდელი გამომუშავებული გამოცდილების (ტრენინგის) მეშვეობით იძენს ინტუიციას სემანტიკური შესაბამისობის ამოცნობაში. დროთა განმავლობაში, როდესაც მოდელი მეტ ფრაგმენტს აერთიანებს, მისი კონტექსტუალური აღქმა ფართოვდება და ყურადღება შეიძლება გამახვილებულ იქნას არა მხოლოდ ლოკალურ, არამედ გლობალურ დონეზე, რათა უზრუნველყოს სტრუქტურული

თანმიმდევრულობა. თანმიმდევრული ასამბლების ამოცანებში (როგორცაა მტკიცებულების აგება ან კომპლექსურ შეკითხვაზე პასუხის დამუშავება), თანამედროვე ტრანსფორმერულ მოდელებს უკვე ახასიათებთ მსგავსი ქცევა: ისინი ამოირჩევენ შესაბამის ცოდნის ფრაგმენტებს და ნაბიჯ-ნაბიჯ აწყობენ პასუხს. თავსატეხის(Jigsaw Puzzle) მიდგომა ნათლად ასახავს ამ პროცესს. ის განასახიერებს ნაწილების ამორჩევასა და თანმიმდევრული განთავსების პრინციპს, სადაც ყოველი შერჩეული ნაწილი იკავებს ადგილს მზარდ სტრუქტურაში.

თავსატეხების ამოხსნის სტრატეგიები, როგორც ცოდნის ინტეგრაციის ევრისტიკა

ათასობით ფრაგმენტის აწყობის სირთულის სამართავად, თავსატეხების შემთხვევაში ადამიანები და თავსატეხების ამოხსნის ალგორითმები ევრისტიკულ სტრატეგიებს ეყრდნობიან. ამ სტრატეგიებს ცოდნის ინტეგრაციის ამოცანებში გამოყენებაც შეიძლება:

ერთ-ერთი ფართოდ ცნობილი სტრატეგია თავსატეხის ამოხსნისას არის პირველ რიგში საზღვრის აგება, ანუ კიდეებზე განლაგებული ნაწილების იდენტიფიცირება და ჩარჩოს ფორმირება, რასაც შემდგომში მოჰყვება შიდა სივრცის შევსება. ეს მნიშვნელოვნად ამცირებს ძიების სივრცეს, რადგან ქმნის მყარ კონტურს, რომლის ფარგლებშიც შიდა ნაწილები უნდა მოექცეს. ამ პრინციპს ემყარებიან კომპიუტერული ამომხსნელებიც: მაგალითად, ისინი ახდენენ კიდეებური ნაწილების გამოვლენას ფორმის მიხედვით და აწყობენ საზღვარს როგორც დახურულ მარყუჟს (ხშირად წარმოჩენილი როგორც ტურისტის ამოცანა - Traveling Salesman Problem). ჩარჩოს აგება ქმნის სტრუქტურულ საყრდენს, რომელზეც შეიძლება შემდგომი ნაწილების განლაგება.

ცოდნის აგების კონტექსტში ანალოგიური მიდგომაა ჯერ სტრუქტურული ჩარჩოს განსაზღვრა. მაგალითად, თუ საქმე ეხება ცოდნის გრაფის ან არგუმენტის აგებას, პირველ რიგში შეიძლება განისაზღვროს ზოგადი სქემა ან გეგმა: ძირითადი ერთეულები და მათი ურთიერთობები, „ჩარჩოს ნაწილები“. ეს შეიძლება ნიშნავდეს ძირითადი კატეგორიების ან მაღალი დონის ონტოლოგიის იდენტიფიცირებას, რომელიც განსაზღვრავს ამოცანის ფარგლებს. „საზღვრის პირობის“ ფიქსირებით, იქნება ეს ფუნდამენტური პრინციპები, შეზღუდვები თუ საერთო ნარატივის

სტრუქტურა. ხელოვნური ინტელექტი ამცირებს გაურკვეველობას იმის მიხედვით, თუ სად უნდა განთავსდეს დარჩენილი ფრაგმენტები. ზუსტად ისე, როგორც თავსატეხის კიდეებზე განლაგებულ ფრაგმენტს აქვს მხოლოდ ერთი მხარე, რომელიც შეიძლება დაუკავშირდეს შიდა ნაწილს (მეორე მხარე მყარად არის მიბმული საზღვარს), ასევე, ცოდნის ფაქტს, რომელიც მიბმულია მაღალი დონის ჩარჩო სტრუქტურასთან, ნაკლები თავისუფლება აქვს ინტერპრეტაციის თვალსაზრისით. ეს სტრატეგია თავიდან გვარიდებს ფრაგმენტების მექანიკურ, კონტექსტის გარეშე დაკავშირებას, რადგან თავიდანვე იქმნება ლიმიტირებული კონტექსტუალური სივრცე.

რთული თავსატეხების შემთხვევაში აუცილებელია დაყავი და იბატონე (divide-and-conquer) სტრატეგიის გამოყენება. იმის ნაცვლად, რომ ალგორითმმა ერთდროულად ყველა ნაწილი მოათავსება სცადოს, ისინი ქმნიან მცირე კლასტერებს ან სეგმენტებს. მაგალითად, ნაწილების ფერის ან ტექსტურის მიხედვით დაჯგუფებით აწყობენ გამოსახულების რომელიმე კონკრეტულ ნაწილს (მაგალითად, ცის ნაწილს ან გარკვეულ ობიექტს) და შემდგომ აერთიანებენ ამ სეგმენტებს უფრო დიდ სტრუქტურებად. Gallagher-ის ალგორითმი ამის მაგალითია, რომელიც იყენებს ხეზე დაფუძნებულ შეკრების მიდგომას: ის ეგოისტურად (greedy) აერთიანებს კომპონენტებს (ნაწილობრივ აწყობილ სტრუქტურებს) გეომეტრიული შეზღუდვების გათვალისწინებით. ანალოგიურად, Pomeranz და თანაავტორები იყენებდნენ ერთი კომპონენტის გრადუალურ ზრდას, პერიოდულად ცვლიდნენ შეკრებილ მონაკვეთს ისე, რომ უკეთ შეესაბამებინა ის ახალ ნაწილებთან.

თუ აღნიშნულ პრინციპს ხელოვნური ინტელექტის ცოდნის ინტეგრაციის კონტექსტში გადავიტანთ, მივიღებთ მოდულური შეკრების იდეას. AI სისტემას შესაძლოა მოუწიოს რთული ცოდნის აგების ამოცანის გადაჭრა მისი შინაარსობრივად თანმიმდევრულ ქვეპრობლემებად დაყოფით: მაგალითად, ანგარიშის აგებისას, ის შეიძლება თავდაპირველად დამოუკიდებლად შეაგროვოს ცალკეული სექციები ან თემები (თითო სექცია წარმოადგენს თავსატეხის რეგიონს), ხოლო შემდეგ მოახდინოს ამ სექციების ერთმანეთთან დაკავშირება. ცოდნის გრაფის კონტექსტში კი, შესაძლებელია პირველად მოხდეს ძლიერი კავშირებით შემდგარი ერთეულების

კლასტერების იდენტიფიცირება (ქვეგრაფების ფორმირება), რომელთა შემდგომი შერწყმა ხდება საერთო კვანძებისა ან ურთიერთობების საფუძველზე.

მოცემული იერარქიული ინტეგრაცია ამცირებს სირთულეს თითოეულ ეტაპზე, რადგან თითოეული მოდული უფრო მცირეა მთლიან სტრუქტურასთან შედარებით. ამასთან, ის საშუალებას იძლევა განსხვავებული სტრატეგიების იქნას გამოყენებული თითოეული ფრაგმენტისთვის (ისევე, როგორც ადამიანი იყენებს ერთ ტაქტიკას “ცის” ნაწილის ასაწყობად და სხვას, ტექსტურით დატვირთული მონაკვეთისთვის). გადამწყვეტია ის ფაქტორი, რომ მოდულების შერწყმა მოითხოვს ინტერფეისების შესაბამისობას, რაც ანალოგიურია ორი აწყობილი რეგიონის კიდების დამთხვევასთან. ცოდნის ტერმინებით, როდესაც ხდება ორი ქვედომენის შერწყმა, მათ გადაკვეთაზე არსებული ცნებები უნდა იყოს შეთანხმებული: მაგალითად, საჭიროა დუბლირებული ერთეულების იდენტიფიცირება, როგორც ერთი და იგივე ობიექტის, ან ორი თეორიის ტერმინოლოგიის თანხვედრა.

ცოდნის ინტეგრაციის პროცესში ყურადღებაზე დაფუძნებული აგების პროცესი ანალოგიურია იმ პროცესისა, როცა ხელოვნური ინტელექტი ფოკუსირდება შემდეგ დასამატებელ ყველაზე მნიშვნელოვან ინფორმაციაზე. მრავალდოკუმენტური რეზიუმეს შექმნისას, AI სისტემები ათვალიერებენ შესაძლო ფაქტებს და ყურადღებას ამახვილებენ მათზე, რომლებიც დაკავშირებულია უკვე შექმნილ ფრაგმენტების შინაარსთან. ცოდნის გრაფის აგების ამოცანაში, AI შეიძლება ფოკუსირდეს ერთ კონკრეტულ ერთეულზე და მოძებნოს ის მეორე ერთეული, რომელიც მოცემულ კონტექსტში საუკეთესოდ დაუკავშირდება მას. ტრანსფორმერულ არქიტექტურებში გამოყენებული ყურადღების მექანიზმები უკვე იყენებენ ასეთ დინამიკურ ფოკუსირების მექანიზმებს: მაგალითად, კითხვაზე პასუხის გაცემის სისტემაში, ყურადღება მიმართულია ტექსტის იმ ფრაგმენტისკენ, რომელიც შეიცავს პასუხს, შემდეგ კი მომდევნო შესაბამის ფრაგმენტისკენ შემდგომი პასუხისთვის, რაც საშუალებას იძლევა ინფორმაცია მრავალ წყაროსგან თანმიმდევრულად აიგოს. თავსატეხის(Jigsaw Puzzle) აწყობის მიდგომა ამ პროცესს მეტ ინტუიციას სძენს: “ყურადღება” ჰგავს იმას, როცა ადამიანი იღებს თავსატეხის ერთ ნაწილს და სხვადასხვა ფრაგმენტს ადარებს, რათა მოძებნოს, თუ სად ჯდება ის საუკეთესოდ. ეს

მეთოდი ასევე მოიცავს შესაბამისობისთვის მახასიათებლების იდენტიფიკაციის მნიშვნელობას: როგორც პაზლის ამწყობი აფასებს ფერსა და ფორმას, ისე AI-ს სჭირდება გამორჩეული მახასიათებლების (ემბედინგები, საკვანძო სიტყვები, ურთიერთობების ნიშნები) აღმოჩენა იმის დასადგენად, უკავშირდება თუ არა ორი ცოდნის ფრაგმენტი ერთმანეთს. მთლიანობაში, ყურადღებაზე დაფუძნებული მიდგომა უზრუნველყოფს იმას, რომ აწყობის პროცესი იმართება ლოკალური კონტექსტის მიხედვით, ანუ იმ მომენტში არსებულ ნაწილობრივ გადაწყვეტას რაც სჭირდება და არა შემთხვევითი გლობალური არჩევნებით, რაც მნიშვნელოვნად ზრდის ეფექტიანობასა და სიზუსტეს საბოლოო შედეგის ფორმირებაში.

გლობალური თანხვედრა და თანმიმდევრულობა

ცოდნის აგების შემთხვევაში გლობალური თანხვედრის და თანმიმდევრულობის მიზანია აგებული ცოდნის გლობალურ ჭრილში უზრუნველყოფა. მას შემდეგ, რაც ინფორმაციის ფრაგმენტები წყვილებად ან მცირე ჯგუფებად არის აგებული, სისტემამ უნდა განახორციელოს „გლობალური შესაბამისობის“ შემოწმება: შეესაბამება თუ არა მთლიანად აგებული ცოდნა საერთო შინაარსობრივ ლოგიკას? ეს შეიძლება მოიცავდეს კონსისტენტობის შემოწმებას (ლოგიკური წინააღმდეგობების არარსებობა, ყველა მითითების გაცხადება), შეზღუდვების დაცვას (თითოეული ფრაგმენტის ერთჯერადი გამოყენება - თავსატეხის ანალოგიური პრინციპი) და საბოლოო სტრუქტურის შესაბამისობას რომელიმე ცნობილ გლობალურ სქემასთან ან ჭეშმარიტებასთან. ერთი კონკრეტული მაგალითია ონტოლოგიების შეჯერება: როდესაც ორი ქვეონტოლოგია გაერთიანდება, გლობალური შეჯერების პროცესი შესაძლოა მოითხოვდეს განსაზღვრებების გადახედვას ისე, რომ დუბლირება და კონფლიქტები აღმოიფხვრას და ტაქსონომია სრულად იყოს გაერთიანებული. ემბედინგის სივრცეში, გლობალური შეჯერება შეიძლება გულისხმობდეს ვექტორების ოპტიმიზაციის გზით კორექციას, რათა სხვადასხვა წყაროდან წამოსული მსგავსი ცნებები მიახლოებულ პოზიციებში აღმოჩნდნენ და სისტემური აცდენა აღმოიფხვრას (ისევე, როგორც მრავალნაწილიანი 3D თავსატეხის შეკრებისას კოორდინატთა ჩარჩოების სწორად შეჯერება აუცილებელია).

ლოკალური შესაბამისობები თავისთავად არ იძლევა საბოლოო სწორ გადაწყვეტას. შეიძლება ყველა კავშირი სწორი იყოს, მაგრამ მთლიანი სურათი მაინც არასწორი ან არასრული დარჩეს. შესაბამისად, AI სისტემებისთვის დიდი მნიშვნელობა აქვს საბოლოო ინტეგრაციის ფაზას, რომელიც განიხილავს შეკრებილ ცოდნას მთლიანად და ოპტიმიზაციას უკეთებს მის გლობალურ თანმიმდევრულობას. ამ პროცესში შეიძლება გამოყენებულ იქნეს გრაფიკული მოდელების ტექნიკები. შესაბამისად, გლობალური შეჯერება წარმოადგენს იმ ეტაპს, რომელიც გვაწვდის საბოლოო შედეგს. ის უზრუნველყოფს, რომ შეკრებილი ფრაგმენტებისგან მიღებული ცოდნა იყოს ერთიანი, თანმიმდევრული და დასრულებულ ფორმაში.

ენტროპია, როგორც გაურკვეველობისა და სირთულის საზომი

თავსატყეხების ამომხსნელები და დიზაინერები ხშირად სვამენ შეკითხვას: რამდენად რთულია მოცემული თავსატყეხი? თამაშის დიზაინის უახლეს კვლევებში გაჩნდა იდეა, რომ ენტროპია წარმოადგენს თავსატყეხის სირთულის ბუნებრივ მეტრიკას. იდეა მდგომარეობს იმაში, რომ შეფასდეს ის გაურკვეველობა, რომელსაც ამომხსნელი აწყდება ყოველი ნაბიჯისას. ფაქტობრივად, არჩევანის „განტოტვის ხარისხი“ და აგრეგირდეს ეს გაურკვეველობა მთლიანი ამომხსნის პროცესზე.

Chen და თანაავტორები განსაზღვრავენ თავსატყეხის ენტროპიას, როგორც იმ ბიტების რაოდენობას, რომელიც საჭიროა იმისათვის, რომ გადაწყვეტა ავუხსნათ ადამიანს, ვინც იცის თავსატყეხის წესები. ალტერნატიულად, ეს არის ის რაოდენობა, რომელიც გამოხატავს ამომხსნელის წინაშე მდგარი თანაბრად შესაძლო ამომხსნის გზების რაოდენობის ლოგარითმს. თუ თითოეული ნაბიჯისას ამომხსნელს ბევრი ალტერნატიული გზა აქვს მოსინჯული, ენტროპია (გაურკვეველობა) მაღალია და თავსატყეხი ითვლება რთულად. მათი ენტროპიაზე დაფუძნებული საზომი, რომელიც გამოყენებული იყო *The Witness* ვიდეოთამაშის თავსატყეხებზე, წარმატებით კორელირებდა მოთამაშეების მიერ შეფასებულ სირთულესთან და, მეტიც, თავსატყეხების მოხმარების სიამოვნებასთან. ის აჭარბებდა უფრო მარტივ საზომებს, როგორიცაა „ამომხსნის ნაბიჯების რაოდენობა“. ფაქტობრივად, ენტროპია ასახავს თავსატყეხის შიდა ბუნდოვანებასა და კომბინატორულ სირთულეს: მაღალი ენტროპიის მქონე თავსატყეხი ბევრ სავარაუდო ნაბიჯს ან ნაწილების განლაგებას

მოითხოვს თითოეულ ეტაპზე, რაც ამოხსნას ართულებს, განსაკუთრებით იმ შემთხვევაში, თუ არ არსებობს სპეციალური სტრატეგიები ან არ ხორციელდება ხანგრძლივი ძიება.

