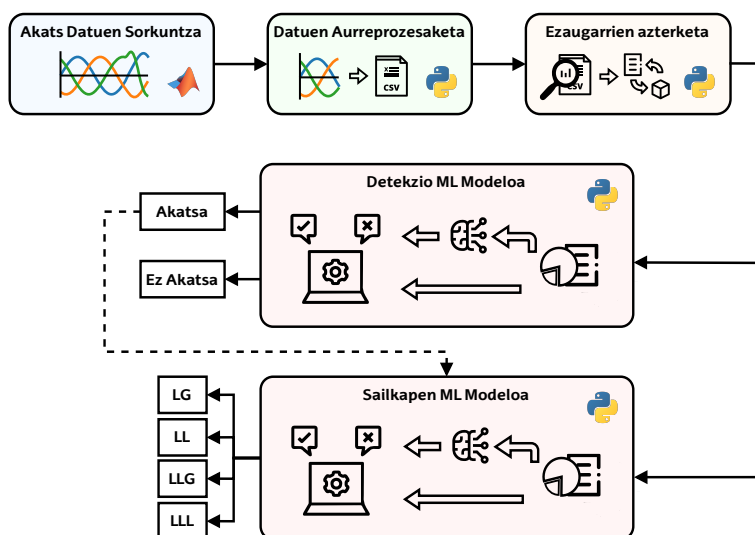


INDUSTRIA INGENIERITZA UNIBERTSITATE MASTERRA

MASTER AMAIERAKO LANA

Akats elektriko detekzioa Adimen Artifizialen estrategia desberdinen bidez



Ikaslea: Jon Xabier Rementeria Renteria

Zuzendaria: Estefanía Planas Fullaondo

Ikasturtea: 2023-2024

Data Bilbo, 2024ko ekainak 6

Laburpen hirueleduna

Izenburua: Akats elektrikoaren detekzioa Adimen Artifizialen estrategia desberdinen bidez.

Proiektu honen helburua sistema elektrikoetan akats elektrikoak detektatu eta sailkatzeko gai den adimen artifizialeko modelo bat garatzea da. Simulazioen bidez, akats elektrikoaren datuak sortzen dira, ondoren prozesatzen direnak, ezaugarri esanguratsuak erauziz eta hautatuz. Azkenik, datu hauek erabiliz, detekzio eta sailkapen ikasketa automatiko modelo desberdinak entrenatu eta ebaluatzen dira, akatsak modu zehatz eta azkar batean detektatzeko, sarearen operazioa optimizatzeko eta mantentze kostuak eta etenaldiak murrizteko.

Gako-hitzak: *Akats elektrikoaren detekzioa, Adimen artifiziala, Ikasketa automatikoa, Datuen prozesamendua, Sare adimendunak.*

Título: Detección de fallos eléctricos mediante diferentes estrategias de Inteligencia Artificial.

El proyecto se centra en el desarrollo de un modelo de inteligencia artificial para la detección y clasificación de fallos eléctricos en sistemas eléctricos. Utilizando simulaciones, se generan datos de fallos eléctricos, que posteriormente se procesan para extraer y seleccionar características significativas. Estos datos se utilizan para entrenar y evaluar diferentes modelos de aprendizaje automático, con el fin de mejorar la precisión y rapidez en la detección de fallos, optimizando la operación de la red y reduciendo costos de mantenimiento y fallos.

Palabras clave: *Detección de fallos eléctricos, Inteligencia artificial, Aprendizaje automático, Procesamiento de datos, Redes eléctricas inteligentes.*

Title: Electrical fault detection through different artificial intelligence strategies.

The project aims to develop an artificial intelligence model capable of detecting and classifying electrical faults in electrical systems. Simulations are used to generate electrical fault data, which are then processed to extract and select significant features. These processed data are used to train and evaluate various machine learning models, aiming to improve the accuracy and speed of fault detection, optimize network operation, and reduce maintenance costs and outages.

Keywords: Electrical fault detection, Artificial intelligence, Machine learning, Data processing, Smart grids.

Gaien Aurkibidea

Laburpen hirueleduna	i
Irudien zerrenda	vii
Taulen zerrenda	ix
Akronimoak	xi
1 Sarrera	1
2 Testuingurua	3
2.1 <i>Global Smart Grids Innovation Hub (GSGIH)</i>	3
2.2 <i>Applied Electronics Research Team (APERT)</i>	3
2.3 <i>Smart Griden</i> beharra	4
2.4 RECON Proiektua	6
3 Helburuak eta Irismena	7
3.1 Helburuak	7
3.1.1 Helburu nagusia	7
3.1.2 Helburu zehatzak	7
3.1.3 Garapen Jasangarriko Helburuak (GJH)	8
3.2 Irismena	9

4	Proiektuaren onurak	11
4.1	Onura tekniko eta ekonomikoak	11
4.2	Onura sozialak	12
5	Artearen Egoera	13
5.1	Sistema elektrikoen analisia	13
5.1.1	Elementuak	13
5.1.2	MGen operazio moduak	17
5.2	Akatsak sistema elektrikoan	18
5.3	Akats elektrikoen detekzioa	20
5.3.1	Metodoak	20
5.3.2	Erronkak	21
5.3.3	Datuetan oinarritutako metodoen literatura azterketa	22
5.4	Adimen Artifiziala	24
5.5	<i>Machine learning</i>	24
5.6	<i>Machine Learning</i> Sailkapen-Teknikak	26
5.6.1	<i>Shallow Learning</i> Teknikak	27
5.6.2	<i>Deep Learning</i> Teknikak	32
5.7	Softwarea	36
5.7.1	Datuen generaziorako softwarea	36
5.7.2	Datuen analisi eta prozesamendurako softwarea	37
6	Alternatiben analisia	41

6.1	Datuen sorkuntzarako softwarea	41
6.2	Datuen prozesamendu eta modeloen entrenamendurako softwarea . . .	42
6.3	<i>Machine Learning</i> modeloen aukeraketa	42
7	Soluzioaren deskribapena	45
8	Metodologia	47
8.1	Akats datuen sorkuntza	47
8.1.1	Sistema elektrikoaren modeloa	47
8.1.2	Simulazioak	57
8.2	Datuen aurreprozesaketa	59
8.2.1	Datuen normalizazioa	60
8.2.2	Datuen etiketatzea	60
8.3	Ezaugarrien azterketa	61
8.3.1	Datuen transformazioa	62
8.3.2	Transformatuaren aplikazioa	67
8.3.3	Ezaugarrien erauzketa	69
8.3.4	Ezaugarrien hautaketa	72
8.4	Detekzio-modeloa	73
8.4.1	Modeloen entrenamendua	73
8.4.2	Modeloaren aukeraketa	74
8.4.3	Modeloaren hiperparametroen doiketa	75
8.5	Sailkapen modeloa	77

8.5.1	Modeloaren aukeraketa	77
8.5.2	Modeloaren hiperparametroen doiketa	78
8.6	Eraitzen balioztatzea	78
9	Planifikazioa	81
9.1	Zereginen deskribapena	81
9.2	<i>Gantt</i> diagrama	84
10	Alderdi ekonomikoak	87
10.1	Barne orduan	87
10.2	Amortizazioak	87
10.3	Gastuak	88
10.4	Kostu totala	88
11	Ondorioak	89
11.1	Etorkizunerako lan-ildo posibleak	90
	Erreferentziak	91
	I. Eranskina	97
	II. Eranskina	101
	III. Eranskina	103
	IV. Eranskina	107

Irudien Zerrenda

2.1	Elektrizitaterik ez duten pertsona kopurua.	5
2.2	Tentsio sinple eta konposatuak.	6
5.1	Transformadoreen nukleo-motak.	15
5.2	Linea elektrikoaren akats-motak.	19
5.3	Akats-korronteak.	20
5.4	ML tekniken sailkapena.	24
5.5	ML tekniken sailkapena.	27
5.6	DT algoritmoen ohiko egitura.	27
5.7	KNN algoritmoaren irudikapena, non $K=3$	28
5.8	SVM algoritmoaren irudikapena.	29
5.9	RF baten diagrama.	31
5.10	GBT baten diagrama.	32
5.11	MLP baten diagrama.	33
5.12	CNN baten egitura.	34
5.13	RNN baten egitura.	35
7.1	Soluzioaren diagrama.	45
8.1	<i>Simulink</i> modelo elektrikoaren diagrama.	47
8.2	10kHz maiztasuneko PWM sorgailu baten tentsioak LC filtroaren aurretik eta ostean.	48

8.3	HIF arku elektrikoaren ereduaren diagrama.	55
8.4	<i>High Impedance Fault</i> (HIF) akatsen korrontek.	56
8.5	Seinale baten <i>Fourier</i> transformatua, non $t = 4$ segundoan c fasean lurrerako akatsa gertatu den.	63
8.6	Seinale baten <i>Wavelet</i> transformatua, non $t = 4$ segundoan c fasean lurrerako akatsa gertatu den.	64
8.7	Transformatuen maiztasun eta denbora espektroen sentsibilitatea.	65
8.8	n mailako seinale baten deskonposaketa.	66
8.9	<i>Mother Wavelet</i> familia desberdinak.	66
8.10	Seinale baten <i>Wavelet</i> transformatua, non $t = 4$ segundoan c fasean lurrerako akatsa gertatu den.	68
8.11	Seinale baten <i>Wavelet</i> transformatua, non $t = 4$ segundoan c fasean lurrerako akatsa gertatu den.	69
8.12	seinale baten <i>Wavelet</i> koefizienteetatik lortutako ezaugarriak.	71
8.13	<i>Cross-validation</i> metodoaren eskema.	74
9.1	Lanaren <i>Gantt</i> diagrama.	85

Taulen Zerrenda

6.1	<i>Shallow</i> eta <i>Deep Learning</i> modeloen arteko konparaketa.	43
8.1	<i>Three-Phase Source</i> blokearen parametroak.	49
8.2	<i>Three-Phase Transformer (Two Windings)</i> blokearen parametroak.	50
8.3	<i>Three-Phase V-I Measurement</i> blokearen parametroak.	50
8.4	<i>Distributed Parameters Line</i> blokearen parametroak.	51
8.5	<i>Three-Phase Transformer (Two Windings)</i> blokearen parametroak.	52
8.6	<i>Three-Phase Series RLC Load</i> blokeen parametroak.	53
8.7	<i>Three-Phase Fault</i> blokearen parametroak.	54
8.8	HIF modeloaren parametroak.	56
8.9	Akats eta ez-akats gertaera desberdinak.	58
8.10	Simulazio parametroak.	58
8.11	Datuen formatua.	61
8.12	Ezaugarri desberdinekin <i>XGBoost</i> modeloan lortutako zehaztasunak.	70
8.13	Ezaugarri desberdinekin <i>XGBoost</i> modeloan lortutako zehaztasunak.	72
8.14	Detekzio modelo desberdinen ebaluazio metrikak.	75
8.15	Detekzio modeloaren hiperparametro balio egokienak.	76
8.16	Sailkapen modelo desberdinen ebaluazio metrikak.	77
8.17	Sailkapen modeloaren hiperparametro balio egokienak.	78
10.1	3.1 Taula: Barne orduak.	87

10.2 3.2 Taula: Amortizazioak.	88
10.3 Taula 9.3: Gastuak.	88
10.4 3.4 Taula: Kostu totala.	88

Akronimoak

ANN *Artificial Neural Network*

APERT *Applied Electronics Research Team*

BPL *Backpropagation Learning*

BPTT *Backpropagation through time*

CNN *Convolutional Neural Network*

CWT *Continuous Wavelet Transform*

DC *Direct Current*

DER *Distributed Energy Resources*

DG *Distributed Generation*

DT *Decision Tree*

DWT *Discrete Wavelet Transform*

EFS *Exhaustive Feature Selection*

ESS *Energy Storage System*

FPGA *Field Programable Gate Array*

FPL *Flexible Power Link*

GBT *Gradient Boosting Trees*

GJH *Garapen Jasangarriko Helburuak*

GRU *Gated recurrent unit*

GSGIH *Global Smart Grids Innovation Hub*

HIF *High Impedance Fault*

KNN *K-nearest neighbor*

LSTM *Long short-term memory*

MAL *Master Amaierako Lana*

MG *Microgrid*

ML *Machine Learning*

MLP *Multilayer Perceptron*

NPC *Neutral Point Clamped*

PV *Photovoltaic*

RF *Random Forest*

RL *Reinforcement Learning*

RMS *Root Mean Square*

RNN *Recurrent Neural Network*

ROC AUC *Receiver Operating Characteristic Area Under Curve*

SG *Smart Grid*

SL *Supervised Learning*

SVM *Support Vector Machine*

UL *Unsupervised Learning*

1. Sarrera

Dokumentu honetan "Akats elektrikoaren detekzioa Adimen Artifizialen estrategia desberdinen bidez" Master Amaierako Lana (MAL) jasotzen da. Lehenengoz, lanaren testuingurua azaltzen da. Bertan, lan hau barne hartzen duen proiektuaren testuingurua azaltzen da. Ondoren, lanaren helburuak eta irismena azaltzen dira. Helburuen barnean, helburu nagusia, helburu zehatzak eta Garapen Jasangarriko Helburuak (GJH) azaltzen dira. Jarraian, helburu hauek betetzeko finkatu den lanaren irismena lantzen da. Gainera, lan honek ekar ditzakeen onura tekniko, ekonomiko eta sozialak azaltzen dira,

Artearen egoeran, sistema elektrikoaren analisia eta akats elektrikoaren detekzioaren inguruko egungo egoera aztertzen da. Metodo desberdinak, erronkak eta adimen artifizialeko tekniken erabilera aztertzen dira. Alternatibaren analisia, akats elektrikoaren datuen sorkuntzarako eta prozesamendurako erabilitako softwarearen analisia egiten da, baita modelo desberdinen hautaketa ere. Ondoren, akats elektrikoaren detekzioa eta sailkapena egiteko proposatutako soluzioa deskribatzen da. Gero, lan honetan erabilitako metodologia zehazten da, hala nola, akats datuen sorkuntza, datuen aurreprozesaketa, ezaugarrien azterketa, eta modeloen entrenamendua eta ebaluazioa.

Planifikazioan, lanean jarraitutako pausuak azaltzen dira, hauen deskribapena erabilitako baliabideak eta iraupena zehaztuz. Jarraian, lanaren alderdi ekonomikoak lantzen dira. Azkenik, lanaren ondorioak azaltzen dira, eta etorkizunerako zabaldu diren lan-ildoak deskribatzen dira.

2. Testuingurua

Lan hau *Global Smart Grids Innovation Hub* (GSGIH) zentroa eta *Applied Electronics Research Team* (APERT) ikerketa-taldearen arteko proiektu batetik eratortzen da. Proiektua, Potentzia-Elektronikarekin Kontrolatutako Sarea, *back-to-back* bihur-gailuen bidez bi sareen arteko konexioa burutzea bilatzen du, kontrol-teknika aurrera-tuak inplementatuz. Proiektuarekin, *Smart Grid* (SG) sareen inplementazioa bideratu eta sustatu nahi da.

2.1 *Global Smart Grids Innovation Hub* (GSGIH)

GSGIH, Iberdrola taldekoa, energia berriztagarrien eta sare adimendunen inguruan berrikuntza bultzatzeko sortu zen. GSGIHren helburua energia-sistemen digitalizazioa eta elektrifikazioa bultzatzea da, horrela, sare elektrikoen efizientzia eta erresilientzia hobetuz. Hub honek hainbat proiektu eta kolaborazio sustatzen ditu, ikerkuntza zentroekin, unibertsitateekin eta enpresa teknologikoekin lankidetzan, teknologia berriak garatu eta merkaturatzeko asmoz. GSGIHeK punta-puntako instalazioak eta baliabideak eskaintzen ditu, sektoreko liderrekin elkarlanean arituz, energiaren etorkizuna hobetzeko.

2.2 *Applied Electronics Research Team* (APERT)

APERT ikerketa-taldea Bilboko Ingeniaritza Eskolan (UPV/EHU) kokatuta dago eta potentzia-elektronika eta sistema digitalen arloan espezializatuta dago. Bere ikerketak larroek belaunaldi berriko eta ahalmen handiko *Field Programmable Gate Array* (FPGA) en erabilera barne hartzen dute, sistema digitalak zirkuitu integratu bakar batean integratzeko eta dispositibo hauen konfigurazio-ahalmena aprobetxatzeko. Honek komunikazio digitalen segurtasuna, fidagarritasuna eta komunikazio-sistemak hobetzea

biltzen ditu.

Potentzia-elektronikaren arloan, APERT-ek energia elektrikoaren sorkuntza, eraldaketa eta biltegitratze sistemak ikertzen ditu, bihurtgailu aurreratuak eta energia-iturri berriztagarriak erabiliz. Propulsiio-sistema eta ibilgailu elektrikoaren karga azpiegiturarako elektronikaren erabilgarritasuna eta ibilgailu elektrikoaren karga azpiegiturarako elektronikaren erabilgarritasuna ere lan egiten dute, errendimendua eta kontrola hobetuz. Azkenik, partikula azeleragailuetan eta sare adimendunetan (Smart grids) erabiltzen diren potentzia bihurtgailuen ikerketa ere egiten dute. Taldearen jarduerak berrikuntza teknologikoa eta garapen jasagarria sustatzen ditu, ingeniariak elektrikoaren arloan ezagutza sakonak erabiliz.

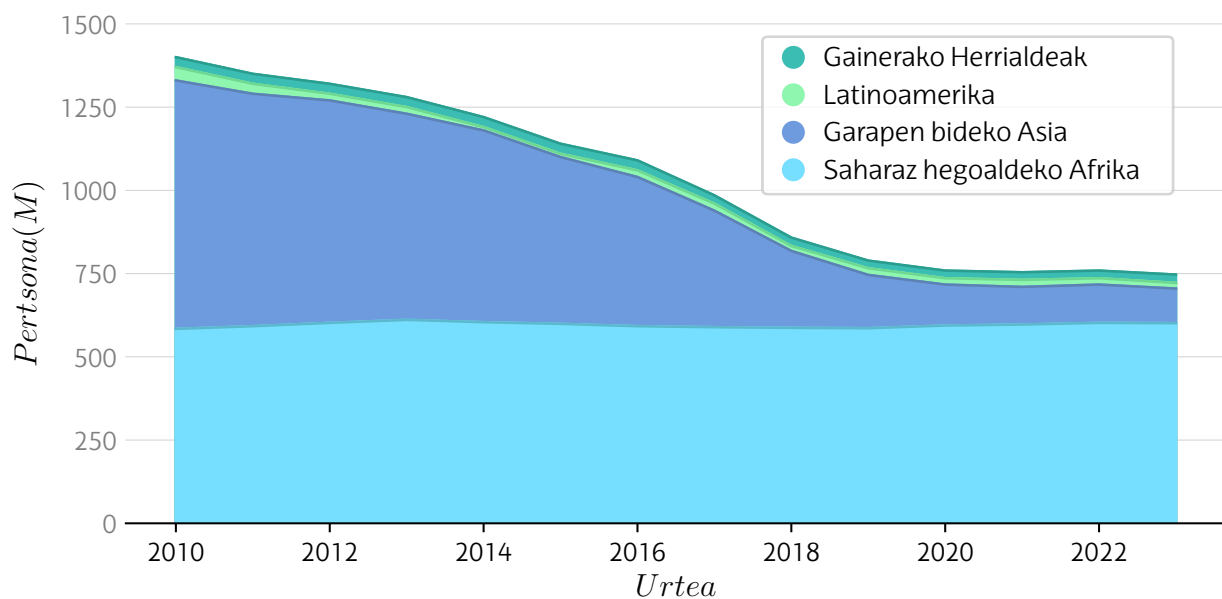
2.3 *Smart Griden* beharra

Azken hamarkadetan, elektrizitatearen eskari handipena eta energia berriztagarri gorabeheratsuen inplementazioa dela eta, sistema elektrikoek gainkarga eta estres baldintzei gero eta sarriago aurre egin behar diete. Gaur eguneko sare tradizionala ez da areagotzen ari den eskariari aurre egiteko eta energia-baliabide berriztagarriak eskala handian integratzeko gai.

Behar hauei aurre egiteko sare tradizionalaren egitura mantenduko balitz, sorkuntza-planta berriak eraiki eta sare elektrikoaren azpiegitura osoan funtsezko aldaketak egin beharko litzateke. Irtenbide hauek ez dira baliagarriak ekonomia-, segurtasun- eta ingurumen-arazoak kontuan hartuta. Ondorioz, sare tradizionalak baliabide banatuak integratzen dituen transformazio barneratzailea jasan beharko du [1].

Aldi berean, munduan elektrizitaterako sarbidea duten pertsonen zenbakia gero eta handiagoa izan arren, oraindik munduko biztanleriaren ehuneko hamarrak (745 milioi biztanle) ez dauka sarbiderik. Gaur eguneko erritmoarekin 2030 urterako 660 milioi biztanlek baldintza hauetan jarraituko dute (2.1 Irudia). Garapen Jasangarrirako 7. Helburua betetzeko joera hau aldatu beharko da [2].

Helburu eta aldaketa hauek bete ahal izateko *Microgrid* (MG) teknologia da apropo-



2.1 Irudia: Elektrizitaterik ez duten pertsona kopurua [3].

sen, energia sorkuntza lokala ahalbidetzen duelako banatutako baliabide berriztagarriak ustiatuz [4]. MGak eskala txikiko sistema elektrikoak dira, *Distributed Generation* (DG) edo sorkuntza sakabanatuko sistemaz, energia biltegitratzeaz eta hainbat kargaz osatuta dagoena.

Elektrizitatearen jarraitasuna bermatzeko, SG teknologiak integratu beharko dira, eskariaren erantzuna, energiaren eraginkortasunaren hobekuntza eta energia berriztagarriekin batera. Elementu hauek batera barneratu ahal izateko MGek rol garrantzitsua jokatzen dute, sorkuntza-planta berrietan inbertsio altuegiak egin barik energia-eskaintza altuagoak lortzea ahalbidetzen duelako [1].

