eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Gradu Amaierako Lana

Informatika Ingeniaritzako Gradua

Konputazioa

Atakeen sailkapena konputagailuen arteko komunikazioetan ikasketa automatikoko teknikak erabiliz

Maider Amuchastegui Alberdi

Zuzendariak
Itziar Irigoyen Garbizu
Basilio Sierra Araujo

2023.eko ekainaren 24

Esker onak

Sei hilabete luzez proiektu honetan lanean ibili ondoren, eta asko ikasi ostean, bai profesionalki eta baita pertsonalki ere, eskerrak eman nahi dizkiet nire gradu amaierako laneko zuzendariei, Itziar eta Basiri, momentu oro eskaini didazuen laguntzagatik eta eman dizkidazuen aholkuengatik. Betiere, nirekin hain ulerberak izatearren.

Gainera, aipamen berezi bat egitea gustatuko litzaidake, Lander Segurolak eskainitako laguntza azpimarratuz, zure laguntza ezinbestekoa izan baita lan hau aurrera eramateko. Zure ezagutzak helaraziz, proiektuaren garapena errazago egitea lortu da.

Laburpena

Zibersegurtasunaren arloa informatikaren munduan dagoen kezka handienetako bat da. Enpresa zein erakunde desberdinetako datuak babestuta mantentzea erronka handia da gaur egun. Beraz, egun, konputagailu sistemetan erasoak saihesteko, edo gutxienez, hauei ahalik eta era eraginkorrean aurre egiteko baliabideak izateak garrantzi handia du.

Proiektu honen helburua, BETH datu-multzoa erabiliz, zibersegurtasuneko datuetan erasoak detektatzea da. Zehazki, ikaskuntza automatikoko teknikak erabiliz, konputagailuen arteko komunikazioak atakeak diren edo ez sailkatu nahi da.

Horretarako, batetik, sailkapen gainbegiraturako teknika klasikoak (k-NN eta Naïve Bayes) erabili dira, eta bestetik, sare egitura kontuan hartzen duen Graph Convolutional Network (GCN) erabili da.

Gaien aurkibidea

Irudien aurkibidea

Taulen aurkibidea

KAPITULUA 1

Sarrera

Atal honetan, burututako proiektuaren sarrera dago, hau egiteko motibazioa, helburuak eta memoriaren antolamendua azalduz. Gainera, proiektuan zehar egindakoaren laburpena ere azaltzen da.

1.1 Motibazioa

Zibersegurtasuna, gaur egun, kezka handia da informatikaren munduan. Enpresa zein instituzio guztiek dute beraien datuak babesteko beharra, hori dela eta, konputagailu sistemetan erasoak saihestea egun dagoen erronka handia da. Izan ere, datu horien filtrazioak enpresa edo erakundeetako bezero zein erabiltzaileen identitatea arriskuan jar dezake, datu horiek iruzurra edo bestelako delituak gauzatzeko erabiliz [1].

Erasotzaileek, atakeen bitartez, beste zenbait kalte ere eragin ditzakete, besteak beste, software gaiztoak erabiliz, ekipamendu informatikoa kaltetu dezakete edo hauetara sarrera blokea dezakete.

Azken urteotan, eta denbora tarte laburrean, zibererasoen kopurua nabarmen hazi da. Horren adibide dira, “Basque CyberSecurity Centre”-ek 2022ko bigarren hiruhilekoan jasotako datuak [2]. Denboraldi horretan, 6.653 zaurgarritasun berri argitaratu ziren, 2022ko lehen hiruhilekoarekin alderatuta %15 gehiago, 5.800 izan baitziren. Aldi berean, zaurgarritasun kritikoen kopurua %157 igo zen, 290 izatetik, 745 izatera pasatuz.

Guzti hori kontutan hartuz, eta gaur egun dauden zibererasoen hazkuntza ikusita, konputagailuen arteko komunikazioetan erasoak detektatzearen garrantzia ikusi da. Beraz, hau izan da proiektua garatzeko arrazoi nagusietako bat.

Hala ere, lan hau egiteko erabili diren datuen berezitasunak aparteko motibazioa gehitzen dio proiektu honi. Zeren eta erabilitako datuak konputagailuetako prozesuen arteko komunikazioetan oinarritzen dira, eta hauek irudikatzeko, datu-egitura bereziak erabili behar dira. Ondorioz, erabili beharreko neurona-sareen arkitekturak ere ohikoak baino bereziagoak izango dira.

Motibazio pertsonalari dagokionez, zeregin honek duen zailtasunak eragina du honen erakargarritasunean. Orain arte erabili ez ditudan datu-motak ulertzeak eta hauek prozesatzeak duen zailtasunak gauza berriak ikastera eraman zaitzake, eta baita bakoitzaren gaitasunak neurtzera ere.

1.2 Helburuak

Proiektu honen helburua BETH datu-multzoa erabiliz [3], zibersegurtasuneko datuetan erasoak detektatzea da, eta hau lortzeko neurona-sare klase egoki bat erabiltzea, sortutako eredua datu-multzo honetarako ahalik eta eraginkorrena izan dadin.

Bide batez, datu-multzo konplexuekin lan egiten ikastea eta tresna desberdinak ezagutzea eta erabiltzen ikastea lortu nahi da.

Honakoak dira, besteak beste, egindako lanaren helburu zehatzak:

- Sareko aktibitateko ezaugarriak ulertzea zibersekuritatearen ikuspuntutik.
  - BETH, zibersegurtasunarekin lotutako datu-multzoa ulertzea.
- Erasoak identifikatzeko ikasketa automatikoko prozedurak proposatzea, bi ikuspegi eskainiz:
  - Erasoen arteko grafo egitura kontuan hartzen ez dutenak.
  - Erasoen arteko grafo egitura kontuan hartzen dutenak.

  Eta bi ikuspegitatik lortutako emaitzak konparatzea.

Gainera, gradu amaierako lan baten helburu orokorrak lortu nahi dira:

- Proiektuaren eskakizunak edo helburuak betetzea, eta garapen prozesuan sortutako zailtasunak gainditzea.
- Egin beharreko lana planifikatzea eta kudeatzea.
- Memoria bat idaztea, proiektua garatzeko egindako lana zehaztasunez azalduz.
- Egindako lana aurkezpenerako erabiliko den dokumentu baten bitartez jendaurrean azaltzea eta defendatzea.

1.3 Proiektuan egindakoaren laburpena

Proiektu honetan, BETH datu-multzoa erabili da konputagailuen arteko komunikazioak erasoak diren edo ez sailkatzeko.

Lan hau aurrera eramateko, lehendabizi, proiektua planifikatu behar izan da, honen kudeaketa zehatz-mehatz definitzeko.

Ondoren, hainbat kontzeptu teoriko barneratu behar izan dira, azterlan honetan zehar egin den esperimentazioa burutu ahal izateko.

Egindako probei dagokienez, hauek garatzeko, BETH datu-multzoak eskaintzen dituen datuak aurreprozesatu eta estandarizatu dira. Ostean, datu horiek egitura desberdinak erabiliz definitu dira eta sailkapenak egin dira.

Datuen egiturari dagokionez, bi multzotan banatu ditzakegu: erasoen arteko grafo egitura kontuan hartzen ez dutenak eta erasoen arteko grafo egitura kontuan hartzen dutenak.

Proiektuan zehar, hainbat proba desberdin gauzatu dira, hala nola, sailkatzaile mota desberdinak erabili dira eta sailkapenetan erabilitako datuen antolamenduarekin eta datu-kopuruarekin ere jokatu da. Zehazki, sailkapenak egiteko, k-NN eta Naïve Bayes bezalako sailkapen gainbegiraturako teknikak eta grafo egitura kontuan hartzen duen Graph Convolutional Network (GCN) arkitektura erabili dira.

Proba hauek egitearen helburua, ahalik eta emaitza onenak lortzea izan da.

1.4 Memoriaren antolamendua

Memoria honi hasiera ematen zaio proiektuaren inguruko sarrera bat eginez. Lehenengo atalean, lan hau egiteko motibazioa eta honen helburuak azaltzen dira. Gainera, proiektuan zehar egindakoa era laburrean azaltzen da.

Bigarren atalean, proiektu hau burutzeko jarraitutako plangintza azaltzen da. Bertan, bete beharreko atazak eta hauen garapen denborak, sortu beharreko emangarriak, erabilitako informazio-sistemak eta komunikatzeko erabilitako baliabideak azaltzen dira.

Hirugarren atalean, erabilitako informatikako tresnak adierazten dira. Tresna bakoitza zertarako erabili daitekeen espezifikatuz.

Laugarren atalean, proiektua garatzeko ulertu behar izan diren kontzeptu teorikoak jaso dira.

Bosgarren atalean, egindako proba zein esperimentuak eta lortutako emaitzak laburbiltzen dira, hauek egiteko jarraitutako planteamenduak adieraziz.

Amaitzeko, lan honetan lortutako emaitzen gainean ateratako ondorioak azaltzen dira seigarren atalean, eta etorkizunean egin daitezkeen lanak edo hobekuntzak ere adierazten dira. Gainera, ondorio pertsonalak ere aurkezten dira.

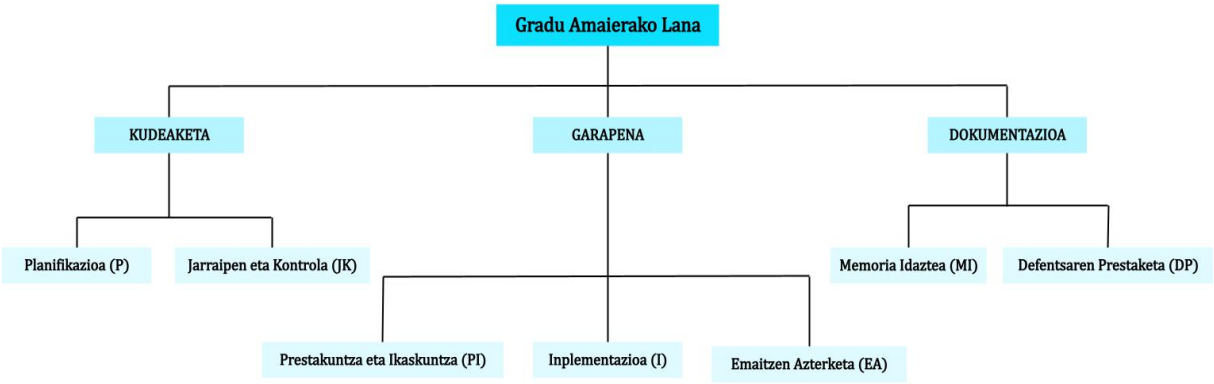
KAPITULUA 2

Lanaren plangintza

Atal honetan, proiektuaren gaineko kudeaketa eta jarraitu diren pausuak azaltzen dira.

2.1 LDE diagrama eta atazen deskonposaketa

Proiektuaren ataza desberdinak definitzeko, 2.1 Irudian azaltzen den Lanen Deskonposaketa Egitura (LDE) diagrama erabili da. Diagrama honen bidez, proiektuan zehar egindako lana deskonposatu da.



2.1 Irudia: Proiektuaren LDE diagrama.

LDE diagraman definitutako lan-paketeak deskribatuta daude jarraian:

- **Kudeaketa:**
  - **Planifikazioa (P):** Lan-pakete honetan, proiektuaren planifikazioarekin zer ikusia duen oro sartzen da. Planifikazioa burutzeko, proiektuaren helburuak kontuan hartu behar dira eta proiektuaren eskakizunak identifikatu. Horrela, egin beharreko lana antolatuko da.

  - **Jarraipen eta Kontrola (JK):** Lan-pakete honek proiektuaren garapen egokia bermatuko duten atazak edukiko ditu, eta konkretuki, Gradu Amaierako Lanaren epeak eta espezifikazioak betetzen direla bermatuko ditu.

- **Garapena:**
  - **Prestakuntza eta Ikaskuntza (PI):** Lan-pakete honek, proiektua garatzeko barneratu beharko diren kontzeptuak eta erabili beharko diren baliabideak ezagutzeko eta ikasteko jarraitutako urratsak barne hartzen ditu. Gainera, proiektua garatzen hasteko beharrezko tresna zein baliabide guztiak prest izateko egindakoak ere barne hartzen ditu.
  - **Inplementazioa (I):** Inplementazioak barne hartzen ditu proiektuan burututako proba zein esperimentu desberdinak kodetzeko jarraitu diren urratsak.
  - **Emaitzen Azterketa (EA):** Atal honetan, inplementazioan egindako probetan lortutako emaitzak aztertzen dira, jarraian ondorioak lortzeko.

- **Dokumentazioa:**
  - **Memoria Idaztea (MI):** Lan-pakete honen baitan daude memoria idazteko egindakoak. Memorian, proiektua garatzeko egindako guztia zehaztasunez azalduko da.
  - **Defentsaren Prestaketa (DP):** Gradu Amaierako Lana azaltzeko erabiliko den aurkezpen-dokumentua eta gidoia prestatuko dira, proiektuan zehar egindako guztiaren azalpen labur bat.

2.2 Emangarriak eta hauen amaiera-datak

Proiektu honen amaieran, honako emangarri hauek egon beharko dira sortuta:

- **Kodea:** Proiektuan zehar burututako esperimentuen kodeen fitxategiak. Hauek eskuragarri jarri dira GitHub plataforman.
- **Memoria:** Proiektuan zehar egindako guztia zehatz-mehatz azaltzen duen dokumentua.
- **Aurkezpen-dokumentua:** GrAL guztian zehar egindakoa modu argi eta zehatzean laburbiltzen duen dokumentua.

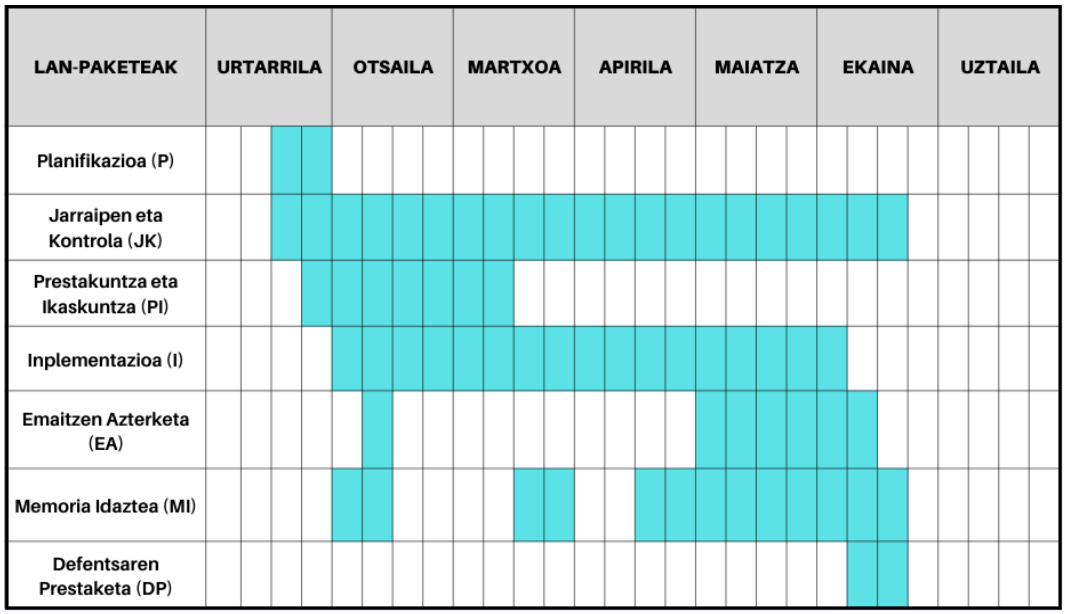
2.1 Taulan adierazten dira emangarri bakoitzaren amaiera-datak:

EMANGARRIA	ENTREGATZE DATA
Kodea	2023ko ekainak 25
Memoria	2023ko ekainak 25
Aurkezpen-dokumentua	2023ko uztailaren 3tik 14ra

2.1 Taula: Emangarri bakoitzaren entregatze datak.

2.3 Atazen garapen denbora tarteak

Proiektuan zehar egindako atazen garapen denbora tarteak adierazteko, 2.2 Irudian azaltzen den Gantt diagrama erabili da.