AI-სთვის ენტროპია შეიძლება განიხილოს როგორც ცოდნის გაურკვევლობის ან აგების სირთულის საზომი. ცოდნის აგების ამოცანა ითვლება მაღალი ენტროპიის მქონედ, თუ ყოველი ეტაპისას არსებობს მრავალი განსხვავებული გზა მომდევნო ფრაგმენტის ინტეგრაციისთვის. მაგალითად, თუ AI ცდილობს ფრაგმენტული ჩანაწერებიდან ისტორიული ქრონოლოგიის აღდგენას და რომელიღაც ეტაპზე არსებობს რამდენიმე შესაძლო ვარიანტი იმასთან დაკავშირებით, თუ როგორ უნდა იყოს ეს ფრაგმენტები დალაგებული ან ერთმანეთთან დაკავშირებული (მაგალითად, გაურკვეველი თარიღების ან კავშირების გამო), მაშინ გაურკვევლობა(ენტროპია) მაღალია. ამის რაოდენობრივი შეფასება ენტროპიით (ბიტებში) საშუალებას იძლევა განისაზღვროს, რამდენი დამატებითი ინფორმაცია - იქნება ეს ახალი მონაცემი თუ შეზღუდვები არის საჭირო ამ გაურკვევლობის აღმოსაფხვრელად. შესაბამისად, ენტროპია შეიძლება იყოს გზამკვლევი AI სისტემისთვის, რათა მან წინასწარ შეაფასოს ცოდნის ინტეგრაციის ამოცანის სირთულე, სრულ ამოხსნამდე. ისევე როგორც თავსატეხის ენტროპია გამოიყენება სხვადასხვა სირთულის თავსატეხების დაგეგმვისთვის, ასევე შესაძლებელია მანქანური სწავლის კურიკულუმის აგება ენტროპიის პრინციპით: დასაწყისში დავალებები შეიძლება იყოს დაბალი ენტროპიის მქონე (ფრაგმენტების მცირე რაოდენობა ან მკაფიო კავშირები), ხოლო შემდგომ ეტაპებზე თანდათან გაიზარდოს სირთულე. გარდა ამისა, ენტროპიამ შეიძლება განსაზღვროს აგების სტრატეგია: მაღალი ენტროპიის მქონე რეგიონები შესაძლოა მოითხოვდეს დამატებითი მინიშნებების მოპოვებას ან სტრატეგიის შეცვლას (ზუსტად ისე, როგორც ადამიანი თავსატეხში ყველაზე ბუნდოვან და რთულ ნაწილებს მაგალითად, ერთგვაროვან ცისფერ ფრაგმენტებს ბოლოსთვის იტოვებს და ჯერ ადვილად ამოცნობადი სეგმენტების აწყობას იწყებენ). ცოდნის გრაფების შემთხვევაში, ის რეგიონები, რომლებიც გამოირჩევიან მაღალი კავშირიანობით და გაურკვევლობით (ბევრი შესაძლო კავშირი), შეიძლება საჭიროებდეს გლობალურ ალგორითმებს, ლოკალურ მიდგომების ნაცვლად, რაც ანალოგიურია იმასთან, როცა

ინფორმაციის თეორიაზე დაფუძნებული ჰეისტიკა ეხმარება სისტემას სტრატეგიის განსაზღვრაში.

ენტროპია ასევე შეიძლება გამოყენებულ იქნას როგორც ინდიკატორი აგებული ცოდნისადმი ნდობის ხარისხის განსასაზღვრად: სრულად ამოხსნილ თავსატეხს აღარ გააჩნია ენტროპია, მხოლოდ ერთი შესაძლო ვარიანტია, რომელშიც ყველა ნაწილი სწორად არის განლაგებული. ანალოგიურად, სრულად აგებული ცოდნის ბაზას ან საბოლოო პასუხს ახასიათებს დაბალი ენტროპია, თითქმის არ არსებობს ბუნდოვნება იმასთან დაკავშირებით, თუ როგორ ერწყმიან ერთმანეთს ფაქტები. აგების პროცესის განმავლობაში, AI სისტემას შეუძლია აკონტროლოს ენტროპიის დონე ფრაგმენტების განთავსებისას: ენტროპიის შემცირება მიუთითებს იმაზე, რომ გადაწყვეტილების სივრცე ვიწროვდება (სისტემა უახლოვდება იდეალურ ინტეგრაციას), მაშინ როდესაც ენტროპიის მუდმივად მაღალი დონე შეიძლება მიანიშნებდეს, რომ სისტემას აკლია მნიშვნელოვანი ფრაგმენტი ან შეზღუდვა ისევე, როგორც თავსატეხი შეიძლება აღმოჩნდეს გადაუჭრელი, თუ გარკვეული ნაწილები დაკარგულია ან ერთმანეთს ძალიან ჰგავს.

4.3 გრაფული მანქანური სწავლების გამოყენება ცოდნის მოდელირებაში

გრაფები წარმოადგენს მათემატიკურ სტრუქტურებს, რომლებიც გამოიყენება ობიექტებს შორის ურთიერთობების აღსაწერად და თითქმის ყველგან გვხვდება ყოველდღიურ ცხოვრებაში. მაგალითად, სოციალური ქსელები, სადაც მომხმარებლები დაკავშირებულნი არიან იმის საფუძველზე, ერთ-ერთი „მიჰყვება“ თუ არა მეორე მომხმარებლის განახლებებს. გრაფები ასევე გამოიყენება რუკების გამოსახატავად, სადაც ქალაქები დაკავშირებულია ქუჩებით. მათ შეუძლიათ აღწერონ ბიოლოგიური სტრუქტურები, ვებგვერდები და ის პროცესებიც კი, რომლებიც ნეიროდეგენერაციული დაავადებების პროგრესიას უკავშირდება. (სტამილე, მარჟულო და დეუსებიო 2021, 3). გრაფებს დიდი ადგილი უჭირავთ მანქანურ სწავლებასა და ცოდნის მოდელირებაშიც.

გრაფული ნეირონული ქსელები (Graph Neural Networks, GNNs) წარმოადგენს ღრმა სწავლის მოდელებს, რომლებიც სპეციალურად შექმნილია გრაფული

სტრუქტურის მქონე მონაცემებზე ოპერირებისთვის. ზოგად დონეზე, GNN-ები სწავლობენ კვანძების (ან მთლიანი გრაფების) ვექტორული წარმოდგენების (ემბედიנגების) გამოთვლას, თითოეული კვანძის მეზობლების ინფორმაციის იტერაციული აგრეგირებით (ლენგეილინგი, და სხვ. 2021, 22).

გრაფები ცენტრალურ როლს ასრულებენ ცოდნის მოდელირებაში, რადგან ისინი ბუნებრივად გამოხატავენ ობიექტებსა და მათ შორის ურთიერთობებს. ცოდნის გრაფი (Knowledge Graph, KG) ფაქტობრივ ცოდნას გამოსახავს გრაფის სახით, სადაც კვანძები წარმოადგენს ობიექტებს, ხოლო სემანტიკურად მონიშნული კავშირები მათ შორის ურთიერთობებს. ჩვეულებრივი რელაციური მონაცემებისგან განსხვავებით, ცოდნის გრაფები ასახავენ მდიდარ სემანტიკას სტრუქტურირებულ ფორმაში. გრაფზე დაფუძნებული მოდელები აძლიერებენ, როგორც ცოდნის წარმოდგენას, ისე დასაბუთებას შესაბამის დომენებში: ემბედიנגებზე დაფუძნებული მეთოდები უზრუნველყოფენ ობიექტებისა და ურთიერთობების განაწილებულ წარმოდგენებს დასკვნის გამოტანისთვის, ხოლო სიმბოლური დასაბუთება იყენებს გრაფის სტრუქტურის ტრავერსირებას სემანტიკური ურთიერთობების განსაზღვრისთვის (ფან, და სხვ. 2023, 2).

გრაფულ მანქანურ სწავლებაზე (Graph ML) დაფუძნებული ტექნიკები ფართოდ გამოიყენება ცოდნის გრაფებზე. მაგალითად, ცოდნის გრაფის ემბედიנגზე დაფუძნებული მოდელები (როგორცაა TransE, DistMult, ComplEx და სხვ.) სწავლობენ ობიექტებისა და ურთიერთობების დაბალგანზომილებიან ვექტორულ წარმოდგენებს, რომლებიც ინარჩუნებენ რელაციურ შაბლონებს. აღნიშნული ემბედინგები გამოიყენება გამოტოვებული კავშირების პროგნოზირებისთვის (ბმულების პროგნოზირება ცოდნის გრაფში) ან ანალოგიების მოსაძებნად. უახლესი მიდგომები მოიცავს გრაფულ ნეირონულ ქსელებს (GNN) მაგალითად, Relational GCN და CompGCN განაზოგადებენ გრაფული კონვოლუციის იდეას მრავალურთიერთობიან მონაცემებზე, ინფორმაციის აგრეგირება კიდეების გასწვრივ (ფან, და სხვ. 2023, 3). ასეთი მოდელები გრაფის სტრუქტურიდან და კვანძთა ატრიბუტებიდან შესწავლით, ახორციელებენ ცოდნის ბაზის შევსებას (knowledge base completion). გარდა ამისა, GNN-ები გამოიყენება მრავალნაბიჯიანი კითხვის პასუხის გაცემისთვის (multi-hop query

answering) ცოდნის გრაფებში: GNN-QE-ს მსგავსი მეთოდები ქმნიან ნეირონულ მოთხოვნის(Query) პრაქტებს, რომლებიც გრაფის სტრუქტურაზე მსჯელობით(Reasoning) რთულ მოთხოვნებს პასუხობენ.

გრაფული მსჯელობა(Reasoning) ცოდნის გრაფებზე (KGs) წარმოადგენს აქტიურად განვითარებად სფეროს. Liang და თანაავტორები აღწერენ ცოდნის გრაფზე მსჯელობას,(Knowledge Graph Reasoning, KGR) როგორც ახალი ფაქტების გამოყვანას უკვე არსებული ფაქტებიდან, ცოდნის გრაფში არსებული ლოგიკური სტრუქტურის საფუძველზე. KGR მოდელები კლასიფიცირდება გრაფის ტიპის მიხედვით: სტატიკური ცოდნის გრაფები, ტემპორალური გრაფები (დროთა განმავლობაში ცვლილებადი ფაქტებით), და მულტიმოდალური გრაფები (ტექსტური ან ვიზუალური ინფორმაციით გამდიდრებული კვანძებით) (ლიანგ, და სხვ. 2023, 3). ზოგადი KGR ამოცანებია: ბმულების პროგნოზირება (ფაქტების შევსება), ტრიპლეტის კლასიფიკაცია (მართებული/მცდარი), და კომპლექსურ შეკითხვებზე პასუხის გაცემა (იმ კავშირების პოვნა, რომლებიც აკმაყოფილებენ გადაცემულ შეკითხვას). ამ ამოცანებში GNN-ები ხშირად გამოიყენება, როგორც ნეირონული საფუძველი (backbone), ზოგჯერ სიმბოლურ წესებსა და ლოგიკასთან კომბინაციით, რაც ნეირონულ-სიმბოლური (neural-symbolic) მიდგომის ფორმას იღებს.

ცოდნის გრაფების (KGs) ფარგლებს გარეთ, გრაფები ასევე გამოიყენება კონცეპტუალური ცოდნის მოდელირებისთვის ისეთ დარგებში, როგორცაა ონტოლოგიის ინჟინერია (კონცეფციის იერარქიები), სემანტიკური ქსელები და კოგნიტური მოდელირება. ყველა ამ შემთხვევაში, გრაფზე დაფუძნებული მანქანური სწავლება (Graph ML) ხელს უწყობს პროცესს ორი ძირითადი გზით: ან გრაფის სტრუქტურის ვექტორულ სივრცეში ემბედირებით, რაც შემდგომ შესაძლებელია მიეწოდოს სასწავლო ალგორითმებს ან სწავლით მიღებული ინფორმაციის პირდაპირი გავრცელებით კიდევების გასწვრივ დასკვნის გამოტანისთვის. მნიშვნელოვანია, რომ გრაფებს ცოდნის მოდელირებაში შემოაქვთ რელაციური ინდუქციური მიკერძობა (relational inductive bias): ისინი ინახავენ ურთიერთობების პირდაპირ გამოსახულებას და სიმეტრიებს, რაც წარმოადგენს სწავლების პროცესისთვის მნიშვნელოვან სტრუქტურულ საფუძველს (ლენგელინგი, და სხვ. 2021, 24). როგორც Fan და

თანავტორები აღნიშნავენ, ცოდნის გრაფები „აგებენ მაღალხარისხიან და სანდო ფაქტობრივ ცოდნას სტრუქტურირებულ ფორმატში“, და ამ მონაცემების ღრმა სწავლებასთან (deep learning) კომბინაცია აძლიერებს მსჯელობას და ამცირებს ისეთ პრობლემებს, როგორცაა ჰალუცინაციები ხელოვნურ ინტელექტში.

საბოლოო ჯამში გრაფულ მანქანურ სწავლებას (Graph Machine Learning) აქვს ორი ძირითადი როლი: ის უზრუნველყოფს როგორც წარმოდგენის ინსტრუმენტებს (ემბედინგები, გრაფული ნეირონული ქსელები, GNNs), ასევე ალგორითმებს (შეტყობინების გადაცემა - message passing, ყურადღება, attention), რომლებიც საშუალებას იძლევა ეფექტიანად გამოვიყენოთ გრაფის სახით წარმოდგენილი სტრუქტურირებული ცოდნა.

4.4 ტენზორული აღრიცხვის გამოყენება ცოდნის აგების პროცესში

ხელოვნური ინტელექტის კონტექსტში, განსაკუთრებით ტექსტუალურ და სემანტიკურ მონაცემებთან მიმართებით ტენზორები შეიძლება ჩაითვალოს მრავალგანზომილებიან „აღრიცხვად“ ან კონტეინერებად, რომლებიც აორგანიზებენ კომპლექსურ, ურთიერთდაკავშირებულ ცოდნას. ბრტყელი მონაცემთა ბაზების ნაცვლად, ცოდნა შეიძლება განიხილებოდეს, როგორც მრავალმხრივი ურთიერთობების ნაკრები (მაგალითად, ტექსტში არსებული სუბიექტი-პრედიკატი-ობიექტი ტიპის ტრიპლეტებად), რომლებიც ბუნებრივად ქმნიან მაღალი განზომილებიან მასივებს. ტენზორული აღრიცხვის ამ მიდგომაში, თითოეული განზომილება (რეჟიმი) შეესაბამება სემანტიკურ ღერძს (ობიექტები, ურთიერთობები, დრო და სხვ.), ხოლო ტენზორული ოპერაციები ახორციელებენ ცოდნის შეკავშირებას და მოთხოვნებს ამ ღერძებზე. მაგალითად, დეკლარაციული წინადადებების ნაკრები შეიძლება წარმოდგენილ იქნას ცოდნის გრაფის სახით, რომელიც თავის მხრივ შეიძლება გადმოიცეს adjacency ტენზორით, რომლის ინდექსები შეესაბამება სუბიექტებს, პრედიკატებსა და ობიექტებს. სემანტიკური მეხსიერების ერთ-ერთ მოდელში, აღნიშნული adjacency ტენზორი აგროვებს ყველა იმ ჭეშმარიტ ფაქტს (ტრიპლეტს), რაც აგენტს ეცოდინება. შესაბამისად, ტენზორზე დაფუძნებული მიდგომა ცოდნას განიხილავს, როგორც მრავალგანზომილებიან აღრიცხვით წიგნს: ტენზორში არსებული ჩანაწერები ასახავს, რომელი ფაქტებია ჭეშმარიტი, ხოლო

ტენზორული ალგებრა (დეკომპოზიცია, კონვოლუცია და სხვ.) ფუნქციონირებს, როგორც ანგარიშგების მექანიზმი, რომელიც ახორციელებს ცოდნის ინტეგრაციასა და ახალი ცოდნის გამომუშავებას.

4.4.1 ტენზორები, როგორც ცოდნის კონტეინერები

ტენზორები ფორმალურად განისაზღვრება როგორც N -განზომილებიანი მასივები (მრავალგანზომილებიანი მატრიცები). ტენზორის თითოეული ჩანაწერი ინდექსირდება მნიშვნელობათა წყვილით. მაგალითად, ცოდნის გრაფში (სუბიექტი, პრედიკატი, ობიექტი) და შეიცავს ინფორმაციას, როგორცაა ფაქტის ჭეშმარიტება ან შესაბამისი ქულა. ენის გაგების კონტექსტში წინადადებათა ნაკრები შეიძლება დაიშალოს SPO ტრიპლეტებად (მაგალითად, (Ana, isFriendOf, Nick)), რის შედეგადაც იქმნება ცოდნის გრაფი. მნიშვნელოვანია ის, რომ ასეთი გრაფი შეიძლება წარმოდგენილ იქნას ტენზორის სახით: სამგანზომილებიანი 0/1 მასივის (ან უფრო მაღალი განზომილებიანი, თუ დრო ან ენტროპიაც მოიზრება), რომლის ელემენტი $Y[s,p,o]=1Y[s,p,o] = 1Y[s,p,o]=1$ მაშინ და მხოლოდ მაშინ, როდესაც ტრიპლეტი (სუბიექტი s , პრედიკატი p , ობიექტი o) ჭეშმარიტია. ამ მიდგომით, ობიექტები და ურთიერთობები ტენზორის განზომილებებს შეესაბამება, ხოლო თავად ტენზორი წარმოადგენს მრავალგანზომილებიან ჩანაწერთა სტრუქტურას. სხვა სიტყვებით, რომ ვთქვათ მთლიანი ცოდნის ბაზა წარმოადგენს ტენზორის ფორმის მქონე აღრიცხვათა წიგნს. ამგვარად, ცოდნაზე ოპერაციები, იქნება ეს დასკვნის გამოტანა, მეხსიერებიდან ამოღება თუ სხვა შეიძლება განიხილებოდეს, როგორც ტენზორულ ოპერაციათა შედეგი.

სტატიკური ცოდნის გრაფის ტენზორი: სტატიკური ცოდნის გრაფი, რომელიც განსაზღვრულია ობიექტთა EEE და ურთიერთობათა PPP სიმრავლით, წარმოქმნის მესამე რიგის adjacency, სადაც $Y_{static}(s,p,o)=1Y_{static}(s,p,o) = 1Y_{static}(s,p,o)=1$ მაშინ და მხოლოდ მაშინ, როდესაც ტრიპლეტი $(s,p,o)(s,p,o)(s,p,o)$ ცნობილია როგორც ჭეშმარიტი. ეს ფიქსირებული ტენზორი იმ ცოდნას, რაც აღიქმება როგორც ჭეშმარიტი ინახავს სემანტიკურ მეხსიერებაში, როგორც ფაქტებს, რომლებიც მუდმივად მიიჩნევა მართებულად.

დროებითი ცოდნის ტენზორი: რეალური სამყაროს ცოდნა ხშირად ცვალებადია. დროებითი ცოდნის გრაფი ასახავს ტრიპლეტებს დროში მათი ცვლილების შესაბამისად. ამან შეიძლება წარმოქმნას მეოთხე რიგის ტენზორი $Y_{temp} \in \{0,1\}^{|E| \times |P| \times |E| \times |T|}$ რომელიც დამატებით ინდექსირდება დროის შტამპით. დროის განზომილებაზე ყოველი „ნაჭერი“ წარმოადგენს სტატიკური ცოდნის გრაფის მდგომარეობას მოცემულ მომენტში. ასეთი ტენზორი საშუალებას იძლევა აღწერილ იქნას მოვლენები (მაგალითად: „(Alice, diagnosed, Flu) დრო t “) და შესაძლებელი ხდება თანმიმდევრულობასა და მიზეზობრივობაზე მსჯელობა.