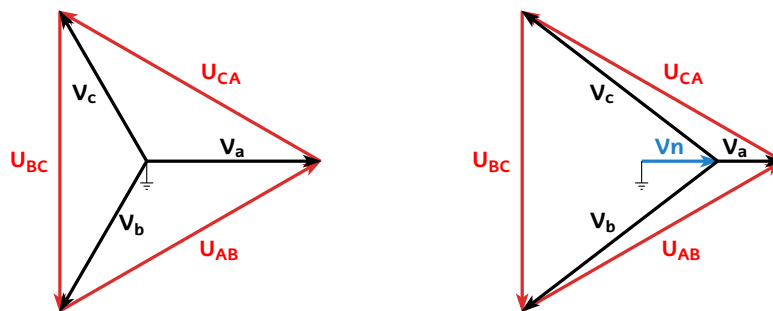
MGak bi modutan lan egin dezakete: sarera konektatuta edo saretik isolatuta. Sare-ra konektatuta lan egiten duenean, MGak sare nagusitik eta DG sorgailuetatik hartzen du energia. Modu isolatuan, MGak saretik deskonektatuta lan egiten du, potentzial-balantzea DG sorgailuekin mantenduz.

MG sareak generazio osagarri bezala erabili izan dira, hauen tentsio- eta potentzial-fluxu ezegonkortasunengatik. Ondorioz, konektatuta dauden linea nagusietan akatsak eman izan direnean ez dute modu isolatuan lan egiteko aukera izan. Ordea, MGen maila altuko integrazioaren bitartez, sare nagusiaren operazio modu eraginkor eta malguagoa eskaini dezakete.

2.4 RECON Proiektua

GSGIH zentroaren barnean, RECON proiektua garatzen ari da APERT eta i-DE-ren artean. Proiektuaren honen barnean hainbat helburu bilatzen dira. Lan hau, potentzia-elektronikaren kontrolaren bidez akats elektrikoak ezabatzeko helburua duen azpi-proiektuaren barnean dago.

Akats elektrikoaren ezabatzea, akatsa dagoen fasearen tentsioa murriztuz egitea birlatzen da. Elektrizitate hornidura mantendu ahal izateko, kargetan dauden tentsioak orekatuta mantendu behar dira. Beraz, fase baten tentsio maila murrizten den bitartean kargen tentsioak orekatuta mantendu ahal izateko, alde batetik, faseen arteko tentsio konposatuak konstante mantendu behar dira, eta beste aldetik, kargak tentsio konposatuekin elikatu behar dira (2.2 Irudia).



2.2 Irudia: Tentsio sinple eta konposatuak.

Baldintza hauek betetzeko, soilik fase baten tentsio-maila murriztu daiteke, eta ondorioz, soilik akats monofasikoak ezabatzeko erabili daiteke.

Azpi-proiektu honen inguruan, hainbat lan burutu dira. Lehenengoz, *back-to-back* bihurgailuaren kontrola garatu zen. Ondoren, plataforma experimental bat garatu zen kontrola aztertzeko. Azkenik, plataforma esperimentalean arku elektrikoak sortu ahal egiteko txartela elektronikoa bat garatu zen. Lan honetan, Adimen Artifizialeko estrategia desberdinen bidez, akats elektrikoaren detekzioa eta sailkapena burutuko da.

3. Helburuak eta Irismena

Atal honetan, lan honen helburuak zehaztuko dira, eta helburu hauek betetzeko lanaren irismena definituko da.

3.1 Helburuak

Helburuak hiru ataletan desberdinduko dira. Lehenik eta behin, lanaren helburu nagusia aurkeztuko da. Ondoren, finkatu diren helburu zehatzagoak ikusiko dira. Azkenik, lan honek jarraitzen dituen GJH-ak azalduko dira.

3.1.1 Helburu nagusia

Lan honen helburu nagusia sistema elektriko baten akats elektrikoak modu zehatz eta azkar batean detektatu eta sailkatzeko gai den Adimen Artifizialeko modelo bat garatzea da. Lanean aztertuko den sistema elektrikoa, i-DE proiektuan garatu den plataforma esperimentaleko sistema izanik.

3.1.2 Helburu zehatzak

Helburu nagusiaz gain, beste helburu zehatzagoak finkatu dira lan honetan:

- *Machine Learning* bidezko akats elektrikoen detekzio eta sailkapenari buruzko ikasketa.
- Akats elektrikoen datu-multzo bat lortzea.
- Modelo desberdinen entrenamendua, lortutako datu-multzoa erabiliz.
- Modelo desberdinen emaitzen ebaluazioa, haietatik hoberena hautatzeko.

3.1.3 Garapen Jasangarriko Helburuak (GJH)

Lan honetan hurrengo GJH jarraitzen dira:

- **GJH 7: Energia irisgarria eta ez-kutsatzailea.**

Aurretik aipatu den moduan, energia irisgarria eta ez-kutsatzailea lortzea ez da bideragarria energia sistema egitura tradizionalekin. Helburu hauek lortzeko, sorkuntza sakabanatu eta SG-etan oinarritzen diren energia sistemak integratu behar dira sarean. Lan honek, SG-ek dituzten zenbait erronkari soluzioa bilatzen die, hauen integrazioari lagunduz.

- **GJH 11: Hiri eta komunitate jasangarriak.**

Aurreko helburuaren harira, SG-ak hiri eta komunitate jasangarriak osatzeko funtsezkoak dira. SG-ek eskaintzen dituzten abantailak direla eta, komunitate energetikoen kontsumo- eta sorkuntza-kudeaketa modu eraginkor baten egin daitezke. Berriztagarriak diren energia iturriak barneratzen dituzte, komunitate energetikoen jasangarritasuna ahalbidetuz. Gainera, lan honen bidez, sistema elektrikoaren azpiegituraren erresilientzia hobetzen da, akats elektrikoaren detekzio azkarra eta zehatza proposatzen delako.

- **GJH 17+1: Hizkuntza eta kultura aniztasunaren bermea.**

17 helburuekin batera, UPV/EHU unibertsitateak hizkuntza eta kultura aniztasuna bermatzeko helburu gehigarri bat barne hartzen du. Lan hau, euskaraz garatu da, unibertsitatean euskararen erabilera sustatuz.

3.2 Irismena

Atal honetan lan honen irismena zehaztuko da. Aurretik finkatu diren helburuak lortu ahal izateko irismena lau ataza desberdinetan banatu da:

1. Hasierako ikerketa

- Akats elektrikoen detekzio eta sailkapen metodo desberdinen inguruko ikerketa, bereziki Adimen Artifizialarekin burutzen direnak.
- *Python*en Adimen Artifizialeko modeloak garatu eta aplikatzen ikasi.

2. Datuen generazioa

- Plataforma experimentalaren eta falta elektrikoen *Simulink* modelo baten garapena.
- Simulazioen bidez datu-multzoa sortzea.

3. Datuen prozesamendua

- Lortutako datuen aurreprozesamendua.
- Transformazioen bidez datuetatik ezaugarriak lortu eta aukeratu.

4. Modeloen entrenamendua eta ebaluazioa

- Hautatuko diren modelo desberdinen entrenamendua.
- Emaizten ebaluazioa eta konparaketa.

4. Proiektuaren onurak

Atal honetan, lanarekin lortutako onura tekniko, ekonomiko eta sozialak azaltzen dira. Onura hauek akats elektrikoak detektatu eta sailkatzeko Adimen Artifizialeko modelo bat garatu eta aplikatzearen emaitza dira.

4.1 Onura tekniko eta ekonomikoak

Ikuspuntu tekniko batetik, lortu diren onurak hurrengoak dira:

- **Sistema elektriko konplexuen akats detekzio zehaztasunaren hobekuntza:** Akatsen detekzio eta sailkapena datuetan oinarritutako metodoen bidez burutuz, akats elektrikoek sistema elektriko konplexutan dituzten patroiak antzeman ahal dira.
- **Sarearen operazioaren optimizazioa:** Akats elektrikoen detekzio eta sailkapen azkar baten bidez, akatsa ezabatzeko interbentzio optimoagoa egin daiteke. Gainera, RECON proiektuan proposatutako kontrolaren aplikazio azkarra ahalbidetzen du, etenik gabeko hornidura elektrikoa eskainiz.
- **Azpiegiturako kalteen prebentzioa:** Detekzio azkar batekin, sistema elektrikoan eman ahal diren kalteak murrizten dira.

Ikuspuntu ekonomikotik, hurrengo onurak jasotzen dira:

- **Etenaldietatik eratorriko kostuen murrizketak:** Proposatutako kontrolaren bidez, sistema elektrikoan ematen diren etenaldiak murriztu daitezke. Honekin batera, etenaldiei lotutako kostuak murrizten dira.
- **Mantentze kostuen murrizketa:** Akats elektrikoen sailkapen zehatz batekin, sare elektrikoen ikuskatze manualen beharra gutxitzen da, batez ere inpedantzia altuko akatsen kasuan.

4.2 Onura sozialak

Lanak ekartzen dituen onura sozialak ondoren azaltzen dira:

- **Hornidura elektrikoaren kalitate hobekuntza:** Akatsen detekzio eta interbentzio arin batekin, hornidura elektrikoaren egonkortasuna eta kalitatea hobetzen da. Batez ere, landa-eremuetako horniduran nabaria izan daiteke, egitura erradiala dela eta.
- **Sektore elektrikoaren ingurumen-inpaktu murrizketa:** SG-en inplementazioa bultzatuz, energia-iturri berriztagarrien erabilera ahalbidetzen da.

5. Artearen Egoera

Atal honetan, sistema elektrikoaren analisiari eta akats elektrikoaren detekzioari buruzko egungo egoera aztertuko da. Lehenik eta behin, sistema elektrikoaren funtsezko kontzeptuak ikusiko dira, eta ondoren, akats elektrikoaren detekzio metodoei buruzko ikuspegi orokorra emango da. Azkenik, Adimen Artifiziala eta *Machine Learning* erabiltzen dituzten tekniken garapena eta aplikazioa azalduko dira.

5.1 Sistema elektrikoaren analisia

Sistema elektrikoak energia elektrikoa sortu, garraiatu eta banatzeko erabiltzen diren sare eta azpiegituren multzoak dira. Ondoren, lan hau burutzeko funtsezkoak diren sareen elementuak eta kontzeptuak landuko dira.

5.1.1 Elementuak

Sorgailua

MG sareetan sakabanatutako energia-baliabideak edo *Distributed Energy Resources* (DER) erabili ohi dira energia-iturri gisa. DER iturrien artean teknologia desberdinak aurkitzen dira. Haietatik erabilienak DG berriztagarriak eta *Energy Storage System* (ESS) izanik. Ordea, DER iturri asko ez dira aproposak zuzenean sarera konektatzeko. Esaterako, *Photovoltaic* (PV) zelulak edo ESS-ak *Direct Current* (DC) energia sortzen dute, edo energia-kalitatean eta maiztasunean hobekuntzak behar dituzten aerosorgailuak. Kasu horietan, potentzia-elektronika behar da DER eta sare nagusiaren arteko konexioa burutzeko [5].

Flexible Power Link (FPL) potentzia-bihurgailuak, bi sare elektriko desberdinen arteko konexioa ahalbidetzen duen gailu elektronikoa da. FPLak bihurgailuaren bi noranz-

koetan potentzia aktiboa eta errektiboa transmititzeko aukera ematen du, sarearen potentzia-fluxua orekatuz eta sareko bi eremuetako tentsio-profilen kudeaketa baimenduz. Horri esker, sarearen egonkortasuna, kalitatea eta eraginkortasuna hobetu daitezke.

Proiektu honen testuinguruan, FPLa *back-to-back* potentzia bihurtailuan oinarritua dago. Potentzia-bihurtailua bi artezgailu/inbertsorez osatuta dago, korrante zuzeneko busetik lotuta.

Transformadorea

Transformadoreak, energia-iturrien tentsio eta sarearen tentsio-mailen arteko aldeagatik erabiltzen dira. Orokorrean, transformadoreek egoera orekatuetan lan egiten dute. Ez hori bakarrik, egoera ez-orekatuetan lan egin ez arren, transformadoreen egitura eta konexio-mota tentsio eta korronteak orekatzeko diseinatuta daude.

Ordea, aurretik ikusi denez, lurreko akats monofasikoa ematen denean, sistemak egoera desorekatuan lan egiten du. Kasu honetan, akats monofasikoa ematekotan egoera desorekatu hau mantenduko da, posible bada, akatsa zuzendu arte. Ondorioz, banaketa-sarean erabili ohi diren transformadoreak ez dira aproposak aplikazio honetarako.

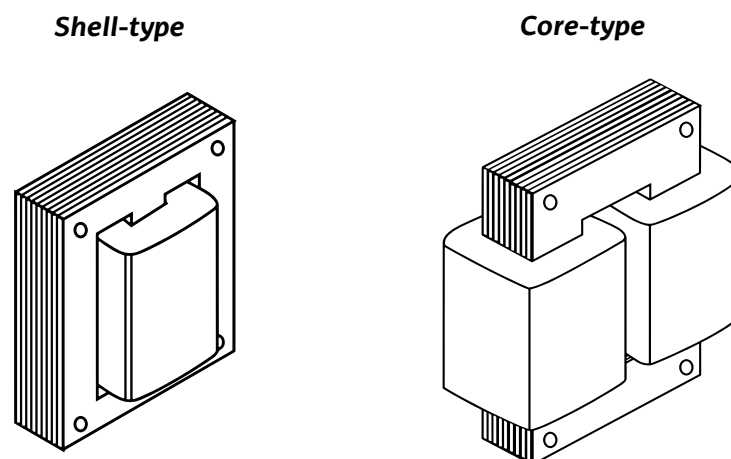
Egoera desorekatuetan sekuentzia homopolarra agertuko da transformadorearen kitzikapenean. Sekuentzia homopolarreko kitzikapen hauen aurrean transformadoreen portaera ohiko portaeraren oso desberdina izan daiteke, eta arriskutsua ere. Transformadorearen portaera egoera desorekatuen aurrean nagusiki harilkatuen konexioaren eta nukleo magnetikoaren konfigurazioaren arabera izango dira.

Alde batetik, harilkatuen konexioa YN motakoa bada, transformadoreak sekuentzia homopolarreko inpedantzia finitua aurkeztuko du, honi bidea emanez. Ordea, beste konexio-taldeek sekuentzia homopolarreko inpedantzia infinitua aurkezten dute, hau da, ez diote bidea ematen. Baieztapen hau ez da kasu guztietarako egia, sigi-saga konexioak dituzten transformadoreak sekuentzia homopolarreko inpedantzia finituak

dituztelako. Ordea, konexio-mota hauek ez dira aztertuko, lur-jartze transformadoreetan erabiltzen diren konexio-motak direlako [6].

Beste alde batetik, nukleo magnetikoa bi motakoa izan daiteke (5.1 Irudia): *shell-type* edo *core-type*.

- **Shell-type:** Mota honetako transformadoreetan nukleo magnetikoa izango da harilkatuak inguratzen dituena. Ondorioz, eremu magnetikoa harilkatuen inguruan egongo da.
- **Core-type:** Kasu honetan, transformadorearen harilkatuak nukleo magnetikoaren inguruan egongo dira. Transformadore trifasikoetan egitura desberdinak daude, nukleo magnetikoak dituen zutabe kopuruaren arabera. Zutabe kopurua 3, 4 edo 5 izan daiteke.



5.1 Irudia: Transformadoreen nukleo-motak.

Sekuentzia homopolarreko kitzikapenak ematen badira transformadorean, hiru faseen fluxuak ez dira haien artean deuseztatuko. Ondorioz, fluxu horrek bueltako bidea hiru faseen zutabeetatik kanpo egin beharko du.

Nukleo magnetikoa *shell* motakoa edo 4 edo 5 zutabeko *core* motakoa bada, fluxu homopolarren bueltako bidea nukleo magnetikotik egin ahalko du. Aurkezten duen inpedantzia homopolarra sekuentzia zuzeneko inpedantziaren antzekoa izango da [7].

Nukleo magnetikoa 3 zutabeko *core* motakoa baldin bada, ordea, fluxu homopolarra ezin izango da bueltatu nukleo magnetikotik. Beharrez, fluxua nukleo magne-

tikoaren goiko aldetik aterako da. Ondoren, aire edo olio tarte handi bat igaroko du. Transformadorearen tankearen geruzara helduko da, eta azkenik, nukleo magnetikoaren beheko aldera bueltatuko da fluxuaren ibilbidea itxiz, eta alderantziz [8].

Fluxua transformadorearen tankea zeharkatzen duenean bi fenomeno ematen dira: alde batetik, tankeak airea baino erreluktantzia baxuagoa izango duenez, fluxua tankeetik handitzeko joera izango du; beste aldetik, tankean *eddi* korronteak induzituko dira, fluxu magnetikora oposatuko direnak, tanketik zeharkatzen duen fluxua murriztuz. Ondorioz, eremu magnetikoa tankearen barneko hormetako lodiera txiki batera mugatuko da [9].

Hortaz, transformadore hauen sekuentzia homopolarreko inpedantzia desberdina izango da. Fluxu homopolarreko bueltako bideak erreluktantzia altua izango duenez, sekuentzia homopolarreko magnetizazio-induktantzia sekuentzia zuzenekoa baino askoz baxuagoa izango da, 0,5 eta 1,5 *p.u.* ordenakoa [10].

Linea elektrikoak

Linea elektrikoak sareetako generazio eta kontsumo nodoen arteko konexioak burutzen dituzten elementu pasiboak dira. Linea elektrikoak diseinatzerakoan hainbat alderdi hartu behar dira kontuan. Alderdi hauen arabera, linea elektrikoak modu desberdinetan sailkatzen dira.

Alde batetik, linea elektrikoak jasan dezaketen potentziaren arabera sailkatu ahal dira. Honen arabera, linea elektrikoaren tentsio maila finkatuko da, lineetan ematen den *Joule* efektua murrizteko. Espainia mailan, arautegiaren arabera, bi tentsio maila be-reizten dira: Goi-tentsioa (BOE-A-2014-6084) eta behe-tentsioa (BOE-A-2002-18099).

Bestalde, linea elektrikoak haien egituraren arabera sailkatu ahal dira: Sare erradialak, eraztun sareak eta sare mailadunak.

- **Sare erradialak:** Sare hauek sinpleenak dira, mutur batetik bakarrik elikatzen direlako. Haien abantaila nagusiak diseinu eta ustiapen erraza, kostu baxua eta babes-sistemen sinpletasuna dira. Ordea, akats elektrikoak ematen direnean

hornidura elektrikoa eten behar da, GSGIH proiektuan proposatzen den kontrol teknika aurreratua erabiltzen ez badira.

- **Eratzun sareak:** Bi muturretatik elikatzen dira, horniduraren jarraitutasuna eta operazio-malguagoa hobetzen dituzte.
- **Sare mailadunak:** Sare hauek konplexuenak dira. Sarearen edozein elementutan akatsak eman arren energia hornidura ez da eten behar.

Gainera, lineak airekoak edo lurpekoak izan daitezke. Aireko lineak distantzia handiko eta landa-eremuko elektrizitate transmisiorako erabiltzen dira. Lurpekoak, ordea, hiri-eremuetan eta ur azpiko transmisiorako erabiltzen dira. Haien arteko desberdintasun nagusienak instalazio mota eta material isolatzaileetatik etortzen dira.

Lina elektrikoak modelizatzeko orduan, metodo desberdinak erabili daitezke, hala nola: Pi zirkuitu baliokidea, lau poloen modeloa eta parametro banatuen modeloa.

- **Pi zirkuitu baliokidea:** Linea elektrikoaren adierazpen sinplea da, non linean serie inpedantzia eta paralelo admitantzien bidez modelizatzen den.
- **Lau poloen modeloa:** Modelo honek zirkuituaren sarrerako eta irteerako tentsioak eta korroneak erlazionatzen ditu, ABCD parametroen bidez.
- **Parametro banatuen modeloa:** Metodo honek linea elektrikoaren parametroak linearen luzeran banatuta daudela kontsideratzen du. Honela, linearen portaera zehatzago aztertzea ahalbidetzen du, batez ere linea luzetan.

5.1.2 MGen operazio moduak

Microgrid baten potentzia-bihurgailuek hiru modutan lan egin dezakete: *grid-feeding*, *grid-forming* eta *grid-supporting*.

1. **Grid-feeding:** Operazio modu honetan *microgrid*ak sare nagusira energia ematen diote, potentzia aktiboa zein erreaktiboa injektatuz. Potentzia-bihurgailua

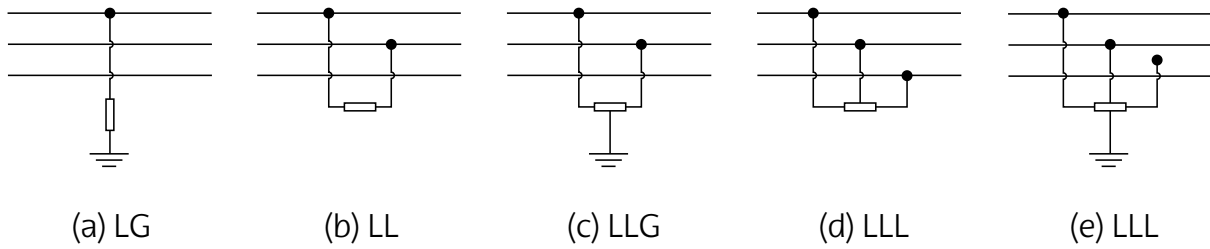
korrante-bihurgailu ideala eta paraleloan inpedantzia altua bezala adierazi daiteke.