2.2 Irudia: Gantt diagrama.

Diagraman, Emaitzen Azterketa (EA) atalak itxura berezia duela esan daiteke. Kontua da, hasieran egindako esperimentuak burutzean prozesua nahiko azkarra izan zela gainerako esperimentuekin alderatuz. Ondorioz, lehenengo esperimentuetako emaitzak prozesu hori amaitzean aztertu ziren, eta hauek aztertu ondoren, Graph Convolutional Network eredua sortu zen. GCN eredua sortzeko denbora tarte handiagoa behar izan zen, hainbat proba egin baitziren. Horregatik, Emaitzen Azterketa atala ez da burutzen otsaila erditik maiatza hasiera arte.

Memoria Idaztea (MI) atala ere, tarteka burutu da, izan ere, proiektu honetan egin diren proba desberdinen kopurua kontuan izanda, beharrezkoa zen hau pixkanaka dokumentatzen joatea. Beraz, memoria otsaila hasieran sortu zen, gehien bat prozesua era informalean dokumentatzen joateko, eta esperimentu desberdinak egin ahala memoria era egokiagoan eratzen joan zen.

2.4 Ataza bakoitzari eman zaion denbora eta desbiderapenak

2.2 Taulan ataza bakoitzari emandako dedikazioen banaketa eta errealitatean emandako dedikazioak adierazten dira.

ATAZAK	ESTIMATUTAKO DENBORA (h)	IGAROTAKO DENBORA (h)
Kudeaketa	30	25
P	10	10
JK	20	15
Garapena	160	175
PI	50	45
I	100	120
EA	10	10
Dokumentazioa	110	120
MI	90	100
DP	20	20
GUZTIRA	300	320

2.2 Taula: Ataza bakoitzari eskainitako denbora.

2.4.1 Plangintzarekiko desbiderapenak

2.2 Taulari erreparatuz, desbiderapenik esanguratsuena Inplementazioa (I) atazako desbiderapena izan da. 100 orduko lana estimatu zen, eta azkenean, 20 ordu gehiago behar izan ziren proiektuko inplementazioa amaitzeko. Honen arrazoi nagusietako bat, proiektuan zehar egin behar izan diren proba desberdinak dira. Proba bakoitza egin ahal izateko inplementazio desberdin bat, edo gutxienez, hasierako inplementazioaren aldaera bat behar zen.

2.5 Informazio-sistema

Proiektuan zehar erabilitako edo sortutako materiala honako biltegietan gorde da:

- **Ordenagailua:** Sortutako dokumentu guztiak ordenagailuan, lokalean, gorde dira.
- **Hodeiko biltegiratzea:**
  - **Google Drive:** Kodea hodeian sortu da, zehazki, Google-ek eskaintzen duen "Google Colab" tresna erabiliz. Eta ondorioz, bertan gorde dira inplementatutako fitxategiak. Gainera, sortutako irudiak ere bertan biltegiratu dira.
  - **Overleaf:** Memoria plataforma honetan idatzi da, eta beraz, bertan gorde da.
- **Kanpo-memoria:** Segurtasun kopiak gordetzeko erabili da, bai kodea duten fitxategiena eta baita memoriaren kopiak.

2.6 Komunikazioak

Proiektuan zehar GrAL-eko zuzendariekin komunikazioa kanal hauen bitartez egin da:

- **Bilerak:**
  - **Presentzialki:** Zuzendariekin bilerak orokorrean presentzialki egin dira, proiektuan unerarte egindakoa erakusteko eta jarraitu beharreko hurrengo urratsak adosteko.
  - **Bideodei bitartez:** Bilerak presentzialki egin ezin izan diren kasuetan, hauek bideodei bidez egin dira, Webex plataformaren bitartez.
- **Mezu elektronikoen bitartez:** Kontsulta puntualak egiteko eta bileretako hitzorduak adosteko erabili da.

2.7 Arriskuen plangintza

Proiektu baten garapenean arriskuak sor daitezke. Batez ere, Gradu Amaierako Lana bezalako luzeradun proiektuetan hainbat faktorek izan dezakete eragina. Horregatik, oso garrantzitsua da arrisku hauek aurrerapenarekin antzematea eta hauei soluzioak bilatzea.

Honakoak dira proiektu honetarako estimatu ziren arriskuak eta hauen soluzioak. Hala ere, ez dira estimatu ziren arrisku guztiak gertatu.

- **Kontzeptu teorikoen ulermena (R1):** Proiektu honetan, orain arte erabili gabeko tresnak erabiltzen ikasi eta kontzeptu berriak barneratu behar izan dira. Kasu hauetan zailtasunak izanez gero, bilera gehiago egin biharko lirateke eta Prestakuntza eta Ikaskuntza (PI) ataza burutzen ordu gehiago igaro beharko litzateke. Hala ere, proiektuan zehar ez da arazo hau azaldu.
- **Erabilitako ordenagailuaren ahalmen falta (R2):** Erabilitako ordenagailuak baliteke datuak prozesatzeko ahalmen nahikoa ez izatea. Beraz, beste makina bat erabili ezinean, prozesatutako datuen kopurua mugatu beharko da. Kasu honetan, arazo hau bai gertatu da proiektuaren garapenean, “Google Colab” tresnak oinarrizko bertsioan memoria mugatua eskaintzen duelako.
- **Beharrezko fitxategi zein dokumentuak ezabatzea eta galtzea (R3):** Arrisku hau saihesteko, erabilitako material denaren segurtasun kopiak gorde beharko lirateke. Eta proiektuan zehar segurtasun kopiak egin direnez, ez da horrelako arazorik egon.

KAPITULUA 3

Erabilitako informatikako tresnak

Proiektu hau aurrera eraman ahal izateko, hainbat baliabide desberdin erabili dira, besteak beste, proiektua kodetzeko softwarea, grafoak irudikatzeko zenbait tresna, neurona-sarea sortzeko eta entrenatzeko liburutegiak, eta ikaskuntza automatikoko ereduaren errendimendua neurtzeko tresnak.

3.1 Softwarea: Python

Proiektu hau *Python* programazio-lengoaia erabiliz kodetu da. Programazio-lengoaia hau asko erabiltzen da web-aplikazioetan [4], softwarearen garapenean, datuen zientzian eta ikaskuntza automatikoan. Garatzaileek Python erabiltzen dute eraginkorra eta ikasteko erraza delako, eta plataforma askotan erabil daitekeelako. Python softwarea doan deskarga daiteke, sistema-mota guztietan ondo integratzen da eta garapenaren abiadura handitzen du.

3.2 Grafoak irudikatzeko tresnak

3.2.1 Gephi

Gephi grafoak aztertu eta ulertzeko tresna da [5]; izan ere, grafoak irudikatu, manipulatu, koloreztatu eta abar egiteko aukera ematen du, eta datu analistei hipotesiak egiten eta patroiak deskubritzen laguntzea du helburu. Gephi kode irekikoa eta doakoa da [6].

Gephi erabilgarria da datu-base batean patroiak eta joerak bilatzeko [7]. Denbora errealeko grafikoak dituen 3D motor errenderizatua erabiltzen du.

Funtzionalitateak [8]:

- Denbora errealean bistaratzea: grafikoen bistaratze-motorrari esker, tamaina handiko grafikoetan ereduak errazago ulertu eta aurkitu daitezke.

- Antolaketa: diseinu algoritmoek forma ematen diote grafikoari. Gephik azken belaunaldiko algoritmoak diseinatzeko algoritmoak eskaintzen ditu, bai eraginkortasunerako, bai kalitaterako.
- Metrika: estatistika eta metriken aukerek, sare sozialak eta eskalarik gabeko sareak aztertzeko metrika ohikoenak eskaintzen dituzte.
- Sareak denboran zehar: erabiltzaileek, sare batek denboran zehar duen bilakaera ikus dezakete denbora-lerroa manipulatuz.

3.2.2 NetworkX

NetworkX paketea Python paketea da [9], sare konplexuen egitura, dinamika eta funtzioak sortu, manipulatu eta aztertzeko.

NetworkX-k grafo oso handietan funtzionatzeko gaitasuna du [10], zehazki 10 milioi nodo eta 100 milioi ertz baino gehiago dituzten grafoetan. Hainbat gauza irudikatzeko datu-egiturak ditu, hala nola grafo sinpleak, grafo zuzenduak eta ertz paraleloak eta self-loops-ak dituzten grafoak. Pakete nagusia BSD lizentziapeko software librea da. BSD lizentzia [11], *Berkeley Software Distribution* izenez ere ezagutzen dena, kode irekiko lizentzia da, softwarearen erabilera, aldaketa eta banaketa librea ahalbidetzen duena.

NetworkX-en erabilera nagusien artean hauek daude:

- Sare sozial, biologiko eta azpiegituren egitura eta dinamika aztertzea.
- Grafoen programazioaren ingurune normalizatua.
- Lankidetza eta diziplina anitzeko proiektuen garapen azkarra.
- C, C++ eta FORTRANen idatzitako algoritmo eta kodeekin integratzea.
- Estandarra ez den datu-multzo handiekin lan egitea.

3.3 Neurona-sarea eta sailkatzaileak eratzeko eta entrenatzeko erabilitako liburutegiak

3.3.1 Pytorch

PyTorch kode irekiko ikaskuntza automatikoko liburutegia da, Torchen liburutegian oinarritua [12]. Ikuspen artifizialerako eta hizkuntza naturalen prozesamendurako erabiltzen da.

Ikaskuntza automatikoa, irudien ezagutza eta ikaskuntza sakonarekin lotutako aplikazioetarako liburutegia da [13].

3.3.1.1 Pytorch-geometric

PyTorch-geometric (PyG) PyTorchen gainean eraikitako liburutegia da [14], eta egitura irregularrak sakon ikasteko erabiltzen da [15], hala nola grafoak, puntu-hodeiak eta gainazalak. PyTorchen esparru ezagunean oinarritzen da, eta GPUk azeleratutako konputazioaren errendimendu optimizatua ematen du tamaina aldakorreko datu urri eta irregularretan.

PyG liburutegian geometria sakoneko ikaskuntzarako hainbat metodo jasota daude, eta hauek, nodoen sailkapena, grafoen sailkapena edo ertzen iragarpena bezalako zereginak egiteko erabil daitezke.

Esparru intuitibo eta ahaltsua da, eta ikertzaileei grafoetan oinarritutako neurona-sareak (Graph Neural Network) erraz ezartzeko aukera ematen die.

3.3.2 Scikit-Learn

Scikit-Learn Python-en doako liburutegia da [16]. Sailkapen, erregresio, clustering eta dimentsio-murrizketako algoritmoak ditu. Gainera, beste Python liburutegi batzuekiko bateragarritasuna du, hala nola `NumPy`, `SciPy` eta `matplotlib` liburutegiak.

Gainera, zenbait funtzio ditu datuak aurreprozesatzeko [17]:

- Normalizazioa: zenbakizko aldagaiak doitzean datza batez bestekoa 0 eta bariantza 1 izan ditzaten, edo [0,1] bezalako tarte batean egon daitezen. Bektoreak normalizatzea ere ahalbidetzen du, 1eko norma izan dezaten.
- Transformazio ez-linealak: kuantiletan eta berretzaileetan oinarrituak, banaketa oso alboratuak dituzten aldagaiak eraldatzeko, adibidez.
- Diskretizazioa: funtzio jarraituak, ereduak, aldagaiak eta ekuazioak kontrako alderdi diskretuetara transferitzeko prozesua da [18]. Muturreko kasu bat aldagai bat bi balio posibleetara soilik bihurtzen denean gertatzen da, binarizazioa izenez ezagutzen dena [17].
- Balio galduak: zenbait azterketetan aldagairen bateko datuak falta direnean (esaterako, erabiltzaile batek ez dio erantzuten inkesta bateko galderaren bati), balio bat eslei dakioke automatiza daitekeen irizpideren baten arabera, adibidez, medianarekin ordezkatu.
- Aldagaien artean interakzioak sortzea polinomioak erabiliz.

3.4 Ereduaren errendimendua neurtzeko tresnak

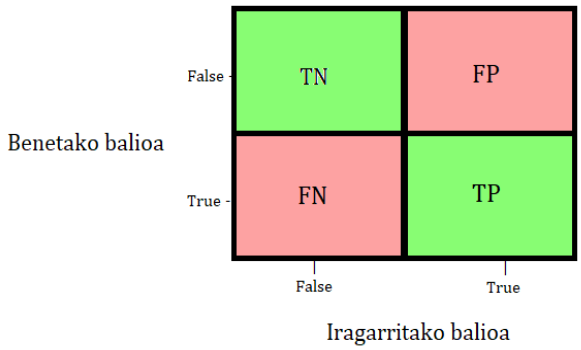
Hasiera batean, asmatze-tasak zehaztasuna erabiliz kalkulatzen ziren, eraikitako modeloaren errendimendua neurtzeko. Baina emaitzak aztertu ahal izateko informazio zehatzagoa beharrezkoa zenez, konfusio-matrizeak erabiltzea erabaki zen. Gainera, doitasuna, estaldura eta F1 metrikak erabili dira lortutako emaitzak aztertzeko.

3.4.1 Konfusio-matrizea (Confusion matrix)

Konfusio-matrizea [19], ikaskuntza automatikoaren sailkapen-problemarako errendimenduaren neurketa da, non produkzioa bi motakoa edo gehiagokoa izan daitekeen. Aurresandako eta benetako balioen 4 konbinazio dituen taula da.

Konfusio-matrizeak etiketa bakarreko zein etiketa anitzeko sailkapenetan erabil daitezke [20]. Etiketaren sailkapenean etiketa bakarra esleitzen da elementu bat sailkatzeko. Banakako etiketaren sailkapen-algoritmoak 2×2 konfusio-matrizean grafikatuko dira. Etiketa anitzeko sailkapenean, etiketa asko eslei daitezke elementu bat sailkatzeko. Sailkapen metodo hori matrize handiago batean grafikatuko da, irudi bakoitzari esleitu beharreko etiketa kopuruaren arabera.

Honako hau etiketa bakarreko sailkapenean lor daitekeen konfusio-matrize baten eredua da, zehazki, proiektu honetan erabili denaren eredua:



3.1 Irudia: Konfusio-matrize baten adibidea.

Sortutako konfusio-matrizeak lau koadrante ditu [21]:

- Benetako negatiboa edo *True Negative* (TN): goiko ezkerreko koadrantea da eta emaitza honek adierazten du ereduak klase negatiboa zuzen iragarri duela [22].
- Positibo faltsua edo *False Positive* (FP): goiko eskuineko koadrantea da eta emaitza honek adierazten du ereduak klase positiboa oker iragarri duela.
- Negatibo faltsua edo *False Negative* (FN): beheko ezkerreko koadrantea da eta emaitza honek adierazten du ereduak klase negatiboa oker iragarri duela.
- Benetako positiboa edo *True Positive* (TP): beheko eskuineko koadrantea da eta emaitza honek adierazten du ereduak klase positiboa behar bezala iragarri duela.