ალბათობითი ცოდნის ტენზორი: როდესაც ფაქტები გაურკვეველია ან სტატისტიკურად მიიღება, ტენზორის ჩანაწერებს შეიძლება მიენიჭოს ალბათობით გაწონასწორებული მნიშვნელობები $\gamma_{s,p,o} \in [0,1]$. ასეთ შემთხვევაში, ალბათობითი ცოდნის გრაფი წარმოდგენილია რეალური რიცხვების მქონე მესამე რიგის ტენზორით $\Gamma_{prob}(s,p,o) = \gamma_{s,p,o}$, რომელიც გამოხატავს თითოეული ტრიპლეტის მიმართ ნდობის ხარისხს ან სიხშირეს. კოგნიტურ ტერმინებში, ასეთი ტენზორი დაკავშირებულია სემანტიკურ მეხსიერებასთან, ტვინის მიერ შენახულ „ფაქტების ცოდნასთან“.

ამ მრავალგანზომილებიანი მასივები რეალურ დომენებში, რა თქმა უნდა, ასტრონომიულად დიდია (მილიარდობით ობიექტი ან კავშირები). სრულ ტენზორებთან მუშაობა პრაქტიკულად შეუძლებელია, თუმცა ტენზორული მოდელები შესაძლებელს ხდის კომპაქტური წარმოდგენების მიღებას. ერთ-ერთი გავრცელებული მიდგომაა **ტენზორის ფაქტორიზაცია**, რომელიც გულისხმობს მაღალი რიგის ტენზორის დაშლას დაბალი განზომილებიან ფაქტორებად (მაგალითად, CP ან Tucker ფაქტორიზაციები). ცოდნის კონტექსტში, ეს ნიშნავს ობიექტებისა და ურთიერთობების ფარული (latent) ემბედიנגების შესწავლას. მაგალითად, RESCAL მოდელი (მატრიცა-ტენზორის ფაქტორიზაცია) თითოეულ ობიექტს i ასახავს ვექტორით a_i , ხოლო თითოეულ ურთიერთობას p მატრიცით R_p , ისე რომ: $Y(s,p,o) \approx a_s^T R_p a_o$. ზოგადად, შესაძლებელია ინდექსური ემბედიנגების გამოყენება: თითოეული ცნება (ობიექტი ან ურთიერთობა) მიიღებს რეალურ რიცხვთა ვექტორს, ხოლო ტენზორის ჩანაწერი განისაზღვრება ამ ვექტორების ფუნქციის საშუალებით.

როდესაც ცოდნის ტენზორი (ცოდნის გრაფის ადიაკენტურობის ტენზორი) იშლება მარტივ კომპონენტებად, ჩვენ ვიღებთ მონაცემთა სტრუქტურირებულ, დაბალგანზომილებიან რეზიუმეს. ეს მსგავსია იმ პროცესისა, როცა აღრიცხვითი სისტემიდან გამოინაყოფება „მნიშვნელოვანი ანგარიშები“. მაგალითად, Canonical Polyadic (CP) დაშლა წარმოსახავს ტენზორს როგორც რანგ-1 კომპონენტების (ვექტორების გარე ნამრავლების) ჯამს. ასეთი დაშლის თითოეული ფაქტორი შეიძლება ინტერპრეტირდეს როგორც ფარული ცნება ან თემა, რომელიც უკავშირებს ობიექტებსა და ურთიერთობებს.

4.4.2 ტენზორული დეკომპოზიცია და ცოდნის ინტეგრაცია

როდესაც ცოდნის ტენზორი იშლება მარტივ კომპონენტებად, ჩვენ ვიღებთ მონაცემთა სტრუქტურირებულ, დაბალგანზომილებიან რეზიუმეს. ეს მსგავსია იმ პროცესისა, როცა აღრიცხვითი სისტემიდან გამოიყოფა „მნიშვნელოვანი ანგარიშები“. მაგალითად, Canonical Polyadic (CP) დაშლა წარმოადგენს ტენზორს, როგორც რანგ-1 კომპონენტების (ვექტორების გარე ნამრავლების) ჯამს. ასეთი დაშლის თითოეული ფაქტორი შეიძლება ინტერპრეტირდეს, როგორც ფარული ცნება ან თემა, რომელიც აკავშირებს ობიექტებსა და ურთიერთობებს. უფრო ზოგადად, ტენზორული ქსელები (tensor networks), როგორცაა Tensor Train ან Tucker, გვთავაზობენ მოქნილ, განაწილებულ წარმოდგენებს. ეს დაშლები საშუალებას იძლევა მოხდეს მრავალდომენიანი ცოდნის ინტეგრაცია და გაერთიანება. ტენზორის დაშლა აყალიბებს მაღალი რიგის მონაცემებს ინტერპრეტირებად კომპონენტებად. მაგალითად, სხვადასხვა მოდალობის მონაცემების (ტექსტი, სურათები, გრაფები) გაერთიანება შესაძლებელია მათი ერთ დიდ ტენზორად ჩამოყალიბებით და შემდგომ დაშლით. ტენზორული ქსელები წარმატებით გამოიყენება დიდი მოცულობის მონაცემთა ამოცანებში, ანომალიის აღმოჩენა, მახასიათებლების გამოყოფა, კლასიფიკაცია, კლასტერიზაცია და პროგნოზირების მოდელირება, რაც ადასტურებს მათ შესაძლებლობებს.

ცოდნის აგების კუთხით, ცოდნის ტენზორის დაშლა არის ცოდნის აღრიცხვის "აუდიტი": ის ამოიცნობს ძირითად ფარულ ფაქტორებს („ანგარიშებს“), რომლებიც

ხსნიან მონაცემთა სტრუქტურას. განსაკუთრებით მნიშვნელოვანია, რომ ტენზორის დაშლები ბუნებრივად ასახავენ მრავალხაზოვანი და მრავალასპექტიანი ურთიერთობებს. მაგალითად, მეოთხე რიგის ტენზორი შეიძლება ერთდროულად მოიცავდეს ურთიერთქმედებებს *პიროვნება×პრედიკატი×ობიექტი×დრო* სიბრტყეებში. მისი დაშლა შესაძლებელს ხდის ისეთი შაბლონების გამოვლენას, როგორცაა ურთიერთობების დროითი ევოლუცია. კოგნიტურ ტერმინებში, ტენზორული მოდელები (ინდექსური ემბედინგებით) დაკავშირებულია სემანტიკური და ეპიზოდური მეხსიერების წარმოშობასთან, სტატისტიკური სტრუქტურებიდან მიღებული ცოდნის სახით. ჩვენს შემთხვევაში, ყოველი გამოყოფილი ფაქტორი წარმოადგენს ერთგვარად შეკრულ ცოდნის ერთეულს ან წესს, რომელიც მოიცავს სხვადასხვა განზომილებებს, ხოლო მათი ხელახალი შერწყმა (ტენზორული ალგებრის საშუალებით) აღადგენს მთლიან ინფორმაციას.

ხელოვნური ინტელექტის (AI) ცოდნის სისტემების გააზრება ტენზორული აღრიცხვის მიდგომით მოიცავს რამდენიმე გაერთიანებულ გადაწყვეტას: ცოდნის ელემენტები გადაიქცევა მრავალგანზომილებიან აღრიცხვით წიგნში შეტანილ ჩანაწერებად, ხოლო სწავლისა და მსჯელობის (reasoning) მეთოდები არითმეტიკად, რომლის მეშვეობითაც ეს ჩანაწერები იშლება, განახლდება და ხელახლა იწყობა. ეს ხდება ერთმანეთთან აკავშირებს სხვადასხვა ტექნოლოგიას, როგორცაა ცოდნის გრაფები, მეხსიერების მოდელები, ნეირონული ქსელები და ტენზორული ქსელები, ერთიან კონცეპტუალურ ჩარჩოში. ტენზორები ბუნებრივად ასახავენ ინტელექტის მრავალმხრივ ბუნებას (ურთიერთობები, დრო, ენტროპია), ხოლო ტენზორული ოპერაციები ორგანიზებას უკეთებენ ამ ინფორმაციას.

4.5 დიდი მონაცემების ინტელექტუალური ანალიზის ფრეიმვორკის არქიტექტურის აღწერა.

დიდი მონაცემების ანალიტიკის არქიტექტურები, როგორც წესი, შრეებად ორგანიზებული ჩარჩოებია, რომლებიც შექმნილია მასიური და ჰეტეროგენული მონაცემთა ნაკრების მიღების, შენახვის, დამუშავებისა და ანალიზისთვის, საბოლოოდ კი დროული და ქმედითი ინტელექტის წარმოების მიზნით. ამ არქიტექტურებმა უნდა უზრუნველყონ დიდი მონაცემების 5V-ის მოცულობის (volume), სიჩქარის (velocity),

მრავალფეროვნების (variety), სიზუსტის (veracity) და ღირებულების (value) ეფექტიანი მართვა და ინფორმაციული მასალის გარდაქმნა ქმედით ცოდნად. პრაქტიკაში, არქიტექტორები დიდ მონაცემებს ორგანიზებულად აყალიბებენ ერთმანეთისგან განსხვავებულ შრეებად ან მოდულებად, რომელთაგან თითოეული მონაცემთა სიცოცხლის ციკლის ერთ-ერთ ფაზას მოიცავს (იხ. სურ. 1). ეს ფაზები მოიცავს მონაცემთა შექმნას (ingestion), შენახვას, დამუშავებას, ანალიტიკას და ანგარიშგებას/ვიზუალიზაციას. ტრადიციული არქიტექტურები, როგორცაა Lambda არქიტექტურა, იყენებენ გამოყოფილ შრეებს პაკეტური და რეალურ დროში მონაცემთა დამუშავებისთვის, ხოლო ახალმა მოდელებმა (მაგ., Kappa არქიტექტურა) მთლიანად სტრიმინგზე გადასვლა არჩია, მიზანი კი გამტარუნარიანობის, ლატენცობის და შეცდომების დაშვებას შორის ოპტიმალური ბალანსის დაცვაა (ჰუა და ჟენგი 2021, 371).

დიდი მონაცემების ინტელექტუალური დამუშავების არქიტექტურის მრავალსაფეხურიანი მაგალითი (Sandia National Labs-ის BDAI Framework): მონაცემთა ნაკადი მრავალწყაროიანი შინაარსიდან (მარცხნიდან) გადადის სტრიმინგ ნაწილში (ცენტრში), შემდეგ კი ანალიტიკის შრეში და საბოლოოდ ანალიზის სისტემებში (მა, და სხვ. 2020, 6).

დიდი მონაცემების არქიტექტურები ხშირად ეფუძნება მრავალშრიან დიზაინს. მაგალითად, Zhu და სხვები (2019) გვთავაზობენ ხუთშრიან დამუშავებისა და ანალიტიკის (BDPA) არქიტექტურას, რომელიც შედგება შემდეგი შრეებისგან: შეგროვების შრე (მონაცემთა შეგროვება და შეყვანა), შენახვის შრე (warehousing/NoSQL), დამუშავების შრე (პაკეტური/სტრიმინგული), ანალიტიკის შრე (მანქანური სწავლება, ბიზნეს ინტელექტის ხელსაწყოები) და აპლიკაციის შრე (მომხმარებელზე ორიენტირებული სერვისები). თითოეულ შრეს გააჩნია საკუთარი, განსაზღვრული ფუნქცია: შეგროვების შრე აგროვებს დაუმუშავებელ ნაკადებსა და ჩანაწერებს; შენახვის შრე უზრუნველყოფს მასშტაბურ და საიმედო საცავებს (მაგ. Hadoop HDFS, განაწილებული მონაცემთა ბაზები); დამუშავების შრე იყენებს შესაბამის ჩარჩოებს (MapReduce, Spark, Flink, Storm) მონაცემთა ტრანსფორმაციისთვის; ანალიტიკის შრე ახორციელებს ალგორითმების (მანქანური სწავლება, გრაფული ანალიტიკა და სხვ.) გამოყენებას ცოდნის ამოსაღებად. ბოლოს, აპლიკაციის შრე

უზრუნველყოფს შედეგების წარდგენას დაფების, ანგარიშებისა და API-ების საშუალებით (ტანგი, ჯულიე და ლი 2019, 41). პრაქტიკაში, არქიტექტურების უმეტესობა აერთიანებს ანალიტიკასა და ვიზუალიზაციას (მაგალითად, მოდელის პროგნოზირების ჩანერგვა ანგარიშგებაში), ან ანაწილებს შუალედურ შრეებად (მაგ., სიჩქარისა და პაკეტური შრეების გამოყოფა Lambda არქიტექტურაში).

ზოგიერთი ჩარჩოს ფარგლებში დამატებითი შრეები მიმართულია კონტროლისა და ორკესტრაციის მექანიზმებისკენ. მაგალითად, Zhang და სხვები (2016) განსაზღვრავენ ორკესტრაციის მოდულს, რომელიც უზრუნველყოფს აპლიკაციური სამუშაო ნაკადების კოორდინაციას, და წვდომის მოდულებს, რომლებიც მართავენ API-ებსა და სერვისის ინტერფეისებს. ყველა შრეში ჯვარედინი საკითხები განიხილება როგორც მოდულები სტანდარტულ საცნობარო დიზაინებში. ყველაზე გავრცელებული ჯვარედინი მოდულებია:

- **უსაფრთხოება** – წვდომის კონტროლის, დაშიფვრისა და შეჭრის გამოვლენის უზრუნველყოფა სისტემის მასშტაბით;
- **მართვა** – სისტემური რესურსებისა და მონაცემთა ციკლების მონიტორინგი და ადმინისტრირება;
- **ფედერაცია/ინტეგრაცია** – უზრუნველყოფს მრავალ დიდმონაცემოვან სისტემას შორის ინტერპერაბელურობასა და მონაცემთა გაზიარების შესაძლებლობას;
- **მასშტაბურობა/ხელმისაწვდომობა** – ყველა კომპონენტის ელასტიურად გაფართოებისა და წარუმატებლობისადმი მდგრადობის უზრუნველყოფა.

მონაცემთა ორგანიზაცია (სქემა, დაყოფის სტრატეგია) და ტექნოლოგიური სტეკის შერჩევა განიხილება როგორც ფუნდამენტური დიზაინერული გადაწყვეტილებები ამ არქიტექტურებში (ქლეინი, და სხვ. 2016, 54 - 55).

დიდი მონაცემების ინტელექტისათვის არაერთი ფორმალური საცნობარო არქიტექტურა იქნა შემოთავაზებული. Zhang და სხვ. (2016) გვთავაზობენ სრულყოფილ საცნობარო არქიტექტურას (Reference Architecture – RA) ეროვნული უსაფრთხოების მიზნით შექმნილი დიდმონაცემოვანი სისტემებისათვის, რომელიც იყოფა ორ ძირითად მოდულად: „აპლიკაციის მომწოდებელი“ (Application Provider) და

„ჩარჩოს მომწოდებელი“ (Framework Provider). აპლიკაციის მომწოდებლის მოდულებში შედის:

- **მონაცემთა შეგროვება** (Collection) – ინფორმაციის მიღება და ჩამოტვირთვა,
- **მომზადება** (Preparation) – მონაცემთა ამოღება, გარდაქმნა და გაწმენდა (ETL),
- **ანალიტიკა** (Analytics) – მოდელირება და მონაცემთა შერწყმა,
- **ვიზუალიზაცია** (Visualization) – მომხმარებელთან ურთიერთობის ინტერფეისი,
- **წვდომა** (Access) – API-ებისა და სერვისების მართვა.

მომწოდებლის ჩარჩოს მოდულები უზრუნველყოფენ იმ საბაზისო ინფრასტრუქტურას (განაწილებული გამოთვლითი რესურსები, საცავები, შეტყობინებების გადაცემა), რომელსაც ეს აპლიკაციები იყენებენ. დამატებით, დამოუკიდებელი **უსაფრთხოების მოდული** (Security module) ზღუდავს და აკონტროლებს წვდომას ყველა მოდულში, ხოლო **მართვის მოდული** (Management module) განაპირობებს სისტემისა და მონაცემთა ციკლის მონიტორინგს. ეს კომპონენტები თანხვედრაშია ინდუსტრიულად მიღებულ შრეებთან (მაგალითად, ELT-პაიპლაინი, მანქანური სწავლების მოდელების სწავლება, ანალიტიკური დაფები), თუმცა ისინი ორგანიზებულია მოდულურ არქიტექტურად, რომელიც ყურადღებას ამახვილებს ხელახალი გამოყენების შესაძლებლობასა და დარგობრივ სპეციფიკაციებზე.

ქმედით ინტელექტზე ორიენტირებულ ერთ-ერთ მაგალითში, Ma და Garcia (2020) აღწერენ პროგრამული უზრუნველყოფის არქიტექტურას, რომელიც მოიცავს მონაცემთა დამუშავების ოთხ დონეს: მონაცემთა წყარო (Data Source), მონაცემთა ნაკადის მილი (Data Pipeline), ანალიტიკის დონე (Data Analytic) და ანგარიშგების დონე (Data Reporting) (მა, და სხვ. 2020). მონაცემთა ნაკადის პაიპლაინი აგებულია *Apache Storm*-ის ტოპოლოგიებზე თითოეული ტიპის შემავალი ნაკადისთვის, ხოლო პროცესების თანმიმდევრობითი დამუშავება ხორციელდება *Apache Kafka*-ს გამოყენებით (**Getter** → **Normalizer** → **Processor** → **Publisher**). ყველა მონაცემთა წყარო ნორმალიზდება საერთო მოვლენათა სქემის (event schema) ფორმატში, რომელიც ტეგირდება დროისა და მდებარეობის მიხედვით, რაც შემდგომ გამოიყენება ინდექსირებისა და შერწყმისათვის. ანალიტიკის დონე ახორციელებს ამ ერთიან

მოვლენათა საცავის მოთხოვნებს და ამუშავებს მანქანური სწავლების მოდელს (მაგალითად, ნეირონულ ქსელს), რათა წარმოიქმნას შერწყმული ინტელექტი. ანგარიშგების დონე წარმოგვიდგენს ამ ანალიტიკურ შედეგებს საინფორმაციო დაფებზე (dashboards) და სიგნალიზაციის მექანიზმების სახით. ეს სრულყოფილი (end-to-end) ჩარჩო წარმოადგენს მაგალითს იმისა, თუ როგორ შეიძლება დიდი მონაცემების არქიტექტურაში ერთდროულად ჩაშენდეს ჰეტეროგენული ნაკადების მიღება, მათი გაწმენდა და ინტელექტუალური ინფორმაციის მიღება ფაქტობრივად რეალურ დროში (მა, და სხვ. 2020, 10).