2. **Grid-forming:** Sarearen tentsioa eta maiztasuna finkatzen dute. Seriean inpedantzia baxua duen tentsio-sorgailu batekin adierazi daiteke. Orokorrean, operazio modu hau *microgrida* irla moduan lan egiten duenean erabiltzen da. Beste *grid-forming* bihurgailuekin lan egin ahal izateko, sinkronizazio-maila altuak behar dituzte, haien artean zirkuitulaburrik ez emateko. Etenik gabeko elikatze-iturri bezala edo tentsio-maiztasun erreferentzi bezala erabili daitezke.
3. **Grid-supporting:** Aurreko bi operazio moduen modura irudikatu daiteke, paraleloan inpedantzia altua duen korrantea iturri bezala edo seriean inpedantzia baxua duen tentsio iturri bezala. Orokorrean, sarearen tentsio anplitudea eta maiztasuna kontrolatzen dute haien korrantea eta tentsioa erregulatuz. Aurreko bi operazio moduak bezala kontrolatu daiteke. Ordea, hauen helburu gorena sarearen tentsio eta maiztasuna kontrolatzea izango da.

5.2 Akatsak sistema elektrikoan

Akats elektrikoak sistema elektrikoaren egoera anomaloak dira, tentsio primarioetan jarduten duten potentzia-sistemen elementuen akats elektrikoetatik eratorriak [11]. Akats hauek desberdinak izan daitezke eta hainbat modutan sailkatu daitezke [12]:

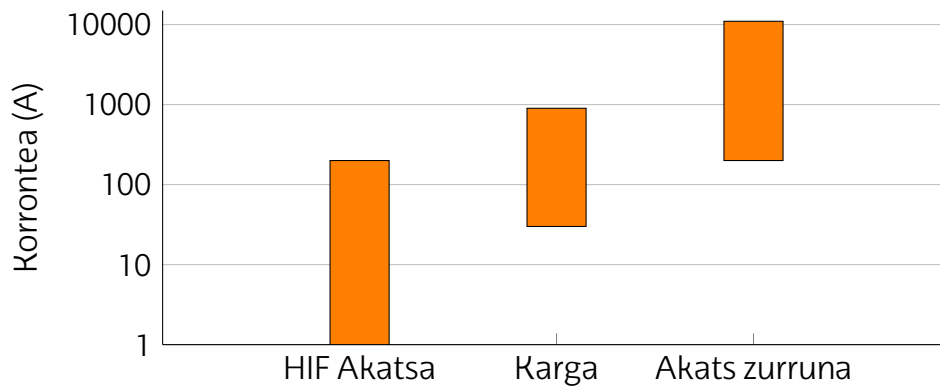
- **Serie edo paralelo:** Akatsak zirkuitu irekien (serie) edo zirkuitu laburren (paralelo) ondorioz eman ahal dira. Zirkuitu irekiak orokorrean eroaleen apurketengatik ematen dira, akats faseen korrante zirkulazioa etenez. Ordea, akats-mota hauek ez dira oso ohikoak eta ez dira proiektu honetan aztertuko [13].
- **Lurra eta fase kopurua:** Akatsak faseen eta lurraren artekoak edota faseen artekoak izan daitezke. Beraz, akatsak fase bakarrekoa lurrera (monofasikoa, LG), bi faseen artekoa (bifasikoa, LL), bi faseen artekoa lurrera (bifasikoa lurrera, LLG), hiru faseen artekoa (trifasikoa, LLL) edo hiru faseen artekoa lurrera (trifasikoa lurrera, LLLG) izan ahal dira (5.2 Irudia).



5.2 Irudia: Linea elektrikoaren akats-motak.

Akats suntsikorrenak akats trifasikoak dira, haien akats-korronteak handienak direlako. Gainera, akats simetriko bakarrak dira, sistema elektrikoa orekatuta mantenduz. Horregatik akats trifasikoak izaten dira zirkuitulaburreko ikasketetarako kontsideratzen direnak. Ordea, akats trifasikoak ezohikoenak dira, banaketasareen akatsen %5 inguru izanik. Akats monofasikoak akatsik ohikoenak dira eta akatsen %80 osatzen dute.

- **Jatorria:** Akats elektrikoak hainbat faktoreren ondorio izan daitezke, hala nola, baldintza meteorologikoen, zuhaitzekin kontaktuen, ekipoetako arazoaren edo giza faktoreen ondorio. Arrazoi meteorologikoak ohikoenak izaten dira, eta haien artean, tximistak, haize bortitzak, izotza metatzea, suteak eta uholdeak daude. Zuhaitzek eragindako akatsak ere ohikoak dira.
- **Korrontea:** Akats elektrikoak akats korronte baxukoak edo altukoak izan daitezke. Korronte hau sistemaren lur-jartzearen eta akats-motaren arabera izango dira. Batetik, sistemaren lur-konexioa inpedantzia handikoa, erresonantzia edo isolatua bada, akats korronteak txikiak izango dira. Bestetik, akats elektrikoak inpedantzia altukoak direnean HIF-ek akats korronte oso baxuak sortzen dituzte (5.3 Irudia). Horrek zaildu egiten du akatsen detekzioa eta lokalizazioa. Lur jartzea zurruna edo inpedantzia baxukoa denean, akats-korronteak magnitude altukoak dira.
- **Iraunkorrak edo iragankorrak:** Akats elektriko iraunkorrak sistemaren elementuren bat kaltetu denean gertatzen dira. Akats hauek ematen direnean, orokorrean, hornidura elektrikoa etetea eragiten dute. Akats elektriko iragankorrak, ordea, ez dute sistemaren elementurik kaltetzen modu iraunkorrean. Akats



5.3 Irudia: Akats-korrenteak [14].

hauek ematen direnean, sistema elektrikoa denbora labur baten eteten bada, akatsa ezabatu daiteke.

5.3 Akats elektrikoen detekzioa

5.3.1 Metodoak

Akats elektrikoak detektatzeko hainbat metodo daude, bi talde nagusitan sailka daitezkeenak: Modeloetan oinarritutako teknikak eta datuetan oinarritutako metodoak [15].

- **Modeloetan oinarritutako metodoak:** Metodo hauetan sistema elektrikoa-ren modelo elektrikoa erabiltzen da. Modelo honen bitartez, ebaluatutako aldagaien neurketak modeloaren balioekin konparatzen dira, haien arteko aldea aztertzeke. Metodo hauen barnean, falten detekziorako teknika desberdinak erabili daitezke.
- **Datuetan oinarritutako metodoak:** Metodo hauek datuen azterketa sakon-ean oinarritzen dira. Sistemaren datuak xehetasunez aztertzean, sarrerako aldagaien eta sistemaren egoeraren arteko erlazioak identifika daitezke. Behar bezala jarduteko, metodo horiek datuen bolumen esanguratsuak behar dituzte. Be-reziki egokiak dira sistema konplexuak aztertzeke, eredu hauen formulazio zeha-tza lortzea zaila izan daitekeelako. Metodo hauen artean, ondoren aztertuko di-ren Adimen Artifizialeko teknikak aurkitzen dira. Metodo hauen lan-fluxua hiru

pausutan laburbildu daiteke:

- Lehengoz, akats elektrikoak barne hartzen dituen datu-multzo bat lortu edo sortu behar da.
- Ondoren, datu gordinen prozesaketa burutu behar da.
- Azkenik, detekziorako modeloa aukeratu, garatu eta balioztatu behar da.

5.3.2 Erronkak

SG bidezko DG sistemen integrazioak zenbait erronka aurkezten ditu falten detekzioan [15],[16]:

- **Akats-korrontearen aldakuntza:** Sistema elektrikoetan, DG sistemen presentziaren ondorioz, sisteman inpedantzia berri bat agertzen da, Z_{DG} . Inpedantzia berri honen ondorioz, sistema elektriko nagusiak akats-korrontera ekartzen duen i_{grid} korronea ez du portaera lineala izango. MGak sare nagusira konektatuta edo sare nagusitik isolatuta lan egiten dutenean, zirkuitu laburreko korronteen balioak aldaketa esanguratsuak dituzte. Sarera konektatuta daudenean, akats-korronteak 10 - 50 *p.u.* baliokoak dira. Ordea, isolatuta daudenean, akats korronteak sorgailuen teknologiaren arabekoak izango dira. Makina birakarietan oinarritutako DG sistemak akats-korronte handiak ematen dituzte. Aldiz, inbertso-reetan oinarritutako DGen akats korronteak 1 - 2 *p.u.* balioetara mugatuta daude. Hau dela eta, gainkorronte-babesgailuak ez dira beti aktibatzen.
- **Noranzko biko fluxua:** Generazio-, karga- eta topologia-aldaketak direla eta, akats-korronteen noranzkoa aldatu daiteke. Ondorioz, noranzkoan oinarritutako sistema erradialen babesgailuek ez dute behar bezala lan egiten.
- **Nahi gabeko isolatzea:** DG sistema bat sare nagusitik isolatzen denean eta nahi gabe kargak elikatzen jarraitzen dituenean, nahi gabeko isolamendua eman dela esaten da [17]. Akatsak ematen direnean, nahi gabeko isolamenduak ematen badira, DG sistemak akatsa elikatzen jarraituko dute.
- **Desarra faltsua edo jarraitzailea:** DG sistema baten alboko sistema baten

akatsa ematen denean, DGen babesgailuak akatsa detektatzea eta sistema isolatzea gerta daiteke.

- **Gain- eta azpi-irismena:** Lehen aipatu den moduan, i_{grid} korrontearen balioa murrizten da DG sistemen presentzian. Sistema hauek sarera konektatuta edo saretik isolatuta lan egiten dutenez, i_{grid} balioa aldakorra izango da. Korronte honen balioa txikiagoa denean, akatsaren zonaldea babesten duten babesgailu guztiek ez dute akatsa detektatuko. Korrontea handiagoa denean, babesgailuen zonaldetik kanpoko akatsak detektatuko dituzte.
- **Birkonexio-arazoak:** Birkonexio-babesgailuak, fusibleekin batera lan egiten dutenean, orokorrean fusibleak baino arinago operatzeko koordinatzen dira. Horrela, akatsa iragankorretan fusibleak galtzea lortu daiteke. Hala ere, DGak ekartzen duen topologia eta akats-korronte aldaketaren ondorioz, babes-elementuen deskoordinazioa ematen da [18].

Erronka hauen ondorioz, MGen integrazio egokirako babes eskema berriak garatzea eta implementatzea beharrezkoa da. Babes eskema hauek MGen lan modu desberdinei egokitu beharko dira, sistemaren elementu desberdinen arteko komunikazioa funtsezkoa rola hartuz; horra hor SGen garrantzia. SGen datu guzti hauen kudeaketa eta analisirako *Big Data* edo datu handien analisisen teknikak aplikatu beharko dira.

5.3.3 Datuetan oinarritutako metodoen literatura azterketa

Azken urteetan, datuetan oinarritutako metodoak gero eta ohikoagoa dira akats elektrikoaren detekzioa burutzen dituzten artikuluetan. Lehen azaldu den moduan, metodo hauek hiru ataletan banatzen dira.

Lehenengo pausua akats elektrikoaren datu-multzoa lortzea da. Akats elektrikoak sistema elektrikoaren gertaera urriak dira. Honen ondorioz, akats elektrikoak dituzten sistema elektriko errealean datuak lortzea zaila izan daiteke. Ikerketa gehienek *Simulik* bidez lortutako datu-multzoak erabiltzen dituzte [19],[20],[21],[22]. Ordea, zenbait artikulutan datu errealek erabiltzen dira [23],[24]. Datu errealek erabiltzen dituzten artikulua as-

ko, *kaggle* webgunean aurkitzen den *VSB Power Line Fault Detection Kaggle Competition* datu-multzoa erabiltzen dute [25].

[24] artikuluan azaltzen den moduan, simulazioen bidez edo fabrikatutako datuetatik lortutako datu-multzoak erabiltzea ikerketen aplikagarritasuna murrizten du, metodo hauekin lortutako datuak datu errealak dituzten arazoak erreplikatzeko ez dituztelako. Datu errealak, laginketa-periodo aldakorak, falta diren datuak eta zarata, besteak beste, izan ahal dituzte. Ordea, datu errealak oso eskasak edo hurbilgaitzak direnez, ikerketa gehienek simulazioen bidez lortzen dituzte datuak.

Jarraitu behar den hurrengo pausua, datuen prozesaketa eta transformazioa da. Transformazio honen bitartez, datuetatik ezaugarriak erauzten dira. Ikerketetan, metodo asko erabiltzen dira pausu honetan. Batzuetan, zuzenean neurtutako korrante eta tentsioen irakurketa datuak edo *Root Mean Square* (RMS) balioak erabiltzen dituzte [19], [20], [23]. Ikerketa hauen artean bi multzo desberdindu daitezke; alde bateatik, datu-multzo sinpleak dituzten ikerketak, modelo sinpleak erabiltzen dituztenak; eta beste aldetik, datu-multzo konplexuak eta modelo konplexuak erabiltzen dituztenak.

Datu-multzo sinpleak, datuen klaseek haien arteko desberdintasun handia dituztenak dira. Akats elektrikoaren testuinguruan, sistemaren gertaera gutxi kontuan hartzen dituzten datu-multzoak dira. Ikerketa askok soilik akats-mota, akats-erresistentzia eta akats-distantzia desberdinak kontuan hartzen dituzte. Hala ere, ez dituzte operazio modu desberdinak, HIF, beste motako iragankorrak, desorekak, etab. kontuan hartzen.

Ordea, beste ikerketa askotan datuen transformazioak aplikatzen dira ezaugarriak lortzeko, batez ere HIF akatsak barne hartzen dituztenak. Haietatik erabilienak *Fourier* Transformatua [26], [27] eta *Wavelet* transformatuak dira [22], [28], [29], [30].

Azkenik, lortutako ezaugarriak erabiliz *Machine Learning* modeloa entrenatu behar da. Ikerketetan erabiltzen diren modeloak anitzak dira. Ordea, gehien erabiltzen dena *Artificial Neural Network* modeloak dira [19], [20], [21], [26], [28]. Bestalde, bestelako hainbat modelo erabiltze dira, hala nola *Space Vector Machine* [24], *Decision Tree* [29] edo *Convolutional Neural Network* modelo konplexuagoak [23]. Orokorrean, modelo guztiekin lortzen da zehaztasun altu bat.

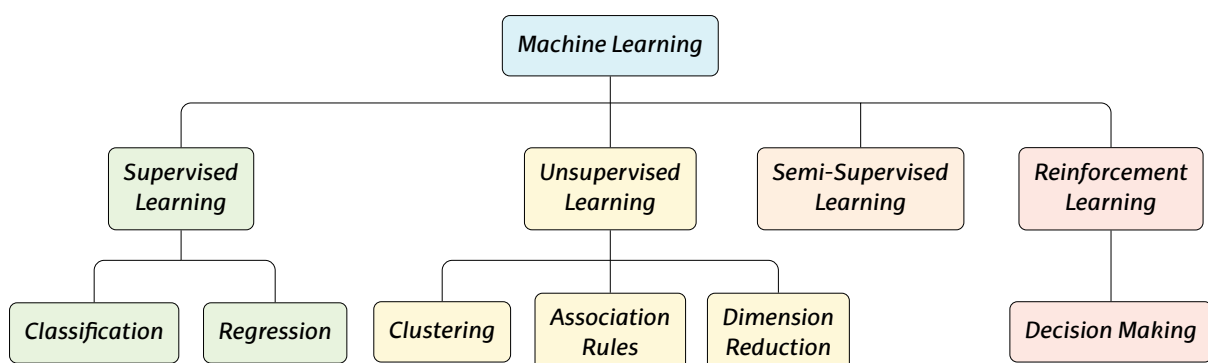
5.4 Adimen Artifiziala

Adimen Artifiziala informatikako arloa bat da, gizakien adimena behar duten zereginak burutu ditzaketen sistemak sortzea du helburu, hala nola ikaskuntza, arrazonamendua eta pertzepzioa [31].

5.5 Machine learning

Machine Learning (ML) teknikak sistemei esperientziatik automatikoki ikasteko eta hobetzeko ahalmena ematen dien Adimen Artifizialeko aplikazioak dira, betetzen dituzten funtzioetarako esplizituki programatuta egon barik. Ikasketa prozesua datuen behaketarekin hasten da, non datuak adibideak, esperientzia zuzena edo aginduak izan ahal diren. Datu hauetan ML algoritmoak patroi esanguratsuak bilatzen ditu, garrantzitsuak ez diren patroiak baztertuz [32],[33].

ML tekniken barruan lau talde desberdindu ahal dira: Ikasketa gainbegiratu, edo *Supervised Learning* (SL); ikasketa ez-gainbegiratu, edo *Unsupervised Learning* (UL); ikasketa erdi-gainbegiratu edo *Semi-supervised Learning*; eta errefortzu bidezko ikasketa edo *Reinforcement Learning* (RL) (5.4 Irudia) [34].



5.4 Irudia: ML tekniken sailkapena.

- **Supervised Learning:** Lehenengo talde honetan, ML algoritmoek gainbegiratu-tako eta etiketatutako datuekin lan egiten dute. Algoritmoaren helburua *input* edo sarrera eta *output* edo irteeren arteko erlazioak ikastea da. Ikasketa hau algoritmoen entrenamenduaren bitartez lortzen da, non entrenamendurako eti-

ketatutako datu talde bat erabiliko den. Datu talde honetan *inputak* neurtuko diren datuak edo ezaugarriak dira, eta *outputak* gainbegirale batek input instantzia bakoitzari finkatutako etiketak dira. Etiketa hauek algoritmotik espero diren emaitzak izango dira. Entrenamenduan, algoritmoak bere parametroak doitzen ditu iragarpenen eta etiketa errearen arteko aldea minimizatzeko, horrela, behar bezala orokortu ditzake etiketatu gabeko datu berriak.

Talde honen barruan bi mota desberdindu daitezke:

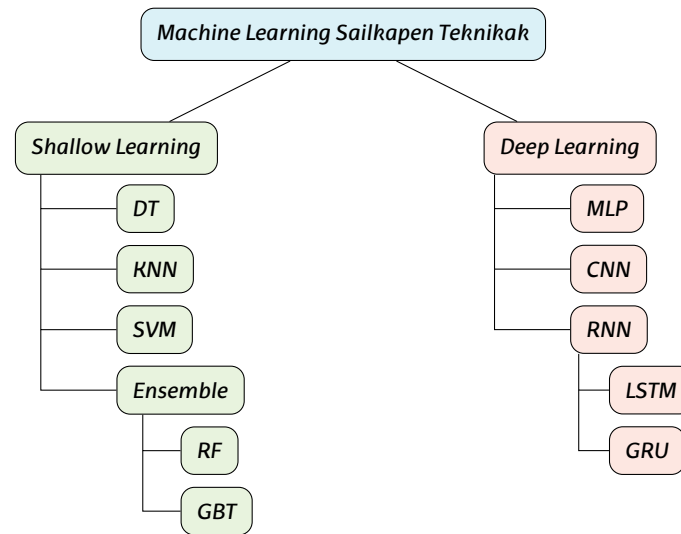
- **Classification:** Sailkapen algoritmoak ezaugarri espezifikoetan oinarritutako taldeetan datuak kategorizatzeko erabiltzen dira. Talde horiek etiketa edo datu-instantzia bakoitzari esleitutako klaseen bidez definitzen dira. Modelo hauetan *inputak* sailkatu nahi den instantziaren ezaugarriak dira, eta *outputa* instantziaren kategoria. Sailkapen problemak etiketa bitarrak izan ditzakete, non datuak bi kategoriatan banatzen diren, edo maila anizkoitzetan, non bi kategoria posible baino gehiago dauden.
- **Regression:** Erregresio-algoritmoak aldagai independenteen (edo ezaugarri) eta menpeko aldagai baten arteko erlazioa ezartzeko erabiltzen dira. Aldagai independenteak ereduaren *inputak* dira; mendeko aldagaia, berriz, aurreikusi nahi den *outputa*. Erregresioaren helburu nagusia da, ereduaren entrenamenduan erlazioa ezarri ondoren, aldagai independenteen datuak erabiliz menpeko aldagaiaren balioa aurreikustea da. Orokorrean denbora serie baten hurrengo balioak aurreikusteko erabiltzen direnez, modelo hauek datu-multzo sekuentzial edo tabularrekin erabili ohi dira.
- **Unsupervised Learning:** Hurrengo taldeko modeloak entrenamendu datuak etiketa edo klase gabekoak direnean erabiltzen dira. ML algoritmo hauen helburua datuen artean ezkutuko patroiak aurkitzea da, datu taldearen azterketaren bidez datuei buruz gehiago ikasi ahal izateko. SL kasuaren kontrari, datuak etiketa gabekoak direnez eta gainbegiralerik ez dagoenez, algoritmo hauekin lortutako emaitzak ez dira zuzenak ez okerrak. Haien artean hiru talde desberdindu daitezke:
 - **Clustering:** datu-multzoetan ezaugarri antzeko taldeak definituz lan egiten

dute taldekatze edo Clustering algoritmoek.

- *Association Rules*: Datuen ezaugarrien arteko erlazioak aztertzeko erabiltzen dira asoziazio-araua edo Association Rule algoritmoak.
- *Dimension Reduction*: Datuen dimentsio-murrizketak egiteko erabiltzen dira, datuen irudikapena, ezaugarri ikasketa edo beste ML algoritmoen zehaztasuna eta errendimendua hobetzeko. Dimentsio murrizketa ahal den neurrian datuen ezaugarrien informazioa galdu gabe egiten da [31].
- ***Semi-supervised Learning***: Talde honetako algoritmoak datuak etiketadunak eta etiketa gabekoak direnean erabiltzen dira. Datu etiketatuak modeloaren kategoriak ikasteko erabiltzen dira, eta etiketa gabekoak, ordea, kategoria haien arteko ugak hobeto definitzeko erabiltzen dira. Algoritmo hauekin lortutako zehaztasuna SL algoritmoekin lortutakoa baino altuagoa da [34].
- ***Reinforcement Learning***: Azken talde honetan, ML algoritmoek ingurunearekin elkar eragiten dute, ekintzen bidez akatsak edo sariak lortzeko. Probak eta akatsak eginez, algoritmoek portaera sarituenak ikasten dituzte haien errendimendua maximizatzen.