3.4.2 Doitasuna, estaldura eta F1

Doitasuna, estaldura eta F1 sailkapeneko metrikak dira, eta datu-zientzialariek ereduaren errendimendua optimizatzeko erabiltzen dituzte. Zehaztasunak adierazten ez digun ereduaren errendimenduaren neurria ematen digute [23].

- **Doitasuna**: iragarpen positiboen kalitateari buruzko informazioa ematen digun metrika da.

$$Precision = \frac{TP}{TP + FP} \tag{3.1}$$

  Doitasunaren balioa 0 eta 1 artekoa da [24].

- **Estaldura** [23]: ereduak egiazko positiboak zenbateraino identifikatzen dituen adierazten digu.

$$Recall = \frac{TP}{TP + FN} \tag{3.2}$$

- **F1** [23]: F1 metrika doitasuna eta estaldura neurriak balio bakar batean konbinatzeko erabiltzen da [25].

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.3}$$

  Metrika honen abantaila nagusia doitasuna eta estaldura balio bakar batean laburtzen dituela da [26]. F1-ek bere maximoa 1ean lortzen du, eta minimoa 0n, non lehen balioak eredu hutsezin bat adierazten duen, eta 0koak algoritmoak denbora guztian huts egiten duela adierazten duen.

KAPITULUA 4

Oinarri teorikoak

Proiektu hau burutzeko, BETH datu-multzoa erabili da zibersegurtasuneko datuetan erasoak detektatzeko. BETH datu-multzoan azaltzen diren prozesuak k-NN eta Naïve Bayes bezalako sailkatzaileak erabiliz sailkatzeaz gain, datu-multzoa grafo bitartez adierazi nahi izan da, eta horregatik, datu hauek prozesatzeko Graph Convolutional Network-ak (GCN) erabili dira.

Proiektu honen helburua zibersegurtasuneko datuetan erasoak detektatzea denez, eta datuak grafo bitartez adierazi direnez, ebatzi nahi izan den problema nodoen sailkapen-problema edo “The Node Classification Problem” izan da.

Horregatik, atal honetan, BETH datu-multzoa zertan datzan azaltzen da, eta k-NN eta Naïve Bayes sailkatzaileen inguruko eta Graph Convolutional Network-en gaineko azalpen labur bat ematen da. Amaitzeko, Graph Convolutional Network-ekin erlazionatuz "The Node Classification Problem" zertan datzan azaltzen da.

4.1 BETH datu-multzoa

BETH datu-multzoa [3], ziurgabetasuna eta sendotasuna ebaluatzeko zibersegurtasuneko datu-multzo handienetako bat.

Datu-multzo honek, zortzi milioi gertaera baino gehiago biltzen ditu eta 23 ostalari (*host*) desberdinen jarraipena eginez lortu da. Ostalari bakoitzak jarduera onbera du eta, gehienez, eraso bakar bat. Datu-multzo hau *Honeypot*-en jarraipen-sistema berri bat erabiliz bildu da. Honeypot bat sistemen erasoei eta erasotzaileen tresnei buruzko informazioa lortzeko diseinatua izan den aplikazio [27], software edo makina multzoa da. Horren bitartez erasotzailearen metodo, teknika, tresna, etab... buruzko informazioa jaso daiteke hauek ekidin ahal izateko.

BETH da, halaber, prozesu- eta sare-erregistroak dituzten datu-multzo bakarretako bat, eta portaera maltzurraren ikuspegi holistikoa ematen du. Esan bezala, datu-multzoa bi sentsore-erregistrok osatzen dute: kernel mailako prozesu-deiek eta sareko trafikoak.

Kernel mailako prozesu-deien erregistroek, prozesuen erregistroak soilik biltzen dituenez, hau da lan hau burutzeko erabili den informazioa.

Prozesu-dei bakoitzak 14 ezaugarri gordin eta 2 etiketa ditu, azken horiek eskuz ezarrita. Bi etiketa hauetako lehenengoak, "sus" etiketak, gertaera bat susmagarria den adierazten du, eta bigarren etiketak, "evil" etiketak, gertaera batean sisteman berezkoa ez den kanpoko presentzia gaiztoa dagoen adierazten du.

Hona hemen prozesu-dei bakoitzak dituen ezaugarriak:

- **timestamp:** `float` motako ezaugarria, sistema abiarazten den momentutik zenbat segundo igaro diren adierazten du.
- **processid:** `int` motako ezaugarria, erregistro hau sortzen duen prozesua adierazten du.
- **threadid:** `int` motako ezaugarria, erregistro hau sortzen duen haria adierazten du.
- **parentprocessid:** `int` motako ezaugarria, erregistro hau sortzen duen prozesuaren gurasoa adierazten du.
- **userid:** `int` motako ezaugarria, erregistro hau sortzen duen erabiltzailea adierazten du.
- **mountnamespace:** `int (long)` motako ezaugarria, prozesu jakin batek hainbat muntatze puntutara duen sarbidea zehazten du.
- **processname:** `string` motako ezaugarria, exekutatutako kate-komandoa adierazten du.
- **hostname:** `string` motako ezaugarria, zerbitzari ostalariaren izena adierazten du.
- **eventid:** `int` motako ezaugarria, erregistro hau sortzen duen gertaera adierazten du.
- **eventname:** `string` motako ezaugarria, erregistro hau sortzen duen gertaeraren izena adierazten du.
- **argsnum:** `int` motako ezaugarria, argumentuen luzera adierazten du.
- **returnvalue:** `int` motako ezaugarria, erregistro honek itzultzen duen balioa (normalean 0 izaten da).
- **stackaddresses:** `int` zerrenda, prozesurako garrantzitsuak diren memoria-balioak adierazten ditu.
- **args:** prozesuari pasatako argumentuen zerrenda da.
- **sus:** `int` motako ezaugarria, gertaera susmagarria den adierazten duen etiketa (1 susmagarria bada, eta 0 ez bada).
- **evil:** `int` motako ezaugarria, gertaera gaiztoa edo erasoa den adierazten duen etiketa (1 erasoa bada, eta 0 ez bada).

Lan honetan Highnam, Arulkumaran, Hanif eta Jennings-en lanari jarraikiz, datuak prozesatzeko "processId", "parentProcessId", "userId", "eventId", "argsNum", "returnValue"eta "evil" ezaugarri zein etiketak erabili dira.

Ezaugarrien aurreprozesatzeari dagokionez, honela eraldatu dira:

- **processId:** prozesuek 0, 1 eta 2 identifikatzaileak izan ditzakete, eta balio hauek sistema eragileak erabiltzen dituenak dira. Gainerako prozesuei ausazko zenbaki bat esleitzen zaie identifikatzaile gisa. Horregatik, lan honetan, prozesuen identifikatzaileak aldagai bitar batekin ordezkatzen dira, identifikatzailea 0, 1 edo 2 den ala ez adierazten duena. Beraz, identifikatzailea 0, 1 edo 2 denean 1eko batekin ordezkatzen da eta bestela 0ko batekin.
- **parentProcessId:** processId ezaugarria bezala aurreprozesatu da.
- **userId:** Linux sistemetan, sistema eragilearen jarduerek 1000tik beherako zenbaki bat izaten dute identifikatzaile gisa (normalean 0). Erabiltzaile arruntei, saioa hasten duten heinean, 1000tik gorako identifikatzaileak esleitzen zaizkie. Ondorioz, erabiltzaileen identifikatzaileak aldagai bitar batekin ordezkatzen dira, identifikatzailea sistema eragilearen jarduerei dagokion edo erabiltzaile arrunt bati dagokion adierazteko. Identifikatzailea 1000 baino txikiagoa bada, 1eko batez ordezkatuko da, eta identifikatzailea 1000 edo handiagoa bada, 0ko batez ordezkatuko da.
- **returnValue:** dei bat behar bezala amaitu den edo ez zehazteko erabil daiteke. ReturnValue-ri 1 balioa esleituko zaio hasierako balioa positiboa denean, 0 balioa esleituko zaio hasierako balioa 0 denean, eta 2 balioa esleituko zaio hasierako balioa negatiboa denean.

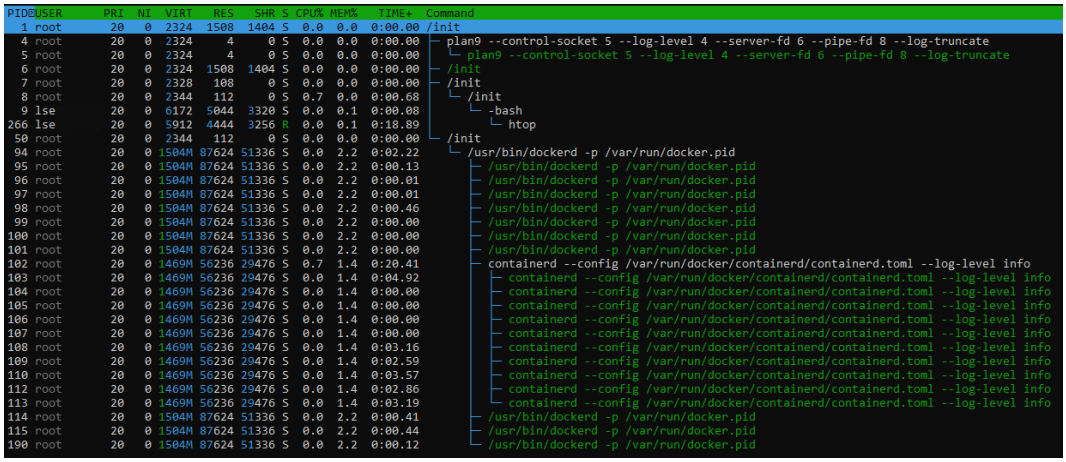
Hasiera batean, prozesu-erregistroak, entrenamendu-, balidazio- eta proba-multzoetan banatuta daude, %60, %20 eta %20 proportzioan gutxi gorabehera, ostalarian sortutako erregistro-kopuruan eta erregistratutako jardueran oinarrituta; proba-multzoan bakarrik sartzen da eraso bat. Entrenamendurako eta balidaziorako multzoetan ez dago erasorik.

Banaketa hori ingurune ez-gainbegiratu baterako da. Baina proiektu hau burutzeko ingurune gainbegiratua erabili denez, datu-multzoaren banaketa desberdin bat erabili da. Izan ere, BETH datu-multzoa, grafo bitartez irudikatu nahi izan da lan honetan. Eta grafoetako nodo guztiei ezaugarriak eta etiketak esleitu zaizkie, modu horretan, ingurune gainbegiratuan lan egiteko.

BETH datu-multzoko prozesu-deien grafo egitura ondorioztatzeko, Linux-en prozesuak nola adierazten diren aztertu da.

Linux sistemetan, prozesu bakoitzari identifikatzaile bakarra (PID) esleitzen zaio. Beste datu batzuk ere izango ditu: prozesu gurasoaren PID-a (prozesua sortu zuen PPID-a), baliabideen kontsumo-denborak, okupatutako memoria-kopurua, prozesua abiarazteko erabili zen komandoa, etab. [28].

4.1 Irudian Linux sistemako prozesuen adibidea azaltzen da. Bertan ikus daiteke `PID` zutabeak prozesuen identifikatzaileak adierazten dituela, eta `Comand` zutabeak prozesu bakoitza zein prozesutik datorren adierazten duela. Kasu honetan, `Comand` zutabean, 266 identifikatzailea duen prozesuari erreparatuz gero, hau 9 identifikatzailea duen prozesutik datorrela ikus daiteke, `bash` prozesutik, eta prozesu hori, 8 identifikatzailea duen `/init` prozesu batetik datorrela. Adibide hau eredu gisa hartuz, BETH datu-multzoan, "processId"-an 266 identifikatzailea izango genuke eta "parentProcessId"-an 9 identifikatzailea izango genuke. Laburbilduz, prozesuen zerrendan, prozesu bakoitzaren aldagaiak azalduko dira, aldagai horietako bat "parentProcessId" izan daitekeelarik.



4.1 Irudia: Linux sistemako prozesuen adibidea.

Beraz, datuen artean prozesu gurasoaren identifikatzailea izanik, prozesu gurasoaren eta prozesuaren arteko erlazioa kontuan har daiteke, prozesu gurasoak eta prozesuak konexio bat izango dutelarik. Horregatik, prozesuen arteko erlazioak grafoko ertzak bezala irudikatu daitezke, grafoko nodoak prozesuak izanik.

4.2 Sailkapen gainbegiraturako teknika klasikoak

4.2.1 k-NN Sailkatzailea

Hurbileneko k bizilagunen algoritmoa [29], KNN edo k-NN izenez ere ezaguna dena, ikaskuntza gainbegiratuko sailkatzaile ez-parametrikoa da, eta hurbiltasuna erabiltzen du banakako datu-puntu baten multzokatzeari buruzko sailkapenak edo iragarpenak egiteko. Erregresio- edo sailkapen-problemetarako erabil badaiteke ere, batez ere sailkapen-algoritmo gisa erabiltzen da.

k-NN algoritmoaren helburua kontsulta-puntu jakin bateko bizilagun hurbilenak identifikatzea da, puntu horri klase-etiketa bat esleitzeko.

Kontsulta-puntu jakin batetik hurbilen zein datu-puntu dauden zehazteko, beharrezkoa izango da kontsulta-puntuaren eta beste datu-puntuen arteko distantzia kalkulatzea.

Distantzia-metrika horiek erabaki-mugak eratzen laguntzen dute, kontsulta-puntuak eskualde desberdinetan banatzen dituztenak.

Distantzia-neurri bat baino gehiago dago, besteak beste:

- **Distantzia euklidearra:** gehien erabiltzen den distantzia da, eta balio errealeko bektoreetara mugatuta dago. Kontsulta-puntuaren eta neurtzen den beste puntuaren arteko lerro zuzen bat neurtzen du.

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2} \tag{4.1}$$

  $\mathbf{x} = (x_1, ..., x_n)$ eta $\mathbf{y} = (y_1, ..., y_n)$ izanik.

- **Manhattan distantzia:** bi punturen arteko distantzia hauen koordenatu kartesiarren diferentzia absolutuen batura da. Hau da, $\mathbf{x}$ eta $\mathbf{y}$ koordenatuen arteko diferentziaren batura totala da [30].

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n}|x_i - y_i|\right) \tag{4.2}$$

- **Minkowski distantzia** [29]: distantzia euklidearraren eta Manhattan distantziaren forma orokorra da. Ondoren adierazten den formulan, p parametroari esker, beste distantzia-metrika batzuk sor daitezke. Formula horren bidez adierazten da distantzia euklidearra p=2 denean, eta Manhattan distantzia p=1 denean.

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n}|x_i - y_i|\right)^{\frac{1}{p}} \tag{4.3}$$

- **Hamming-distantzia:** Teknika hori bektore boolearrekin edo kate-bektoreekin erabili ohi da, bektoreak bat ez datozen puntuak identifikatuz. Horren ondorioz, gainjartze-metrika ere deitzen zaio. Hamming-distantziak datu guztiak aztertzen ditu, eta datu-puntuak antzekoak edo desberdinak diren guneak aurkitzen ditu [31]. Metrika honen emaitzak, zenbat ezaugarri diren desberdinak adierazten du. Demagun luzera bereko bi kate ditugula, "ABCDE" eta "AGDDF", eta haien arteko hamming-distantzia aurkitu nahi dugula.