Hua და Zeng (2021) გვთავაზობენ არქიტექტურას, რომელიც სპეციფიკურად მორგებულია მეცნიერებისა და ტექნოლოგიის (S&T) ინტელექტის ანალიზზე. ავტორები ხაზს უსვამენ სამ ძირითად კომპონენტს:

- მრავალწყაროიანი მონაცემთა შეგროვება (მაგ. სამეცნიერო სტატიები, პატენტები, სიახლეები, სოციალური მედია, სისტემური ჟურნალები და სხვა);
- დიდი მონაცემების პლატფორმა მონაცემთა ინტეგრაციისა და ანალიზისთვის;
- ინტელექტის აპლიკაციები, რაც მოიცავს ისეთ შესაძლებლობებს, როგორებიცაა ამოცნობა, ტრეკინგი, შედარება, შეფასება და პროგნოზირება.

მოცემული მოდელის მიხედვით, მონაცემები იკრიბება API-ებისა და ვებ-მაძიებლების (web crawlers) მეშვეობით და ინახება განაწილებულ პლატფორმაზე. მაგალითად, სტრიმინგ ტიპის სიახლეები და სოციალური მედიის მონაცემები იკრიბება *Kafka* ან *Flume*-ის საშუალებით ვებსაიტებიდან, მაშინ როდესაც სტრუქტურირებული ჩანაწერები (სტატიები, პატენტები) მოდის მონაცემთა ბაზებიდან. მონაცემთა დამუშავება იყენებს *Hadoop*-ს პაკეტური ამოცანებისთვის, *Storm* ან *Spark*-ს ნაკადების მართვისთვის და სპეციალიზებულ ხელოვნური ინტელექტის ინსტრუმენტებს (მაგ. *TensorFlow*, *PyTorch*) ანალიტიკისთვის. ყველა მონაცემი ერთიანდება ე.წ. "შუალედური მონაცემთა პლატფორმის" (data middle platform) საშუალებით – ეს არის საერთო ინფორმაციული სქემა, მეტამონაცემთა რეესტრი და კატალოგების სისტემა, რომელიც ხელს უწყობს სხვადასხვა წყაროდან მიღებული ჩანაწერების კორელაციას. ამ ერთიანი მონაცემთა ფენის თავზე სისტემა უზრუნველყოფს ინტელექტუალურ სერვისებს, როგორიცაა ტექნოლოგიური

ტენდენციების მონიტორინგი, კონკურენტული ანალიზი და სცენარების პროგნოზირება ამრიგად, არქიტექტურა აერთიანებს დიდი მონაცემების ინსტრუმენტებს ინტელექტუალური ანალიზის სამუშაო პროცესში, რომელიც მოიცავს შემდეგ ეტაპებს: *მონაცემთა მიღება → ინტეგრაცია → ანალიზი → ინტელექტუალური პროდუქტი* (ჰუა და ჟენგი 2021, 372).

დიდი მონაცემების ინტელექტუალური ანალიზის ჩარჩო ჩვეულებრივ წარმოადგენს მრავალფენიან არქიტექტურას, რომელიც აერთიანებს: მონაცემთა შეგროვებას, მასშტაბირებად საცავს, განაწილებულ დამუშავებას, მოწინავე ანალიტიკს და Front-end აპლიკაციებს. აღნიშნული სისტემა უნდა შეესაბამებოდეს ჰეტეროგენული (მრავალფეროვანი) წყაროების მონაცემებს და უზრუნველყოფდეს როგორც პაკეტურ, ისე ნაკადური რეჟიმების მხარდაჭერას. რეფერენს-არქიტექტურები (მაგალითად, NIST, SEI) ამ ფუნქციებს ანაწილებენ მოდულებად და ამავდროულად ასახავენ ჯვარედინ მოთხოვნებს, როგორებიცაა უსაფრთხოება და მასშტაბურობა. პროგრამულ პრაქტიკაში, ეს ფენები ხორციელდება თანამედროვე დიდი მონაცემების პლატფორმებით, როგორებიცაა *Hadoop, Spark, Kafka, Storm, NoSQL* საცავები და ხელოვნური ინტელექტის ხელსაწყოები. შედეგად იქმნება მოქნილი მონაცემთა გადამამუშავებელი არხები (pipelines), რომლებიც დროულ, მონაცემებზე დაფუძნებულ ინტელექტს ქმნიან.

4.6 ფრაგმენტებისგან ცოდნის აგება გენეტიკური ალგორითმის გამოყენებით

4.6.1. ცოდნის გრაფის ენტროპიის მინიმიზაციის ამოცანა

მოცემულ პარაგრაფში ჩვენ განვიხილავთ ცოდნის აგების ჩვენეულ კონცეპციას, რომელიც პრინციპულად წარმოადგენს მოლეკულური გენერაციის სხვადასხვა პარადიგმების, კერძოდ, როგორიცაა drug design, docking, jigsaw ერთგვარ გადმოტანას ცოდნის ინჟინერიის სფეროში. ეს განპირობებულია მოლეკულური გეომეტრიული მსგავსებით ცოდნის გრაფულ წარმოდგენასთან. მოგეხსენებათ, რომ ყოველი მოლეკულა არის გრაფული სტრუქტურა, რომელიც რაღაც პარამეტრებითა და თვისებებით ხასიათდება. თავის მხრივ, მოლეკულის გრაფული მოდელი შედგება უფრო მცირე მოლეკულურ-ატომარული კომპონენტებისგან, რომელთა შეერთების

მიხედვითაც განისაზღვრება ყოველი მოლეკულის თვისებები. სწორედ ამ კომპონენტებისა და მათი შეერთების კონფიგურაციის შერჩევით ხდება სხვადასხვა მოლეკულების აწყობა-გენერაცია.

თუ გარკვეულ პარალელებს გავავლებთ, ცოდნის ინჟინერიის კუთხით, დავინახავთ, რომ ეს პროცესი მართლაც რაღაცით წააგავს მოლეკულურ-ატომარულ კომპონენტებს შორის ურთიერთშეკავშირების ანუ ბმების პროცესს, როცა თავსებადობა შეიძლება დახასიათდეს ამ ურთიერთშეერთებათა გარკვეული ენერგეტიკული წინააღმდეგობით.

ცოდნის ინჟინერიის თვალსაზრისით, ჩვენი კონცეპციის შემთხვევაში, როდესაც ვსაუბრობთ ცოდნის გრაფულ მოდელზე, როცა ხდება დიდი მონაცემების, კერძოდ ტექსტური მასივის დანაწევრება (Splitting) და მონაცემთა თითოეული ქვესიმრავლისთვის ცოდნის გრაფული ფრაგმენტების მიღება, შემდგომ პროცესის დასკვნით ფაზაში უკვე აუცილებელია ცოდნის ამ ფრაგმენტების „შეკოწიწება“ (Assembly) და ერთიანი ცოდნის აგება. სწორედ, ამ ფაზაში ცოდნის ფრაგმენტების ურთიერთქმედება შეიძლება შეფასდეს გარკვეული ენტროპიით, რაც ცოდნის სხვადასხვა ფრაგმენტის გრაფული მოდელის მწვერვალებს შორის ინფორმაციული შეუთავსებლობის ექვივალენტურია.

როგორც მოლეკულურ მოდელირებაში, ისე ცოდნის ინჟინერიაში, გრაფულ მოდელში ყოველი წიბო-რკალის წარმოდგენის დროს შეიძლება გამოყენებულ იქნას ე.წ. აქსონ-დენდრიტული სინაპსური მიდგომა. ეს საშუალებას იძლევა როგორც მცირე მოლეკულურ-ატომარული კომპონენტების შეკავშირების, ისე მიღებული ცოდნის ფრაგმენტების „შეპირაპირების“ (docking) პროცესისთვის განისაზღვროს შეთავსებულობის (ან შეუთავსებლობის) ფუნქცია, როგორც გარკვეული ენერგეტიკული წინააღმდეგობის ან ინფორმაციული ენტროპიის, იგივე თითოეული წიბო-რკალის აქსონ-დენდრიტული სინაპსების მნიშვნელობა.

ორივე სფეროსთვის იკვეთება ოპტიმიზაციის, კერძოდ მინიმიზაციის ამოცანა როგორც მცირე მოლეკულურ-ატომარულ კომპონენტებს შორის ენერგეტიკული წინააღმდეგობის, ისე ცოდნის ფრაგმენტების მწვერვალებს შორის არსებული ენტროპიის მინიმიზაციის კრიტერიუმით.

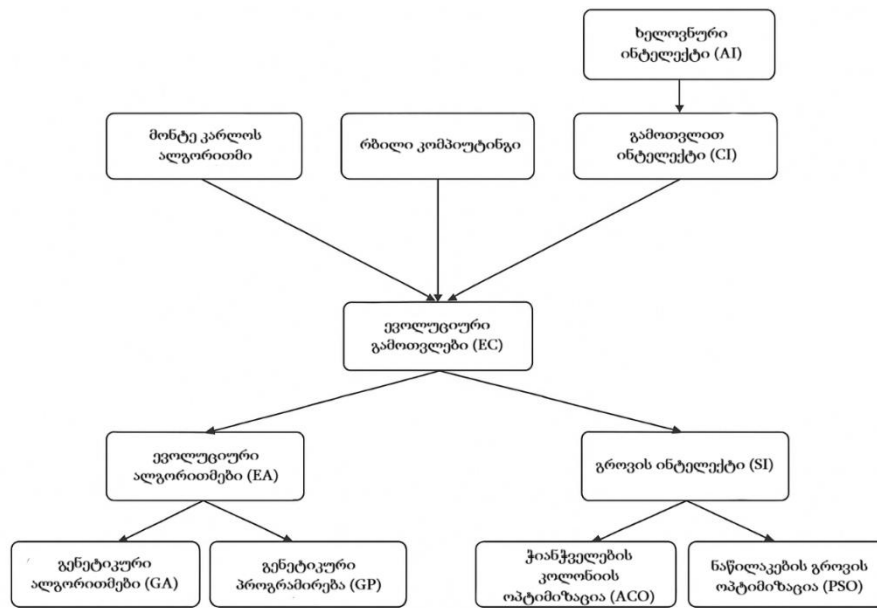
ამგვარად, მოცემული პარაგრაფი უნდა მიედგინას თეორიაში კარგად ცნობილ ე.წ. jigsaw puzzling მიდგომას, როგორც ფრაგმენტების შეერთებათა მრავლობითი ვარიაციების გადარჩევის პროცესს. აქ, ფაქტობრივად, ცოდნის აგების ჩვენეული კონცეპცია გულისხმობს ფრაგმენტების შეერთებათა თითოეული კომბინაციის (გრაფის კონფიგურაციის) შეფასებას მისი ოპტიმიზაციის ანუ ენტროპიის მინიმიზაციის თვალსაზრისით. აღნიშნული ამოცანა მიეკუთვნება დიდი განზომილების, მრავალექტრემუმიან, არაწრფივი ოპტიმიზაციის კლასს, რომლის დეტერმინირებული მეთოდებით (მათ შორის, დინამიური პროგრამირების მეთოდის გამოყენების შემთხვევაშიც) გადაწყვეტა დაკავშირებულია დიდ მოცულობის გამოთვლით სირთულეებთან.

4.6.2. ევოლუციური ალგორითმების კლასიფიკაცია

ხელოვნური ინტელექტი (AI) აერთიანებს მრავალგვარ ოპტიმიზაციის ალგორითმს, რომელთაც გააჩნიათ უნარი სწავლასა და დამოუკიდებელი გადაწყვეტილებების მიღებაზე. ევოლუციური ალგორითმები (EA) წარმოადგენს პოპულაციაზე დაფუძნებულ მიდგომებს, რომლებიც იყენებენ ბიოლოგიური ევოლუციის პრინციპებს, მათ შორის გამრავლება, მუტაცია, რეკომბინაცია, ბუნებრივი გადარჩევა და „ძლიერთა გადარჩენა“. EA ალგორითმები ინდივიდებს განიხილავენ, როგორც პრობლემის შესაძლო ამონახსნებს და იყენებენ მათ პარალელური ანალიზისთვის. საბოლოო მიზანია აარჩიოს ოპტიმალური ამონახსნი ძლიერი და სუსტი ინდივიდების შედარებით.

ზოგადად, ოპტიმიზაციის ალგორითმები ორ ძირითად კატეგორიად იყოფა: დეტერმინისტული (Deterministic) და ალბათური (Probabilistic). დეტერმინისტული მოდელები გამოიყენება იმ შემთხვევაში, როდესაც არსებობს ზუსტი და პირდაპირი კავშირი შემავალ პარამეტრებსა და მიღებულ ამონახსნებს შორის. ხოლო როდესაც მსგავსი კავშირი არ არის მკაფიო, ალგორითმის კონსტრუქციაში გამოიყენება ალბათური მიდგომა.

ნახ. 4.6.1.-ზე მოცემულია ევოლუციური ალგორითმების კლასიფიკაცია.



ნახ. 4.6.1. ევოლუციური ალგორითმების კლასიფიკაცია (წყარო: ავტორი).

4.6.3. გენეტიკური ალგორითმი (GA)

გენეტიკური ალგორითმი (GA) ეფუძნება ბუნებრივი ევოლუციის პრინციპებს და წარმოადგენს ეფექტურ მეთოდს რთული, მრავალგანზომილებიანი, მრავალკრიტერიუმიანი ამოცანების ოპტიმალური ამონახსნის სწრაფად მოსაძებნად. მისი წარმატება გამომდინარეობს მონაცემთა პარალელური დამუშავებიდან, რაც სწრაფ და შედეგიან ანალიზს უზრუნველყოფს. GA ალგორითმები აგებულია მემკვიდრეობითობისა და ევოლუციის კლასიკურ მოდელებზე, რის შედეგადაც წარმოადგენენ ადაპტურ სისტემას, რომელიც ბაძავს ცოცხალი ორგანიზმების განვითარების მექანიზმებს.

გენეტიკური ალგორითმი (GA) წარმოადგენს მეთოდს, რომელიც მიზნად ისახავს ამოცანის საპრობლემო არეში მიახლოებითი საუკეთესო ამონახსნის ძიებას. იგი ხელმძღვანელობს გენეტიკური მოდელით, რომელიც ეყრდნობა ბუნებრივი ევოლუციის პრინციპებს. GA-ში თითოეული ამონახსნი აღწერილია როგორც ქრომოსომა-გენების ერთობლიობა, რომელიც წარმოადგენს ამონახსნის კოდირებულ ფორმას. ქრომოსომაში კოდირების მეთოდის შერჩევა წარმოადგენს ალგორითმის კონსტრუქციის საწყის ეტაპს და გავლენას ახდენს პროცესის ეფექტიანობაზე. ამ მიზნისთვის გამოიყენება რამდენიმე ტიპის კოდირების მიდგომა:

- ორობითი კოდირება წარმოადგენს მეთოდს, სადაც თითოეული ქრომოსომა წარმოდგენილია ფიქსირებული ზომის ბიტურ სტრიქონებად. ეს ნიშნავს, რომ ამონახსნები გადადის ორობით სისტემაში, მხოლოდ ნულებისა და ერთების გამოყენებით. ამ სტრიქონებზე შესაძლებელია მარტივი ბიტური ოპერაციების შესრულება, როგორცაა: NOT, OR, AND, და XOR.

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

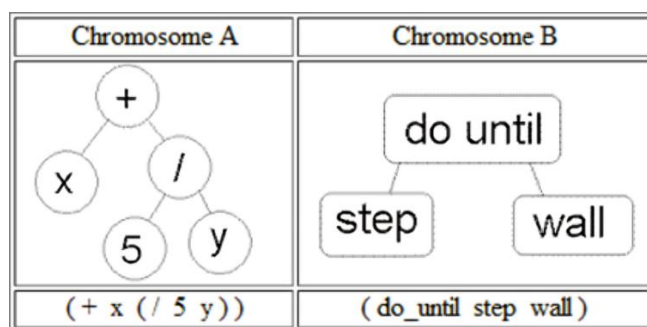
ნახაზი 4.6.2. გენეტიკური ალგორითმში ორობითი კოდირება (წყარო: ავტორი).

- ალფაბეტური და რიცხვითი კოდირება წარმოადგენს გენეტიკური ალგორითმის მეთოდს, რომელშიც ქრომოსომა ფორმირდება, როგორც სიტყვების ან ნამდვილი რიცხვების სტრიქონი. ამ ტიპის კოდირებით, თითოეული გენეტიკური ამონახსანი კონკრეტულ სიმბოლოებს ან რიცხვებით გამოხატულ მონაცემებს შეიცავს, რაც განსაზღვრავს მის ფუნქციურობას.

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

ნახაზი 4.6.3. გენეტიკური ალგორითმში ალფაბეტური და რიცხვითი კოდირება (წყარო: ავტორი).

- ხისებრი სტრუქტურით კოდირება არის მეთოდი, რომლის მიხედვითაც თითოეული ქრომოსომა წარმოდგენილია როგორც ხის სტრუქტურა, კომპონენტების იერარქიული განლაგებით. ეს მოდელი საშუალებას იძლევა, რომ ინდივიდის ქცევები, ლოგიკა ან გამოთვლითი ფორმულები აღწერილ იქნას სტრუქტურულად, რაც ქმნის მოქნილ და ადაპტურ პროცესს.