5.6 *Machine Learning* Sailkapen-Teknikak

Akats elektrikoaren detekzioarako ML sailkapen teknika edo algoritmo desberdinak erabili daitezke. Entrenamendurako erabiliko diren datuak etiketadunak izaten direnez, aztertuko diren teknikak SL motakoak dira. Algoritmo hauek bi multzotan sailkatu ahal dira: Azaleko ML teknikan oinarritutako metodoak (edo *Shallow Learning*) eta ML teknika sakonetan oinarritutako metodoak (edo *Deep Learning*) (5.5 Irudia).

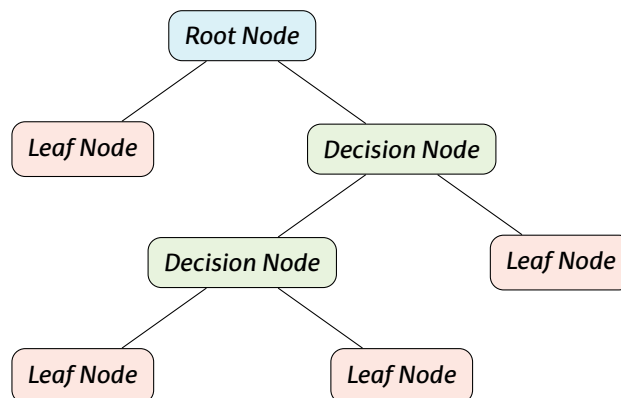


5.5 Irudia: ML tekniken sailkapena.

5.6.1 Shallow Learning Teknikak

Decision Tree (DT)

Decision Tree (DT) edo erabaki-zuhaitzen modeloak, izenak esaten duen moduan, zuhaitz hierarkikoen egitura hartzen dute. DT modeloek datuak sailkatzen dituzte datuen ezaugarriari baldintza logikoak ezarriz. Baldintza hauek erabaki nodoetan ezartzen dira, eta hauetatik adar bi edo gehiago ateratzen dira. Hasierako erabaki nodoa *Root* edo sustrai nodoaren izena hartzen du, eta bukaerako nodoak *leaf* edo hosto nodoak dira (5.6 Irudia).



5.6 Irudia: DT algoritmoen ohiko egitura.

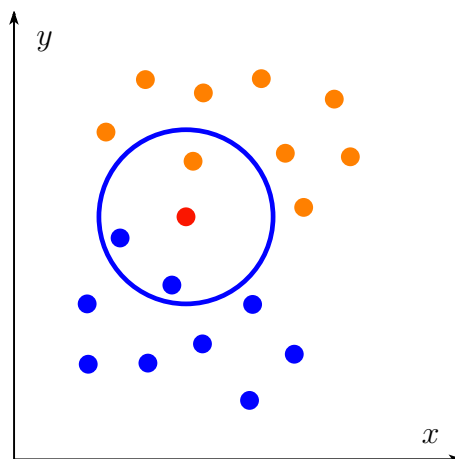
DT modeloen egitura ez dago finkatuta. Entrenamenduan zehar zuhaitzen adarrak hasten dira problemaren konplexutasunaren arabera. Entrenamendurako algoritmo desberdinak erabili daitezke, hala nola ID3, C4.5, eta J48 algoritmoak [33].

K-Nearest Neighbor

K auzokide hurbilena edo *K-nearest neighbor* (KNN) gainbegiraturako ML sailkapen-eta erregresio-teknika sinplea da. Algoritmo ez-parametrikoa da, hau da, ez ditu entrenamendu datuen distribuzioan azterketarik egiten. KNN algoritmoek datu berrien klasea hurbileko entrenamendu datuen klasearen arabera determinatzen dute. Datuen hurbiltasuna haien arteko distantzia neurketen arabera definitzen da, hala nola distantzia euklidearraren arabera (5.1 Ekuazioa). Teknika hau bere sinpletasunagatik eta bere konputazio denbora murriztagatik erabiltzen da.

$$d(x_j, x_t) = \|x_j - x_t\|^2 \quad (5.1)$$

Algoritmoan etiketa gabeko datu berri bat sartzen denean, KNN bi operazio burutzen ditu. Lehenengoz, datu berriaren K puntu hurbilenak aztertzen ditu. Horretarako, puntu berriaren eta entrenamendu datu guztien arteko distantzia euklidearra kalkulatzen da. Ondoren, hurbileko puntu horien klasea erabiliz, datu berriaren klasea ezartzen da (5.7 Irudia). Algoritmo honen erabilerarako, funtsezkoa da entrenamendu datuen normalizazioa.



5.7 Irudia: KNN algoritmoaren irudikapena, non $K=3$.

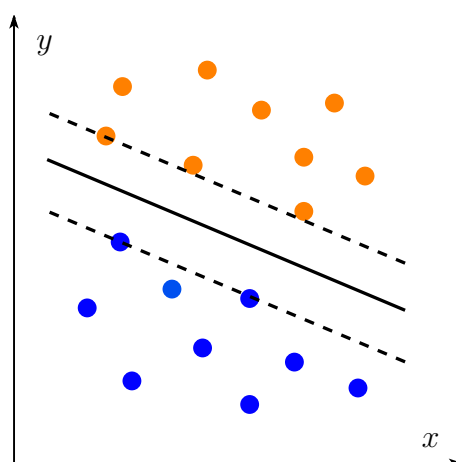
Algoritmoaren zehaztasuna K parametroaren arabera izango da. K parametroa txikia, edo unitatearen baliokoa denean, zehaztasun handiagoa lortuko du entrenamendu datuekin. Ordea, modu honetan algoritmoa gaindoituta egongo da eta honen zehaztasuna neurketen zaratarekiko oso sentikorra izango da. Ondorioz, K balio altuagoak gomendatzen dira, klaseen arteko mugen definizioa galdu arren.

Beste alde batetik, entrenamendu datu-multzoa handiegia denean, KNN modelook sailkatzaile eraginkor batek baino biltegitatze memoria handiagoa behar du. Baita, datu berrien klasea aurreikusteko denbora handiagoa egiten da datu-multzo handiekin.

Modelo honen desabantailak murrizteko KNN modeloaren aldaerak erabili daitezke, hala nola *Weight Adjusted KNN* edo *KNN with K-Means* algoritmoak [35].

Support Vector Machine

Euskarri bektoredun makinak edo *Support Vector Machine* (SVM) sailkapenerako eta erregresiorako erabiltzen den gainbegiratutako ML teknika bat da. SVM algoritmoak bi datu klase desberdin banatzen dituen hiperplano (edo erabaki muga) optimoena bilatzen du. Hiperplanorik optimoena, hiperplanoaren eta bi kasetako datuen arteko distantzia maximizatzen duena izango da, gaizki sailkatuta dauden puntuen kopurua minimizatzen duen bitartean (5.8 Irudia).



5.8 Irudia: SVM algoritmoaren irudikapena.

Datuak ez-linealak baldin badira, SVM ez-lineala aplikatu daiteke. Algoritmo honekin sarrerako datuak "ezaugarri espazio" deituriko dimentsio handiagoko espazio baten transformatzen dira. Espazio berri honetan datuak linealak bilakatzen dira, non SVM lineala aplikatu daiteke. Problema ez linealak ebazteko gaitasun hau SVM algoritmoen indar nagusia da. Gainera, SVM algoritmoen entrenamenduarekin beti lortzen da soluzio global bat, hau da, honen optimizazioa ez da minimo lokal batera doitzen, baizik eta minimo globalera.

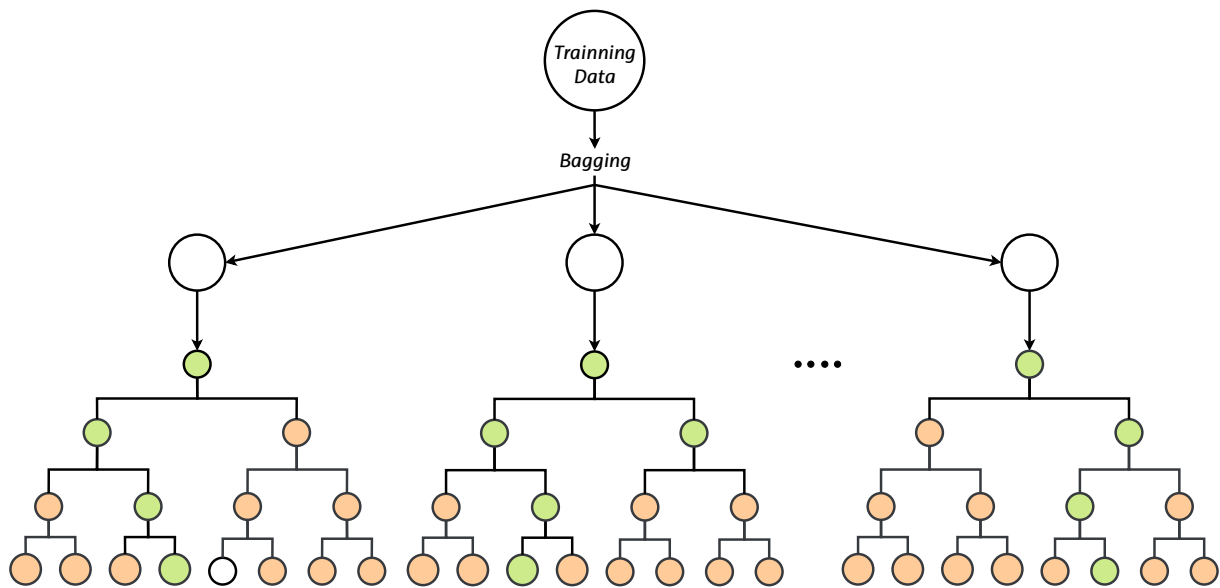
Ordea, problema ez-linealak ebazteko ahalmena baita haien ahultasun nagusia da [36]. Alde batetik, lortutako modelo ez linealak ez dira interpretagarriak, eta kutxa beltzen modura hartzen dira. Beste alde batetik, problema linealizatzeko *kernel* egokia hautatzea zaila izan daiteke. *Kernel* hauek lineala, polinomiala, RBF edo MLP izaten dira. Era berean, entrenamendu denbora beste modeloekin alderatuta altua izan daiteke, batez ere entrenamendu datu-multzo handiekin.

Random Forest

Random Forest (RF) algoritmoa *bootstrap aggregation* edo *bagging* teknikan oinarritutako ikasketa multzokatuko edo *ensemble* teknika da, datuen sailkapen edo erregresiorako erabili daitekeena. RF algoritmoak DT algoritmoetan ematen den gaindoitzea gutxitzen du problema berdinerako hainbat zuhaitz eraikiz.

Bagging multzokatze teknikarekin, entrenamendu datuen *bootstrap* datu-multzoak sortzen dira. datu-multzo hauek jatorrizko entrenamendu datuen zorizko datuak hautatzen dituzte, jatorrizko datuen dimentsioak mantenduz. Haien desberdintasun nagusia jatorrizko datuekin bikoiztutako datuak dituztela da. Behin n *bootstrap* datu-multzo sortu direla, n DT eraikitzen dira datu-multzo horiei doitu. Ondoren, eraikitako n DT algoritmoek datu berrien klasea definitzen da, DT guztien artean datuaren klasea bozkatuz (5.9 Irudia).

Bagging prozedurarekin modeloaren errendimendua hobetzen da modeloaren bariantza murrizten delako, honen *bias* edo alborapena handitu gabe [37]. DT bakarra erabilita, lortutako modeloaren gaindoikuntza handia da. Ondorioz, modeloaren



5.9 Irudia: RF baten diagrama.

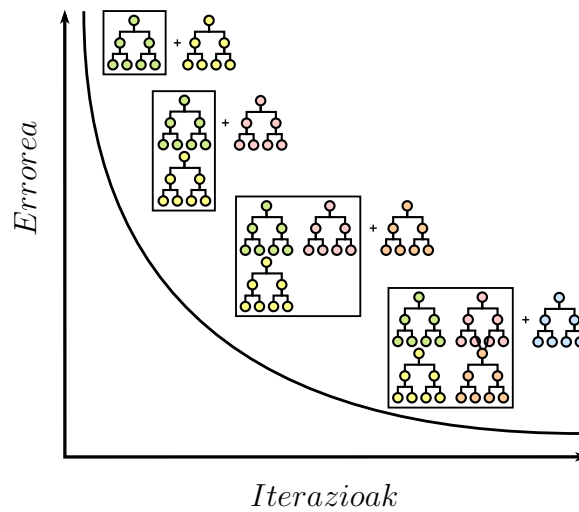
zehaztasuna sarrera datuen zarataren menpekotasun altua daukate. Ordea, haien artean korrelazio ez duten hainba zuhaitz erabilita, menpekotasun hau asko murrizten da. Bootstrap laginketaren bitartez zuhaitzen arteko korrelazioa murriztea lortzen da.

Gradien Boosting Trees

Gradient Boosting Trees (GBT) *boosting* tekniketako oinarritutako ikasketa multzokatu metodoa da. *Boosting* teknikaren helburua da, zehaztasun gutxiko eta sinpleak diren arau orokor askoren konbinazioaren bitartez, iragarpen modelo oso zehatza sortzea [38].

Bagging teknikaren aldera, non aldi berean DT guztiak doitzen dira paraleloan, *boosting* teknikarekin entrenamendu pausu bakoitzean sailkatzaile bat gehitu eta doitzen da. Sailkatzaile bakoitzaren entrenamendurako jatorrizko entrenamendu datuen azpimultzo bat erabiltzen da. Entrenamendu pausu bakoitzean DT sailkatzaileak gehitzen dira, aurreko zuhaitzen errorea murrizteko helburuarekin (5.10 Irudia).

GBT algoritmoekin lortutako zehaztasuna oso altua izaten da. Ordea, modelo hauek gaindoitzeko joera daukate, hauen errendimendua datuen zarataren menpekoak izanik. GBT modeloen desabantailak murrizteko eta hauen eraginkortasuna hobetzeko, aldaera desberdinak garatu dira, *XGBoost*, *LightBoost* edo *CatBoost* esaterako.



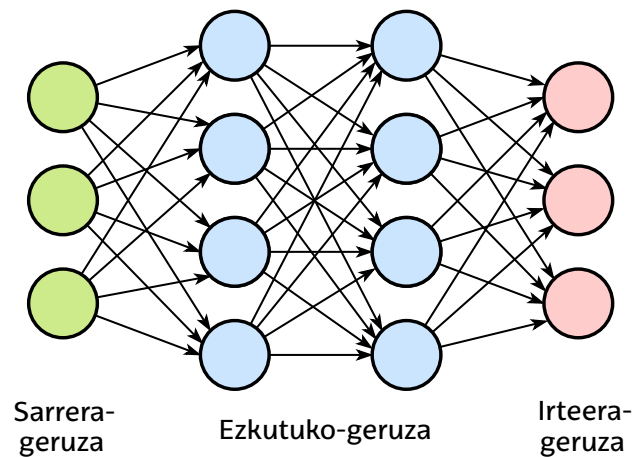
5.10 Irudia: GBT baten diagrama.

5.6.2 Deep Learning Teknikak

Multi Layer Perceptron

Multilayer Perceptron (MLP) guztiz konektatutako hainbat geruza dituen *feedforward* neurona-sare artifiziala edo *Artificial Neural Network* (ANN) da. MLP modeloa hiru geruza nagusiz osatuta dago: sarrera-geruza, gutxienez ezkutuko-geruza bat eta irteera-geruza. Alboko geruzak elkarren artean guztiz konektatutako neuronak dituzte. Hala ere, geruza bereko neuronak ez daude elkarri konektatuta, ezta haien buruari ere.

Sarrera-geruzaren tamaina sarrerako datuen araberakoa izango da. Irteera-geruzaren tamaina problemaren araberakoa da: MLP modeloa datuak sailkatzeko erabiltzen bada, irteera-geruzaren tamaina klaseen kopurua izango da; eredu erregresioa egiteko erabiltzen bada, irteera-geruzak neurona bakarra egongo da. Ezkutuko-geruzaren tamaina problemaren konplexutasunaren araberakoa izango da. Hau askotan saiakuntza eta errorearen bitartez determinatzen da. Ezkutuko- eta irteera-geruzako neuronak aktibazio funtzioak dituzte (5.11 Irudia).

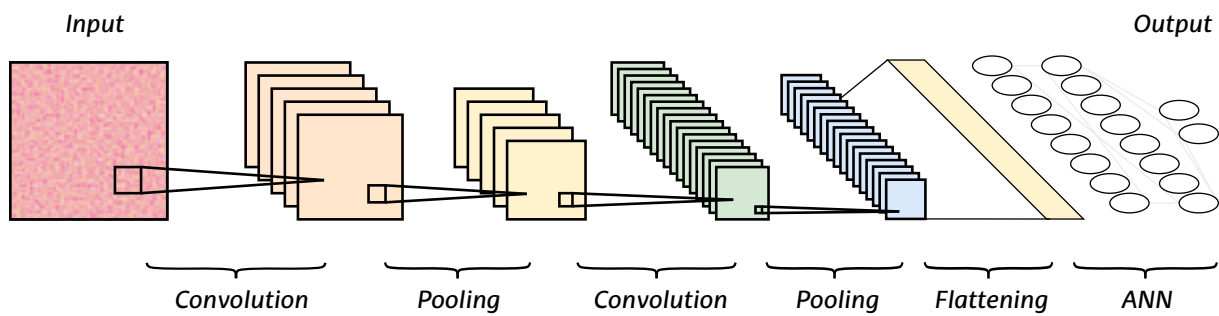


5.11 Irudia: MLP baten diagrama.

MLP modeloen entrenamenduan *Backpropagation Learning* (BPL) algoritmoa erabiltzen da. BPL algoritmoa oso erabilia den algoritmo iteratiboa da. Iterazio bakoitzean BPL benetako irteeraren eta aurreikusitako irteeraren arteko diferentzia kalkulatu du, eta ondoren MLP modeloaren geruza bakoitzeko pisuak eta alborapenak aldatzen dira. Honen bidez, kalkulatuako errorea minimizatuz, MLP modeloaren ikasketa prozesua egiten da. Orokorrean, BPL algoritmoan gradiente jaitsiera optimizazio teknika erabiltzen da.

Convolutional Neural Network

Convolutional Neural Network (CNN) *feedforward* neurona-sarea da, metodo konboluzionalak aplikatuz, sarrera datuetatik ezaugarriak atera ditzakeena. CNN algoritmoak datu-multzoen irudien prozesamendu eta analisiak egiteko erabiltzen dira. Ordea, seinaleak baita erabili daitezke sarrera datu gisa. CNN algoritmoak hainbat geruzatan banatuta dago. Geruza hauek, orokorrean, hurrengoak dira: konboluzionala, multzokatze edo *pooling* geruza, guztiz konektatutako geruza eta irteera geruza (5.12 Irudia).



5.12 Irudia: CNN baten egitura.

- **Konboluzionala:** Konboluzio geruza, sarrerako datuak prozesatzeko erabiltzen da. Datu hauetatik, X_i sarrera datuen eta F_j konboluzio filtroen artean konboluzio operazioa aplikatuz, y_j ezaugarriak ateratzen dira. Geruza honen irteera ezaugarri mapa bezala ezagutzen da. Honen tamaina, sarrera datuen tamainaren eta konboluzio filtroaren araberakoa da.

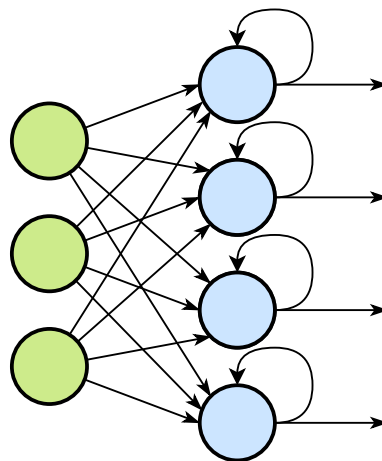
$$y_j = \sigma \left(\sum F_j * X_i + B_j \right) \quad (5.2)$$

Non σ aktibazio funtzioa eta B_j alderapenak diren.

- **Multzokatzea:** Multzokatze edo *pooling* geruza, konboluzio geruzan lortutako ezaugarri maparen tamaina murrizteko erabiltzen da. Ezaugarri mapak ezaugarri asko ditu, gaindoitze arazo batera eramanez. Ondorioz, ezaugarri hauen tamaina murrizten da konpresioaren bitartez. Honetarako, bi multzokatze teknika erabili daitezke: *max pooling*, non multzokatze geruzan lortutako ezaugarri maparen maximoak hartzen diren; eta *average pooling*, non ezaugarri maparen batezbestekoak hartzen diren.
- **Gutziz konektatutako geruza:** Gutziz konektatutako geruza, aurreko geruzetatik lortutako datuen ezaugarriak prozesatzeko erabiltzen den neurona-sarea da. Azken multzokatze geruzan datuak dimentsio bakarreko bektore gisa ateratzen dira, *flattening* prozesuaren pausuaren bitartez. Bektore hau guztiz konektatutako geruzaren sarrera izango da.
- **Irteera geruza:** Irteera geruza, guztiz konektatutako geruzaren irteera datuak sailkatzeko erabiltzen den geruza da. Geruza honen tamaina sailkapen klase kopuruaren araberakoa izango da.

Recurrent Neural Network

Recurrent Neural Network (RNN) neurona-sarea, sekuentzia datuen edo denbora serieen prozesamendu eta analisiak egiteko erabiltzen da. *Feedforward* neurona-sareekin alderatuta, RNN sareetan neurona geruzak ez daude soilik hurrengo geruzetara konektatuta. Geruza berdineko neuronak, haien irteera datuak berriro sarrera datu gisa erabiliz, geruza berdineko neuronekin eta haien buruarekin konektatuta daude. Beraz, RNN sareen irteera, sarrera datuen eta sarearen aurreko egoeraren menpekoa izango da (5.13 Irudia). Honek, RNN sareei memoria izateko gaitasuna ematen die. RNN sareak normalean *Backpropagation through time* (BPTT) erabiltzen dute entrenamendu prozesuan. BPTT metodoa erabiltzerakoan, *vanishing and exploding gradients* arazoak eman daitezke [39]. Arazo hauen saihesteko, RNN algoritmoen aldaerak erabiltzen dira, *Long short-term memory* (LSTM) eta *Gated recurrent unit* (GRU) algoritmoak.