  ABC**D**E eta **A**GD**D**F

  Kate hauek konparatzean, ikusten da urdinez markatutako bi letrak berdinak direla eta gainerakoak desberdinak direla. Beraz, Hamming-distantzia 3 izango da, hiru letra direlako desberdinak.

k-NN algoritmoko k balioak zehazten du zenbat bizilagun egiaztatuko diren kontsulta-puntu jakin baten sailkapena zehazteko. Adibidez, $k = 1$ bada, instantzia hurbilen duen bizilagunaren klase berari esleituko zaio. $k > 1$ bada, hurbilen dituen k instantzien klaseak begiratuko dira, eta horregatik, k definitzea oreka-ekintza bat izan daiteke, balio desberdinek gehiegi edo gutxiegi doitzea eragin baitezakete. k-ren balio txikienek bariantza handia izan

dezakete, baina alborapen txikia, eta k-ren balio handienek alborapen handia eta bariantza txikiagoa sor dezakete. k aukeratzea, neurri handi batean, sarrerako datuen araberakoa izango da; izan ere, balio atipiko edo zarata gehien duten datuek agian hobeto funtzionatuko dute k-ren balio altuagoekin. Oro har, bi klasetako problementzat k zenbaki bakoitia izatea gomendatzen da, sailkapenean berdinketarik egon ez dadin.

4.2.2 Naïve Bayes Sailkatzailea

Naïve Bayes ereduak gainbegiratutako ikaskuntza-algoritmoen multzo bat dira [32], eta Bayesen teoreman oinarritzen dira.

Eredu hauetan aldagai iragarleak elkarrekiko independenteak direla onartzen da. Bestela esanda, datu-multzo batean ezaugarri jakin bat agertzeak ez du inolako loturarik beste edozein ezaugarriren presentziarekin [33].

Hori lortzeko, A gertaera jakin bat gertatzeko 'a posteriori' edo ondorengo probabilitatea kalkulatzen da, 'a priori' edo aurretiko probabilitate batzuk emanda.

Bayesen teoremak erlazio hau ezartzen du, y klase-aldagaia eta menpeko ezaugarri-bektorea emanda $(x_1, ..., x_n)$ [32]:

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \tag{4.4}$$

Naïves-en baldintzapeko independentzia-hipotesia erabiliz:

$$P(x_i|y, x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i|y), \qquad i = 1, ..., n \tag{4.5}$$

Erlazio hau honela sinplifikatzen da:

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, ..., x_n)} \tag{4.6}$$

$P(x_1, \ldots, x_n)$ sarrera konstantea denez, honako sailkapen-arau hau erabil dezakegu:

$$P(y|x_1, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \tag{4.7}$$

$$\Rightarrow$$

$$\hat{y} = arg\ max_y P(y)\prod_{i=1}^{n} P(x_i|y) \tag{4.8}$$

eta egiantz handieneko estimatzailea erabil dezakegu $P(y)$ eta $P(x_i|y)$ estimatzeko; lehena y klaseak entrenamendu multzoan duen maiztasun erlatiboa da.

Naïve Bayes sailkatzaile desberdinak, nagusiki, ez datoz bat $P(x_i|y)$-en banaketari buruz egiten dituzten suposizioengatik.

4.2.2.1 Gaussian Naïve Bayes

Gaussian Naïve Bayes Naïve Bayesen aldaera bat da, Gaussiar banaketa normala jarraitzen du eta datu jarraituak bermatzen ditu [34].

Datu jarraituekin lan egiten denean, askotan onartzen da klase bakoitzari lotutako balio jarraituak banaketa normal (edo gaussiar) baten arabera banatzen direla. Ezaugarrien probabilitatea hauxe dela suposatzen da:

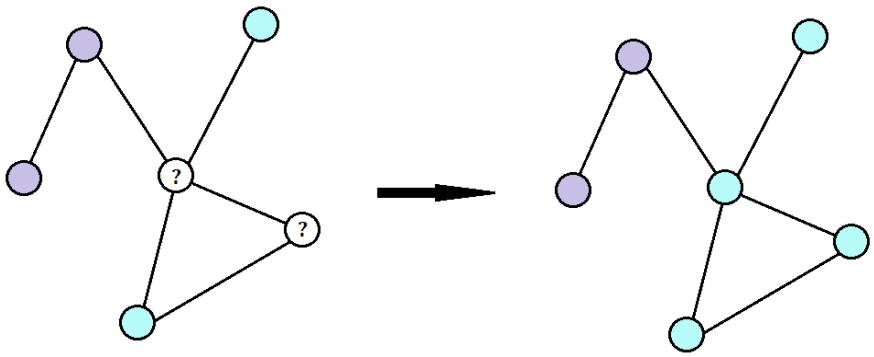
$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma^2}_y} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{4.9}$$

4.3 Grafoan oinarritutako neurona-sare konboluzionala (Graph Convolutional Network)

Graph Convolutional Network (GCN) ereduak neurona-sare arkitektura mota bat dira [35], zehazki, Graph Neural Network-aren (GNN) aldaeretako bat dira [36]. GCN-ek adierazpen-ahalmen handia dute grafoen irudikapenak ikasteko [35].

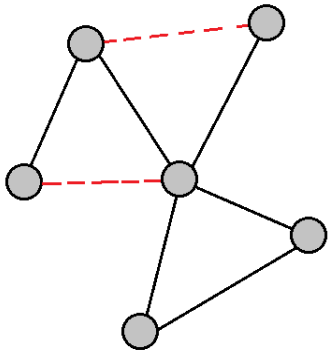
Graph Neural Network eta Graph Convolutional Network ereduak, grafoekin erlazionatutako ikasketa automatikoko hainbat problema desberdin ebazteko erabili daitezke [37]:

- **Nodoen sailkapena:** nodo-motak edo -etiketak iragartzea. Adibidez, zibersegurtasuneko sarean iruzurrezko erakundeak detektatzea nodoen sailkapenarekin lotutako arazo bat izan daiteke.



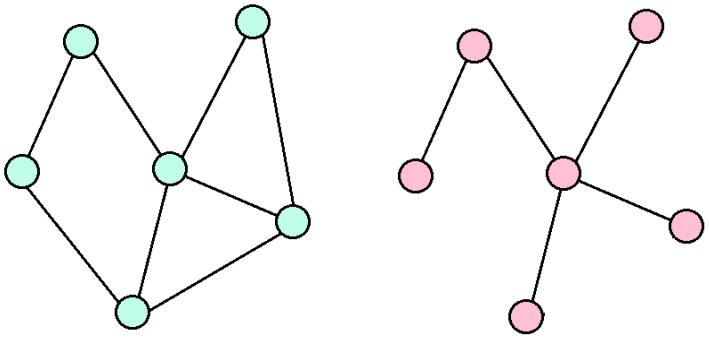
4.2 Irudia: Nodoen sailkapenaren problemaren irudikapena.

- **Ertzak iragartzea:** nodoen artean konexio potentzialak (ertzak) dauden aurresatea. Adibidez, sare sozialen zerbitzu batek sareko datuetan oinarritutako lagun-konexioak iradokitzen ditu.



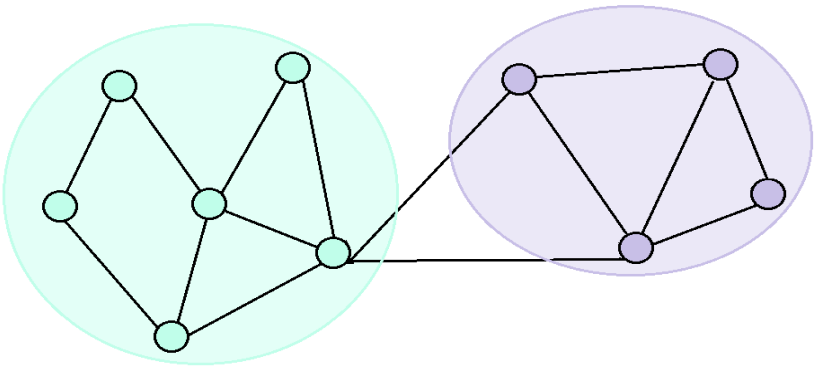
4.3 Irudia: Ertzen iragarpen problemaren irudikapena.

- **Grafoen sailkapena:** grafo bat kategoria desberdinetan sailkatzea. Adibidez, konposatu kimiko bat toxikoa edo ez-toxikoa den zehaztea, haren grafo egiturari begira.



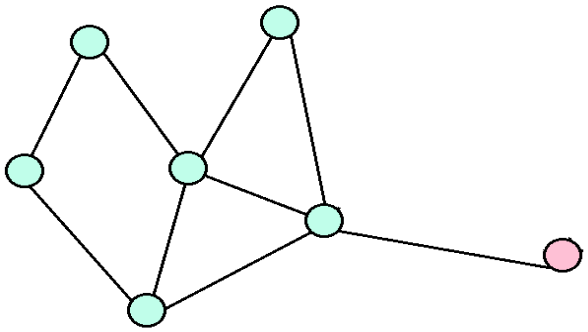
4.4 Irudia: Grafoen sailkapen problemaren irudikapena.

- **Komunitate-detekzioa:** nodoak klusterretan banatzea. Adibide bat gizarte-grafo batean komunitate desberdinak aurkitzea da.



4.5 Irudia: Komunitate-detekzioaren problemaren irudikapena.

- **Anomalien detekzioa:** gainbegiratu gabeko grafo batean nodo anomaloak aurkitzea. Datuak etiketarik gabe daudenean erabili daiteke.



4.6 Irudia: Anomalien detekzioaren problemaren irudikapena.

Gainbegiratzearen ikuspegitik, grafoen ikaskuntza lanak ere hiru entrenamendu-ingurunetan sailka ditzakegu [36]:

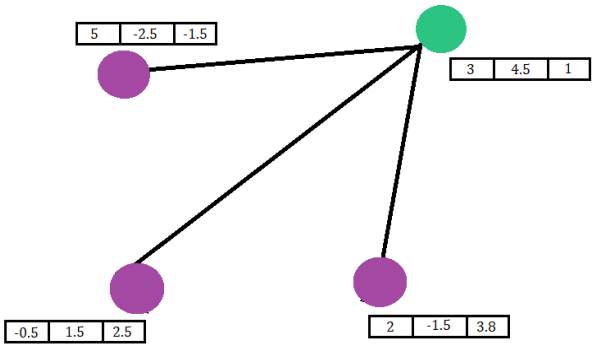
- Gainbegiratutako inguruneak entrenamendurako datu etiketatuak ematen ditu.
- Ingurune erdi-gainbegiratuak etiketa gutxi batzuk eta entrenamendurako etiketatu gabeko datu ugari ematen ditu. Nodoak eta ertzak sailkatzeko ataza gehienak erdi-gainbegiratuak dira.
- Gainbegiratu gabeko inguruneak etiketatu gabeko datuak bakarrik eskaintzen ditu patroiak aurkitzeko. Nodoen clustering-a gainbegiratu gabeko ikaskuntzaren ohiko zeregina da.

4.3.1 Graph Convolutional Network eta nodoen sailkapena

Nodoen sailkapen problema edo "The Node Classification Problem" grafoetan oinarritutako datu-analisian erabiltzen den ikaskuntza automatikoko ataza da [38], non helburua nodoen propietateetan eta haien arteko erlazioetan oinarritutako grafo batean nodoei etiketak esleitzea den.

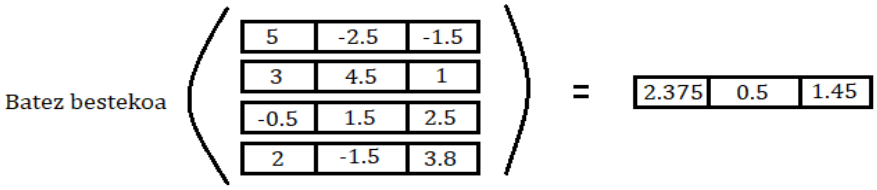
GCN erabiltzean [39], nodo bakoitzerako, nodo horren eta auzokide guztien ezaugarrien informazioa lortzen da. Informazioa lortu ondoren, informazio guztia agregazio funtzio baten bidez bateratzen da, esaterako batezbesteko bidez. Gauza bera egiten da nodo guztiekin. Azkenik, neurona-sare batean elikatzen dira batezbesteko balio horiek, eta lortutako bektorea jatorrizko nodoaren balioak eguneratzeko erabiliko da.

4.7 Irudia adibide gisa hartuz, har dezagun nodo berdea jatorrizko nodotzat. Lehenik, auzokideen ezaugarrien balio guztiak lortzen dira, baita bere buruarenak ere.



4.7 Irudia: Grafo bateko nodoen ezaugarrien bektoreen irudikapena.

Agregazio funtzio gisa batezbestekoa erabiliz, 4.8 Irudian azaltzen den bektorea lortuko da:



4.8 Irudia: Adibideko ezaugarrien batezbestekoaren kalkulua.

Emaitza hau neurona-sare batetik pasatuko da, eta honek, ondoriozko bektore bat itzuliko du. Ondoren, bektore hau erabiliko da jatorrizko nodoaren balioak eguneratzeko.

Azaldu berri den bezala, Graph Convolutional Network erabiltzean jarraitzen diren urratsak honela laburbildu daitezke:

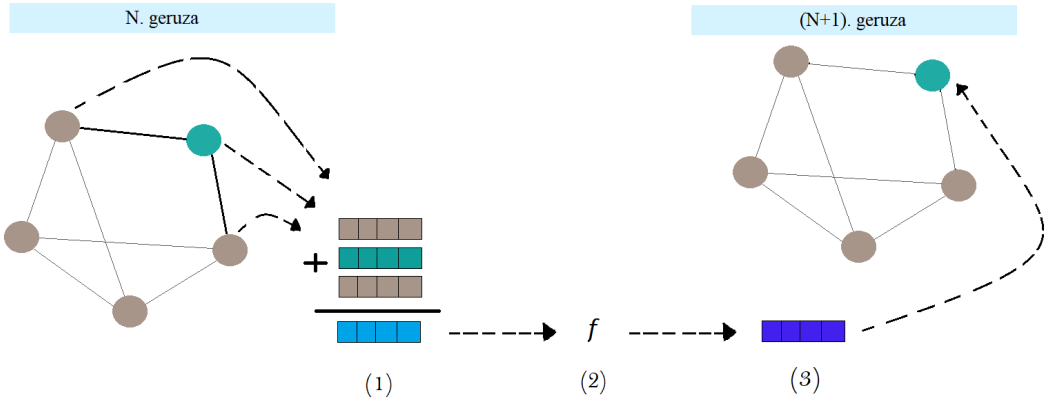
1. Nodo bakoitzerako, nodoaren eta honen auzokide guztien ezaugarrien informazioa lortzen da. Ezaugarri hauek bektoreetan adierazita egongo dira.

2. Behin, ezaugarri guztiak izanda, informazio hau bateratu egiten da, agregazio funtzio baten bidez (batezbestekoa, batuketa, maximoa, minimoa edo besteren bat izan daiteke). Modu horretan, hainbat bektore izan beharrean, bektore bakarra egongo da.

3. Bektore horrekin, jatorrizko nodoaren ezaugarriak eguneratzen dira, normalean, agregatutako ezaugarriak neurona-sare batetik pasatuz. Neurona-sare horrek, bektore eraldatua itzuliko du, ondoren, jatorrizko nodoari esleituko zaiona.