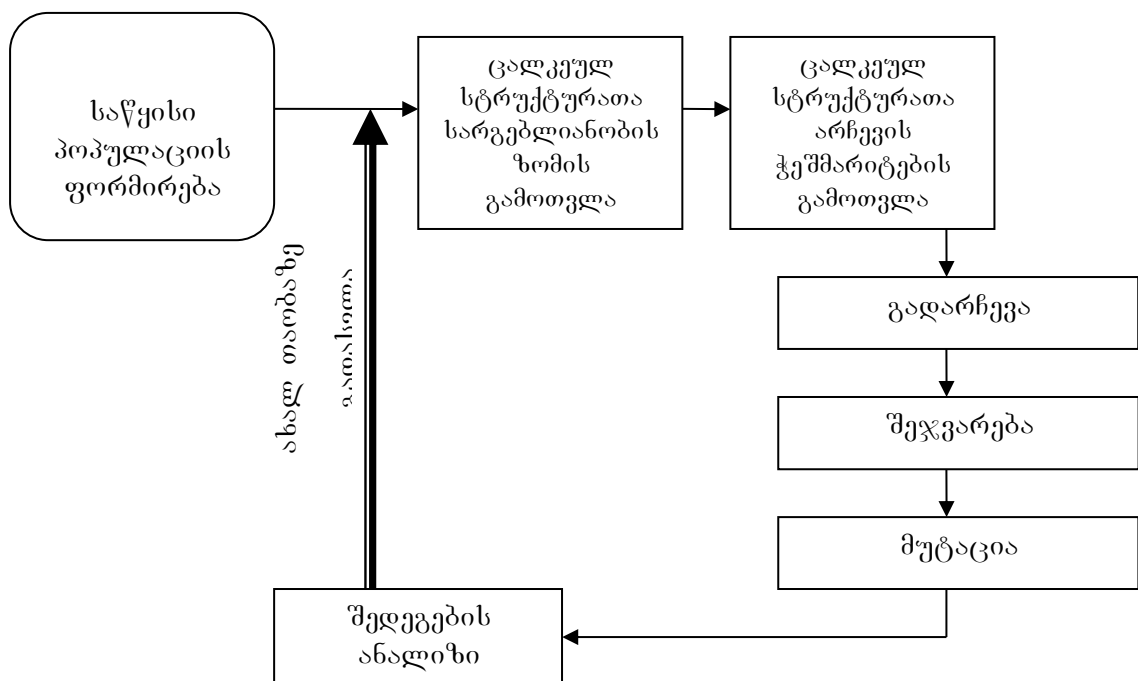


ნახაზი 4.6.4. GA-ში ხისებრი სტრუქტურის კოდირება (წყარო: ავტორი).

გენეტიკურ ალგორითმებში პოპულაციური გენეტიკის დინამიკა ხორციელდება სტრუქტურათა პოპულაციების ორგანიზაციის გზით, რომელიც დროთა განმავლობაში განიცდის ევოლუციას სამუშაო გარემოში მისი სტრუქტურის ქცევის შესაბამისად. თითოეული სტრუქტურა ალგორითმში წარმოდგება, როგორც მისი შემადგენელი ნაწილების (გენოტიპის) მიმდევრობა, რომელსაც ამუშავებს ძიების ოპერატორები.

სტრუქტურის კონკრეტული ინტერპრეტაცია ალტერნატიული ამონახსნების სივრცეში იძლევა ერთადერთ წერტილს განსახილველი პრობლემისათვის (ფენოტიპი), რომელიც შემდგომ შეიძლება ჩაირთოს ევოლუციურ პროცესში და რომელიც შეიძლება მივიღოთ სარგებლიანობის ზომად. ძიება ხორციელდება მიმდინარე საძიებო არეიდან სტრუქტურების მუდმივი ამორჩევით, სარგებლიანობის ხარისხის საფუძველზე.

არჩეულ სტრუქტურებზე ხორციელდება გენეტიკური ოპერაციები, რომლებიც ევოლუციაში ქმნიან ახალ სტრუქტურებს, შთამომავლობას. გენეტიკური ალგორითმები ზოგადად შეიძლება აღიწეროს შემდეგი სქემის სახით (ნახ.4.6.5.):



ნახაზი 4.6.5. გენეტიკური ალგორითმის ზოგადი სქემა (წყარო: ავტორი).

არჩევის ჭეშმარიტების გამოთვლა უზრუნველყოფს სელექციურ შერჩევას სტრუქტურების სასარგებლოდ, მოცემული სიმრავლიდან აიღება შედარებით უკეთესი სტრუქტურები. ასეთი არჩევითობის პრინციპით კარგი სტრუქტურები თანდათან იკავებენ მეტ ადგილს პოპულაციაში.

ძიების ადაპტური სტრატეგიის არსი მდგომარეობს არა ცალკეული სტრუქტურების აპრობაციაში, არამედ აპრობაციის შედეგად მიღებული ინფორმაციის გამოყენებაში. კარგი სტრუქტურები ინახება. იგი თავის მხრივ ქმნის უფრო დიდი რაოდენობის მსგავსი კონფიგურაციების შექმნის წინაპირობას. ეს სტრუქტურული კონფიგურაციები შეიძლება განვიხილოთ როგორც სივრცეში რეგულარული. ერთხელ თუ შეგვხვდა, ისინი ასრულებენ ე.წ. `საშენი ბლოკების` როლს ახალი სტრუქტურების შექმნაში.

ახალი სტრუქტურების შექმნის პროცესი ორიენტირდება საძიებო სივრცის მეტად პერსპექტიულ არეზე. გარდა ამისა სივრცის ასეთი გამოკვლევა არაცხადად მიმდინარეობს პარალელური სახით.

კროსოვერი გახლენს მოცემულ სტრუქტურულ კონფიგურაციას, თუ არჩეული გახლენის წერტილები მოხვდება ამ კონფიგურაციის კომპონენტების ორ მნიშვნელობას შორის. გახლენების დროს პოპულაციების შექმნის გარკვეული ტემპის მიღწევა დამოკიდებულია იმ კომპონენტის მნიშვნელობათა კონკრეტულ კონფიგურაციაზე, რომლებიც განსაზღვრავენ გახლენვას. აქედან შესაძლებელია დამშლელი ეფექტების ანალიზი, რომლებიც იწვევენ ახალი პოპულაციების წარმოქმნას. კროსოვერის ოპერატორს გააჩნია გადარჩენის გარკვეული ტემპის შენარჩუნების უნარი, რაც განპირობებულია გახლენვის დროს მცირედ განსაზღვრული ფრაგმენტების შენარჩუნებით.

არსებობს, აგრეთვე გახლენვის დროს დიდი განმსაზღვრელი ფრაგმენტების რღვევის ტენდენცია, მაგრამ რამდენადაც სტრუქტურები, რომლებიც ეკუთვნის სპეციფიკურ გახლენვას, უზრუნველყოფენ ფუნქციონირების მაღალ ხარისხს, მცირე განმსაზღვრელი ფრაგმენტებით თანდათან იწყებენ დამკვიდრებას პოპულაციაში, ხდება სხვა გახლენების განმსაზღვრელი ფრაგმენტების რიცხვის ეფექტური შემცირება, რაც ასუსტებს კროსოვერის ოპერატორის დამშლელ მოქმედებას.

მუტაციის ოპერატორი ამორჩევის ფაქტორზე არ ახდენს მნიშვნელოვან ზეგავლენას, რამდენადაც მას ძიებაში ენიჭება მხოლოდ ფონური როლი.

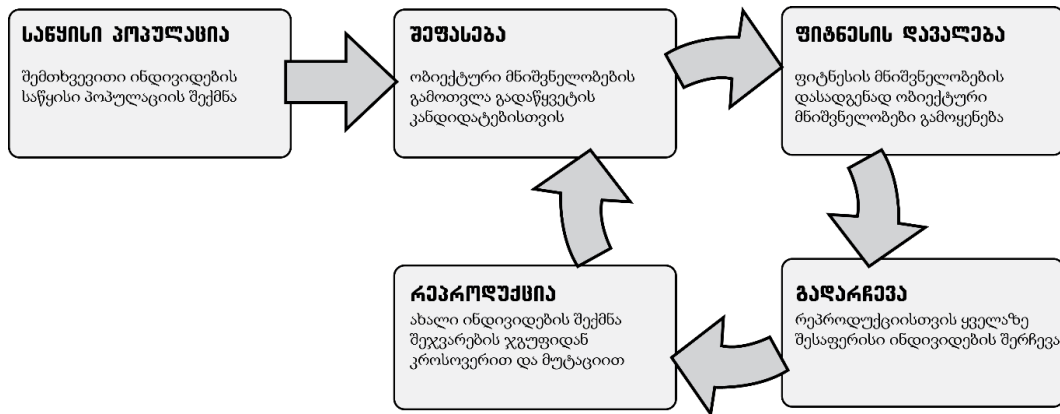
ასეთ ძიებაში შეიძლება აღიძვრას სიმძლეები, თუ არსებითი გახლეჩვა მოიცავს გრძელ განმსაზღვრელ ფრაგმენტებს. ეს პრობლემა წარმოსახვასთან არის დაკავშირებული და წარმოიშვება სტრუქტურული კომპონენტების მიმდევრობის წარუმატებელი ამორჩევისას. ამ შემთხვევაში ძიების პროდუქტიულობა ინვერსიის გამოყენების გზით მაღლდება. სტრუქტურაში კომპონენტების მიმდევრობის შეცვლის წყალობით, მისი გამოყენება ქმნის დიდი სიგრძის განმსაზღვრელ ფრაგმენტებში სიგრძის შემცირების ტენდენციას, ამის წყალობით კროსოვერის გამოყენების ფართო შესაძლებლობები ვლინდება.

ასე რომ, გენეტიკური ალგორითმის შესაძლებლობები დაკავშირებულია მის უნართან ცხადი გამოთვლებისა და დამახსოვრების გარეშე პარალელურად შეისწავლოს სტრუქტურული კომპონენტების თანმიმდევრობის კომბინაციათა დიდი რაოდენობა. ამის დახმარებით მიმდინარეობს საძიებო არეს კონცენტრირებული შესწავლა, ყურადღება მახვილდება იმ არეებზე, რომლებიც შეიცავენ საშუალოზე მაღალი სარგებლიანობის სტრუქტურებს. მეტნაკლებად პოპულაცია ხდება ფართოდ განსაზღვრული სივრცეში, რომელიც ძიებას არ აძლევს რომელიმე ლოკალურ მინიმუმზე გაჩერების საშუალებას.

მართალია გენეტიკური ალგორითმები გათვალისწინებულია სიმბოლოების სტრიქონებთან მუშაობისათვის, მაგრამ ის გამოიყენება აგრეთვე ფრეიმებზე ორიენტირებულ წარმოდგენებში. გენეტიკური ალგორითმი მუშაობს დიდ პოპულაციებთან, რომლებშიც ცალკეული წევრების სიმრავლე მკვეთრად არ განსხვავდება ერთმანეთისაგან.

გენეტიკური ალგორითმი ინტენსიურად იკვლევს პერსპექტიულ ან მიზნობრივ არეებს საძიებო სივრცეში, რამდენადაც მრავალჯერადი გამრავლებისა და შეჯვარების შედეგად ამ არეში გროვდება ჯაჭვების კიდევ და კიდევ უფრო დიდი რაოდენობა. მშობლის რანგში ალგორითმი ირჩევს საუკეთესო მიმდევრობას და ამ ჯაჭვის მიხედვით შემდგომ თაობაში წარმოიშვება უფრო მეტი შთამომავალი.

ალგორითმი შედგება რამდენიმე ძირითადი ბიჯისგან (ნახაზი 4.6.6). განვიხილოთ თითოეული მათგანი.



ნახაზი 4.6.6. გენეტიკური ალგორითმის მოქმედების ძირითადი ციკლი (წყარო: ავტორი).

ინიციალიზაცია

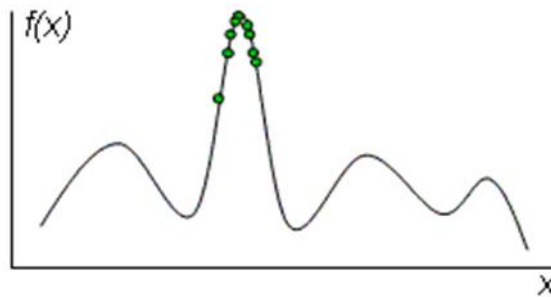
გენეტიკური ალგორითმის საწყისი პოპულაცია გენერირდება შემთხვევითი პრინციპით, ანუ თითოეული ინდივიდი (ქრომოსომა) საძიებო არის სხვადასხვა შესაძლო ამონახსნიდან ირჩევა შემთხვევით. პოპულაციის ზომა განისაზღვრება კონკრეტული ამოცანის სირთულისა და დამუშავების მოთხოვნების მიხედვით. როგორც წესი, იგი შედგება რამდენიმე ასეული ან ათასობით ინდივიდისგან.

ევოლუცია და „ფიტნეს ფუნქციის“ განსაზღვრა

თითოეული ქრომოსომა გენეტიკურ ალგორითმში ატარებს შესაბამის მნიშვნელობას, რომელსაც უწოდებენ ფიტნეს ფუნქციას (Fitness function). ეს ფუნქცია განაპირობებს, რამდენად მნიშვნელოვანი და ეფექტური ამონახსანია ქრომოსომა საძიებო სივრცეში არსებული ყველა შესაძლო ვარიანტს შორის. ფაქტობრივად, იგი ზომავს ინდივიდის „გენეტიკურ ძალას“ ანუ, რამდენად წარმატებულად პასუხობს ინდივიდი დაყენებულ ამოცანას. მიუხედავად შემთხვევითი გენერირების საწყის ეტაპზე, მაღალი ფიტნესის მქონე ინდივიდებს აქვთ დიდი შანსი გადარჩნენ და მოხვდნენ მომავალ პოპულაციებში. ეს ნიშნავს, რომ სელექციის პროცესში უპირატესობა ენიჭებათ ძლიერ გენებს, რომლებიც უფრო ეფექტური გადაწყვეტილების მიღებას უზრუნველყოფენ.

პოპულაციის ცხოვრების ციკლი არის რამდენიმე შემთხვევითი შეჯვარება და მუტაცია, რომლის შედეგადაც პოპულაციას ემატება ახალი ინდივიდუმების

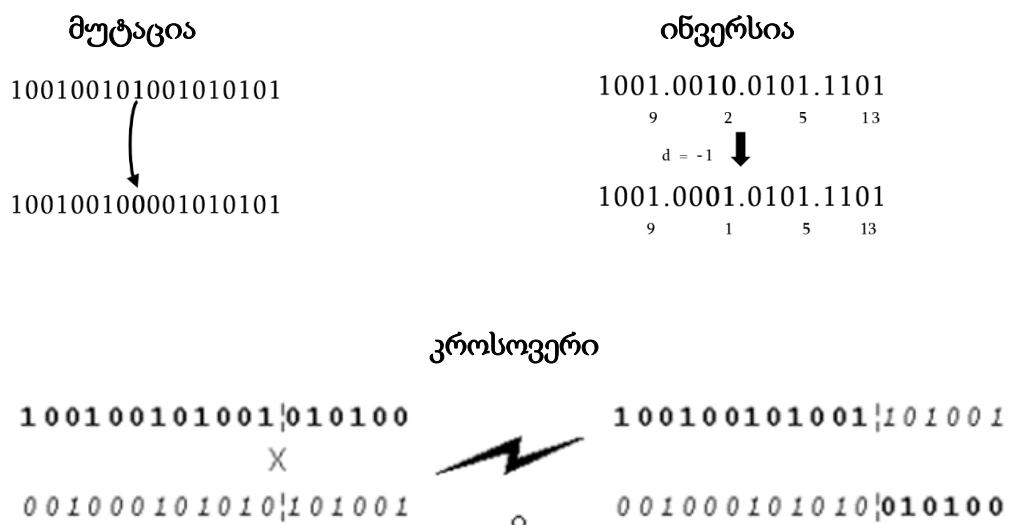
გარკვეული რაოდენობა. გადარჩევის დროს ხდება ძველიდან ახალი პოპულაციის ფორმირება, რომლის შედეგადაც ძველი პოპულაცია ილუპება. ამის შემდეგ ახალ პოპულაციაზეც ვრცელდება კროსოვერის, მუტაციის და გადარჩევის ოპერაციები. მომდევნო პოპულაცია ფორმირდება მიზნობრივი ფუნქციის შესაბამისად. რაც უფრო შეგუებადია ინდივიდუმი, მისი კროსოვერში მონაწილეობისა და გამრავლების უფრო მეტი ჭეშმარიტება არსებობს. გა-ს გაჩერების კრიტერიუმად ჩაითვლება პოპულაციის კრებადობა (*convergence*), ეს ის მდგომარეობაა, როდესაც ყველა სტრიქონი თავს მოიყრის ოპტიუმის არეში და შესაბამისად, მათ აქვთ ერთნაირი მნიშვნელობა(ნახ.4.6.7.):



ნახაზი 4.6.7. პოპულაციის კრებადობა (წყარო: ავტორი).

აღნიშნული მდგომარეობა აღნიშნავს, რომ მიღწეულია ოპტიმალურთან მიახლოებული ამონახსნი. საბოლოო ამონახსნად შეიძლება ჩაითვალოს ბოლო თაობის ყველაზე მაღალი შემგუებლობის მქონე არსება.

გენეტიკური ოპერატორები. ისინი იღებენ მიმდინარე საძიებო არედან სტრუქტურებს და ქმნიან ახალ სტრუქტურებს.



ნახაზი 4.6.8. გენეტიკური ოპერატორები (წყარო: ავტორი)

ახალი სტრუქტურების აგებისას მათი შემდგომი აპრობაციისათვის, კროსოვერის ოპერაცია გამოიყენებს ინფორმაციას, რომელიც არსებობს მიმდინარე სტრუქტურებში. თუ კონკრეტული ინფორმაცია არ არის, რომელიც შეიძლება დაიკარგოს წინა იტერაციის ეტაპზე გადარჩევის პროცესში, მაშინ ამ ოპერატორს არ შეუძლია შექმნას მისი შემცველობის ახალი სტრუქტურა. პოპულაციაში ახალი ინფორმაციის შესატანად გათვალისწინებულია მუტაციის ოპერატორი, რომელიც თავისუფალი სახით ცვლის არჩეული სტრუქტურის ერთ ან რამდენიმე კომპონენტს.