5.13 Irudia: RNN baten egitura.

LSTM algoritmoen esker, neurona sareak gai dira aurreko uneko eta honen aurreko uneetako sarearen egoeraren datuak erabiltzeko eta datu horietatik zein erabili erabakitzeko. LSTM algoritmoek hiru atez osatuta daude: ahazteko atea, sarrera atea eta irteera atea.

GRU algoritmoa LSTM algoritmoaren aldaera oso erabilia da. GRU algoritmoak ez dauka egoera gelaxkarik, bere egitura bi atez osatuta dago: eguneratze atea eta ete-naldi atea. Hortaz, GRU algoritmoen egitura sinpleagoa da eta prozesamendu denbora murriztagoa dute [40].

5.7 Softwarea

Akatsak detektatzeko erabiliko den modeloa garatzeko, laburbilduz, bi urrats jarraitu beharko dira. Alde batetik, datu-multzo bat sortu beharko da, sistema elektrikoa-ren egoera normala eta akats egoera jasoko dituenena. Bestalde, datu horiek prozesatu eta ML ereduak entrenatu beharko da. Pausu bakoitzerako, software desberdinak erabili ahalko dira.

5.7.1 Datuen generaziorako softwarea

Datuen generaziorako, sistema elektrikaren modelizazioa eta simulazioa egiteko gaiten softwarea beharko da. Ezaugarri hauek eskaintzen dituzten hainbat software daude. Haien artean, hurrengoak aztertuko dira: MATLAB, PowerFactory eta PowerWorld.

MATLAB

MATLAB programazio eta zenbakizko kalkulurako ingurunea da. Eragiketa matematikoak egiteko, datuak bistaratzeko eta arazo konplexuak ebazteko aukera ematen du. Bestalde, *Simulink*, *MATLAB*en luzapen gisa, bloke diagramen bidez sistema dinamikoak modelatzeko eta simulatzeko erabiltzen da. Bereziki erabilgarria da kontrol-sistemak diseinatzeko, industria-prozesuak aztertzeko eta gehiago. Bi tresna horiek aplikazio ugari eskaintzen dituzte, eta funtsezkoak dira ikertzaileentzat eta ingeniariarentzat.

Simulink ingurunean sistema elektrikoen modelizaziorako *Simscape Electrical* paketea erabiltzen da. Pakete honen barruan *Specialized Power Systems* liburutegiak sistema elektrikoetarako espezifikoki garatutako osagaiak eta teknologia eskaintzen ditu [41],[42].

PowerWorld

PowerWorld potentzia-sistemen simulaziorako zabalki erabili izan den softwarea da. Software hau, batez ere, energia kontsumoa aurreikusteko eta etenaldiak baretzeko

erabiltzen da. Beste aplikazio batzuk sistemaren potentzia fluxu optimoaren simulazioa eta animazioa da. Software honen populartasuna, beste softwareen artean, potentzia fluxuen, tentsio eta korronteen egoeraren animaziorako gaitasunagatik eman da. Honi esker, sistemaren egoera konplexuak begirada batekin ulertzea baimentzen du. Gainera, potentzia fluxuen analisirako eta akats egoeren azterketarako tresna ahaltzua da [43],[44].

PowerFactory

DigSILENT PowerFactory sistema elektrikoaren simulazio eta analisirako erabiltzen den softwarea da, hainbat sistemen modeloak eta kontrolak erantsiak dituen. Oso ezaugarria da energia industriaren transmisio, banaketa, sorkuntza eta merkatu sistemen simulazioak eta analisiak egiteko duen gaitasunagatik. *PowerFactory* softwareak modelizazio integraleko ingurune bat eskaintzen du, analisi-ahalmen ugari dituen. Besteak beste, gai da potentzia fluxu, zirkuitulaburren eta harmonikoaren analisiak egiteko, egonkortasun zein iragankortasunen simulazioak egiteko, egoera egonkorrean zein desorekatuetan. Erabiltzeko erraztasunean eta malgutasunean oinarrituta, *PowerFactory* tresna baliagarria da energia-sistemen plangintzaren eta eragiketen errorei aurre egiteko. *DigSILENT* (DPT) programazio lengoaiari esker azterketen eta funtzionalitate desberdinen automatizazioa burutu daiteke [45],[46].

5.7.2 Datuen analisi eta prozesamendurako softwarea

Datuen analisirako eta prozesamendurako, eta ML modeloaren entrenamendurako baita ere, software multzo desberdin bat aztertuko da.

Python

Python maila altuko helburu orokorreko programazio-lengoaia da. *Python*en erabilerak, datuen zientziaren arloan, inoiz baino altuagoa da, bereziki aske dauden tresna eta liburutegiaren artean. *Python*, data zientzia komunitatean, programazio lengoaiarik era-

biliena bilakatu da, batez ere, honen erraztasun erlatiboagatik, hainbat liburutegiz osatutako duen ekosistema handiagatik eta algoritmo anitz zuzenean aplikatzeko duen gaitasunagatik.

Pythonen, testuinguru honetan erabiltzen diren liburutegiak hurrengoak dira. Alde batetik, datuen irakurketetarako orokorrean *Pandas* erabiltzen da. Datuak aurreprozesatzeko *NumPy* eta *SciPy* liburutegiak erabiltzen dira. Hauek n dimentsioko *array*etan operazioak egiteko eta algoritmo matematikoak aplikatzeko erabiltzen dira. Datuen irudikapenerako *Matplotlib* eta *Plotly* erabiltzen dira. ML aplikazioetarako *scikit-learn* da liburutegi erabiliena. Azkenik, *Deep Learning* aplikazioetarako *Google Brain* taldeak garatutako *TensorFlow* liburutegia erabiltzen da, eta honen inguruan eraikitako liburutegi, *Keras*. *PyTorch*, *Facebook*ek garatua, aplikazio berdinetarako erabiltzen ari den liburutegia da [47].

R

R prozesala, funtzionala eta objektuetan orientatutako programazio lengoia da. Milaka pakete ditu eskuragarri, *I/O*, datuen analisia, datuen irudikapenak eta bestelakoak burutu ahal izateko. *Pythonekin* batera, datuen zientzien arloan *R* programazio lengoia erabilienetakoa da [48].

R lengoian, datuen analisirako *tidyverse* paketea erabiltzen da. ML algoritmoak aplikatzeko *h2o*, *ranger*, *xgboost* eta abar erabiltzen dira. ML modeloen interpretaziorako *iml*, *pdp*, *vip* paketeak erabiltzen dira, besteak beste [49]. *R* lengoian, *TensorFlow* eta *keras* erabiltzeko aukera dago, baita [50].

MATLAB

MATLAB, inguru akademikoan oso erabilia da algoritmoen sorkuntza eta konputaziorako. *MATLAB*en, ML-rako erabiltzen den liburutegia *Statistics and Machine Learning Toolbox* deiturikoa da. Honen bitartez, ML algoritmo gehienak aplikatu daiteke datuen deskribapena, analisi eta modelizaziorako. Bestalde, *Deep Learning Toolbox* liburute-

gien bitartez, *Deep Learning* algoritmoak erabili daitezke. Liburutegi honek, Gainera, *TensorFlow*, *PyTorch* eta *ONNX* modeloen inportazioa eta esportazioa ahalbidetzen du.

6. Alternatiben analisia

Atal honetan, sistema elektriko baten akats elektrikoaren detekzio eta sailkapenerako metodologia eta software desberdinak aztertuko dira, haien arteko aldeak, abantailak eta desabantailak kontuan hartuta.

6.1 Datuen sorkuntzarako softwarea

Proiektuaren plataforma esperimentalean, momentu honetan, ez da bideragarria entseguen bidez beharrezkoak diren datu guztiak lortzea. Horregatik, sistema elektrikoaren akatsak simulatzea eta datuak lortzea lan honetarako funtsezkoa den pausua da. Lan honetan hiru software desberdin kontsideratu dira pausu hau burutzeko: *MATLAB/Simulink*, *PowerWorld* eta *PowerFactory*.

Azkenik, **MATLAB/Simulink erabiltzea erabaki da** hurrengo arrazoiengatik:

- **Simulazioen beharrak:** Burutu beharko diren simulazioak sistema elektriko nahiko simple batetan burutuko dira, eta ez dira beste softwareak eskaintzen dituzten funtzionalitate aurreratuak beharko.
- **Proiektuaren aurretiko erabilera:** Proiektuan aurretik garatutako modelo elektriko guztiak *MATLAB/Simulink* ingurunean garatu dira.
- **ML teknikak implementatzeko gaitasuna:** Behar izanez gero, *MATLAB/Simulink* ingurunean *ML* modeloak entrenatu eta implementatzeko aukera dago.
- **Formakuntza denbora:** Software hau testuinguru akademikoetan maiz erabiltzen denez, lana bertan burutu ahal izateko beharreko formakuntza denbora murriztagoa izango da.



6.2 Datuen prozesamendu eta modeloen entrenamendurako softwarea

Datuen prozesamendurako hiru software edo programazio-lengoi aztertu dira: *MATLAB*, *Python* eta *R*. Pausu honetarako hirurak baliagarriak direnez, modeloen entrenamendurako erabiliko den software berdina erabiltzea erabaki da.

Modeloen **entrenamendu eta ebaluaziorako Python erabiliko dela erabaki da**. Hona hemen arrazoiak:

- **Erabiltzeko erraztasuna:** *Python*-en sintaxia argia eta irakurgarria da, eta horrek ikasteko eta erabiltzeko erraztasuna ematen dio, batez ere programazioan esperientzia gutxi dutenentzat.
- **Dokumentazioa eta tutoretzak:** Dokumentazio ugari eta kalitate handikoa dago eskuragarri, baita lineako ikastaroak eta tutoretzak ere.
- **Kode publiko eta laguntza:** *Python*-ek komunitate zabala eta aktiboa du, eta kode irekiko proiektu ugari daude eskuragarri. Gainera, laguntza-foroak, dokumentazio eta Adimen Artifizialeko programazio laguntzaile asko daude sarean eskuragarri.
- **Bateragarritasuna:** *Python* hainbat tresnekin eta sistema eragileekin ondo integratzen da. *TensorFlow*, *Keras* eta *PyTorch* bezalako liburutegi ezagunak ditu, eta horiek *Machine Learning* eta *Deep Learning* aplikazioak errazten dituzte.

6.3 Machine Learning modeloen aukeraketa

Lan honetan ML modelo desberdinak erabili eta ebaluatuko dira. Ordea, lehen ikusi den moduan, modeloen artean bi multzo bereizi daitezke.

Alde batetik, *shallow learning* modeloak aurkitzen dira. Modelo hauek sinpleagoak dira eta kostu konputazional murriztagoa dute. Hala ere, ez dira dimentsio handiko problemak edo datuetan patroi konplexuak antzemateko gai. Horregatik, modelo hauek

erabiltzean, datuen transformazio teknikak aplikatu beharko dira ezaugarriak datuetatik erauzteko.

Beste aldetik, *deep learning* modeloak daude. Modelo hauek aurrekoak baino konplexuagoak dira, eta ondorioz, kostu konputazional altuagoa dute. Ordea, honi esker datuetan patroi konplexuagoak bereizteko gai dira, datuetan egin beharreko transformazio eta ezaugarri erauzketa pausuak kenduz edo sinplifikatuz.

Bi modelo mota desberdinek metodologia desberdinak behar dituztenez, haietako mota bakarra aukeratu da. Hauetatik bat aukeratzeko, haien arteko konparaketa egingo da (6.1 Taula).

6.1 Taula: *Shallow* eta *Deep Learning* modeloen arteko konparaketa.

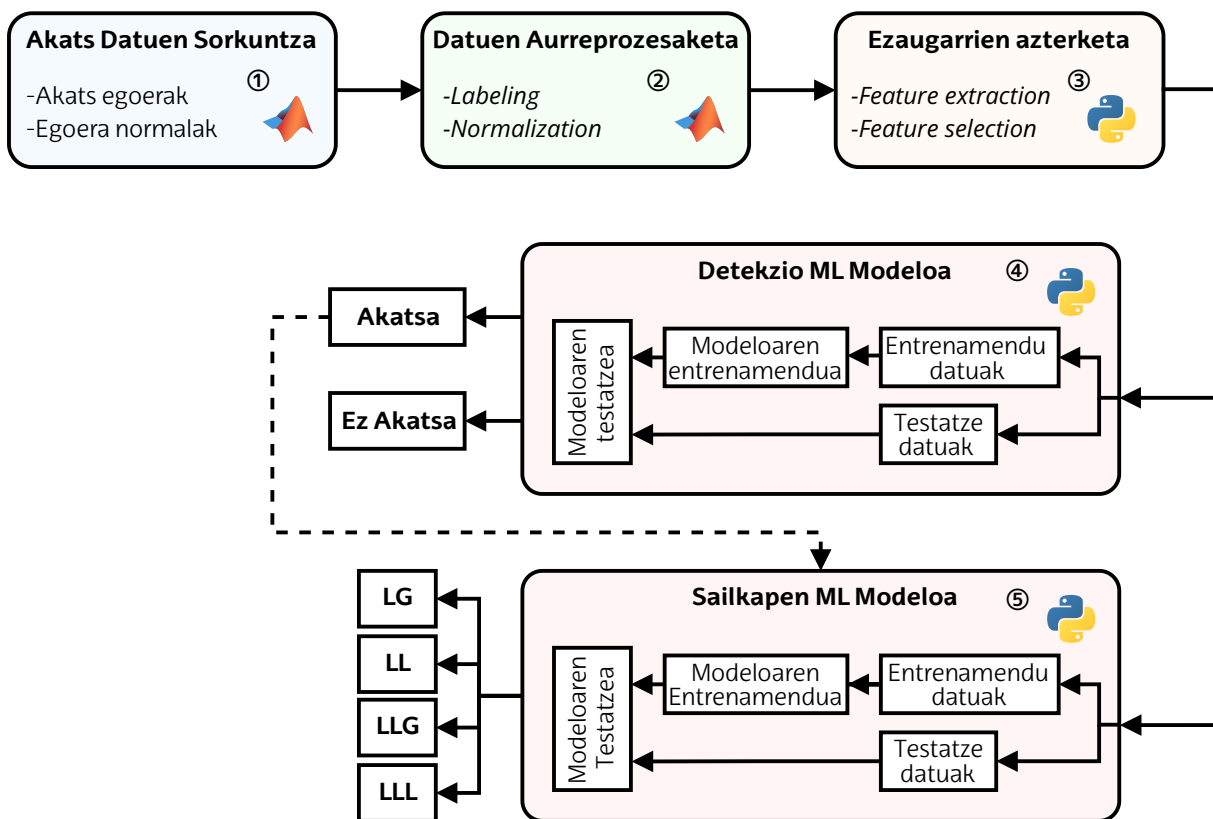
Irizpidea	Modelo Mota	
	Shallow Learning	Deep Learning
Konplexutasuna	Baxua	Altua
Kostu Konputazionala	Baxua	Altua
Dimentsio Handiko Datuak	Ez eraginkorra	Eraginkorra
Patroi Konplexuak Antzematea	Ez da gai	Gai
Kanpo ezaugarri erauzketa	Beharrezkoa	Hautazkoa
Entrenamendu Denbora	Murrizta	Luzea
Interpretagarritasuna	Erraza	Zaila

Metodo desberdinen konparaketa burutu ondoren, lan honetan ***shallow learning* modeloak eta datuen transformazio metodologia hautatu da**. Modelo hauek aplikatzeko errazagoak dira eta ez dituzte eskatzen adimen artifizial eta ML arloko eza-gutza aurreratuak. Bestalde, datuen transformazio mota aukeratuz, datuetatik erauzi nahi diren ezaugarriak hautatu daiteke.

Honek, proiektua garatzeko beharrezko baliabideak eta denbora optimizatzea ahalbidetzen du; baita, lan honen testuinguruan, emaitza fidagarriak eta eraginkorrak lortzea ere.

7. Soluzioaren deskribapena

Atal honetan proposatutako soluzioaren deskribapena azaltzen da. Akats elektrikoaren detekzioa eta sailkapena burutzeko, proiektuaren soluzioa 5 pausutan bananduko da (7.1 Irudia). Pausu hauek proiektuan zehar jarraituko diren pausuak definituko dituzte.



7.1 Irudia: Soluzioaren diagrama.

- ① **Akats datuen sorkuntza:** Lehenengo pausu honetan, ML modeloak entrenatzeko behar den datu multzoa sortuko da. Horretarako, *MATLAB/Simulink* softwarearen bidez sistema elektrikoaren modela garatuko da. Sistema elektrikoan mota desberdineko akats elektrikoak aplikatuko dira, eta hainbat simulazio burutuko dira. Simulazio bakoitzean sistema elektrikoaren sarearen fase bakoitzeko tentsio eta korrante neurketak egingo dira. Akats elektriko hauek, modeloaren funtzionamendua orokortu ahal izateko, baldintza desberdinetan ezarri behar

dira. Horretarako distantzia, akats erresistentzia, karga maila eta bestelako parametroak aldatuko dira simulazioetan.

- ② **Datuen aurreprozesaketa:** Pausu honetan, sistema elektrikoaren tentsio eta korrante neurketa gordinak aurreprozesatuko dira, datuak hurrengo pausuetan errazago maneiatu ahal izateko. Horretarako simulazio neurketak simulazio datuekin batera gordeko dira, ondoren erabiliko diren SL tekniketarako behar diren etiketadun datuak sortuz. Datu hauek sistema elektrikoaren eta akats mota desberdinen parametroak izango dira, hala nola, karga maila, akats distantzia, akats erresistentzia eta bestelako parametroak. Gainera, tentsio eta korrante neurketen eskalak oso desberdinak izan daitezkeenez, datuen normalizazioa burutuko da, neurketak *per unit* balioetan jarriz.
- ③ **Ezaugarrien azterketa:** Aurreprozesatutako datuetatik abiatuta, ML modeloek datuen artean multzo desberdinak bereizteko, datuen ezaugarriak definitu behar dira. Problemaren konplexutasunaren arabera, zuzenean tentsio edo korrante neurketak nahikoak izan daitezke *shallow* ML modelo sinpleen bidez detekzioa eta sailkapena burutzeko. Problema konplexuagoa bada, esaterako HIF detekzioa burutzen bada, datuen artean ezaugarriak bilatu beharko dira, eta hauen artean egokienak hautatu.
- ④ **Detekzio ML modeloa:** Behin datuak prest izanda, detekziorako ML modeloa garatuko da. Modelo desberdinen entrenamendua burutuko da, datuak entrenamendu eta testatze multzoetan bananduz. Zehaztasun handiena duten modeloak hautatuko dira, eta ondoren, modelo hauen hiperparametroen bilaketa burutuko da. Azkenik, modelo egokiena hautatuko da.
- ⑤ **Sailkapen ML modeloa:** Sailkapen modeloa hautatzeko pausu berdinak jarraituko dira, aurreko pausuan lortutako emaitzak kontuan hartuta. Ordea, akatsen detekzioa sailkapen bitarreko arazoa da, eta ordea, akats moten sailkapen arazoa hainbat klase edo etiketako problema izan daiteke. Beraz, arazoa ebazteko planteamendurik aproposena bilatuko da.

8. Metodologia

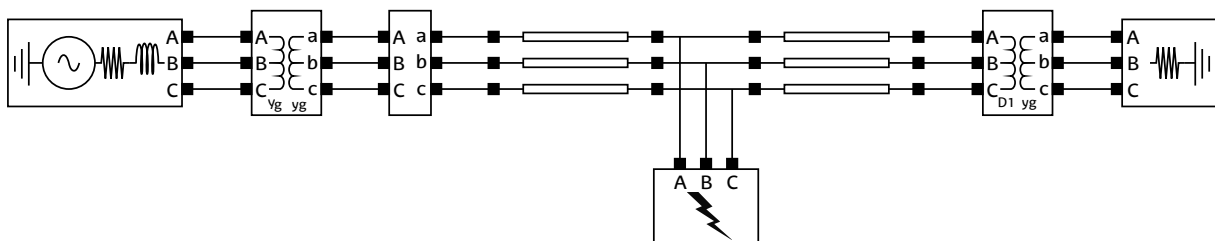
8.1 Akats datuen sorkuntza

Atal honetan akats-datuak sortzeko jarraitu diren pausuak azaltzen dira. Lehenik eta behin, sistemaren modelo elektrikoa osatzeko erabili diren *Simulink*-blokeak eta hauen parametroak aurkezten dira. Bigarrenik, datu-multzoa lortzeko burutu diren simulazio desberdinak azaltzen dira.

8.1.1 Sistema elektrikoaren modeloa

Akats-datu multzoaren sorkuntzarako lehenengo pausia sistema elektrikoaren modelizazioa izan da. Modeloa, proiektuko plataforma experimentalaren sistema elektrikoan oinarrituta dago. Ordea, aldaketa batzuk burutu dira, modeloaren portaera MG baten portaerara gehiago hurbiltzeko. Modelo hau *Simulink* aplikazioan garatu da, *Specialized Power Systems* liburutegiko elementuak erabiliz.