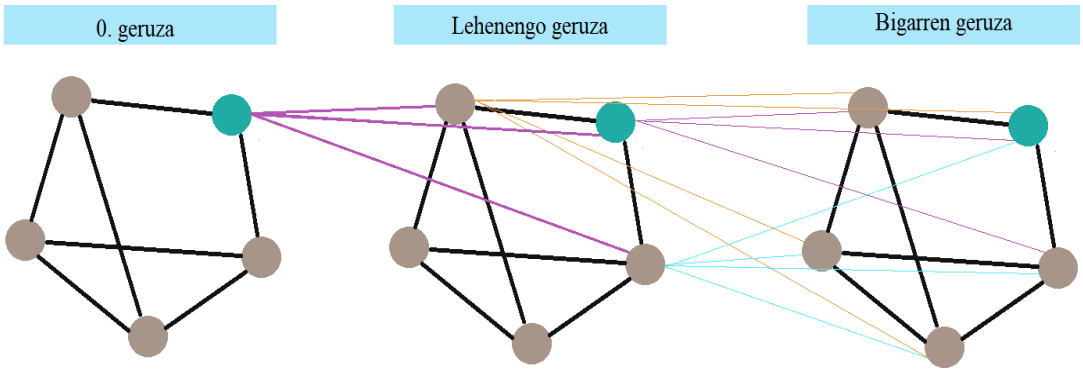
Urrats hauek biltzen dituen prozesua *Message Passing* izenez ezagutzen da. Message Passing-a, neurona-sareko geruza bakoitzean aplikatzen da [40], eta prozesu hori paraleloan errepikatzen da grafoaren nodo guztietarako [41].

4.9 Irudian adierazten da neurona-sare bateko N.geruzatik (N+1).geruzara nola aplikatzen den prozesu hau. Lehenengo urratsean (1) jatorrizko nodoaren eta honen auzokideen ezaugarriak bateratzen dira batuketa erabiliz; bigarren urratsean (2), bateratutako ezaugarriak eguneratze-funtzio batetik pasatzen dira, oro har, neurona-sare bat izaten dena; eta azkenik (3), neurona-sareak itzultzen dituen balioekin jatorrizko nodoaren ezaugarriak eguneratzen dira.



4.9 Irudia: GCN erabiltzen duen neurona-sare batean geruzen artean aplikatzen diren urratsak.

Esan bezala, prozesu hau neurona-sareko geruza guztietan gertatzen da. Hala ere, neurona-sareko geruzetan aurrera joan ahala, nodo baten ezaugarriak eguneratzeko kontuan hartuko diren nodoen kopurua handiagotzen joaten da [39]. GCN erabiltzen duen neurona-sare baten geruza kopuruak, nodoaren ezaugarriek bidaiatu dezaketen distantziarik urrunena adierazten du. Esaterako, GCN geruza batekin, nodo batek auzokideen informazioa bakarrik lor dezake; GCN bi geruzarekin, nodo batek auzokideen eta auzokide hauen auzokideen informazioa lor dezake. Honen adibide gisa, 4.10 Irudian adierazten da neurona-sare bateko GCN geruzen bilakaera. Lehenengo geruzaren irteera bigarren geruzaren sarrera da.



4.10 Irudia: Neurona-sare bateko GCN geruzen bilakaera, adibide gisa nodo bakar bat hartuta.

Hala ere [40], grafo bateko nodoen sailkapen bitarra egin nahi bada, hau da, bi klase desberdinetan sailkatu nahi badira, ez da nahikoa GCN geruzak erabiltzearekin. Behin grafoak nodoei buruzko informazioa duenean, neurona-sarearen azkeneko geruzan, nodo bakoitzerako, sailkatzaile lineal bat aplikatu beharko da. Geruza lineal horren bidez, nodo bakoitzak klase bakoitzean egoteko duen probabilitatea lortuko da. Eta balio horiei esker, nodoak klase batean zein bestean sailkatu ahal izango dira.

Aurretik aipatu den bezala, GCN-ek adierazpen-ahalmen handia dute grafoen irudikapenak ikasteko eta errendimendu handia lortu dute zeregin eta aplikazio ugaritan [35].

Esaterako, GCN-ak minbiziaren farmakoaren erantzuna aurresateko erabili dira [42]. "DeepCDR: a hybrid graph convolutional network for predicting cancer drug response" azterlanean, minbiziaren aurkako medikamentuen erantzun zehatza aurresateko, GCNak oso lagungarriak izan ziren sendagaien informazio estrukturala antzematen.

KAPITULUA 5

Egindako esperimentazioa eta emaitzak

Atal honetan, proiektuan zehar egindako esperimentazioa eta lortutako emaitzak azaltzen dira. Lehendabizi, sailkapen gainbegiraturako teknika klasikoekin egindako probak azaltzen dira, zehazki, k-NN eta Naïve Bayes sailkatzaileekin; eta ondoren, grafo egitura erabiliz lortutako emaitzak azaltzen dira.

5.1 Datuen antolamendua eta aurreprozesaketa

Proiektuan zehar prozesuak sailkatu nahi izan dira, zehazki, prozesu hauek erasotzaileak diren edo ez adieraztea lortu nahi izan da.

Horregatik, sailkapenak egiteko erabili ziren ezaugarriak ez ziren BETH datu-multzoak eskaintzen dituen datuak bere horretan. Guztira, 19032 ezaugarri zerrenda erabili ziren. Ezaugarri zerrenda bakoitzak, BETH datu-multzoan dagoen prozesu bat irudikatzen zuen. Horregatik, prozesu bat gurasoa zen kasuetan, datu-multzoan ez zuen ezaugarririk izango, eta honen ezaugarriei "-1000" balioa esleitu zitzaien. Balio hau erabili zen, prozesuen ohiko ezaugarriek ez dutelako "-1000" balioa izango; beste edozein zenbaki handi eta negatibo erabili zitekeen.

Gainera, ezaugarriak aurreprozesatuta zeuden, eta ezaugarri zerrenda horietako balioen diferentziak kontuan izanda, datu hauek estandarizatzeko edo normalizatzeko beharra zegoen. Kasu hauetarako, `sklearn` liburutegiko `MinMaxScaler` funtzioa erabili zen [43].

Funtzio horrek, ezaugarriak eraldatzen ditu, ezaugarri bakoitza tarte jakin batera aldatuz, kasu honetan [0-1] tartera. Aipatutako funtzioak ezaugarri bakoitza bakarka ebaluatu eta itzultzen du, eta, beraz, adierazitako tartean egongo dira, adibidez, zero eta bat artean.

5.1.1 Datuen antolamendua eta aurreprozesaketa probatzeko sailkatzaileak

Datuen antolamendua eta aurreprozesaketa egokiak ziren probatzeko erabili ziren sailkatzaileak k-NN (kasu honetan, 1-NN, 3-NN eta 5-NN) eta Naïve Bayes izan ziren.

k-NN sailkatzailea distantzia euklidearra erabiliz kalkulatu zen. Eta Naïve Bayes sailkatzailearen kasuan, Naïve Bayes Gaussiarra (GNB) erabili zen.

Proba kasuetan lortutako emaitzak dira 5.1 Taulan ageri direnak. Aipatu bezala, sailkatzaileen errendimendua neurtzeko konfusio-matrizeak, eta doitasuna, estaldura eta F1 metrikak erabili ziren. Orokorrean, emaitzak onak dira.

Aurretik aipatu den bezala, ezaugarriak aurreprozesatuta zeuden, eta ezaugarri-zerrendetan antolatuta. Train eta test multzoak eratzeko ezaugarri-zerrenda hauen ordena kontuan hartu zen arren, prozesu erasotzaileen eta ez-erasotzaileen kopurua train zein test multzoetan orekatua izateko, orden hau pixka bat moldatu zen.

Guztira, 19032 ezaugarri-zerrenda izanik, train-erako 4514.ezaugarri-zerrendatik 9026.zerrendara eta 12126.zerrendatik 19032.era erabili ziren. Test-erako, tartean erabili gabe geratu ziren ezaugarri-zerrendak erabili ziren (lehenengo 4513 ezaugarri-zerrendak eta 9026.etik 12126.era).

Laburbilduz, entrenamendurako 11419 ezaugarri-zerrenda erabili ziren, eta hauetatik 6906 ziren erasotzaileak. Eta testerako 7613 ezaugarri-zerrenda erabili ziren, hauetatik 3099 izanik erasotzaileak.

METODOA	**DOITASUNA**		**ESTALDURA**		**F1**	
	0 klasea	**1 klasea**	**0 klasea**	**1 klasea**	**0 klasea**	**1 klasea**
1-NN	1	0.81552632	0.84470536	1	0.91581602	0.89839107
3-NN	1	0.82114467	0.85046522	1	0.91919071	0.90178961
5-NN	1	0.82027528	0.84957909	1	0.91867289	0.90126509
GNB	1	0.84464432	0.87372619	1	0.93260818	0.91578014

5.1 Taula: k-NN eta Naïve Bayes sailkatzaileekin lortutako emaitzak.

5.1 Taulan adierazten diren emaitzei esker ikus dezakegu planteatu den ezaugarrien aldaera erabilgarria dela. Emaitza onenak ematen dituen sailkatzailea Gaussian Naïve Bayes den arren, lau sailkatzaileek nahiko emaitza onak ematen dituzte proba kasuetan. Hau islatuta ikusten da doitasuna eta estaldura balioetan, non hauek 0.8-tik gorakoak diren. Gainera, F1 balioei erreparatuz, ia guztiek 0.9-tik gorako balioa dute. Izan ere, 1 klaseari dagokion estaldura balioek adierazten dute, eredu guztiek egiazko positiboak ehuneko ehunean identifikatzen dituztela. Beraz, sailkatzaile guztiek, prozesu erasotzaile guztiak ondo sailkatzen dituzte.

5.2 Grafo egiturak erabiliz egindako probak

5.2.1 Denbora leihoak eraikita egindako esperimentuak

Abiapuntu gisa, BETH datu-multzoa grafo bitartez irudikatu zen, denbora leihoak eraikiz, soilik nodoen arteko egiturak kontuan hartuz, ezaugarriak erabili gabe.

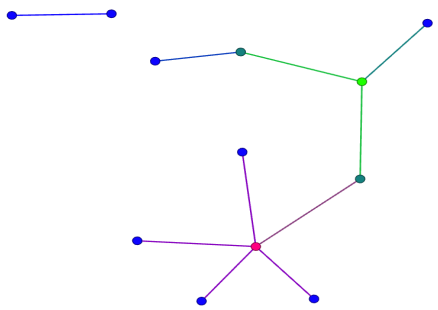
BETH datu-multzoa prozesu- eta sare-erregistroetan banatuta dago, eta proiektu hau burutzeko prozesu-erregistroak erabili dira.

Prozesuen arteko erlazioak grafo bitartez irudikatzeko, prozesu gurasoen eta prozesu umeen arteko erlazioak erabili ziren. Grafo hauek irudikatzeko, *gephi* tresna erabili zen. Tresna honek aukera ematen du nodoekin jokatzeko, koloreztatzeko eta abar.

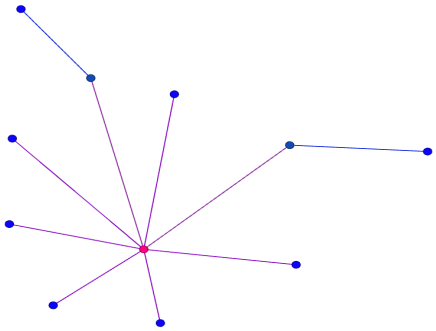
Hasierako planteamenduan, beraz, "processId"-ak eta "parentProcessId"-ak bi dimentsioko grafoetan irudikatu ziren, hauen arteko erlazioak grafoko ertzak izanik. Identifikatzaile hauek moldatu gabe erabili ziren, hau da, hasierako csv-an azaltzen ziren moduan.

Hasieran, 10 ertzeko grafoak eraiki ziren, hau da, datu-multzoa 10 timestamp-ero zatitu zen, inolako teilakatzerik gabe. Grafoak nodoen graduen arabera koloreztatu ziren eta ez zitzaien inolako identifikatzailerik jarri. Koloreztatzeko erabili zen moduari dagokionez, nodoak kolore urdina hartzen zuen irudiko gainerako nodoak baino ertz gutxiago bazituen; eta kolore arrosak adierazten zuen, uneko nodoak irudiko gainerako nodoak baino ertz gehiago zituela. Kolore berdeak tarteko kasuak adierazten zituen. Beraz, nodoak, zituzten ertz kopuruen arabera koloreztatu ziren.

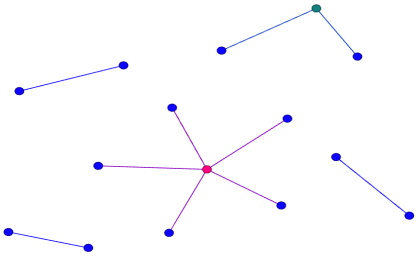
Hala ere, lortutako grafoetan ez zen erasorik nabaritzen. Honen adierazle dira 5.1 Irudia, 5.2 Irudia, 5.3 Irudia eta 5.4 Irudia. Irudi hauek, denbora tarte batean prozesuen arteko loturek duten bilakaera adierazten dute.



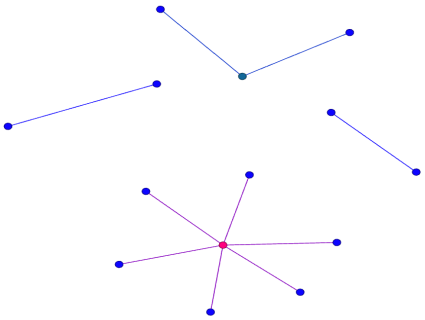
5.1 Irudia: Hamar ertzeko grafoa, 1. timestamp-etik 10. timestamp-era.



5.2 Irudia: Hamar ertzeko grafoa, 11.timestamp-etik 20.timestamp-era.



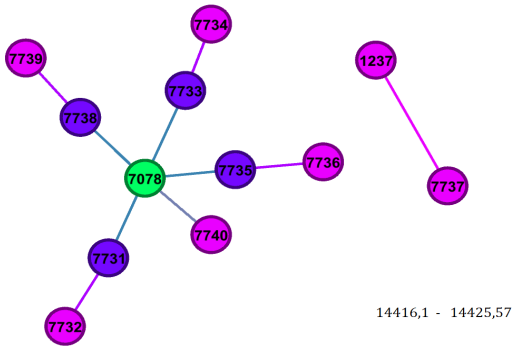
5.3 Irudia: Hamar ertzeko grafoa, 21.timestamp-etik 30.timestamp-era.



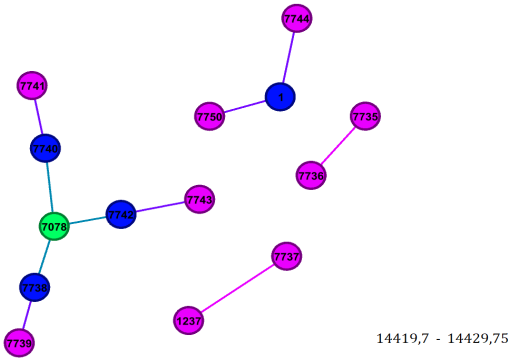
5.4 Irudia: Hamar ertzeko grafoa, 31.timestamp-etik 40.timestamp-era.