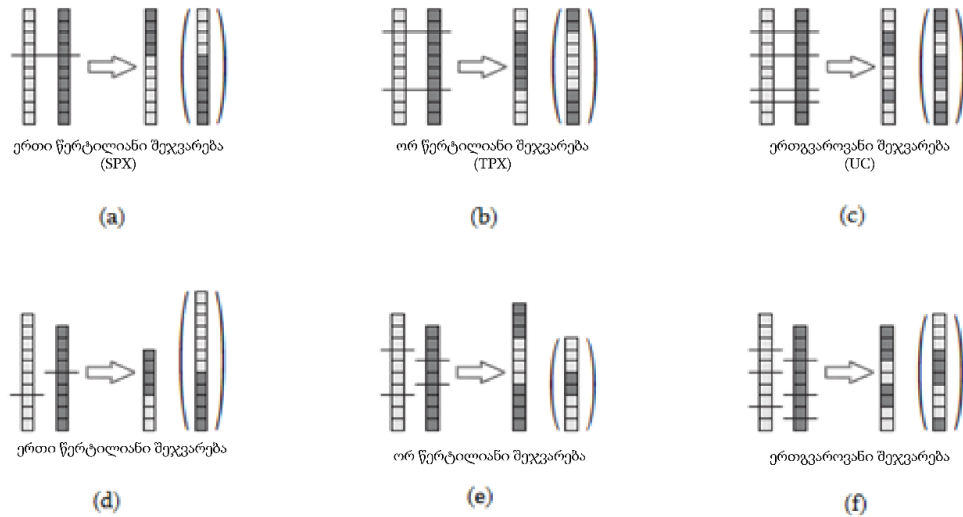
სელექცია

სელექცია არის პროცესი, რომლის დროსაც პოპულაციის ინდივიდები ირჩევიან რეპროდუქციისთვის, ფიტნეს ფუნქციის გათვალისწინებით, რაც მიზნად ისახავს შემდეგი თაობის ფორმირებას. ამ პროცესის შედეგად სავარაუდოდ, ახალი თაობის ინდივიდები, უფრო ძლიერი გენების მატარებლები იქნებიან, მათ წინამორბედებთან შედარებით. სელექციის პროცესში აუცილებელია რამდენიმე მნიშვნელოვანი ასპექტის გათვალისწინება. თავდაპირველად, საჭიროა განისაზღვროს, თუ რამდენი შთამომავალი უნდა წარმოიქმნას მომდევნო თაობისთვის. ასევე მნიშვნელოვანია სელექციის ინტენსივობის დაბალანსება: ზედმეტად მკაცრი სელექცია იწვევს გენეტიკური მრავალფეროვნების შემცირებას, რაც ართულებს მომავალში ადაპტაციისა და ევოლუციის შესაძლებლობებს; ხოლო ზედმეტად სუსტი სელექცია აფერხებს ევოლუციურ პროგრესს და ანელებს სისტემის ოპტიმიზაციას.

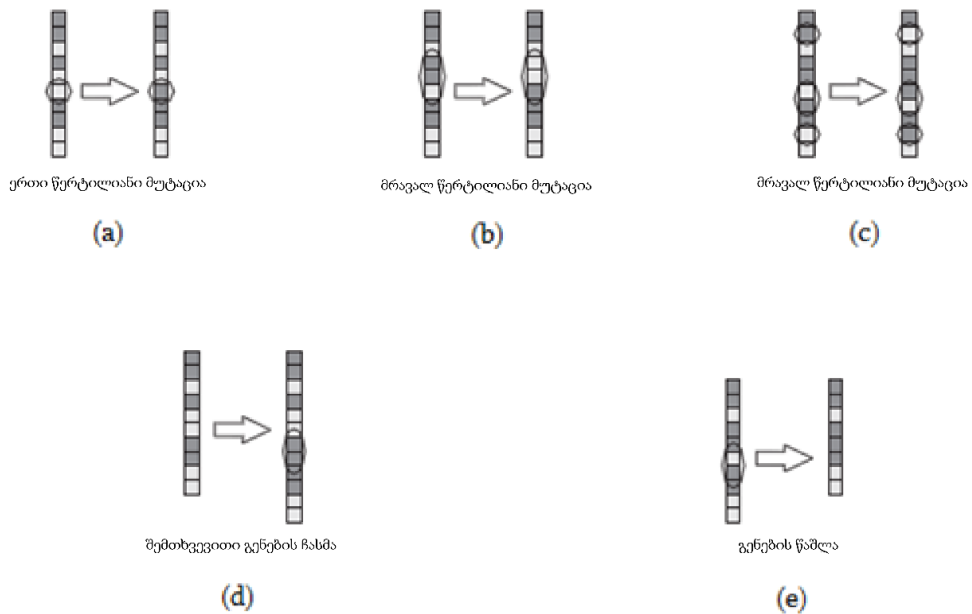
რეპროდუქცია და პოპულაციის შეცვლა

რეპროდუქცია არის პროცესი, რომლის საშუალებითაც წარმოიქმნება შემდეგი თაობის პოპულაცია წინა თაობის ინდივიდების საფუძველზე. ეს პროცესი ხორციელდება გენეტიკური ოპერატორების გამოყენებით, რაც გულისხმობს, რომ ახალი ინდივიდი ყალიბდება წინა თაობის ერთი ან რამდენიმე ინდივიდისგან კონკრეტული გენეტიკური ოპერაციის შესრულების გზით. გენეტიკურ ალგორითმებში გამოიყენება სხვადასხვა ტიპის ოპერატორები, როგორცაა ინვერსია, გენის დუბლირება, გადაჯგუფება, კოლონიზაცია (colonization), მიგრაცია და სხვა. თუმცა, კლასიკური გენეტიკური ალგორითმის მოდელში ძირითადად ორი ოპერატორია გამოყენებული: შეჯვარება (*Crossover*) და მუტაცია (*Mutation*), რომლებიც

შესაბამებთან ნახაზებს 4.6.9 და 4.6.10. რეპროდუქციის შედეგად მიღებული ახალი თაობა ანაცვლებს წინა თაობას, რაც განსაზღვრავს ევოლუციური პროცესის უწყვეტობას და სისტემის გაუმჯობესების პოტენციალს.



ნახაზი 4.6.9. Ga-ში ქრომოსომების შეჯვარება (წყარო: ავტორი)



ნახაზი 4.6.10. გენეტიკურ ალგორითმში ქრომოსომების მუტაცია (წყარო: ავტორი)

ალგორითმის დასრულება

ალგორითმი გრძელდება მანამ, სანამ „დასრულების პირობა“ არ შესრულდება, ანუ არ შესრულდება ჩამოთვლილთაგან ერთ-ერთი:

- ნაპოვნია პასუხი, რომელიც აკმაყოფილებს მინიმალურ კრიტერიუმებს;

- ალგორითმმა წინასწარ განსაზღვრული რეპროდუქციების რიცხვს გადააჭარბა;
- ალგორითმმა წინასწარ განსაზღვრულ „ბიუჯეტს“ (დრო/სხვა რესურსები) გადააჭარბა;
- შესაძლო ამონახსნის „ფიტნეს ფუნქციის“ მნიშვნელობამ მაქსიმუმს მიაღწია ან შემდეგი იტერაციების შედეგად აღარ უმჯობესდება (ჯანელიძე, მეფარიშვილი და ხაჩიძე 2023, 57);

ალგორითმის შესრულება გრძელდება მანამ, სანამ არ დაკმაყოფილდება დასრულების პირობა (termination condition), რაც გულისხმობს შემდეგ წინასწარ განსაზღვრულ კრიტერიუმთაგან ერთის ან მეტის შესრულებას:

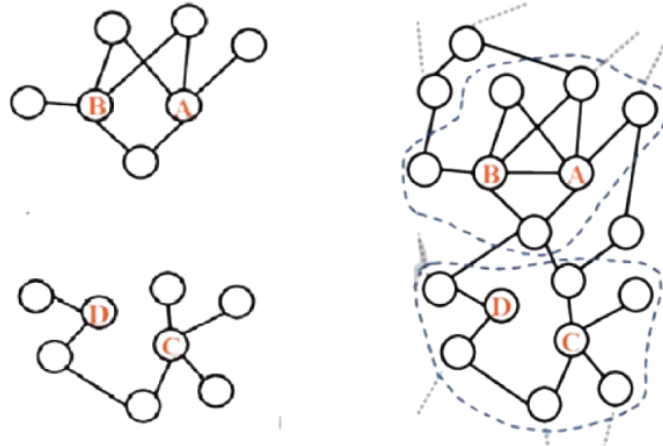
- მიღწეულია ამონახსნი, რომელიც აკმაყოფილებს მინიმალურ მისაღებ მოთხოვნებს;
- შესრულებული რეპროდუქციების რაოდენობამ გადააჭარბა წინასწარ განსაზღვრულ ზღვარს;
- ამოიწურა წინასწარ დადგენილი რესურსული ლიმიტი, როგორცაა დრო ან გამოთვლითი ბიუჯეტი;
- ამონახსნის ფიტნეს ფუნქციის მნიშვნელობამ მიაღწია მაქსიმუმს, ან ზედიზედ რამდენიმე იტერაციის განმავლობაში გაუმჯობესება აღარ ფიქსირდება.

ეს პირობები უზრუნველყოფს ალგორითმის ეფექტიან დასრულებას ისე, რომ თავიდან აიცილოს უსასრულო გამოთვლითი პროცესი ან რესურსების არარაციონალური გამოყენება.

4.6.4. ფრაგმენტებისგან ცოდნის აგება გენეტიკურ ალგორითმის გამოყენებით

მოცემულ კონტექსტში, უფრო მიზანშეწონილად შეიძლება წარმოვიდგინოთ ევოლუციური, ევრისტიკული ალგორითმები, კერძოდ გენეტიკური ალგორითმები, რომელიც ხასიათდება ოპტიმუმის პოვნის პროცესში სწრაფი კრებადობით. ამრიგად, განვიხილავთ გენეტიკურ ალგორითმზე (GA) დაფუძნებულ მიდგომას, როგორც ფრაგმენტების შეერთებათა მრავლობითი ვარიაციების გადარჩევის ეფექტურ პროცესს, რომლის შედეგად რამდენიმე იტერაციის მეშვეობით შესაძლებელია ცოდნის ფრაგმენტების უცნობი კონფიგურაციის, ზომებისა და ორიენტაციის შემთხვევაშიც ცოდნის გრაფის აწყობა ენტროპიის მინიმიზაციის კრიტერიუმით.

ჩვენი მიდგომა ძირითადად ეფუძნება ახალ კროსოვერის ტექნიკას, ცოდნის სხვადასხვა ფრაგმენტის მწვერვალების დაკავშირების ვარიაციების გადარჩევას გენეტიკური ოპერაციების მეშვეობით.



ნახაზი 4.6.11. ცოდნის ფრაგმენტების მწვერვალების დაკავშირება (წყარო ავტორი).

ჩვენ ვვარაუდობთ, რომ არ არსებობს ფრაგმენტების ორიენტაციის შესახებ წინასწარი ცოდნა. GA მიდგომის დროს ელემენტების (მაგ., ქრომოსომის წარმოდგენა, კროსოვერის ოპერატორი და ფიტნეს ფუნქცია) საჭიროა, რომ ყოველი გადაწყვეტა უნდა იყოს ვალიდური, ანუ ცოდნის აწყობის ყველა ფრაგმენტი უნდა გამოჩნდეს ერთხელ და მხოლოდ ერთხელ, დაკარგული და/ან დუბლირებული ფრაგმენტების გარეშე. იტერაციათა გადარჩევის პროცესში მოხდება ცოდნის მიმდინარე გრაფის თანდათანობითი, ევოლუციური გაუმჯობესება.

გრაფის სრული კომბინატორული სიმრავლე წარმოადგენს ფრაგმენტების მწვერვალთა რაოდენობის დეკარტულ ნამრავლს. მაგალითად, თუ გვაქვს ცოდნის გრაფის n ფრაგმენტი $FG_1, FG_2, \dots, FG_n,$

სადაც: თითოეული ფრაგმენტი $FG_i = (V_i, E_i)$, მაშინ მათი დეკარტული ნამრავლი ზოგადი ფორმით იქნება:

$$FG_1 \square FG_2 \square \dots \square FG_n,$$

ხოლო მწვერვალების სიმრავლე იქნება:

$$V(FG) = V_1 \times V_2 \times \dots \times V_n$$

ანუ ყოველი წვერო არის n -ელემენტიაანი მწკრივი $(v_1, v_2, \dots, v_n), v_i \in V_i$.

მართალია, ჩვენს მიერ შემუშავებული გენეტიკური ალგორითმი ეფუძნება ევოლუციური თეორიის ფუნდამენტურ პრინციპებს, თუმცა მისი სტრუქტურა და

მოქმედების მექანიზმები მნიშვნელოვნად განსხვავდება ტრადიციული ბიოლოგიური ევოლუციის მოდელისგან. შემოთავაზებული ალგორითმი შედგება ორი პრინციპული ფაზისგან: წინასწარი დამუშავება (Preprocessing) და საკუთრივ გენეტიკური ოპერაციების შესრულება.

წინასწარი დამუშავების დანიშნულებას წარმოადგენს გრაფის სრული კომბინატორული სიმრავლედან წინასწარ დადგენილი განზომილების მუშა სიმრავლის ანუ პოპულაციის ფორმირება (ფილტრაცია) გამოთვლითი ოპერაციების რაოდენობის შემცირების მიზნით.

ჩვენს შემთხვევაში, პოპულაცია წარმოადგენს ე.წ. ქრომოსომების მასივს, სადაც თითოეული ქრომოსომა შეესაბამება ფრაგმენტების კონკრეტულ შეერთებას, კომბინაციას, რომელიც ფასდება შეერთების თავსებადობის ანუ ენტროპიის გარკვეული მნიშვნელობით.

ჩვენს მიერ შემოთავაზებული ევოლუციური ალგორითმი შეიძლება წარმოვიდგინოთ შემდეგი ეტაპების სახით:

1. ფილტრაციის ანუ მუშა პოპულაციის ფორმირებისთვის ცოდნის გრაფის სრული კომბინატორული სიმრავლიდან რანდომიზებულად შეირჩევა ნებისმიერი ქრომოსომა, რომლისთვისაც განისაზღვრება საწყისი ფიტნეს-ფუნქცია.
2. შემდეგ, სრული კომბინატორული სიმრავლიდან კვლავ რანდომიზებულად შეირჩევა ახალი ქრომოსომა, რომლისთვისაც ასევე განისაზღვრება ფიტნეს-ფუნქცია.
3. თუ ახლად გენერირებული ქრომოსომის ფიტნეს-ფუნქცია არ აღემატება საწყისი ქრომოსომის ფიტნეს-ფუნქციას, მაშინ ეს უკანასკნელი ჩართული იქნება მუშა პოპულაციის ფაილში. ხოლო საპირისპირო შემთხვევაში იგი აღარ განიხილება.
4. შემოთავაზებული ალგორითმის წინასწარი დამუშავების ანუ ფილტრაციის პროცესი ფაზა გაგრძელდება მუშა პოპულაციის განზომილების ბოლომდე.
5. გენეტიკური ოპერაციების შესრულების ფაზაში უკვე ფორმირებული მუშა პოპულაციის ყოველი ქრომოსომისთვის განისაზღვრება ე.წ. ფიტნეს-ფუნქცია, რომელიც ენტროპიის შებრუნებულ სიდიდეს წარმოადგენს.

6. ამის შემდეგ, მიზნობრივი ფუნქციის შეფასებისთვის თითოეული ქრომოსომისთვის გამოითვლება ე.წ. შეგუებადობის ალბათობა:

$$P_{sel}(i) = \frac{f(i)}{\sum_{i=1}^n f(i)}$$

7. ამ ეტაპზე ხდება მუშა პოპულაციის რანჟირება, ქრომოსომათა შეგუებადობის ალბათობების კლებადობის მიხედვით. რანჟირება ითვალისწინებს ქრომოსომისთვის რანგის მინიჭებას. რანგი ანუ ადგილი მუშა პოპულაციაში ავლნიშნოთ r_j ; ($j = \overline{1, M}$), სადაც: M მუშა პოპულაციის განზომილებაა.

8. მოცემული რანჟირების პროცესი უზრუნველყოფს მუშა პოპულაციის სტრუქტურირებულ დაყოფას სამ ქვეჯგუფად: მაღალი ფიტნესის მქონე ინდივიდები, ე.წ. ლიდერთა ჯგუფი, რომლებიც გამოყოფილნი არიან მაღალი ადაპტაციური უნარის საფუძველზე და შემდგომ ექვემდებარებიან კროსოვერის (Crossover) ოპერატორს ახალი თაობის ფორმირების მიზნით; დაბალი ფიტნესის მქონე ინდივიდები, ე.წ. აუტსაიდერთა ჯგუფი, რომელთა მონაწილეობა შემდგომ ევოლუციურ პროცესებში წყდება, ვინაიდან მათი გენეტიკური მახასიათებლები აღარ განიხილება ოპტიმალური გადაწყვეტილების მიღწევის გზაზე; საშუალო ფიტნესის მქონე ინდივიდები, რომლებიც, გარკვეული პირობების შემთხვევაში, შეიძლება აირჩიონ, როგორც დამატებითი რესურსი შერჩევისა და კროსოვერისათვის, რათა შენარჩუნდეს გენეტიკური მრავალფეროვნება და შეინარჩუნოს ევოლუციური პროცესის დინამიკა.

კროსოვერის ეტაპი გულისხმობს ლიდერთა ჯგუფში შემავალი ქრომოსომების დაწყვილებას მათი სორტირების მიხედვით. აუცილებელია გაითვალისწინოს, რომ იმ შემთხვევაში, თუ ლიდერთა ჯგუფში ინდივიდების რაოდენობა კენტია, წყვილების ფორმირება ხორციელდება საშუალო ფიტნესის მქონე ქრომოსომების ჩართვით, რათა ყველა წყვილს მიეცეს კროსოვერის ოპერაციის გავლაზე შესაძლებლობა.