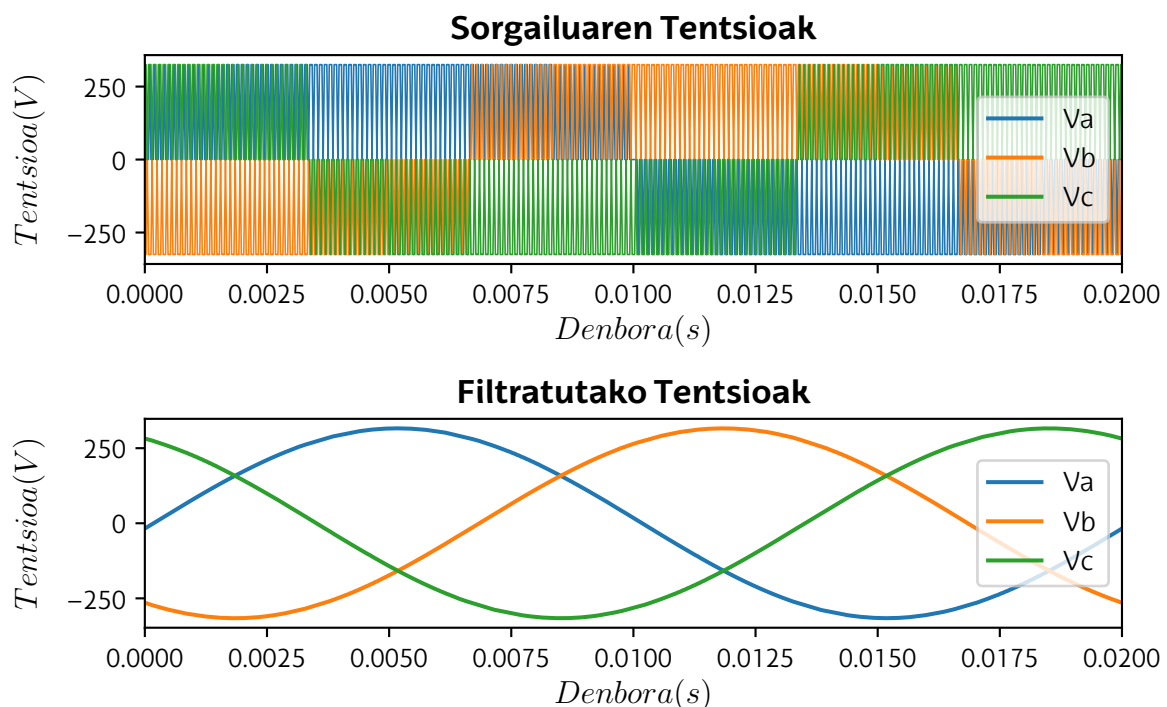
Sistema elektrikoaren elementuak hurrengoak dira: Sorgailua, *YNyn* transformadorea, neurgailua, linea elektrikoa, *Dyn* transformadorea, kargak eta falta elektrikoaren modeloak (8.1 Irudia).



8.1 Irudia: *Simulink* modelo elektrikoaren diagrama.

Sorgailua

Plataforma experimentalaren sorgailua *Neutral Point Clamped* (NPC) motako alderanzgailua da. Ordea, alderanzgailuaren konmutazio maiztasuna 10kHz da, eta ostean, LC filtro baten bidez sorgailuaren tentsio eta korronteak filtratzen dira (LCL filtroa, transformadorearen harilkatuen inдукtantzia kontuan hartuta). Ondorioz, filtro edo transformadore ostean aurkezten diren tentsio eta korronte uhinak ia harmonikorik gabeko uhin sinusoidalak dira (8.2 Irudia).



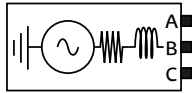
8.2 Irudia: 10kHz maiztasuneko PWM sorgailu baten tentsioak LC filtroaren aurretik eta ostean.

Ondorioz, modeloaren konplexutasuna murrizteko eta simulazioen exekuzio-denbora murrizteko, sorgailua eta filtroa zuzenean sorgailu ideal baten bidez modelizatzea aukeratu da. Erabiliko den *Simulink* blokea *Three-Phase Source* da.

Sorgailuaren parametroak plataforma experimentalaren sorgailuaren arabera finkatu dira. Barne-inpedantzia kalkulatzeko parametroak finkatu dira sorgailuaren portaera egokirako (8.1 Taula).

8.1 Taula: *Three-Phase Source* blokearen parametroak.

Parametroak	
<i>Configuration</i>	$\sqrt{3}$
<i>Phase-to-phase voltage (Vrms)</i>	400
<i>Phase angle of phase A (degrees)</i>	$360 \cdot rand$
<i>Frequency (Hz)</i>	50
<i>Short-circuit level (VA)</i>	100k
<i>Base Voltage (Vrms ph-ph)</i>	400
<i>X/R ratio</i>	0.5



Phase angle of phase A parametroan *rand* funtzioa gehitu da, simulazio bakoitzean akats elektrikoak faseen une desberdinetan gerta daitezten.

YNyn transformadorea

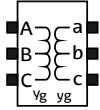
Transformadore hau, lehen ikusi denez, *YNyn* motakoa izan behar da, kontrolaren bitartez aplikatuko diren tentsioak sare elektrikoan finkatu ahal izateko, edo, beste modu baten esanda, tentsio eta korrante homopolarrak transformadorea zeharkatu ahal izateko.

Plataforma esperimentalean dagoen transformadorea hiru zutabeko *YNyn* transformadorea da, 400V/400V transformazio erlazioarekin. Ordea, hiru zutabeko *YNyn* transformadoreen inpedantzia homopolarraren arazoa saihesteko, simulazioetan transformadorearen bost zutabeko modeloa erabili da. Transformadorearen *Simulink* blokea *Three-Phase Transformer (Two Windings)* da.

Bloke honen parametroak plataforma esperimentalean dagoen *Tecnotrafo* transformadorearen *datasheet*etik atera dira (8.2 Taula).

8.2 Taula: *Three-Phase Transformer (Two Windings)* blokearen parametroak.

Parametroak	
<i>Winding 1 connection</i>	<i>Yg</i>
<i>Winding 2 connection</i>	<i>Yg</i>
<i>Type</i>	<i>Five-limb core</i>
<i>Nominal power (VA) and frequency (Hz)</i>	[10k, 50]
<i>V1 Ph-Ph (Vrms), R1 (ohm), L1 (H)</i>	[400, 0.32, 2.7m]
<i>V2 Ph-Ph (Vrms), R2 (ohm), L2 (H)</i>	[400, 0.191, 2.84m]
<i>Magnetization resistance Rm (Ohm)</i>	31100
<i>Magnetization inductance Lm (H)</i>	0.102

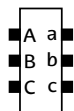


Neurgailua

Datu-multzoa sortzeko hartuko diren neurketak transformadorearen ostean, linea elektrikoaren hasieran, fase bakoitzeko tentsio eta korronteak izango dira. Horretarako *Three-Phase V-I Measurement* blokea erabili da. Ondoren azalduko den moduan, bloke honetan sistemaren balio nominalak adieraziko dira, neurketak zuzenean *per unit* balioetan adierazterako.

8.3 Taula: *Three-Phase V-I Measurement* blokearen parametroak.

Parametroak	
<i>Voltage measurement</i>	<i>phase-to-ground</i>
<i>Voltages in pu</i>	✓
<i>Current measurement</i>	<i>yes</i>
<i>Current in pu</i>	✓
<i>Base power (VA 3 phase)</i>	10k
<i>Nominal voltage (Vrms phase-phase)</i>	400



Linea elektrikoa

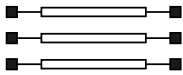
Plataforma esperimentalean aireko eta lurpeko linea elektrikoak inplementatu dira. Linea elektrikoak PI zirkuitu baliokideen bidez eraiki dira, hau da, serieko erresistentzia eta induktantziekin, eta paraleloko bi kapazitantziekin.

Alde batetik, zirkuitu laburreko akats elektrikoak aireko lineetan eman ohi direnez, soilik aireko linea modelizatu da. Beste alde batetik, akats elektrikoaren ondorioz linean eman daitezkeen maiztasun handiko iragankorrak simulatu ahal izateko, parametro sakabanatuen lineako modeloa erabiltzea erabaki da.

Akats elektrikoa linea elektrikoaren luzeran emango da, luzera totala 10 kilometro izanik. Ondorioz, linea elektrikoa bi blokeen bidez modelizatuko da: alde batetik, $YNyn$ transformadoretik akats elektrikora dagoen linea; eta beste aldetik, akats elektrikitik Dyn transformadorera dagoen linearen zatia. Akats elektrikoa, bi bloke hauen artean ezarriko da.

Linearen parametroak plataforma esperimentalean inplementatutako linearen parametroetatik atera dira (8.4 Taula).

8.4 Taula: *Distributed Parameters Line* blokearen parametroak.

Parametroak	
	<i>Number of phases</i> 3
	<i>Line length (km)</i> L_f ; 10 - L_f
	<i>Frequency (Hz)</i> 50
	<i>Resistance (Ohms/km)</i> [0.02232, 0.04464]
	<i>Inductance (H/km)</i> [0.142m, 0.5345m]
	<i>Capacitance (F/km)</i> [134.4n, 89.6n]

Dyn transformadorea

Banaketa-sareetan kargak elikatzeke Dyn transformadoreak erabiltzen dira. Plataforma esperimentalean, ordea, transformadore hau ez da inplementatu. Aldiz, karga

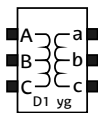
elektrikoak zuzenean hirukian konektatu dira, kargetan linea elektrikoaren tentsio konposatuak finkatzeko.

Proiektu honetan karga desorekatuak simulatuko dira. Suposatuz MGen desoreka mailak altuak eman daitezkeela, *Dyn* transformadorea ez erabiltzekotan, desoreka hauek linea elektrikoaren tentsioetan duten eragina oso altua da. Hala ere, *Dyn* transformadorea erabiliz, transformadoreak berak kargen tentsio desorekatuak linea elektrikoan izango duten eragina murriztuko dute, tentsio hauek maila baten orekatuz. Fenomeno hau gerta dadin, *Dyn* transformadorea simulazioetarako erabiltzea erabaki da. Blokea, aurreko transformadorearen kasuan bezala, *Three-Phase Transformer (Two Windings)* izango da.

Transformadore honen parametroak zuzenean Iberdrolak eskainitako jatorrizko *Simulink* fitxategitik hartu dira (8.5 Taula).

8.5 Taula: *Three-Phase Transformer (Two Windings)* blokearen parametroak.

Parametroak	
<i>Winding 1 connection</i>	D1
<i>Winding 2 connection</i>	Yg
<i>Type</i>	<i>Five-limb core</i>
<i>Nominal power (VA) and frequency (Hz)</i>	[10k, 50]
<i>V1 Ph-Ph (Vrms), R1 (pu), L1 (pu)</i>	[400, 0.002, 0.02]
<i>V2 Ph-Ph (Vrms), R2 (pu), L2 (pu)</i>	[400, 0.002, 0.02]
<i>Magnetization resistance Rm (pu)</i>	500
<i>Magnetization inductance Lm (pu)</i>	500

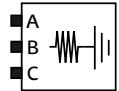


Karga

Plataforma esperimentalean karga elektrikoak erresistentzien bidez inplementatu dira. Simulazioetarako modeloan baita erresistentziak erabiliko dira karga elektrikoaren modura. Honetarako, *Three-Phase Series RLC Load* blokea erabiliko da. Bloke honetan sistemaren parametroak finkatzen dira, eta kargaren potentzia kontsumoa finkatzen

da. Datu hauekin, potentzia kontsumo hori emateko beharko den erresistentzia kalkulatu du, erresistentzia balio hau finkatuz.

8.6 Taula: *Three-Phase Series RLC Load* blokeen parametroak.

Parametroak	
	
<i>Configuration</i>	Y_g
<i>Nominal $V_a V_b V_c$ (Vrms)</i>	230
<i>Frequency (Hz)</i>	50
<i>Active powers (W)</i>	$[P_a, P_b, P_c]$
<i>Inductive reactive powers (var)</i>	$[0, 0, 0]$
<i>Capacitive reactive powers (var)</i>	$[0, 0, 0]$

Inpedantzia baxuko akats elektrikoa

Inpedantzia baxuko akats elektrikoak simulatzeko *Three-Phase Fault* blokea erabili da. Bloke honetan akats-erresistentzia, lurreko erresistentzia eta akatsa ezartzeko erabiltzen den etengailuaren *snubber* zirkuituaren parametroak finkatu ahal dira. Akats-erresistentziaren balioa kenduta, beste parametroetarako blokearen balio lehenetsiak erabili dira.

Inpedantzia baxuko akats elektrikoetarako erabili beharko diren erresistentzia balioak sarearen tentsioaren eta karga mailaren arabera dira. Inpedantzia baxuko akatsak, akats-korronte altuak eragiten dituzten akatsak dira. Hau da, gainkorronte handiak eragiten dituzten akatsak.

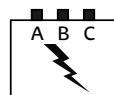
Plataformaren sare elektrikoa 400V-ko tentsio maila duenez, bi fenomeno kontuan hartu beharko dira akats erresistentzia tartea finkatzeko.

Alde batetik, akats elektriko bat ematen denean, akats-korrontea, laburbilduz, sorgailuaren eta falta puntuaren lurraren arteko inpedantzia totalaren menpekota da, non *Ohmen* legea betetzen den. Hortaz, erresistentzia maila berdinerako sareko tentsioa altuagoa bada, akats korrontea altuagoa izango da. Banaketa-sareetako tentsio mailak 1kV eta 35kV artekoak izaten dira. Sareko tentsioa 400V-koa denez, gain-korronte al-

tuak eman ahal izateko, akats erresistentziak txikiak izan beharko dira, ohiko banaketa sareetako akats erresistentzia balioekin alderatuz.

Bestalde, banaketa-sareetako tentsio-mailak altuak izaten dira, sareetatik dabilen korrontea murrizteko. Beraz, inpedantzia baxuko akats-korrontek sareen korrontean suposatzen duten gainkorrontea handiagoa da. Beraz, berriro ere, antzeko gainkorrontek simulatu ahal izateko inpedantzia balio baxuagoak erabiliko dira 400V-ko sarean.

8.7 Taula: *Three-Phase Fault* blokearen parametroak.

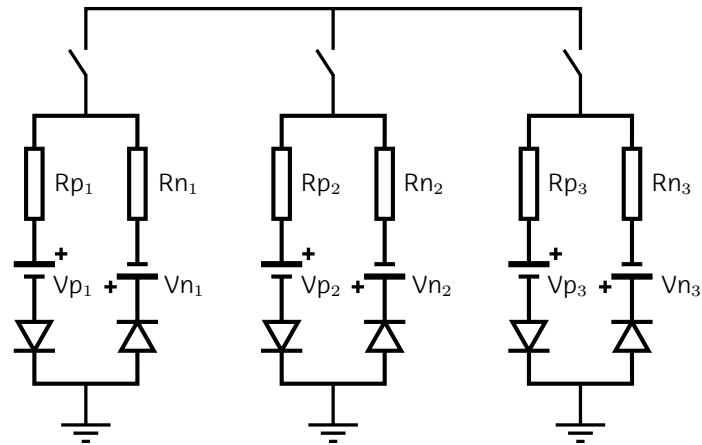


Parametroak

<i>Fault resistance (Ohm)</i>	R_f
<i>Ground resistance (Ohm)</i>	0.01
<i>Snubber resistance (Ohm)</i>	1M
<i>Snubber capacitance (F)</i>	inf

HIF akats elektrikoa

HIF edo inpedantzia altuko falta gehienak arku elektrikoak izaten dira. Arku elektrikoak modelizatzeko metodo ugari daude, baina gehienak Emanuelen arku elektrikoaren eredu oinarritzen dira [51]. Proiektu honetan, paraleloan hainbat Emanuel arku-eredu erabiltzen oinarritzen den metodoa erabiliko da [52]. Metodo honetan, erresistentzia aldakorrak erabili ordez, balio desberdinak dituzten hainbat Emanuel arku-eredu erabiltzen dira paraleloan, bakoitza denbora tarte batzuetan lan egiten (8.3 Iru-dia).



8.3 Irudia: HIF arku elektrikoaren ereduaren diagrama.

Ereduan paraleloan soilik hiru modulu erabiltzea erabaki da. Akatsa denbora gehiagoan zehar simulatzekotan modulu gehiago erabiltzea komeni da, arkuaren portaera ez lineala eta aldakorra islatzeko. Falta eman ondoren bakarrik bi ziklo simulatu direnez, falta eman eta 0.04 segundo ostean aktibatzen diren moduluak ez dira erabili.

Ereduko elementuen balioak [52] artikuluan eskainitako balioetatik abiatuta finkatu dira. Artikuluko balioak 20kV-eko sare baterako finkatu direnez, balio hauek murriztu beharko dira 400V-ko sarean erabili ahal izateko. Aplikatu den murrizketa-faktorea zuzenean bi sareen arteko tentsio erlazioa izan da:

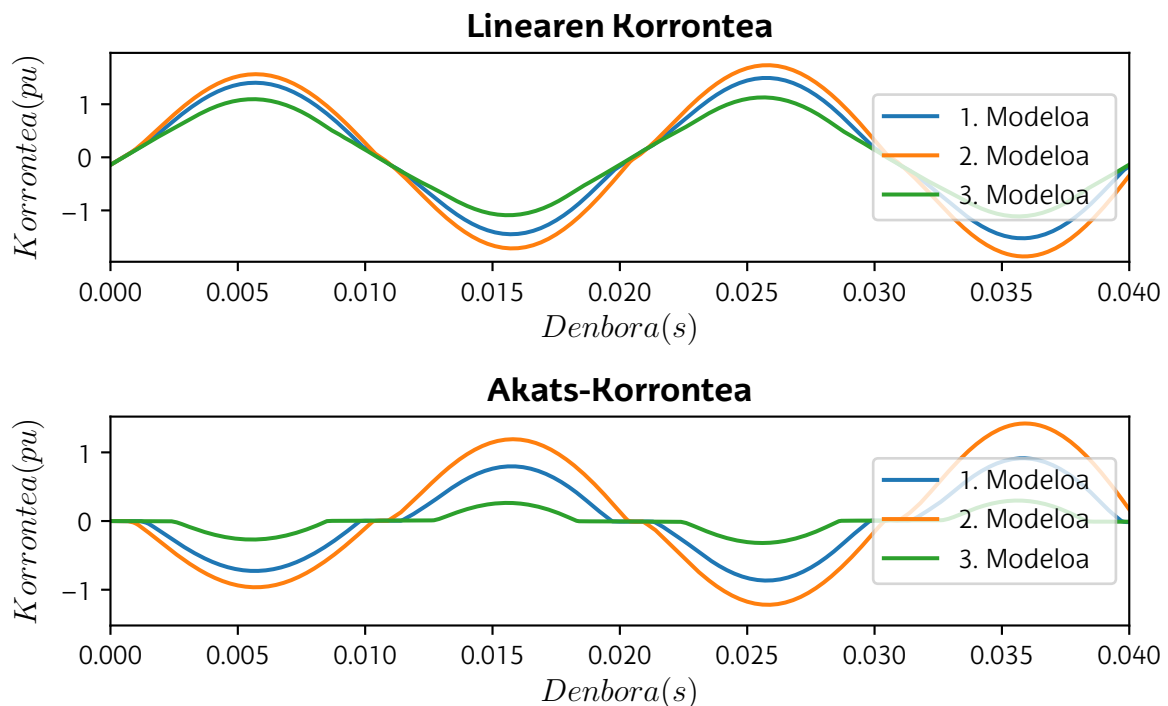
$$c = \frac{20.000V}{400V} = 0.02 \quad (8.1)$$

Erabili diren parametroak (8.8 Taula) eta zuzenketa faktorea zuzenak izan direla egiaztatzeko, HIF mota bakoitzeko simulazioak egin dira. Ondoren, lortutako akats korronteak eta linearen korronte neurketak (8.4 Irudia) aztertu dira, lortutako emaitzak arku elektrikoek sortutako akats korronteekin konparatuz.

8.8 Taula: HIF modeloaren parametroak.

Modeloa	t (s)	Rp (Ohm)	Rn (Ohm)	Vp (V)	Vn (V)
1. Modeloa	0.04	1000	1050	4000	4500
	0.05	900	905	1800	1900
	0.07	1500	1505	9000	10000
2. Modeloa	0.04	2900	3000	8000	8100
	0.05	2800	2850	3000	3500
	0.07	9000	9600	8000	9000
3. Modeloa	0.04	3500	3550	7500	7600
	0.05	2500	2550	4000	4500
	0.07	6500	7000	11000	11050

(Balioak murrizketa-faktorea aplikatu aurrekoak dira.)



8.4 Irudia: HIF akatsen korrontekak.

8.1.2 Simulazioak

Akats datu-multzoa lortzeko azken urratsa simulazioak egitea izango da. Datu-multzo zabal eta osoa lortzeko parametro aldakorrak eta ziurtasunik gabeko gertaerak hartu behar dira kontuan. ML modeloak bere funtzionamendua orokortu eta edozein egoeratan behar bezala jardun ahal izateko, ezinbestekoa da simulazioetan zehar parametro hauek aldatzea.

Ziurtasunik gabeko gertaerak bi taldetan banatzen dira: akats-gertaeren taldea eta ez-akats gertaeren taldea.

- **Akats-gertaerak:** Gertaera hauek akatsak izan ahal dituzten ezaugarri desberdinak dira. Hauen artean akats mota, akats distantzia, akats erresistentzia eta inpedantzia altuko akatsak topatzen ditugu.
- **Ez-akats gertaerak:** Gertaera hauek sistemaren gertaera arruntak dira, akats egoerekin erlazionatuta ez daudenak. Haien artean karga maila desberdinak, karga desorekak, sistemaren topologia aldaketak, kommutazio gertaerak eta MG operazio modu desberdinak daude.

Lan honetan hurrengo gertaerak kontsideratu dira (8.9 Taulan laburbilduta).

- **Akats mota:** LG, LL, LLG eta LLL akats mota guztiak kontsideratuko dira, totalen 10 akats mota izanik. LLLG akats motak ez dira simulatuko, LLL eta LLLG akats moten arteko bereizketa ez delako egiten akatsak detektatzeko orduan.
- **Akats erresistentzia:** Akats erresistentzia 0,01 - 15 tarteko balioetan finkatzea erabaki da. Akats erresistentzia zenbat eta handiagoa izan, orduan eta gainkorronte txikiagoak ematen dira.
- **HIF akatsak:** Akats elektrikoaren artean HIF akatsak baita simulatuko dira.
- **Akats distantzia:** Akats distantzia 1 eta 9 kilometroen artean finkatuko da, kilometro bateko jauziak eginez.
- **Karga maila:** Sistemaren karga maila karga nominaleko %20 eta %100 balioen

artean finkatuko da, %20-ko jauziekin.