Ondoren, berriro ere, datu-multzoa 10 timestamp-ero zatitu zen, baina 5naka teilakatuz, eta nodoei beraien identifikatzailea esleitu zitzaien. Gainera, grafo bakoitzaren azpian timestamp tartea adierazi zen eta borobil hori bat txertatu zitzaien irudian erasorik bazegoen.

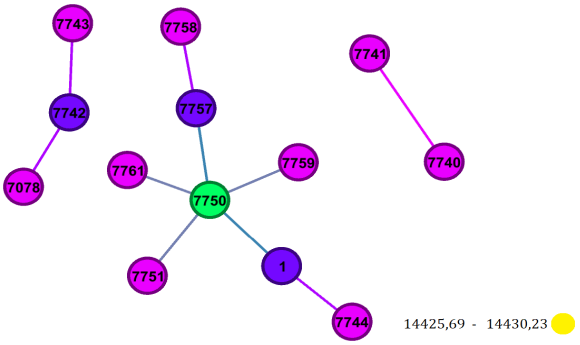
10 ertzekin eta 5eko teilakatzearekin oraindik ez zen erasorik nabaritzen begi-bistara.



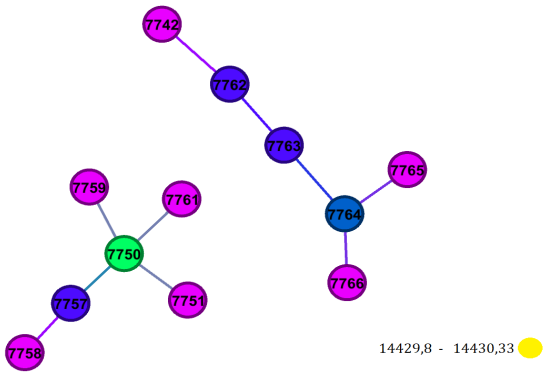
5.5 Irudia: Hamar ertzeko grafoa, erasorik gabe.



5.6 Irudia: Hamar ertzeko grafoa, erasorik gabe.

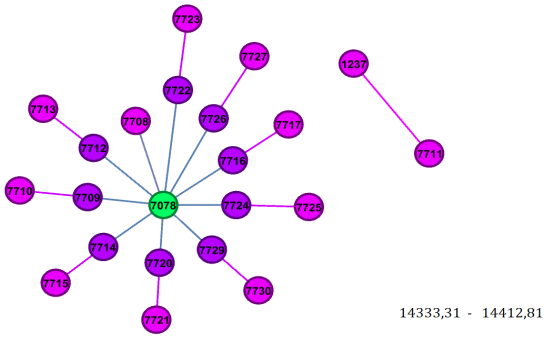


5.7 Irudia: Hamar ertzeko grafoa, erasoarekin.

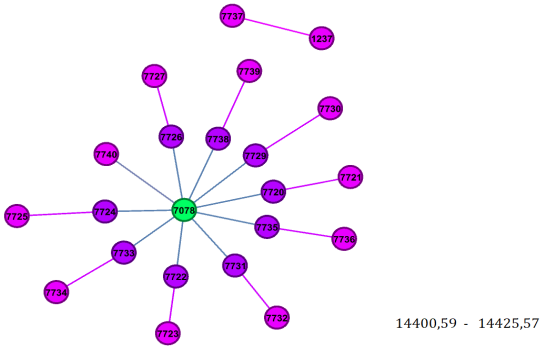


5.8 Irudia: Hamar ertzeko grafoa, erasoarekin.

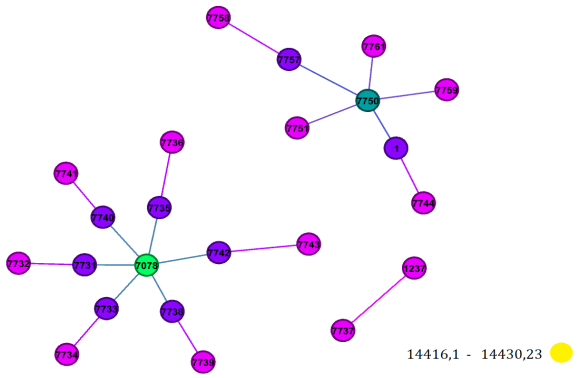
Amaitzeko 20 ertzekin egin ziren grafoak, 10naka teilakatuta. Baina, hala ere, ez ziren erasoak nabaritzen.



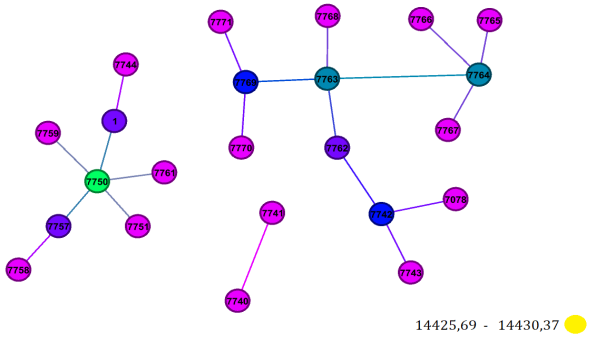
5.9 Irudia: 20 ertzeko grafoa, erasorik gabe.



5.10 Irudia: 20 ertzeko grafoa, erasorik gabe.



5.11 Irudia: 20 ertzeko grafoa, erasoekin.



5.12 Irudia: 20 ertzeko grafoa, erasoekin.

Kasu batzuetan prozesu ume bat prozesu guraso batekin behin baino gehiagotan azal zitekeen erlazionatuta csv-an, eta errepikatuta zeudenez, ertz hauek bakarrik behin hartzen ziren kontuan, errepikatuak ezabatuz. Kontua da, bi nodoren arteko erlazioa erasoa izan zitekeela gertaera batean eta beste batean ez. Beraz, ez zen nahikoa prozesuen identifikatzaileen informazioarekin grafoak osatzeko, hau da, nodoen inguruko informazio gehiago behar zen erasoak detektatu ahal izateko. Beraz, bi dimentsioko grafoak erabili beharrean, hiru dimentsioko grafoak erabili beharko ziren.

5.2.2 Denbora leihorik eraiki gabe

5.2.2.1 Aldagaiak eta grafo egitura sartuta

Grafoekin erlazionatutako hasierako planteamendua nahikoa ez zela ikusita, datu-multzoa grafo bitartez adierazteko ideiarekin jarraituz, hiru dimentsioko grafoak erabiltzea erabaki zen. Grafo hauetan, prozesuen arteko erlazioak mantenduz, erlazio hauek zituzten ezaugarriak ("processId", "parentProcessId", "userId", "eventId" eta abar) adierazi nahi ziren.

Erlazioen ezaugarriak adierazi nahi zirenez, Graph Convolutional Network erabiltzea erabaki zen, Graph Neural Network-en klase bat.

Hasiera batean, grafoak sortzeko Python-ek duen NetworkX paketea erabili zen. Pakete hau sare konplexuen egitura [9], dinamika eta funtzioak sortzeko, eta sare hauek manipulatzeko eta aztertzeko erabiltzen da. Pakete honek sortzen dituen objektuak `Graph` edo `DiGraph` motatakoak dira, grafoa ez zuzendua edo zuzendua denaren arabera. Lan honetan, prozesuen gaineko datuak izanik, `DiGraph` motako objektuak erabili dira, prozesuen arteko erlazioak adierazteko.

Ikaskuntza automatikoko eredua sortzeko, python-ek duen pytorch-geometric liburutegia erabili zen. Pytorch-geometric liburutegiak `Data` motako objektuak erabiltzen ditu grafoak irudikatzeko. Pytorch-geometric-eko `Data` objektuek [44], grafo homogeneoak adierazten dituzte. Objektu hauek, atributuak izan ditzakete nodo-mailan, ertz-mailan eta grafo-mailan.

`Data` objektuak hainbat parametro ditu [45]:

- x (torch.Tensor, hautazkoa): Nodoen ezaugarrien matrizea honako dimentsioekin: $[num\_nodes, num\_node\_features]$. (default: None)
- $edge\_index$ (LongTensor, hautazkoa): Grafoaren konektibitatea, COO formatuan eta honako dimentsioekin: [2, num_edges]. (default: None)
- $edge\_attr$ (torch.Tensor, hautazkoa): Grafoko ertzen ezaugarrien matrizea honako dimentsioekin: $[num\_edges, num\_edge\_features]$. (default: None)
- y (torch.Tensor, hautazkoa): Grafo-mailako edo nodo-mailako benetako etiketak. (default: None)
- pos (torch.Tensor, hautazkoa): Nodoaren posizio-matrizea, honako dimentsioekin: $[num\_nodes, num\_dimensions]$. (default: None)
- $**kwargs$ (hautazkoa): Atributu gehigarriak.

Proiektu honetan x, $edge\_index$ eta y parametroak erabili dira. Hauetaz gain, entrenamendua grafo bakarrarekin egin denean $train\_mask$ eta $test\_mask$ maskarak erabili dira (hauek ere `Data` objektuaren parametroak dira) nodoak entrenamendu- eta probamultzoetan sailkatzeko.

Momentura arte eraikitako grafoak `DiGraph` motako objektuak zirenez, grafoak `Data` motako objektuetara eraldatu behar izan ziren.

Grafoak eraikitzeko moduari dagokionez, bi modu daude GCN modelo bat entrenatzeko [41]:

- **Transduktiboa:** Entrenamendurako eta probetako datuak grafo berean daude. Multzo bakoitzeko nodoak elkarrekin konektatuta daude. Entrenamenduan, probarako nodoetako etiketak ezkutatu egiten dira, entrenamendurako nodoetako etiketak ikusten diren bitartean. Hala ere, nodo guztien ezaugarriak ikusgarri daude GCNarentzat. Entrenamendurako nodo horiek beraien ezaugarriak eta etiketak adierazten dituzte; probarako nodoek, aldiz, beraien ezaugarriak baino ez dituzte adierazten. Probako etiketak modeloarengandik ezkutatuta daude. Maskara bitarrak behar dira entrenamendurako nodoak eta probarako nodoak desberdintzeko.

- **Induktiboa:** Entrenamendurako grafoak eta probarako grafoak bereizita daude, eta bata bestearengandik ezkutatuta daude. Kasu induktiboa *machine learning* erregularraren antzekoa da, non ereduak entrenamenduan entrenamendurako ezaugarriak eta etiketak baino ez baititu ikusten, eta testean testerako ezaugarriak bakarrik. Entrenamendua eta proba bi grafo bereizitan eta isolatutan egiten dira. Entrenamendurako nodoetatik soilik erabiltzen ditu GCNak ezaugarriak eta etiketak. Kasu honetan, ez dago maskara bitarrak erabiltzeko beharrik.

Proiektu honetan, GCN eredua transduktiboki eta induktiboki entrenatu da. GCN eredua grafo bakarrarekin entrenatzean transduktiboki egin da, eta hainbat graforekin entrenatzean induktiboki.

5.2.2.1.1 Train eta testerako grafo bakarra erabilita

Lehendabizi, train, test eta balidaziorako datu-multzoetan datuak desorekatuta zeudenez, berrantolatu egin ziren, eta horretarako, erabilgarri zeuden csv motako taula denak batean batu ziren.

Hainbat grafo sortu beharrean, grafo bakar bat osatu zen, eta grafo honetako nodoak train eta test multzoetan banatu ziren.

Esan bezala, entrenamendurako eta probetako datuak grafo berean zeuden. Beraz, maskarak erabiltzen ziren entrenamendurako nodoak eta probarako nodoak desberdintzeko.

ERREPIKATUTAKO ERTZAK EZABATUZ

Hasiera batean, datu kopurua oso murriztua zen, errepikatuta zeuden erlazio guztiak baztertzen zirelako. Hau da, bi prozesuren arteko erlazioa grafoan existitzen bazen, ezin zen ertz hau errepikatu.

Ondorioz, ezaugarri gehienak baztertzen ziren. Izan ere, prozesu batek, erregistro bakoitzean ezaugarri desberdinak izan ditzake. Horregatik, ertz bat errepikatzen zenean, hau baztertu beharrean, jada nodoek ezaugarriak zituzten arren, hasieran zituzten ezaugarriei ezaugarri berriak gehitzen zitzaizkien, ondoren batez besteko bidez bateratzeko.

Honela, ertz guztiak mantentzen ziren eta nodoari ezaugarri guztien batez bestekoa esleitzen zitzaion ezaugarri gisa. Eta ertzetako bat erasoa zenean, zuzenean nodoari “evil=1” balioa esleitzen zitzaion, azken finean prozesu hori erasotzailea zelako.

Gainera, ezaugarriei aldagai bat gehitu zitzaien, ertz bakoitza zenbat aldiz azaltzen zen kontatzeko.

Esan bezala, prozesuen erlazioak prozesu gurasoaren eta prozesu umearen artekoak dira. Baina csv-etan prozesu umearen ezaugarriak azaltzen dira. Beraz, soilik prozesu gurasoak diren prozesuek ez dituzte ezaugarriak izango. Horregatik, aurretik azaldu den bezala, prozesu gurasoaren ezaugarriei “-1000” balioa esleitzen zitzaien aldagai guztietan, ertz kopurua kontatzeko erabilitako aldagaian eta erasotzailea den adierazten duen etiketan

izan ezik.

Sortutako grafoak, guztira 1799 nodo zituen. Entrenamendurako 1081 nodo zituen, hauetatik 3 erasotzaileak, eta testerako 718 nodo, hauetatik 11 erasotzaileak. Banaketa hau gutxi gorabehera nodoen %60 (train) eta %40koa (test) izan zen.

Eredua grafo honekin entrenatu ostean, lortutako emaitzak ez ziren batere onak izan.

Entrenamendurako nodoen artean, hau da, 1081 nodotatik 3 besterik ez ziren erasotzaileak. Eredua entrenatu ostean, 1081 nodotatik, 1078 ongi sailkatzen zituen, 0'9972ko asmatze-tasa lortuz. Baina kontua da, gaizki sailkatzen zituela erasotzaileak ziren 3 nodo horiek, beraz, nodo guztiak 0 klasean ("False" gisa) sailkatzen zituen, hau da, nodo on gisa sailkatzen zituen.

GCN eredua entrenatu ostean, probatu egin zen. Test multzoa 718 nodok osatzen zuten, hauetatik 11 erasotzaileak izanik. Test multzoa ebaluatu ostean, asmatze-tasa 0'98468koa izan zen. 718 nodotik 707 ongi sailkatu zituen. Baina test kasuan ere, nodo guztiak 0 klasean sailkatzen zituen.

5.2 Taulan laburbiltzen dira train eta test kasuetan lortutako doitasuna, estaldura eta F1 balioak.

	DOITASUNA		**ESTALDURA**		**F1**	
	0 klasea	**1 klasea**	**0 klasea**	**1 klasea**	**0 klasea**	**1 klasea**
Train	0.99722479	0	1	0	0.99861047	0
Test	0.98467967	0	1	0	0.9922807	0

5.2 Taula: Train eta testerako grafo bakarra erabilita eta errepikatutako ertzak ezabatuz lortutako emaitzak.

Asmatze-tasa altuak lortu arren, ez ziren emaitza onak. Ereduak, grafoko nodo guztiak ez-erasotzailetzat sailkatzen zituen. Hau islatuta ikusten da 1 klaseari dagokion doitasun, estaldura eta F1 balioetan, kasu guztietan 0 balioa dutelako, metrika hauek izan dezaketen baliorik baxuena. Eta honek adierazten du algoritmoak denbora guztian huts egiten duela.