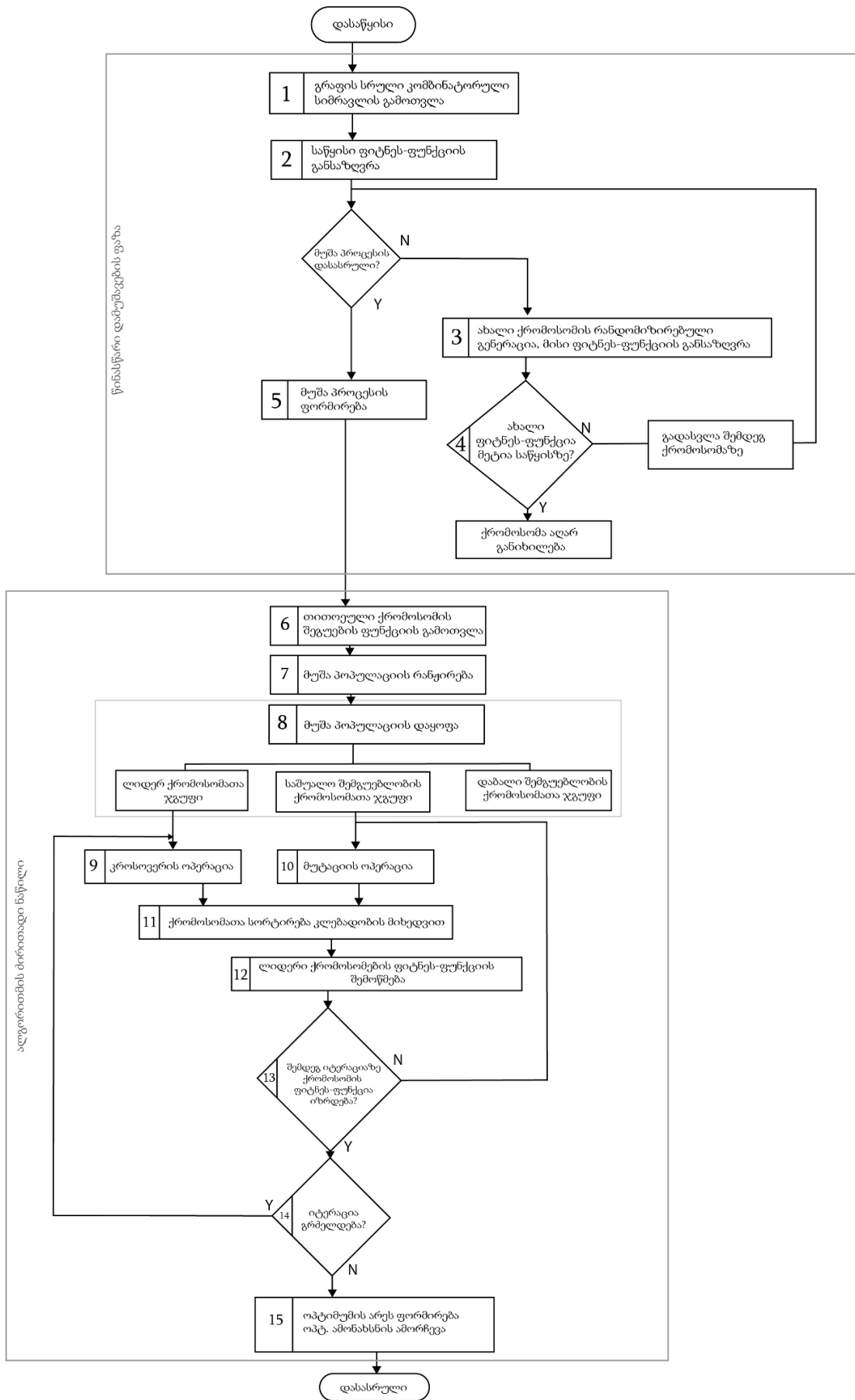
მუტაციის ეტაპზე, საშუალო ფიტნესის მქონე ჯგუფში შემავალი თითოეული ქრომოსომის წინასწარ შერჩეულ პოზიციაზე ხორციელდება თანრიგის ბიტური მნიშვნელობის ცვლილება. ეს პროცესი ემსახურება გენეტიკური მრავალფეროვნების შენარჩუნებას და ხელს უწყობს ახალ გენეტიკურ კომბინაციათა წარმოქმნას. მუტაციის

შესრულება ეფუძნება გადარჩევის (selection) მექანიზმს, რომლის მიხედვითაც განისაზღვრება, თუ რომელი ქრომოსომები დაექვემდებარებიან მუტაციას და როგორ გავრცელდება ცვლილება მათ სტრუქტურაში

დახარისხების ეტაპზე, შერჩევის, შეჯვარებისა და მუტაციის ოპერაციების შესრულების შემდეგ, პოპულაციაში არსებული ქრომოსომები ხელახლა სორტირდება კლებადობის მიხედვით, ფიტნეს ფუნქციის მნიშვნელობიდან გამომდინარე. ამ პროცესის შედეგად გამოიკვეთება ლიდერი ინდივიდი ქრომოსომა, რომელიც ფლობს მაქსიმალურ ადაპტაციურ უნარს მიმდინარე პოპულაციაში და წარმოადგენს ამ ეტაპზე საუკეთესო შესაძლო ამონახსნს.

9. **ალგორითმის დასრულების ეტაპზე** მოწმდება ლიდერი ქრომოსომის ფუნქციური (ფიტნეს) მნიშვნელობა. თუ მომდევნო იტერაციისას აღნიშნული მნიშვნელობა აგრძელებს ზრდას, ალგორითმი აგრძელებს შესრულებას და გადადის შემდეგ იტერაციაზე. იმ შემთხვევაში კი, თუ ლიდერის ფუნქციური მნიშვნელობა სტაგნაციას განიცდის ან კლებას იწყებს, სისტემა მიმართავს მუტაციის ოპერაციას ამ კონკრეტული ქრომოსომის მიმართ. თუ მუტაციის შედეგად მიზნობრივი ფუნქციის მნიშვნელობა კვლავ არ იზრდება, პროცესი წყდება, რაც ნიშნავს, რომ მიღებულია ოპტიმალური ან მის მიახლოებული ამონახსნი. აღსანიშნავია, რომ რამდენიმე იტერაციის შემდეგ პოპულაციის ინდივიდები ფიტნესის მნიშვნელობების თვალსაზრისით იწყებენ კონცენტრირებას ერთ განსაზღვრულ არეში ე.წ. ოპტიუმის ზონაში, რომლის ფარგლებიდანაც საბოლოოდ აირჩევა ოპტიმალური ამონახსნი.

ახლა უფრო დეტალურად განვიხილოთ ცოდნის აგების ალგორითმის ბლოკები (ნახაზი 4.6.12), რომელიც დიდი მოცულობის მონაცემების, კერძოდ ტექსტური მასივის დანაწევრების შედეგად (Splitting) მიღებული მონაცემთა თითოეული ქვესიმრავლისთვის ცოდნის გრაფული ფრაგმენტების „შეკოწიწებას“ (Assembly) და ერთიანი ცოდნის გრაფის აგებას ახორციელებს. როგორც ზემოთ ავღნიშნეთ, ალგორითმი შედგება ორი პრინციპული ფაზისგან: წინასწარი დამუშავება (Preprocessing) და გენეტიკური ოპერაციებზე დაფუძნებული ალგორითმის ძირითადი ნაწილის შესრულება.



ნახაზი 4.6.12. ცოდნის აგების ალგორითმის ბლოკები(წყარო: ავტორი).

წინასწარი დამუშავების ფაზა შედგება შემდეგი ბლოკებისგან:

ბლოკი 1. გრაფის სრული კომბინატორული სიმრავლე (სკს) მიიღება მაშინ, როდესაც გამოთვლილია ფრაგმენტთა თითოეული მწვერვალის რაოდენობათა დეკარტული ნამრავლი

ბლოკი 2. წინასწარ დადგენილი განზომილების მუშა სიმრავლის ანუ პოპულაციის ფორმირება (ფილტრაცია). ამისათვის, მუშა პოპულაციის ფორმირებისთვის ცოდნის გრაფის სრული კომბინატორული სიმრავლიდან რანდომიზებულად შეირჩევა ნებისმიერი ქრომოსომა, რომლისთვისაც განისაზღვრება საწყისი ფიტნეს-ფუნქცია.

ბლოკი 3. სრული კომბინატორული სიმრავლიდან კვლავ რანდომიზებულად შეირჩევა ახალი ქრომოსომა, რომლისთვისაც ასევე განისაზღვრება ფიტნეს-ფუნქცია.

ბლოკი 4. თუ ახლად შექმნილი ქრომოსომის ფიტნეს-ფუნქციის მნიშვნელობა საწყისი ქრომოსომის მნიშვნელობას ვერ აჭარბებს, მაშინ სწორედ საწყისი ქრომოსომა დარჩება სამუშაო პოპულაციის ფაილში; წინააღმდეგ შემთხვევაში, იგი გამოვირიცხება და აღარ გაითვალისწინება.

ბლოკი 5. ალგორითმის წინასწარი დამუშავების პროცესი ფაზა გაგრძელდება მუშა პოპულაციის განზომილების ბოლომდე ანუ პირობის შესრულებამდე ვბრუნდებით ბლოკ 3-ში.

მუშა პოპულაციის ფორმირების შემდეგ გადავდივართ ალგორითმის ძირითადი ნაწილის ბლოკების შესრულებაზე:

ბლოკი 6. ჩამოყალიბებული სამუშაო პოპულაციის ყოველი ქრომოსომისთვის განისაზღვრება *ფიტნეს-ფუნქცია*, რომელიც ენტროპიის ინვერსიას (შებრუნებულ სიდიდეს) წარმოადგენს. ამავე მიზნით, მიზნობრივი ფუნქციის შეფასებისას თითოეულ ქრომოსომაზე ითვლება მისი ადაპტაციის, ალბათობა.

ბლოკი 7. სამუშაო პოპულაციის ქრომოსომები ლაგდება ადაპტაციის ალბათობის შემცირების კლებადობით, ანუ უბეგრდება რანჟირება, რომლის ფარგლებშიც თითოეულ ქრომოსომას საკუთარი რანგი ენიჭება

ბლოკი 8. რანჟირების დასრულების შემდეგ სამუშაო პოპულაცია სამ სეგმენტად იყოფა: **ლიდერთა ჯგუფი** – მაღალი ადაპტაციის მქონე ქრომოსომები, რომლებიც

კროსოვერის ოპერატორს გადაეცემა; **აუტსაიდერთა ჯგუფი** – დაბალი ადაპტაციის მქონე ინდივიდები, რომლებიც შემდგომ ევოლუციურ პროცესში აღარ მონაწილეობენ; **საშუალო ჯგუფი** – საშუალო ადაპტაციის მქონე ქრომოსომები, რომლებსაც მხოლოდ მუტაციის ოპერატორი შეიძლება შეეხოს.

ბლოკი 9. კროსოვერის ეტაპზე სორტირების შედეგად დალაგებული ლიდერთა ჯგუფის ქრომოსომები წყვილებად ფორმირდება და გადადის შეჯვარებაზე. თუ პოპულაციაში წევრთა რაოდენობა კენტია, შეუწყვილებელ ლიდერს პარტნიორად საშუალო ჯგუფიდან შერჩეული ქრომოსომა ემატება.

ბლოკი 10. მუტაციის საჭიროებისას, საშუალო ჯგუფის თითოეულ ქრომოსომაში წინასწარ განსაზღვრულ პოზიციაზე მდებარე ბიტი (თანრიგის მნიშვნელობა) იცვლება.

ბლოკი 11. როგორც კი შერჩევის, შეჯვარებისა და მუტაციის ეტაპები დასრულდება, ქრომოსომები კვლავ კლებადობის მიხედვით დალაგდება, რის შედეგადაც გამოიკვეთება ის ლიდერი, რომელსაც ყველაზე მაღალი შემგუებლობის მაჩვენებელი ექნება.

ბლოკი 12. ამ ეტაპზე ხდება ლიდერი ქრომოსომის ფიტნეს-ფუნქციის მნიშვნელობის შემოწმება.

ბლოკი 13. თუკი მომდევნო იტერაციაში ლიდერის მიზნობრივი ფუნქციის მნიშვნელობა მატულობს, ძიება გრძელდება და ალგორითმი გადადის შემდეგ ეტაპზე.

ბლოკი 14. თუ ლიდერის მიზნობრივი ფუნქციის მნიშვნელობა ზრდას წყვეტს ან მისი მაჩვენებელს კლებისკენაა მიმართული, მაშინ შესაბამის ქრომოსომის კროსოვერის ეტაპზე გადავდივართ.

ბლოკი 15. თუ კროსოვერის შედეგად მიზნობრივი ფუნქციის მნიშვნელობა აღარ უმჯობესდება, ალგორითმი წყვეტს მუშაობას, რაც იმას ნიშნავს, რომ უკვე მიღწეულია ოპტიმალური ამონახსნი. რამდენიმე იტერაციის შემდეგ პოპულაციის წევრები მოექცევიან მნიშვნელობათა ერთ-ერთი „ოპტიმუმის არეში“, და სწორედ იქიდან აირჩევა უკეთესი ამონახსნი, რის შემდეგაც ალგორითმის შესრულება სრულდება.

დასკვნა

ცოდნის მოდელირება თანამედროვე სამყაროში წარმოადგენს კრიტიკულად მნიშვნელოვან კომპონენტს, რადგან ის შესაძლებელს ხდის კომპლექსური ინფორმაციის სტრუქტურირებასა და ინტერპრეტაციას და წარმოადგენს პროცესს, რომელიც აუცილებელია ინტელექტუალური სისტემებისთვის (მაგალითად: ხელოვნური ინტელექტი, ექსპერტული სისტემები, რეკომენდაციის ძრავები და სხვ.). იგი ასევე უზრუნველყოფს მსჯელობისა და სწავლის პროცესების ავტომატიზაციას, რაც აძლიერებს სისტემების ადაპტაციისა და გადაწყვეტილებების მიღების უნარს, ასევე უწყობს ხელს ინფორმაციის ხელახალ გამოყენებასა და გაზიარებას. შედეგად, ეს პროცესი განაპირობებს ცოდნის რესურსების (მაგალითად, ცოდნის გრაფები, ონტოლოგიები) სტანდარტიზაციას. გარდა ამისა, ცოდნის მოდელირება მნიშვნელოვან როლს ასრულებს თანამედროვე ხელოვნური ინტელექტის მოდელებში ჰიბრიდული ინტეგრაციის ხელშეწყობით, სადაც სიმბოლური და ქვესიმბოლური მეთოდები ერთმანეთს ერწყმის. დღევანდელ სამყაროში, სადაც ინფორმაცია ერთ-ერთი ყველაზე ღირებული რესურსია, ცოდნის მოდელირება წარმოადგენს აუცილებელ წინაპირობას ინტელექტუალური სისტემების განვითარებისათვის. ასეთმა სისტემებმა მნიშვნელოვანი წინსვლა განაპირობა ისეთ სფეროებში, როგორცაა მედიცინა, ინდუსტრიული ავტომატიზაცია, განათლება, მონაცემებზე დაფუძნებული გადაწყვეტილებების მიღება და სხვა.

კვლევის თემატიკა და პუბლიკაციები: დისერტაციაში განხილულია ისეთი თემები, როგორცაა განაწილებული ევოლუციური ალგორითმები, გენეტიკური პროგრამირება, გრაფული მოდელირებისათვის ცოდნის ინჟინერიის მიდგომები, ნაკადური მონაცემების დამუშავება, ცოდნისადმი ბიოინფორმატიკული მიდგომები და სხვა. კვლევის კომპონენტი მოიცავს გამოქვეყნებულ სამეცნიერო სტატიებს, რომლებიც შემდეგ საკითხებს ეხება:

- განაწილებული გენეტიკური ალგორითმების გამოყენება ნაკადური დიდი მონაცემების დამუშავებისთვის;
- ნაკადური მონაცემების განაწილებული დამუშავება გენეტიკური პროგრამირების გამოყენებით;

- ცოდნის ფილოსოფიის ახლებური გააზრება;
- ცოდნის ინჟინერიის ზოგიერთი ახალი ასპექტი;
- ცოდნის ამოღება დიდი ტექსტური მონაცემთა ნაკრებიდან;

თითოეული სტატია წარმოადგენს კვლევის შედეგს, რომელიც ამდიდრებს დისერტაციას ახალი მიდგომებით და იძლევა კვლევითი მიზნების მიღწევისათვის მნიშვნელოვან თეორიულ და პრაქტიკულ საფუძველს.

განაწილებული გენეტიკური ალგორითმი ნაკადური მონაცემებისთვის: დისერტაციაში წარმოდგენილია ერთ-ერთი ინოვაციური მიდგომა - გენეტიკური ალგორითმის განაწილებული ვერსია, რომელიც განკუთვნილია ნაკადური მონაცემების ოპტიმიზაციისთვის. კვლევა დეტალურად აღწერს აღნიშნული ალგორითმის სრული ოპერაციული ციკლს. ლამბდა არქიტექტურისა და განაწილებული გენეტიკური ალგორითმის კომბინაცია შესაძლებელს ხდის დიდი მოცულობის ნაკადურ მონაცემებზე ფართო მასშტაბის ოპტიმიზაციის ამოცანების გადაჭრას მაღალი წარმადობისა და დაბალი დაყოვნების პირობებში.

მსგავსი პრინციპი გამოყენებულია **ნაკადური მონაცემების განაწილებულ დამუშავებაშიც, გენეტიკური პროგრამირების გამოყენებით:** შემუშავდა ალგორითმი, რომელიც შემაჯავლი ნაკადიდან ოპტიმალური ამონახსნის პოვნის საშუალებას იძლევა. გენეტიკური პროგრამირების გამოყენება აღწერილია როგორც სრულყოფილი პროცესი, რომელიც უზრუნველყოფს რეალურ დროში ნაკადურ მონაცემებზე დაბალი დაყოვნებით რეაგირებას.

ცოდნის ინჟინერიის ახალი მიდგომები: დისერტაციაში განხილულია ცოდნის ინჟინერიის ახალი ასპექტებიც, სადაც მოცემულია ცოდნის წარმოდგენის ახალი ხედვა. ამ მიდგომის არსი მდგომარეობს იმაში, რომ ინფორმაცია უნდა განიხილებოდეს არა მატერიალურ, არამედ არამატერიალურ ფორმაში, კვანტურ ფორმაში, უწყვეტი ჰოლოგრამის სახით. გარდა ამისა, ნაშრომში წარმოდგენილია დოკინგისა და ასემბლინგის ალგორითმი, რომელიც განკუთვნილია სხვადასხვა წყაროდან მიღებული ცოდნის ფრაგმენტების ინტეგრაციისთვის. აღნიშნული ალგორითმი იძლევა საშუალებას გენერირდეს და სტრუქტურულად შეიკრას ცოდნის

ფრაგმენტები, რაც უზრუნველყოფს ცოდნის აგების პროცესის მოქნილობას და სიმბოლური-ქვესიმბოლური სტრუქტურების ეფექტიან შერწყმას.