- **Desoreka maila:** Desoreka maila %10 eta %90 balioen artean egongo da.

8.9 Taula: Akats eta ez-akats gertaera desberdinak.

Gertaerak	
Akats mota	AG, BG, CG, AB, BC, AC, ABG, BCG, ACG, ABC
Akats erresistentzia	0,01 0,25 2,5 5 10 15
HIF akatsak	✓
Akats distantzia	1 - 9
Karga maila	%20 - %100
Desoreka maila	%10 - %90*

(*Balio nominaletik adierazitako ehunekoa.)

Akats mota bakoitzeko, akats erresistentzia, HIF akats modelo eta akats distantzia konbinazio guztiak simulatuko dira. Gainera, konbinazio bakoitzean karga orekatuko simulazio bat eta karga desorekatuko beste simulazio bat burutuko da. Karga orekatuko simulaziorako, karga maila balio desberdinen artean balio bat hautatuko da zoriz. Karga desorekatuko simulazioetarako, fase bakoitzerako kargaren %10 eta %100 arteko balio bat hautatuko da zoriz. Totalean, 1620 simulazio burutu dira.

Simulazioaren parametroak, *Powergui* blokean eta *Solver Configuration* atalean definitu dira (8.10 Taula).

8.10 Taula: Simulazio parametroak.

Parametroak	
<i>Simulation type</i>	<i>Discrete</i>
<i>Simulation time (s)</i>	1 μ
<i>Solver type</i>	Fixed
<i>Solver</i>	auto
<i>Fized-step size</i>	1 μ

Neurketa datuak laginketa periodo konstante batekin lortzeko, simulazio mota diskretua eta laginketa periodo finkoa definitu dira. Neurketak, 10kHz-eko laginketa maiztasunarekin hartu dira, 0,08 segundotan zehar. Akats elektrikoa 0.04 segundoan ezarri da, simulazio bakoitzean akats gabeko bi ziklo eta akatsarekin beste bi ziklo lortuz.

Beraz, simulazio guztiak burutu ondoren, totalen 1.296.000 datu lortu dira. Haietatik, 648.000 akats elektriko gabeko datuak dira, 432.000 inpedantzia baxuko akatseko datuak eta 216.000 HIF akatseko datuak.

Ondoren azalduko den moduan, simulazio bakoitzarekin batera datuen aurreprozesaketa burutu da, *Simulink* blokeen bidez eta simulazioak exekutatzeko erabili den *script*arekin. Datu-multzoa lortzeko erabili den *MATLAB* kodea l. Eranskinean aurkitzen da.

8.2 Datuen aurreprozesaketa

Datuen aurreprozesaketa ML modeloen funtzionamendu egokirako oinarritzkoa den pausua da. Sistema elektrikoaren datuak gehienetan maiztasun finko batera hartutako denbora-segida seinaleak dira. Datu horiek falta diren datuak, laginketa periodo ez-konstanteko datuak edo neurgailuek sortutako zarata izan ditzakete. Era berean, tentsio eta korrante neurketen arteko eskala asko alda daiteke.

Horrenbestez, ML tekniketarako erabiliko diren datuak aurreprozesaketa etapa bat behar dute. Pausu honetan datuen etiketatzea, normalizazioa, falta diren datuen tratamendua eta bestelako prozesuak burutzen dira.

Orokorrean, datuen aurreprozesaketa ML modeloen zehaztasuna hobetu dezake, modeloaren sarrera datuak alderatzea eta ebaluatzea errezoago eginez.

Proiektu honetan, datuak *Simulink* simulazioen bidez lortu direnez, datuen aurreprozesaketaren pausua ez dauka hainbesteko garrantzia. Alde batetik, lortutako datu guztiak laginketa periodo berdinarekin lortutako datuak dira eta ez daude falta diren daturik. Beste alde batetik, datuek ez daukate zaratarik, erabilitako korrante eta ten-

tsio neurgailua ideala delako. Ondorioz, datuei aplikatu beharko zaizkien aurreprozesaketarako pausuak gutxi izango dira.

8.2.1 Datuen normalizazioa

Lehen azaldu den moduan, korrante eta tentsio neurketak eskala desberdinetan daude. Hau da, tentsio neurketak $230V_{r.m.s}$ balioko uhin sinusoidalak izango dira. Ordea, korrante neurketak anpere gutxiko uhinak izango dira, karga mailaren arabera, inpedantzia baxuko akatsekin handiagoak egingo direnak. Ondorioz, ML modeloak neurketa guztiei garrantzi berdina eman ahal izateko, eta entrenamendua errazteko, tentsio eta korrante neurketak eskala berean finkatzea komeni da. Eskala hau 0 - 1 arteko eskala izan ohi da. Ordea, hurrengo pausuan datuen ezaugarriak lortu ahal izateko datuak transformatu direnez, neurketak *per unit* balioetan jarrita nahikoa, momentuz datuak antzeko eskala batera izateko.

Horretarako, aurretik aipatutako *Three-Phase V-I Measurement* blokean neurketak *per unit*etan finkatuko dira. Honekin tentsioak eta falta aurreko korranteak 0 - 1 tarteko balioetan mugatzea lortuko da, akats korrante neurketak eskala horretatik kanpo geratuz.

8.2.2 Datuen etiketatzea

SL teknikak aplikatu ahal izateko, aurretik azaldu denez, datuak etiketak izan behar dituzte. Etiketa hauek, ML modeloak aurreikusiko edo sailkatuko dituen balioak izan behar dira. Etiketa hauetaz gain, simulazioen beste parametro batzuk izan daitezke, simulazioetan definitutako gertaeren balioak, adibidez. Simulazioen datu hauek eskura izanda, datuen analisi eta azterketa sakonagoak egin daitezke.

Beraz, oinarritzkoak izango diren etiketak hurrengoak izango dira:

- **F:** Datuak akats egoerakoak edo ez diren zehaztuko du. Etiketa honekin akatsen detekzioa burutuko duen modeloa entrenatuko da.

- **A, B, C eta G:** Akatsak ze faseetan eman diren zehaztuko du. Etiketa hauekin akats motak sailkatuko dituen modelo entrenatuko da.

Gainera, hurrengo etiketak gehitu dira:

- **R:** Inpedantzia baxuko akatsen erresistentzia, edo inpedantzia altuko akatsen erabilitako modeloaren zenbakia (1, 2 edo 3) adierazten du.
- **L:** Akatsen distantzia zehazten du.
- **U:** Kargaren desoreka maila adieraziko du. Etiketa honen balioaren kalkulurako, edozein bi faseen arteko karga maila alderik handiena kalkulatu da.
- **HIF:** Inpedantzia altuko akatsak diren adierazten du.

Azkenik, datuak csv fitxategi baten gorde dira hurrengo formatuarekin (8.11 Taula).

la	lb	lc	Va	Vb	Vc	F	A	B	C	G	L	R	U	HIF
-0.83	0.73	0.10	-0.87	0.71	0.16	0	1	0	0	1	1	0.01	0.0	0
-0.17	0.34	-0.17	-0.24	0.92	-0.58	0	1	1	0	0	6	15	0.7	0
-0.54	-0.06	0.82	-0.72	-0.23	0.88	1	1	0	1	1	5	3	0.3	1

8.11 Taula: Datuen formatua.

8.3 Ezaugarrien azterketa

Datuen aurreprozesamendua amaitu ondoren, funtsezkoa da datuetatik ezaugarriak erauztea eta hautatzea. Pausu hauek ML modeloaren eraginkortasuna eta zehaztasuna hobetzeko funtsezko urratsak dira. Datu gordinek, berez, sarritan ez dute informazio nahikoa ematen *shallow learning* ML modeloak behar bezala funtziona dezaten. Gainera, aldiuneko neurketak ez dira nahikoa izango modeloaren funtzionamendu egokirako, baina, adibidez, azken zikloan neurtutako datu guztiak erabiltzeak modeloaren sarrerako datuen gehiegizko dimentsioa ekarriko luke *shallow learning* eredu bantentzat. Horregatik, funtsezkoa da datu horiek transformatzea, seinale elektrikoei da-

gozkien konplexutasun eta patrioiak atzeman ditzaketen ezaugarri garrantzitsuak atertzeko.

Garrantzitsua da adieraztea datuen transformazioak eta ezaugarrien hautaketa aurreprozesamenduaren parte izan daitezkeela, baina lan honetan etapa desberdinetan bereizi dira, urrats bakoitzerako erabilitako metodologia eta tresna desberdinak (*MATLAB* eta *Python*) erabiltzeagatik.

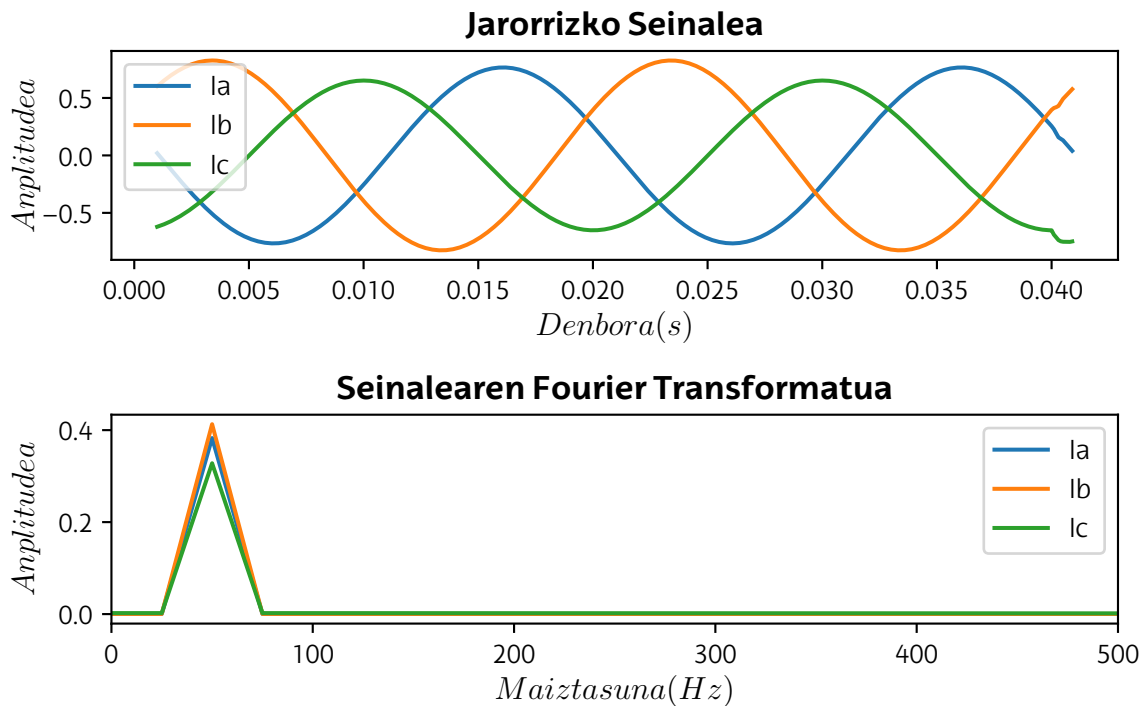
8.3.1 Datuen transformazioa

Lehenik eta behin datuen transformazio-teknika desberdinak aztertuko dira. Datuen transformazioaren bitartez, seinale bat domeinu batetik beste batera alda daiteke, honen errepresentazioa forma batetik beste batera aldatuz. Seinaleen transformaziorako hainbat teknika daude, baina lan honetan hurrengoak aztertuko dira: *Fourier* transformatua eta *Wavelet* transformatua.

- **Fourier transformatua:** Transformatu honen bidez seinaleen datuak denbora domeinutik maiztasun domeinura transformatzen dira. Seinaleak maiztasun eta anplitude desberdineko seinale sinusoidalenetan deskonposatzen dira. Beraz, *Fourier* transformatu baten irteera balioak seinale baten aurkitzen diren harmonikoen balioak adierazten ditu, haien maiztasuna eta anplitudea adieraziz.

Seinale datuen transformazio hau batez ere erabilgarria da seinaleen maiztasun espektroa geldikorra denean, hau da, seinaleetan agertzen diren harmonikoak denboran zehar aldatzen ez direnean. Adibidez, aztertzen ari den seinaleak akats elektrikoa gertatu den lehen unea hartzen badu, seinale osoaren puntu bakarra uhin sinusoidaletik kanpo geratuko da. *Fourier* transformatua aplikatuz, azken uneko perturbazioa ziklo oso bateko harmonikotan deskonposatuko da, anplitudea nabarmen gutxituz (8.5 Irudia).

Desabantaila hauek ekiditeko *Short-Time Fourier* transformatua erabili daiteke. Transformatu hau seinale osoa harmonikoetan deskonposatu ordez, seinalearen atal txikiagoak deskonposatu eta gainezartzen ditu. Ordea, denbora domeinuan daukan sentsibilitatea murriztua da [53].

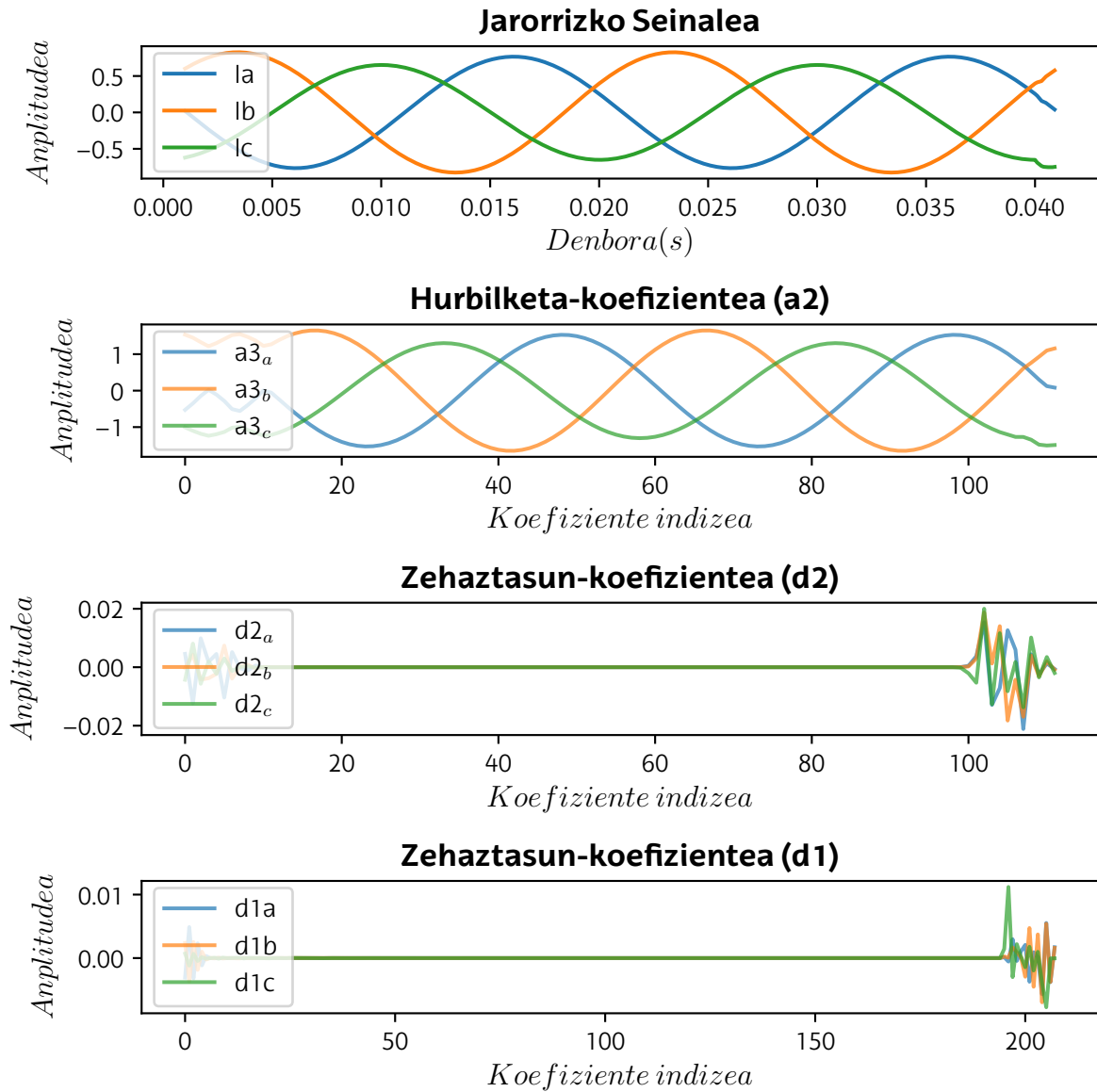


8.5 Irudia: Seinale baten *Fourier* transformatua, non $t = 4$ segundoan c fasean lurrerako akatsa gertatu den.

- **Wavelet transformatua:** Transformatu hau seinaleen maiztasun espektroa dinamikoki aztertzen ditu. Transformatuak erresoluzio handia du maiztasun eta denbora domeinuan. Hau lortzeko, *Wavelet* deituriko seinale baten eta jatorrizko seinalearen artean konboluzioa aplikatzen da. *Wavelet* uhinaren eskala eta forma aldatuz, jatorrizko seinalean aurkitzen diren harmoniko desberdinak eta hauen kokapena denboran lortu daitezke.

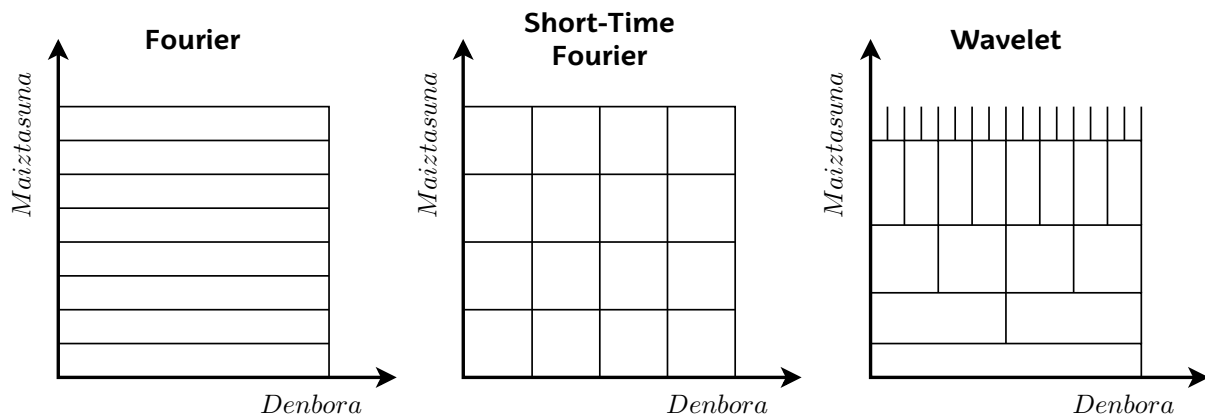
Ondorioz, lehen ez bezala, uhin baten azken aldiunean perturbazio bat ematen bada, *Wavelet* transformatua aplikatuz honen presentzia askoz nabarmenagoa izango da transformatutako datuetan (8.6 Irudia).

Wavelet transformatuen artean bi mota aurkitzen dira: *Continuous Wavelet Transform* (CWT) eta *Discrete Wavelet Transform* (DWT). Bien arteko desberdintasun nagusia *Wavelet* uhina eskalatu eta transladatzeko eran dago. DWT transformatuan eskala eta translazio faktoreak diskretuak dira. Ondorioz, jatorrizko uhinak eskala desberdinetan deskonposatzen dira.



8.6 Irudia: Seinale baten *Wavelet* transformatua, non $t = 4$ segundoan c fasean lurrerako akatsa gertatu den.

Akats elektrikoaren detekzioarako, akatsa ematen den unean lortutako neurketak ez dituzte denboran konstanteak diren perturbazioak izango. Ondorioz, denbora domeinuan sentsibilitatea duen transformatu bat erabiltzea komeni da aplikazio honetarako (8.7 Irudia) [53].



8.7 Irudia: Transformatuen maiztasun eta denbora espektroen sentsibilitatea.

Aurrekoa kontuan hartuz, eta transformatu desberdinen maiztasun eta denbora espektroan daukaten sentsibilitatea ikusita, ondorioztatu egin da akatsen detekziorako interesgarrien izan daitekeen transformatua *Wavelet* transformatua dela. Hain zuzen ere, lan honetan DWT transformatua erabiliko da.

Discrete Wavelet Transformatua

Uhinak, oszilakorrak diren seinaleak dira, denboran mugatuak ez daudenak. Adibidez, *Fourier* transformatuan uhin sinusoidalak erabiltzen dira seinaleak deskonposatzeko. Ordea, *wavelet* denboran luzera mugatua duten uhin txikiak dira. *Wavelet* transformatuak seinaleak deskonposatzeko uhin txiki hauek erabiltzen ditu.

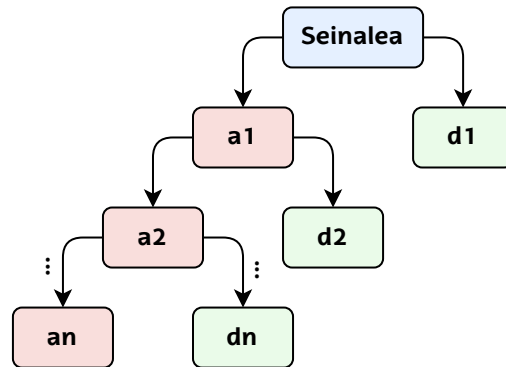
DWT transformatuak bereizmen (edo maiztasun) desberdineko mailatan deskonposatzen ditu seinaleak. Seinaleen deskonposaketa hurrengo formularekin adierazten da (8.2 Ekuazioa):

$$x(t) = \sum_m \sum_n d_m(n) \psi_{mn}(t) \quad (8.2)$$

Non $x(t)$ jatorrizko uhina, d deskonposizio-koefizienteak, $\psi(t)$ *mother wavelet* uhina, m eskala-koefizientea eta n translazio-koefizienteak diren.