Emaitza hauek train/test multzoen banaketa desorekatuaren ondorio izan zitezkeen, guztira 14 nodo erasotzaile bakarrik zeudelako 1799 nodoko grafo batean. Beraz, datumultzoen banaketa orekatuago baten beharra zegoen.

DATU-MULTZOA HANDITUZ

Lehenengo emaitzak aztertu ondoren, grafoen nodo kopurua handiagotzea beharrezkoa zela bistakoa zen.

Beraz, nodo kopurua handiagotzeko, ertz bat errepikatuta zegoenean, prozesu umeari ezaugarriak aldatu beharrean, nodo berri bat sortzen zen uneko prozesuaren ezaugarriekin. Esaterako, grafoan jada 1→0 ertza izanda eta hurrengo ertza 2→0 izanda, 0 nodoari identifikatzailea aldatzen zitzaion eta modu horretan ezaugarrien zerrenda mantentzen zen. Nodo berri bat sortzean, ausaz ertz desberdinak esleitzen zitzaizkion, honek ertz bakarra izan ez zezan.

Ezaugarriak eraldatzeko planteamendu hau, k-NN eta Naïve Bayes sailkatzaileetan erabilitakoaren berdina da, baina grafo egitura emanez.

Modu horretan, existitzen ez ziren identifikatzaileak esleituz, nodo kopurua handiagotzea lortzen zen, batez ere, erasotzaileak ziren nodoen kopurua handiagotzea, eta nodo hauen ezaugarriak errealak izango ziren.

Kasu honetan ere, ezaugarriak normalizatu egin ziren.

Horrela, 19032 nodoko grafoa eraiki zen, eta nodo hauetatik 10005 ziren erasotzaileak. Gainera, train eta test multzoetarako banaketa orekatua egiten saiatu zen, beti ere, banaketa hau gutxi gorabehera nodoen %60 (train) eta %40koa (test) izanik.

Entrenamendurako 11419 nodo erabili ziren, eta hauetatik 6906 ziren erasotzaileak. Eta testerako 7613 nodo erabili ziren, eta hauetatik 3099 ziren erasotzaileak.

	DOITASUNA		**ESTALDURA**		**F1**	
	0 klasea	**1 klasea**	**0 klasea**	**1 klasea**	**0 klasea**	**1 klasea**
Train	0	0.6047815	0	1	0	0.75372442
Test	0	0.40706686	0	1	0	0.57860344

5.3 Taula: Datu-multzoa handituz lortutako emaitzak.

Emaitza hauei erreparatuz, ikus daiteke nodo erasotzaileen kopurua handitzea ez zela nahikoa modeloak datuetatik ikas zezan. Honen adierazle dira 5.3 Taulan azaltzen diren doitasunaren, estalduraren eta F1 metrikaren balioak, non 0 klaseari dagokionean, hiru metrikek 0 balioa duten, metrika hauek izan dezaketen baliorik baxuena. Hau, ereduak nodo guztiak 1 klasean ("True" gisa) sailkatzen zituelako zen, hau da, erasotzaile gisa. Hori dela eta, planteamendu hau k-NN eta Naïve Bayes sailkatzaileekin erabilgarria izan arren, GCN ereduak errendimendu oso baxua izaten jarraitzen zuen datu-multzoa handitu arren.

5.2.2.1.2 Train-erako hainbat grafo erabilita

Grafo bakar bat osatuta emaitzak onak ez zirela ikustean, entrenamendurako hainbat grafo erabiltzea erabaki zen. Kasu honetan, entrenamendurako grafoak eta probarako grafoak bereizita zeuden.

Nodoen kopurua train eta testerako grafo bakarra erabilita handitzen zen bezala handiagotu zen, eta ezaugarriak normalizatu egin ziren.

Hainbat proba egin ziren tamaina desberdineko grafoekin, 5.4 Taulan adierazten dira probetako batek emandako emaitzak.

Entrenamendurako 852 grafo erabili ziren, eta grafo bakoitzak 20 nodo zituen. Train-erako, guztira 17040 nodo inguru erabili ziren, eta nodo hauetatik 4174 ziren erasotzaileak. Testerako 824 nodoko grafoa erabili zen, eta nodo hauetatik 106 ziren erasotzaileak.

	DOITASUNA		**ESTALDURA**		**F1**	
	0 klasea	**1 klasea**	**0 klasea**	**1 klasea**	**0 klasea**	**1 klasea**
Test	0.87135922	0	1	0	0.93125811	0

5.4 Taula: Entrenamendurako hainbat grafo erabilita lortutako emaitzak.

5.4 Taulako emaitzei erreparatuz, ikus daiteke ereduak nodo guztiak ez-erasotzaile gisa sailkatzen dituela. Izan ere, taulan, doitasuna, estaldura eta F1 metrikek, 1 klaseari dagokionean, 0 balioa dute, metrika hauek izan dezaketen baliorik baxuena. Beraz, modeloak datuetatik ikasi gabe jarraitzen du entrenamendurako hainbat grafo erabili arren. Horregatik, ondorioztatu daiteke, entrenamendua grafo bakarrarekin edo hainbat graforekin eginda, lortutako emaitzek ez dutela alde handia izango.

KAPITULUA 6

Ondorioak

Zibersegurtasunean erasoak antzemateko, oinarrizkoa da prozesu-deiak erasoak diren detektatzen duen eredu eraginkorra izatea. Eredu hau entrenatzeko, oso baliagarria da BETH datu-multzoa erabiltzea. Batez ere, honek eskaintzen duen prozesu-deien erregistroa erabiltzea.

Proiektu hau burutzean, hainbat bide desberdin jarraitu dira. Lehendabizi, k-NN eta Naïve Bayes bezalako sailkapen gainbegiraturako teknika klasikoak erabili dira, eta ondoren, grafo egiturak erabili dira.

k-NN eta Naïve Bayes sailkatzaileekin lortutako emaitzak nahiko onak izan dira. 5.1 Taulan biltzen dira sailkatzaile hauekin lortutako doitasuna, estaldura eta F1 balioak. Doitasuna metrika erabiliz lortutako balioek adierazten dute positibotzat, edo 1 klasean, sailkatu diren datuen artetik %80tik gora sailkatu direla ondo. Eta negatibotzat, edo 0 klasean, sailkatu diren datuen artetik denak sailkatu direla ondo. Estaldura metrika erabiliz lortutako balioek adierazten dute, ereduak egiazko positiboak ehuneko ehunean identifikatzen dituela, eta egiazko negatiboak, %80tik gora.

Sailkatzaile hauen artetik, emaitza onenak eman dituena Gaussian Naïve Bayes sailkatzailea izan da.

Emaitza hauek lortzeko, BETH datu-multzoko datu eraldatuak erabili dira. Horregatik, emaitza hauek adierazten dute datuak moldatzeko pentsatutako planteamendua erabilgarria izan daitekeela. Hala ere, lortutako emaitza hauek, onak izan arren, hobetu daitezke.

Emaitza hauek hobetze aldera, grafo egiturak erabiltzea erabaki zen. Lehendabizi, 2 dimentsioko grafoak erabili ziren, denbora leihoak eraikita, soilik prozesuak eta hauen prozesu gurasoen arteko erlazioak grafo batean irudikatuz, inolako informazio gehiago gabe. Hala ere, ez zen emaitzak hobetzea lortu.

Lortutako irudiei erreparatzen badiegu, argi ikusten da, begi-bistara ez dela erasorik nabaritzen, eta irudi hauek sailkatuz ez genukeela emaitza egokirik lortuko. Gainera, prozesuen arteko konexioak ez dira nahikoa erasoa dagoen edo ez erabakitzeko, zeren eta bi prozesuren arteko konexioa, lotura horrek duen ezaugarrien arabera erasoa izan daiteke

edo ez.

Adibide bat jarriz, demagun $0 \rightarrow 1$ prozesuen arteko lotura dugula, erregistroak sortzen dituen prozesua 1 izanik eta 0 honen gurasoa. Sistema abiarazi eta 2 minuturen buruan 1 prozesua erregistratzen da, 0 gurasoarekin eta zenbait ezaugarrirekin. Hori gertatu eta 5 minuturen buruan, berriz ere, 1 prozesuak erregistro bat sortzen du, 0 gurasoa izango du baina gainerako ezaugarriak guztiz desberdinak izango dira. Lehenengo erregistroan, 1 prozesuak ez du erasorik egin, baina agian bigarren erregistroan bai.

Honekin ondorioztatu daiteke, zibersegurtasuneko datuen kasuan ezin dela prozesu bat erasotzailetzat edo ontzat sailkatu bakarrik prozesuen arteko loturek sortzen duten grafo egiturari erreparatuz.

Ondoren, 3 dimentsioko grafoak erabili ziren, grafo egitura erabiltzeaz gain, nodoei ezaugarriak esleitzeko aukera ematen zuelako. 3 dimentsioko grafoekin egindako esperimentuetan, eredua grafo bakarrarekin zein askorekin eta tamaina anitzeko grafoekin entrenatu zen.

Emaitzei erreparatuz, GCN eredua erabiliz lortutako emaitza guztiak antzekoak izan dira. Ereduak test-erako erabilitako nodo guztiak klase batean edo bestean sailkatzen ditu. Honek esan nahi du, sortutako ereduak ez duela ikasten, eta honen arrazoia erabilitako ezaugarriak izan zitezkeen. Hala ere, k-NN eta Naïve Bayes sailkatzaileetan lortutako emaitzen arabera ondorioztatu daiteke hau ez dela arrazoia.

6.1 Etorkizuneko lanak

Proiektuan zehar lortutako emaitzak aztertu ondoren, esan daiteke ikerketa honi jarraipena eman beharko zitzaiola emaitzak hobetze aldera.

Lehendabizi egin beharreko lana, emaitza hauen zergatia argitzea izan daiteke. Grafo egiturak erabiltzea ideia ona den edo ez erabaki beharko litzateke.

Gainera, grafo egiturak erabiltzea egokia dela ondorioztatuz gero, grafoekin lan egiteko dauden neurona-sare mota desberdinak probatu daitezke. Azterlan honetan, Graph Convolutional Network-ak erabili dira, hala ere, GCN-ez gain, GraphSAGE eta Graph Attention Network (GAT) ere erabili daitezke, besteak beste.

Beste alde batetik, lan honetan grafoetako nodoak sailkatu dira, baina ertzak edo grafoak ere sailkatu ahal izango lirateke. Izan ere, denbora tarteak eraikiz, erasoek duten portaera era orokor batean irudikatzeko grafo desberdinak erabili daitezke, eta ondoren, grafo hauek sailkatu daitezke.

Horrez gain, badirudi egin diren proba desberdinetan, gainbegiratu gabeko ikaskuntza erabiliz lortutako emaitzak nahiko onak izan dira. Hortaz, bide horretatik jarraitzea ere aukera bat izan daiteke.

Azkenik, k-NN eta Naïve Bayes sailkatzaileek, batez ere, Gaussian Naïve Bayes sailkatzaileak, lortutako emaitzez baliatuz, grafo egiturak alde batera utzi eta sailkatzaile

desberdinen arteko konbinazioak proba daitezke.

6.2 Ondorio pertsonalak

Sei hilabeteko lan honen ostean, lortutako emaitzak atzean dagoen ahalegina islatzeko egokienak ez diren arren, egindako lanarekin pozik nagoela adieraz dezaket.

Proiektuaren gaia eskaini zidatenean, hasiera batean zalantzan jarri nituen lan hau aurrera eramateko nituen gaitasunak. Hala ere, GrAL-ean zehar aurkeztutako egoera desberdinak gogotsu hartu ditut momentu oro, eta garapen prozesuan aurrera joan ahala, nire trebetasunetan gehiago sinesten joan naiz.

Prozesuan zehar gauza berriak ezagutzen eta ikasten joan naiz, eta proiektuaren amaieran lortu beharreko helburuak erdietsi ditudala uste dut.

Eskuratutako emaitzei dagokienez, hasiera batean gogogabetu egin ninduten. Hala ere, edozein proiektutan lortzen diren emaitzak dena delakoak izanda ere, emaitza guztiek ematen dute informazio baliagarria. Baliteke, emaitza eskasak lortzeak proiektua beste bide batetik zuzendu behar dela adieraztea, edota planteamendua pixka bat aldatu behar dela. Honekin esan nahi dudana da, proiektua amaitzean oso argi geratu zaidala egindako lana, lortutako emaitzak egokiak zein eskasak izan, beti izango dela erabilgarria.

A Eranskina

Proiektuan zehar egindako beste proba batzuk

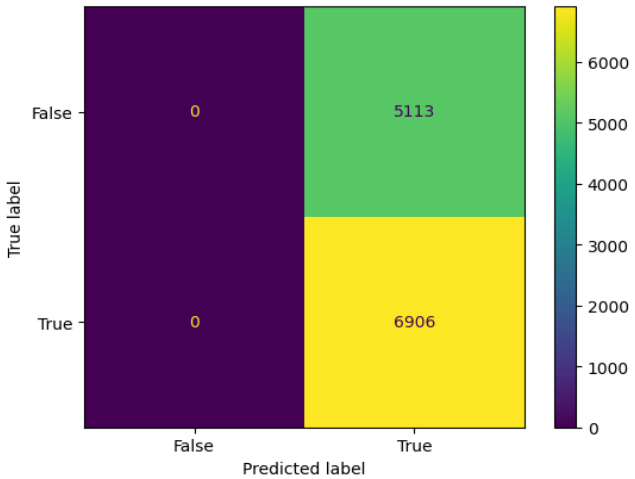
Atal honetan, proiektuan egindako proba gehiago azaltzen dira. Esperimentu hauetan lortutako emaitzak azaltzen dira, zehazki, doitasuna eta estaldura metrikekin lortutako emaitzak, eta lortutako konfusio-matrizeak. Proba hauek denbora leihorik eraiki gabe, eta aldagaiak eta grafo egitura sartuta egin ziren.

Entrenamendurako eta probarako grafo bakarra erabiliz

Lehenengo proba honetan, 20032 nodoko grafoa eraiki zen, eta nodo hauetatik 10005 ziren erasotzaileak.

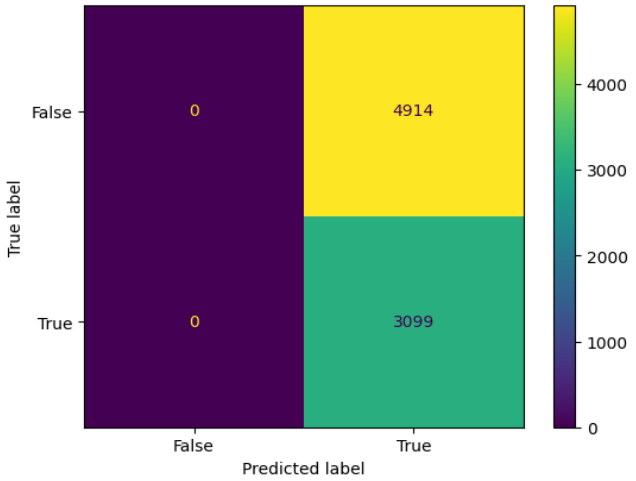
Entrenamendurako 12019 nodo erabili ziren, hauetatik 6906 izanik erasotzaileak. Eta testerako 8013 nodo erabili ziren, eta hauetatik 3099 ziren erasotzaileak.