ცოდნის ფრაგმენტების დოკინგი და ასემბლინგი: დისერტაციის მთავარ სიახლეს წარმოადგენს ალგორითმი, რომელიც განკუთვნილია ცოდნის ფრაგმენტების დოკინგისა და ცოდნის ასემბლინგისათვის. ამ ნაწილში წარმოდგენილია ცოდნის აგების კონცეფცია, რომელიც ეფუძნება მოლეკულური გენერაციის პარადიგმების კერძოდ, drug design-ის, მოლეკულური დოკინგისა და jigsaw puzzle-ის გადმოტანას ცოდნის ინჟინერიის სფეროში. აღნიშნული მიდგომა მოტივირებულია მოლეკულების გეომეტრიულ მსგავსებასა და ცოდნის გრაფულ წარმოდგენას შორის არსებული პარალელით. ამ კონცეფციის მიხედვით, ფრაგმენტების შეერთების თითოეული კომბინაცია (ანუ გრაფის კონფიგურაცია) შეფასდება ოპტიმიზაციის თვალსაზრისით, კერძოდ ენტროპიის მინიმიზაციის პრინციპის საფუძველზე. მოცემული ამოცანა მიეკუთვნება მაღალი განზომილების, მრავალექსტრემუმიან და არაწრფივი ოპტიმიზაციის ამოცანათა კლასს, რომლისთვისაც დეტერმინისტული მეთოდები (მათ შორის დინამიური პროგრამირებაც) დაკავშირებულია მაღალ გამოთვლილ სირთულეებთან.

თეორიული და პრაქტიკული ღირებულება: დისერტაციაში შემუშავებული გრაფული მანქანური სწავლების მოდელები და ალგორითმები იძლევა როგორც თეორიულ, ასევე პრაქტიკულ ღირებულებას ინფორმაციული სისტემებისთვის, როგორც დიდი მოცულობის, ისე რეალურ დროში დამუშავებადი მონაცემების კონტექსტში. აღნიშნული კვლევების ერთობლიობა, განაწილებული გენეტიკური ალგორითმები, გენეტიკური პროგრამირების მოდელები და ცოდნის ფრაგმენტების გრაფული ინტეგრაცია შესაძლებელს ხდის შემოთავაზებული მეთოდების ფართოდ ადაპტირებას თანამედროვე მონაცემთა მართვის სისტემებსა და ინტელექტუალური ანალიზის სფეროებში.

გამოყენებული ლიტერატურა

1. ა. დისესა, სმიტ III, ჯონ პ.,როშელი, ჯ. 1993. "Misconceptions reconceived: a constructivist analysis of knowledge in transition." *Journal of the Learning Sciences*, 3(2) 115 - 163.
2. tutorialspoint.com. 2022. www.tutorialspoint.com. 07 01. Accessed 05 25, 2025. https://www.tutorialspoint.com/data_mining/index.htm.
3. ალხამისი, ბ., ლი, მ., ჩელიკილმაზი, მ., დიაბი, მ., . 2022. "A Review on Language Models as Knowledge Bases." *arXiv - Computer Science > Computation and Language* 1 - 21.
4. ანილ, კ. 2015. *Business Intelligence and Data Mining*. New York: Business Expert Press, LLC .
5. ბერტრანი, პ, ფ. 2014. *archive.org*. 03 15. Accessed 05 25, 2025. <https://web.archive.org/web/20190312082929/http://www.datasalt.com/2014/01/lambda-architecture-a-state-of-the-art/>.
6. ბეჰერა, ლ., ჩილუკური, ვ. 2024. "End-to-End Data Pipelines: Redefining the Architecture of Data Engineering in Cloud Environments." *ESP International Journal of Advancements in Science & Technology* 26 -33.
7. გოგარე, ა. 2024. "Next-Generation Data Pipeline Designs for Modern Analytics : A Comprehensive Review." *International Journal of Scientific Research in Computer Science Engineering and Information Technology* 1 - 8.
8. დელგრანდი, ჯ., გლიმი, ბ., მაიერი თ., ტრუმბინსკი, მ., ვოლტერ, ფ. 2023. "Current and Future Challenges in Knowledge Representation and Reasoning." *arXiv* 4 - 36.
9. დერი, ლ., მენგისტუ, რ., ავე, ო. 2017. "Neural Combinatorial Optimization for Solving Jigsaw Puzzles:." *Stanford University* 1 - 8.
10. დეჰალი, რ., შარმა, მ., რაჯაბი, ე. 2025. "Knowledge Graphs and Their Reciprocal Relationship with Large Language Models." *MDPI* 1-26.
11. ვაენერი, ვ. 2022. www.kai-waehner.de. 11 19. Accessed 05 28, 2025. <https://www.kai-waehner.de/blog/2021/09/23/real-time-kappa-architecture-mainstream-replacing-batch-lambda/#:~:text=match%20at%20L351%20architecture%20in,seems%20like%20it%20would%20be%E2%80%9D>.
12. ვანგი, მ., ქიუი, ლ., წანგ, ქ. 2021. "A Survey on Knowledge Graph Embeddings for Link Prediction." *Symmetry* 1 - 29.
13. ვებერ, ნ., რუდინგერი, რ., ვან დურმი, ბ. 2020. "Causal Inference of Script Knowledge." *Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. 7583–7596.

14. ვებ-ონტოლოგიის-ენა. 2012. *www.w3.org*. 12 11. Accessed 05 25, 2025.
<https://www.w3.org/OWL/>.
15. ვოსნიადუ, ს. 2019. "The Development of Students' Understanding of Science." *Front. Educ.* 1 - 6.
16. თეიკენ, დ. 2022. *The Science of EDXML*. 9 15. Accessed 05 29, 2025.
<https://edxml.org/science/#:~:text=Emergence>.
17. იეონგ, კ., ზოუ ჟ., ჰუანგ ვ., ჯონგ, მ. 2024. "Structure-aware Attention based on Vector Symbolic Architectures." *ICLR 2025 Conference*. Singapore: ICLR 2025.
18. კლეიკო, დ., დევისი, მ., ფრედი, ე., კანერვა კ. 2023. "Vector Symbolic Architectures as a Computing Framework for Emerging Hardware." *arXi - Computer Science > Hardware Architecture* 1 - 31.
19. კოლელოვი, ბ., რეგლი, უ. 2024. "Neuro-Symbolic AI in 2024: A Systematic Review." *arXiv* 1-19.
20. კუმარი, ა., დინაკარანი, ს. 2020. "TEXTBOOK TO TRIPLES: Creating knowledge graph in the form of triples from AI TextBook." *Arxiv* 1 - 5.
21. ლი, დ. 2019. *databricks.com*. 09 21. Accessed 05 28, 2025.
<https://www.databricks.com/glossary/lambda-architecture#:~:text=and%20eventually%20corrects%20the%20data,has%20two%20very%20important%20functions>.
22. ლიანგ, კ., მენგი, ლ., ლიუ, მ., ლიუ, ი., ვანგ, ს... 2023. "A Survey of Knowledge Graph Reasoning on Graph Types: Static, Dynamic, and Multi-Modal." *arXiv:2212.05767v7* 1 - 20.
23. ლინ, ლ., ლინგ, ლ., ლუ., ჯ., ჟენგი, ჩ., სონგ, ლ. 2003. "Case-Based Reasoning Integrating with Direct-Case-Unkage for TacitKnowledge Management." *7th Pacific Asia Conference on Information Systems*. Adelaide: 7th Pacific Asia Conference on Information Systems. 1724-1733.
24. მეფარიშვილი, ჯანელიძე, ხაჩიძე. 2023. "Some New Aspects of Knowledge Engineering." *INTERNATIONAL SCIENTIFIC PRACTICAL CONFERENCE MODERN CHALLENGES AND ACHIEVEMENTS IN INFORMATION AND COMMUNICATION TECHNOLOGIES - 2023*. Tbilisi: Georgian Technical University. 28 - 35.
25. მონაცემთა-მოპოვება-კლასიფიკაცია. 2023. *geeksforgeeks.org*. may 06. Accessed 05 25, 2025. <https://www.geeksforgeeks.org/basic-concept-classification-data-mining/>.
26. მუვა, ს. 2021. "DATA PIPELINE ORCHESTRATION AND AUTOMATION: ENHANCING EFFICIENCY AND RELIABILITY IN BIG DATA ENVIRONMENTS." *nternational Journal of Core Engineering & Management* 316-321.

27. ნაზარუკსი, ვ., ოსისი, ი. 2021. "An Overview of Knowledge Representation With Frames." *Research Gate* 46 - 63.
28. ნაფადე, დ. 2025. "The Evolution and Modernization of Data Pipeline." *European Journal of Computer Science and Information Technology* 42 - 53.
29. ჟუ, ჯ., ტენგი, ბ. 2019. "A five-layer architecture for big data processing and analytics ." *Big Data Intelligence, Vol. 6, No. 1* 38 - 49.
30. რავალ. კ., განატრა. ტ. 2022. "Basics, types and applications of molecular docking: A review." *IP International Journal of Comprehensive and Advanced Pharmacology* 12 - 16.
31. რუზგასი, ტ., იაკუბელიენე, ტ., ბუვიტე, ა. 2016. "Big Data Mining and Knowledge Discovery." *Journal of Communications Technology, Electronics and Computer Science* 5.
32. სანჩეს-ლენგელინგი, ბ., რეიფი, ე., პირსი, ა., ვილჩკო, ა. 2021. "A Gentle Introduction to Graph Neural Networks." *Distill* 1 - 31.
33. სტამილე, კ., მარზულო, ა., დეუსებო, ე. 2021. *Graph Machine*. Birmingham : Packt Publishing.
34. ფანი, ქ., ვანგი, შ., ჰუნგ, ჯ., ჩენი, ჟ., სონგ, ი... 2023. "Graph Machine Learning in the Era of Large Language Models (LLMs)." *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, SUBMISSION 2023* 1 - 21.
35. ქლეინ. ჯ., ბუგლაკი. რ., ბლოკოვი. დ., ვუტკი. ტ., კუპერი. ბ.. 2016. "A Reference Architecture for Big Data Systems in the National Security Domain." *2nd International Workshop on BIG Data Software Engineering* 51 - 57.
36. შიაოჯუნ, ჩ., ლიუ, ი. 2024. "Integration of Fragmented Knowledge Based on Knowledge Graph." *International Conference on Education, Knowledge and Information Management*. Chengdu: ICEKIM. 547 - 555.
37. შმუელი, გ., ბრიუს, პ., ნიტინ, რ. 2018. *DATA MINING FOR BUSINESS ANALYTICS Concepts, Techniques, and Applications in R*. New Delhi: John Wiley & Sons, Inc.
38. შოლომონი, დ., დევიდ, ო., ნეთანიაჰუ, ნ. 2013. "AGenetic Algorithm-Based Solver for Very Large Jigsaw Puzzles." *Computer Vision Foundation* 1767 - 1774.
39. შრამი, ს., ვენერი, კ., შმიდი, უ. 2024. "Comprehensible Artificial Intelligence on Knowledge Graphs: A Survey." *Journal of Web Semantics* 1 - 23.
40. წითაშვილი, ლ., ჯანელიძე, გ., ხაჩიძე, ი. 2023. "ნაკადური მონაცემების განაწილებული დამუშავება გენეტიკური პროგრამირების გამოყენებით." *საერთაშორისო პერიოდული სამეცნიერო ჟურნალი ინტელექტი*, 12 18: 99 - 104.
41. ხაჩიძე, ი. 2024. "ცოდნის მოპოვება დიდი ტექსტური მონაცემთა ნაკრებიდან." *საერთაშორისო სამეცნიერო ჟურნალი "ინტელექტი"* 90-98.

42. ხაჩიძე, ი., ჯანელიძე, გ., მეფარიშვილი, ბ. 2024. "A New Understanding of the Philosophy of Knowledge." *International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)*, 03 24: 62 - 67.
43. ჯ მა, ტ., ჯ გარსია, რ., დანფორდი, ფ., პატრიცი, ლ., გალასო, ჯ., ლოიდ, ჯ. 2020. "Big data actionable intelligence architecture." *Journal of Big Data* 1 - 19.
44. ჯანელიძე, გ., მეფარიშვილი, ბ., ხაჩიძე, ი. 2023. "განაწილებული გენეტიკური ალგორითმების გამოყენება ნაკადური დიდი მონაცემების დამუშავებისთვის." *შრომები - მართვის ავტომატიზირებული სისტემები*, 05 14: 54 - 59.
45. ჰუა, ბ., ზენგი, ზ. 2021. "Architecture of S&T Intelligence Analysis SystemBased on Big Data." *International Conference on Digital Twins and Parallel Intelligence (DTPI)* 370 - 373.

ცხრილების ნუსხა

1. ცრილი. 1.1. მონაცემთა მოპოვების მნიშვნელოვანი ტექნიკები -----25.
2. ცრილი. 1.2. დაბნეულობის მატრიცა (Confusion Matrix) -----32.

სურათების /დიაგრამების/გრაფიკების ნუსხა

1. სურათი 1.1. მონაცემთა მოპოვების პროცესი ბიზნეს ანალიტიკის კონტექსტში -
-----21.
2. სურ.1.2. მონაცემთა მოპოვების სისტემების არქიტექტურა -----22.
3. სურ.1.3. მონაცემთა მოპოვების კლასიფიკაციის კრიტერიუმები -----24.
4. სურ.1.4: მონაცემთა მოპოვების ძირითადი პრობლემური საკითხები -----34.
5. სურ.2.1. სიტყვების ემბედინგების 3D ვიზუალიზაცია -----52.
6. სურ.3.1. მაღალი დონის მონაცემთა ნაკადი ლამბდა არქიტექტურაში -----72.
7. ნახ.3.2. განაწილებული გენეტიკური ალგორითმი -----94.
8. ნახ.3.2. მონაცემთა ნაკადში ოპტიმალური ამონახსნის პოვნა -----97.
9. ნახ.3.3. მშობელი ხეების დაწყვილება -----97.
10. ნახ.3.4. ხეებში კვების წერტილის პოვნისა და შეჯვარების ოპერაცია -----98.
11. ნახ.3.5. მუტაციის ოპერაცია -----99.
12. ნახ.3.6. ალგორითმი „წონითი კოეფიციენტების“ გამოყენებით -----99.
13. სურ.3.7: ცოდნის იერარქიული წარმოდგენა -----112.
14. სურ.3.8: ცნების ონტოლოგიური „ატომური“ მოდელი -----114.
15. სურ. 3.9: ონტოლოგიური ტრიპლეტი -----115.
16. სურ.3.10: მეტაცნობიერების ჰიპერგრაფი -----116.
17. სურ.3.11: ა) ნეირონული მოდელის ელემენტები. ბ) ნეირონული მოდელის
მაგალითი -----117.
18. სურ.3.12: ორმხრივი ინფორმაციის მიმოცვლა -----118.
19. სურ.3.13: ცოდნის აღმოჩენის პროცესი -----120.
20. სურ.3.14: ცოდნის მენეჯმენტის პროცესი -----121.

21. სურ.4.1. DIKW პირამიდა (DIKW Pyramid – მონაცემი, ინფორმაცია, ცოდნა და სიბრძნე) -----	124.
22. სურ.4.2. მონაცემთა ტიპები დებებში -----	125.
23. სურ. 4.3. ცოდნის ჰიპერგრაფი -----	128.
24. სურ. 4.4. ცოდნის მოპოვების პროცესის ოთხი ფაზა -----	129.
25. სურ. 4.5. KDD პროცესი -----	130.
26. სურ. 4.6. jigsaw puzzles -----	131.
27. სურ. 4.7. აქსონურ-დენდრიტული მოდელები -----	131.
28. სურ. 4.8. დოკინგის პროცესის სიმულაცია -----	133.
29. სურ. 4.9. ცოდნის გრაფი ერთი აბზაცისთვის -----	136.
30. სურ. 4.10. „საკეტი და გასაღების“ მოდელი -----	138.
31. სურ. 4.11. „ინდუცირებული მორგება“ -----	139.
32. ნახაზი. 4.6.1. საცნობარო არქიტექტურის მოდულებად დაშლა -----	172.
33. ნახაზი 4.6.2. GA-ში ორობითი კოდირება -----	173.
34. ნახაზი 4.6.3. GA-ში ალფაბეტური და რიცხვითი კოდირება -----	173.
35. ნახაზი 4.6.4. GA-ში ხისებრი სტრუქტურის კოდირება -----	174.
36. ნახაზი 4.6.5. გენეტიკური ალგორითმის ზოგადი სქემა -----	175.
37. ნახაზი 4.6.6. გენეტიკური ალგორითმის მოქმედების ძირითადი ციკლი -----	177.
38. ნახაზი 4.6.6. გენეტიკური ალგორითმის მოქმედების ძირითადი ციკლი -----	178.
39. ნახაზი 4.6.8. გენეტიკური ოპერატორები -----	179.
40. ნახაზი 4.6.9. Ga-ში ქრომოსომების შეჯვარება -----	180.
41. ნახაზი 4.6.10. Ga-ში ქრომოსომების მუტაცია -----	180.
42. ნახაზი 4.6.11. ცოდნის ფრაგმენტების მწვერვალების დაკავშირება -----	182.
43. ნახაზი 4.6.12. ცოდნის აგების ალგორითმის ბლოკები -----	183.

აბრევიატურები

1. DM - მონაცემთა მოპოვება;
2. BI - ბიზნეს ანალიტიკა;
3. GIGO - ნაგვის შეტანა, ნაგვის გატანა;
4. TP - ჭეშმარიტად დადებითი;
5. TN - ჭეშმარიტად უარყოფითი;
6. FP - ცრუ დადებითი;
7. FN - ცრუ უარყოფითი;
8. ANN - ხელოვნური ნეირონული ქსელი;
9. GNN - გრაფული ნეირონული ქსელი;
10. GAT - გრაფული ყურადღების ქსელი;
11. GA - გენეტიკური ალგორითმები;
12. GP - გენეტიკური პროგრამირება;
13. OWL - ვებ ონტოლოგიის ენა;
14. RDF - რესურსების აღწერის ჩარჩო;
15. VSA - ვექტორული სიმბოლური არქიტექტურა;
16. HRR - ჰოლოგრაფიული შემცირებული წარმოდგენა;
17. MAP - გამრავლება-შეკრება-გადანაცვლება.

Abbreviations

1. DM - Data Mining;
2. BI - Business Intelligence;
3. GIGO - Garbage In, Garbage Out;
4. TP - True Positive;
5. TN - True Negative;
6. FP - False Positive;
7. FN - False Negative;
8. ANN - Artificial Neural Network;
9. GNN - Graph Neural Network;
10. GAT - Graph Attention Network;
11. GA - Genetic Algorithms;
12. GP - Genetic Programming;
13. OWL - Web Ontology Language;
14. RDF - Resource Description Framework;
15. VSA - Vector Symbolic Architecture;
16. HRR - Holographic Reduced Representation;
17. MAP - Multiply-Add-Permute.