- **Entrenamendua:**
  - Doitasuna: [0, 0.57459023]
  - Estaldura: [0, 1]
  - Konfusio-matrizea:



1 Irudia: 20032 nodoko grafoaren entrenamenduko konfusio-matrizea.

- **Proba:**
  - Doitasuna: [0, 0.38674654]
  - Estaldura: [0, 1]
  - Konfusio-matrizea:



2 Irudia: 20032 nodoko grafoaren probako konfusio-matrizea.

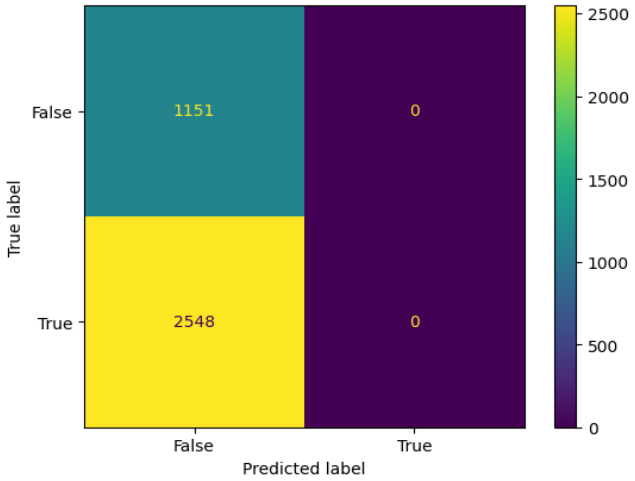
Entrenamendurako hainbat grafo erabiliz

Proba hauetan, entrenamendurako hainbat grafo erabili ziren, eta honako egoera hauek aztertu ziren:

1. **Egoera:** Entrenamendurako 852 grafo erabili ziren, eta grafo bakoitzak 20 nodo zituen. Beraz, guztira 17040 nodo inguru erabili ziren entrenamendurako, eta hauetatik, 4174 ziren erasotzaileak. Testerako 3699 nodoko grafoa erabili zen, eta nodo hauetatik 2548 ziren erasotzaileak.

   Honako hauek dira test-ean lortutako emaitzak:

   - Doitasuna: [0.31116518, 0]
   - Estaldura: [1, 0]
   - Konfusio-matrizea:

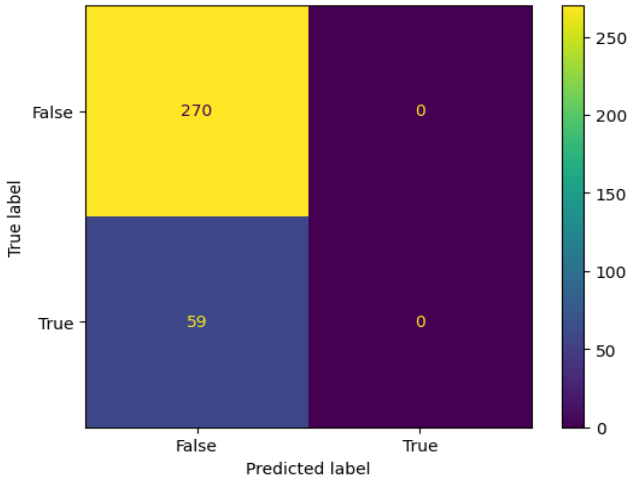


3 Irudia: 1.egoeraren konfusio-matrizea.

2. **Egoera:** Entrenamendurako 852 grafo erabili ziren, eta grafo bakoitzak 20 nodo zituenez, guztira 17040 nodo inguru erabili ziren entrenamendurako. Nodo hauetatik, 4174 ziren erasotzaileak. Testerako 329 nodoko grafoa erabili zen, eta nodo hauetatik 59 ziren erasotzaileak.

   Honako hauek dira test-ean lortutako emaitzak:

   - Doitasuna: [0.82066869, 0]
   - Estaldura: [1, 0]
   - Konfusio-matrizea:

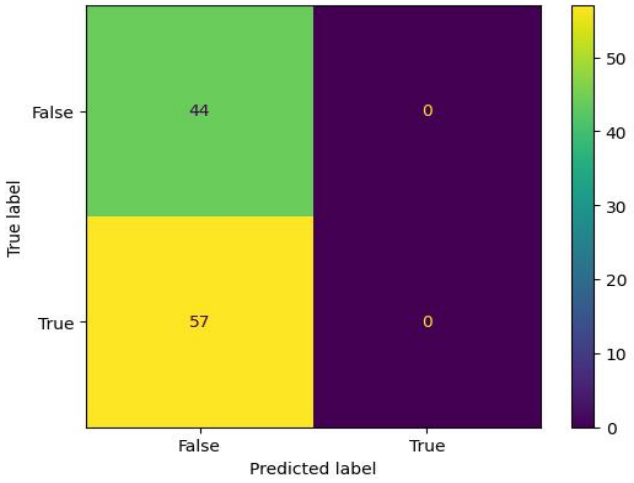


4 Irudia: 2.egoeraren konfusio-matrizea.

3. **Egoera:** Entrenamendurako 852 grafo erabili ziren, eta grafo bakoitzak 20 nodo izanda, guztira 17040 nodo inguru erabili ziren entrenamendurako. Nodo hauetatik, 4174 ziren erasotzaileak. Testerako 101 nodoko grafo bakarra erabili zen, eta nodo hauetatik 57 ziren erasotzaileak.

   Honako hauek dira test-ean lortutako emaitzak:

   - Doitasuna: [0.43564356, 0]
   - Estaldura: [1, 0]
   - Konfusio-matrizea:



5 Irudia: 3.egoeraren konfusio-matrizea.

B Eranskina

Proiektuaren inplementazioa

Proiektu honetan burututako esperimentazioan sortutako kodea atzigarri dago honako webgunean: https://github.com/maideramutxastegi/Atakeen_sailkapena.git

Bibliografia

[1] "¿Qué es la ciberseguridad y su importancia?," *Revista Seguridad 360*, 2021. [Online]. Eskuragarri: https://revistaseguridad360.com/destacados/que-es-la-ciberseguridad. Ikusi 1 orrialdea.

[2] "Situación de la ciberseguridad en Euskadi - 2º trimestre 2022," tech. rep., Basque Cybersecurity Centre, 2022. Ikusi 1 orrialdea.

[3] K. Highnam, K. Arulkumaran, Z. Hanif, and N. R. Jennings, "BETH Dataset: Real Cybersecurity Data for Anomaly Detection Research," *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2021. Ikusi 2, 15 orrialdeak.

[4] "¿Qué es Python?," datarik gabe. [Online]. Eskuragarri: https://aws.amazon.com/es/what-is/python/. Ikusi 9 orrialdea.

[5] L. Gracia, "¿Qué es Gephi?," *Un poco de Java*, 2014. [Online]. Eskuragarri: https://unpocodejava.com/2014/09/11/que-es-gephi/. Ikusi 9 orrialdea.

[6] "The Open Graph Viz Platform," datarik gabe. [Online]. Eskuragarri: https://gephi.org/. Ikusi 9 orrialdea.

[7] D. Urrutia, "Qué es Gephi," *Arimetrics*, 2023. [Online]. Eskuragarri: https://www.arimetrics.com/glosario-digital/gephi. Ikusi 9 orrialdea.

[8] D. P. Estrada, "Gephi," *Edutools*, datarik gabe. [Online]. Eskuragarri: https://edutools.tec.mx/es/colecciones/tecnologias/gephi. Ikusi 9 orrialdea.

[9] "NetworkX, Network Analysis in Python," datarik gabe. [Online]. Eskuragarri: https://networkx.org/. Ikusi 10, 34 orrialdeak.

[10] "What is NetworkX?," datarik gabe. [Online]. Eskuragarri: https://www.nvidia.com/en-us/glossary/data-science/networkx/. Ikusi 10 orrialdea.

[11] S. Gonzalez, "¿Qué es la licencia BSD?," *AppMaster*, 2023. [Online]. Eskuragarri: https://appmaster.io/es/blog/que-es-la-licencia-bsd. Ikusi 10 orrialdea.

[12] "Pytorch," *Wikipedia, la enciclopedia libre*, 2022. [Online]. Eskuragarri: https://es.wikipedia.org/wiki/PyTorch. Ikusi 10 orrialdea.

[13] "Aprende sobre Pytorch con cursos online," datarik gabe. [Online]. Eskuragarri: https://www.edx.org/es/aprende/pytorch. Ikusi 10 orrialdea.

[14] *PyG Documentation*, datarik gabe. [Online]. Eskuragarri: https://pytorch-geometric.readthedocs.io/en/latest/. Ikusi 11 orrialdea.

[15] N. Malingan, "Pytorch Geometric," *Scaler Topics*, 2023. [Online]. Eskuragarri: https://www.scaler.com/topics/deep-learning/pytorch-geometric/. Ikusi 11 orrialdea.

[16] "Scikit-Learn, herramienta básica para el Data Science en Python," 2019. [Online]. Eskuragarri: https://www.master-data-scientist.com/scikit-learn-data-science/. Ikusi 11 orrialdea.

[17] "Espacio de recursos de ciencia de datos," datarik gabe. [Online]. Eskuragarri: http://datascience.recursos.uoc.edu/es/preprocesamiento-de-datos-con-sklearn/. Ikusi 11 orrialdea.

[18] "Discretización," *Wikipedia, la enciclopedia libre*, 2022. [Online]. Eskuragarri: https://es.wikipedia.org/wiki/Discretización. Ikusi 11 orrialdea.

[19] S. Narkhede, "Understanding Confusion Matrix," *Medium*, 2018. [Online]. Eskuragarri: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62. Ikusi 12 orrialdea.

[20] J. Gallagher, "What is a Confusion Matrix? A Beginner's Guide," *Roboflow Blog*, 2022. [Online]. Eskuragarri: https://blog.roboflow.com/what-is-a-confusion-matrix/. Ikusi 12 orrialdea.

[21] "Python Machine Learning - Confusion Matrix," datarik gabe. [Online]. Eskuragarri: https://www.w3schools.com/python/python_ml_confusion_matrix.asp. Ikusi 12 orrialdea.

[22] S. Tayabali, "A simple guide to building a confusion matrix," 2020. [Online]. Eskuragarri: https://blogs.oracle.com/ai-and-datascience/post/a-simple-guide-to-building-a-confusion-matrix. Ikusi 12 orrialdea.

[23] N. Selvaraj, "Confusion matrix, precision, and recall explained," *KDnuggets*, datarik gabe. [Online]. Eskuragarri: https://www.kdnuggets.com/2022/11/confusion-matrix-precision-recall-explained.html. Ikusi 13 orrialdea.

[24] V. Jayaswal, "Performance Metrics: Confusion matrix, Precision, Recall, and F1 Score," *Medium*, 2021. [Online]. Eskuragarri: https://towardsdatascience.com/performance-metrics-confusion-matrix-precision-recall-and-f1-score-a8fe076a2262. Ikusi 13 orrialdea.

[25] J. M. Heras, "Precision, Recall, F1, Accuracy en clasificación," *IArtificial.net*, 2020. [Online]. Eskuragarri: https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion. Ikusi 13 orrialdea.

[26] Jesús, "Más Allá del Accuracy: Precision, Recall y F1," *DataSmarts Español*, 2019. [Online]. Eskuragarri: https://datasmarts.net/es/mas-alla-del-accuracy-precision-recall-y-f1/. Ikusi 13 orrialdea.

[27] "Honeypot," *Wikipedia, entziklopedia askea.*, 2020. [Online]. Eskuragarri: https://eu.wikipedia.org/wiki/Honeypot. Ikusi 15 orrialdea.

[28] A. S. Corbalán, "Procesos en Linux. Información y administración." [Online]. Eskuragarri: https://sanchezcorbalan.es/procesos-en-linux-informacion-y-administracion/, 2021. Ikusi 17 orrialdea.

[29] "¿Qué es el algoritmo de k vecinos más cercanos?," *IBM*, datarik gabe. [Online]. Eskuragarri: https://www.ibm.com/es-es/topics/knn. Ikusi 18, 19 orrialdeak.

[30] G. R. Sahani, "Euclidean and Manhattan distance metrics in Machine Learning," *Medium*, 2020. [Online]. Eskuragarri: https://medium.com/analytics-vidhya/euclidean-and-manhattan-distance-metrics-in-machine-learning-a5942a8c9f2f. Ikusi 19 orrialdea.

[31] S. A. Gokte, "Most Popular Distance Metrics Used in KNN and When to Use Them," *KDnuggets*, datarik gabe. [Online]. Eskuragarri: https://www.kdnuggets.com/2020/11/most-popular-distance-metrics-knn.html. Ikusi 19 orrialdea.

[32] "Naive Bayes," datarik gabe. [Online]. Eskuragarri: https://scikit-learn.org/stable/modules/naive_bayes.html. Ikusi 20 orrialdea.

[33] V. Roman, "Algoritmos Naive Bayes: Fundamentos e Implementación," *Medium*, 2019. [Online]. Eskuragarri: https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementaci%C3%B3n-4bcb24b307f. Ikusi 20 orrialdea.

[34] P. Majumder, "Gaussian Naive Bayes," *OpenGenus IQ: Computing Expertise & Legacy*, 2020. [Online]. Eskuragarri: https://iq.opengenus.org/gaussian-naive-bayes/. Ikusi 21 orrialdea.

[35] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, no. 11, 2019. [Online]. Eskuragarri: https://doi.org/10.1186/s40649-019-0069-y. Ikusi 21, 26 orrialdeak.

[36] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, *Graph neural networks: A review of methods and applications*, vol. 1, pp. 57–81. AI open, 2020. [Online]. Eskuragarri: https://doi.org/10.1016/j.aiopen.2021.01.001. Ikusi 21, 23 orrialdeak.

[37] T. Masui, "Graph Neural Networks with PyG on Node Classification, Link Prediction, and Anomaly Detection," *Medium*, 2022. [Online]. Eskuragarri: https://towardsdatascience.com/graph-neural-networks-with-pyg-on-node-classification-link-prediction-and-anomaly-detection-14aa38fe1275. Ikusi 21 orrialdea.

[38] "Node Classification," datarik gabe. [Online]. Eskuragarri: https://paperswithcode.com/task/node-classification. Ikusi 23 orrialdea.

[39] C. Pham, "Graph convolutional networks (gcn)," *TOPBOTS*, 2020. [Online]. Eskuragarri: https://www.topbots.com/graph-convolutional-networks/. Ikusi 23, 25 orrialdeak.

[40] B. Sanchez-Lengeling, E. Reif, A. Pearce, and A. B. Wiltschko, "A Gentle Introduction to Graph Neural Networks," *Distill*, 2021. [Online]. Eskuragarri: https://distill.pub/2021/gnn-intro. Ikusi 24, 26 orrialdeak.

[41] R. Anand, "Math Behind Graph Neural Networks," 2022. [Online]. Eskuragarri: https://rish-16.github.io/posts/gnn-math/. Ikusi 24, 34 orrialdeak.

[42] Q. Liu, Z. Hu, R. Jiang, and M. Zhou, *DeepCDR: a hybrid graph convolutional network for predicting cancer drug response*, vol. 36, pp. i911–i918. 2020. Ikusi 26 orrialdea.

[43] Scikit-learn, *sklearn.preprocessing.MinMaxScaler*, datarik gabe. [Online]. Eskuragarri: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html. Ikusi 27 orrialdea.

[44] PyTorch, *torch_geometric.data*, datarik gabe. [Online]. Eskuragarri: https://pytorch-geometric.readthedocs.io/en/latest/modules/data.html. Ikusi 34 orrialdea.

[45] PyTorch, *torch_geometric.data.Data*, datarik gabe. [Online]. Eskuragarri: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.data.Data.html. Ikusi 34 orrialdea.