Wavelet transformatuaren koefizienteak lortzeko, *mother wavelet* eta jatorrizko uhi-

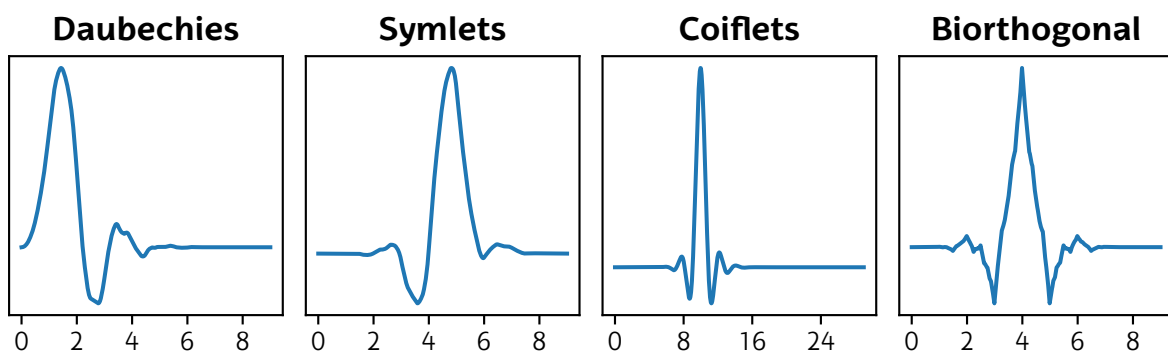
naren arteko konboluzioa aplikatzen da. *Mother wavelet* eskala desberdinetara txikituz edo handituz, maiztasun desberdineko koefizienteak lortzen dira (8.8 Irudia), non a hurbilketa-koefiziente den, eta d zehaztasun-koefizienteak diren. Transformatua diskretua denez, eskala kopurua finitua da, transformatuaren mailaren arabera.



8.8 Irudia: n mailako seinale baten deskonposaketa.

DWT transformaturako *mother wavelet* desberdinak hautatu daitezke (8.9 Irudia). *Mother wavelet* desberdinak erabiliz, seinaleetatik ezaugarri desberdinak erauziko dira.

Sistema elektrikoaren iragankorrak aztertzeko eta, kasu honetan bezala, akats elektrikoaren detektziorako *Daubechies wavelet* familia erabili ohi da. *Daubechies* familiako *wavelet* desberdinak dbN izenarekin ezagutzen dira, non N honen ordena adierazten duen. Lan honetan $db4$ waveleta erabiliko da.



8.9 Irudia: *Mother Wavelet* familia desberdinak.

8.3.2 Transformatuaren aplikazioa

Erabiliko den transformatua behin finkatuta, datuetan zelan aplikatuko den finkatu behar da. Kasu honetan, transformatua datuen leihokor baten aplikatzea erabaki da. Leihokor ziklo bateko luzera izango du, eta simulazio berdineko datu guztietan zehar mugituz joango da. Simulazio bakoitzak 800 datu dituenaz, leihokor 200 datuokor zabalera izango du. Ondorioz, lehenengo transformatua lehenengo 200 datuetan (1 - 200) aplikatuko da, bigarrena hurrengo datua barnebiltzen duen zikloa (2 - 201), azken datua heldu arte (601 - 800). Ondorioz, simulazio bakoitzeko 601 transformatu aplikatuko dira.

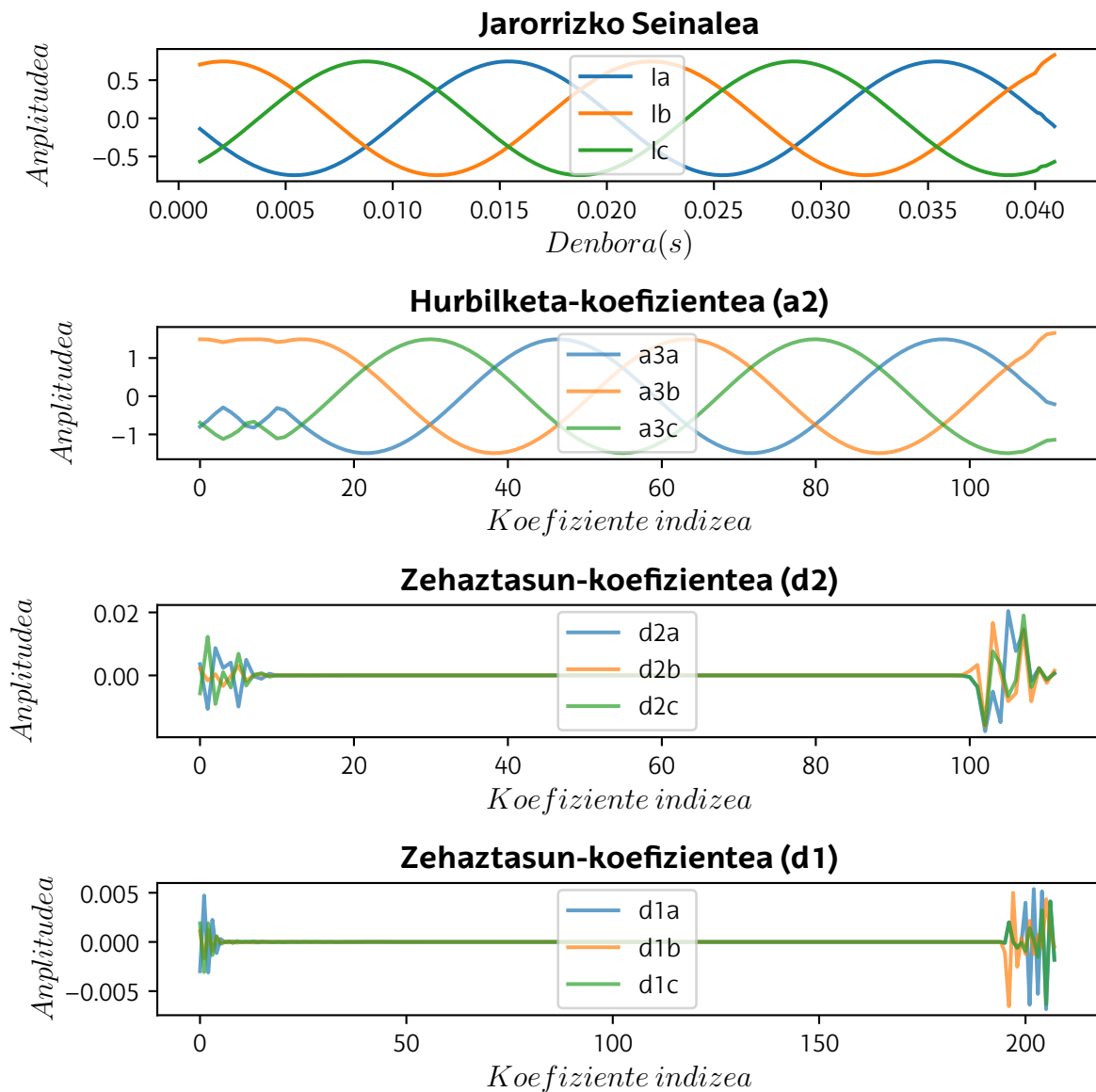
Datuen transformazioa *Python* softwarearekin burutu da. Honetarako *pywt* liburutegia erabili da. Liburutegi honetan komando desberdinak erabili daitezke transformatu aplikatzeko. Maila bakarreko deskonposaketa lortzeko, *dwt* komandoa erabiltzen da. Maila anitzekoa egiteko, *wavedec* komandoa.

Lan honetan, maila anitzeko deskonposaketa erabiliko denez edo, gutxienez, aztertuko denez, *wavedec* funtzioa erabiliko da. Funtzio honek 5 parametro ditu; derrigorrezkoak diren bi, *data*, *wavelet*, eta hautazko beste hiru, *mode*, *level* eta *axis*.

Datu-multzo baten *wavedec* komandoa erabilita, *wavelet* = *'db4'* eta *level* = 2 parametroekin, aztertu da *edge* efektua bezala ezagutzen den fenomenoak gertatzen dela (8.10 Irudia) [54].

Datu-multzoaren aurreko eta ondorengo datuak ezagutzen ez direnez, hasierako eta amaierako datu hauen transformatuan ezegonkortasunak agertzen dira. 8.10 Irudian, batez ere, hasierako datuetan nabarmentzen da efektu hau. Jatorrizko uhinetan perturbazioarik egon ez arren, transformatutako datuetan zarata agertzen da.

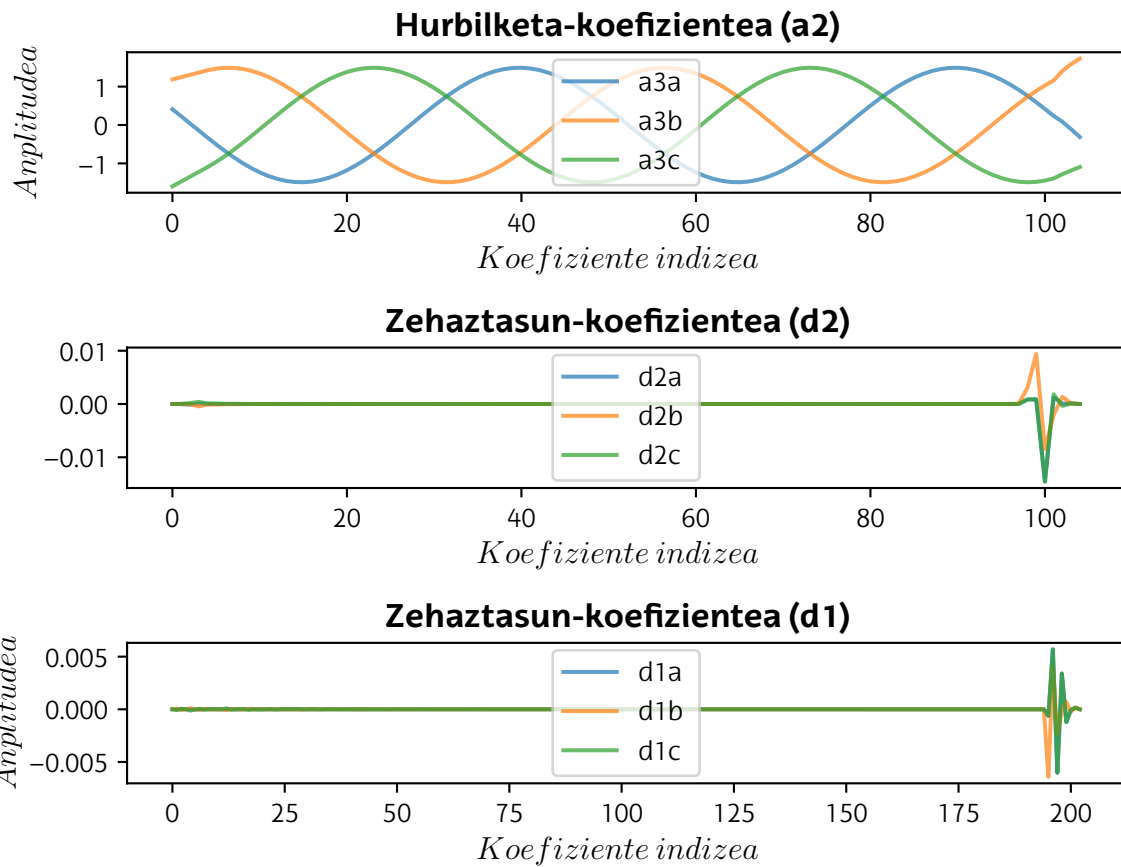
Hau ekiditeko, *wavedec* komandoaren *mode* parametroa erabiliko da, non ertzetako datuak luzatzeko modua finkatzen den. Hauen artean *zero-*, *constant-*, *symmetric-*, *reflect-*, *periodic-*, *smooth-*, *anti-symmetric* eta *anti-symmetric-reflect padding* moduak erabili eta aztertu dira.



8.10 Irudia: Seinale baten *Wavelet* transformatua, non $t = 4$ segundoan c fasean lurre-rako akatsa gertatu den.

Aztertu diren guztietatik soilik *smooth-padding* moduarekin lortu da *edge* efektua ezabatzea (8.11 Irudia). Modu honekin lehenengo eta azken datuen deribatuak kalkulatzen dira eta lortutako malda erabiliz seinaleak lerro zuzen batekin luzatzen dira.

Datuen transformaziorako erabili den *Python* kodea II. Eranskinean aurkitzen da.



8.11 Irudia: Seinale baten *Wavelet* transformatua, non $t = 4$ segundoan c fasean lurre-rako akatsa gertatu den.

8.3.3 Ezaugarrien erauzketa

Datu-leihoetan transformatuak aplikatu eta gero, leiho eta maila bakoitzeko, koefizienteen seinale berriak lortu dira. Seinale berri hauetatik ezaugarriak erauztea funtsezkoa da, ML modeloa datu esanguratsuekin lan egin ahal izateko.

Koefizienteen seinaleetatik hainbat ezaugarri desberdin atera daitezke. Lan honetan haietako zenbait erabili dira, ondoren emaitza hoberenak ematen dituen hautatzeko. Seinaleetatik aterako diren ezaugarriak hurrengoak dira: maximoa, batezbestekoa, deribatuen batezbestekoa, mediana, 25 eta 75 perzentilak, RMS, desbideratze estandarra eta bariantza.

Ezaugarri bakoitzarekin lortutako emaitzak aztertzeko ML modeloen entrenamenduaren ataletan azalduko den prozedura jarraitu da, ezaugarri desberdinak hasierako

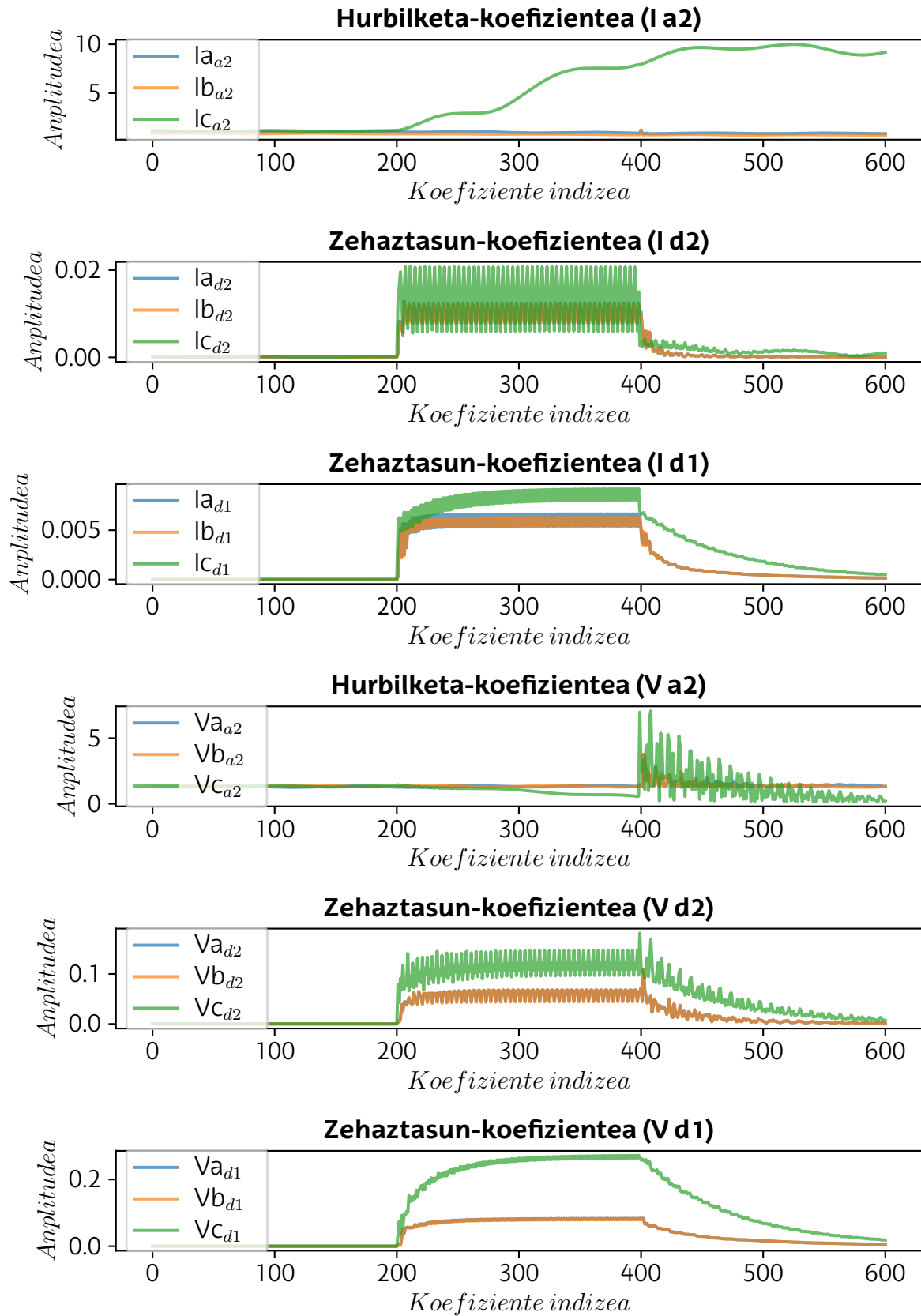
modelo baten entrenamendu eta ebaluazioa egiteko. *XGBoost* izan da hautatu den modeloa pausu honetarako, entrenamendurako behar duen denbora murriztagatik. Prozesua sinplifikatzeko, erabaki da ezaugarri bakoitzarekin lortutako emaitzak ebaluatzeko soilik zehaztasunaren (8.3 Ekuazioa) parametroa erabiltzea.

Ezaugarri hauekin batera, transformatu gabeko datu gordinak erabili dira, datuen transformazioaren eragina ikusi ahal izateko (8.12 Taula).

8.12 Taula: Ezaugarri desberdinekin *XGBoost* modeloan lortutako zehaztasunak.

Ezaugarria	Zehaztasuna	
	Detekzioa	Sailkapena
Maximoa	0,9993	0,9939
Batezbestekoa	0,9980	0,9366
Deribatuen Batezbestekoa	0,9976	0,8438
Mediana	0,9946	0,7517
25 Pertzentila	0,9984	0,9277
75 Pertzentila	0,9983	0,9293
RMS	0,9994	0,9902
Desbideratze estandarra	0,9994	0,9905
Bariantza	0,9994	0,9902
Datu Gordinak	0,9423	0,8116

Zehaztasunik altuena transformatuen maximoa eta desbideratze estandarrekin lortzen da. Transformatuaren koefiziente seinaleen maximoa erabiliz akats moten sailkapenean zehaztasun altuagoa lortzen da, eta desbideratze estandarra erabiliz, ordea, akatsen detekzioan. Ordea, seinaleen balio maximoa jatorrizko seinalearen zaratara sentikorra izango dela kontsideratu denez, desbideratze estandarra erabiltzea erabaki da (8.12 Irudia).



8.12 Irudia: seinale baten *Wavelet* koeffizienteetatik lortutako ezaugarriak.

8.3.4 Ezaugarrien hautaketa

Ondoren, lortutako ezaugarri guztietatik esanguratsuenak direnak bilatuko dira, haien artean erredundanteak diren ezaugarriak egon daitezkeelako. Datuen transformazioaren bidez hamazortzi ezaugarri lortu dira (hiru koefiziente tentsio eta korrante neurketa bakoitzeko). Hauen artean, datuen klaseari buruz informazio gehien ematen duten ezaugarriak bilatuko dira, lortutako zehaztasuna ia murriztu barik.

Horretarako, *Exhaustive Feature Selection* (EFS) metodoa erabiliko da *XGBoost* modeloan. Metodo honen bitartez, ezaugarrien konbinazio guztiekin lortzen den zehaztasuna ebaluatzen da. Iterazio kopurua murrizteko, ezaugarriak sei multzotan batu dira: korranteen hiru koefizienteak eta tentsioen hiru koefizienteak. Ondorioz, ezingo da fase bakarraren koefiziente bat hautatu, hiru faseak batera hautatu beharko dira.

Prozesu hau detekzio eta sailkapen modeloetan jarraitu da, berriz ere modeloen zehaztasuna erabiliz ezaugarri multzo desberdinak ebaluatzeko (8.13 Taula).

8.13 Taula: Ezaugarri desberdinekin *XGBoost* modeloan lortutako zehaztasunak.

Detekzioa		Sailkapena	
Ezaugarriak	Zehaztasuna	Ezaugarriak	Zehaztasuna
I_{d2}	0,9869	I_{d1}, V_{d1}	0,8794
I_{d2}, I_{a2}	0,9993	I_{d1}, V_{d1}, I_{a2}	0,9866
I_{d2}, I_{a2}, V_{a2}	0,9993	$I_{d1}, V_{d1}, I_{a2}, V_{d2}$	0,9872
$I_{d2}, I_{a2}, V_{a2}, V_{d1}$	0,9993	$I_{d1}, V_{d1}, I_{a2}, V_{d2}, I_{d2}$	0,9878

Ikusi daitekeenez, akatsen detekziorako beharko den ezaugarri kopurua murrizta izango da. Soilik bi ezaugarri multzorekin (fase guztiak kontuan hartuta sei ezaugarri totalen) ezaugarri guztiak erabilia lortzen den zehaztasun oso antzeko bat lortzen da (-0.0001).

Akatsen sailkapenerako, ordea, ezaugarri gehiago behar dira. Hiru ezaugarri multzorekin zehaztasun antzeko bat lortzen da (-0.0039), baina gehiago erabiliz zehaztasuna pixka bat hobetzen da.

8.4 Detekzio-modeloa

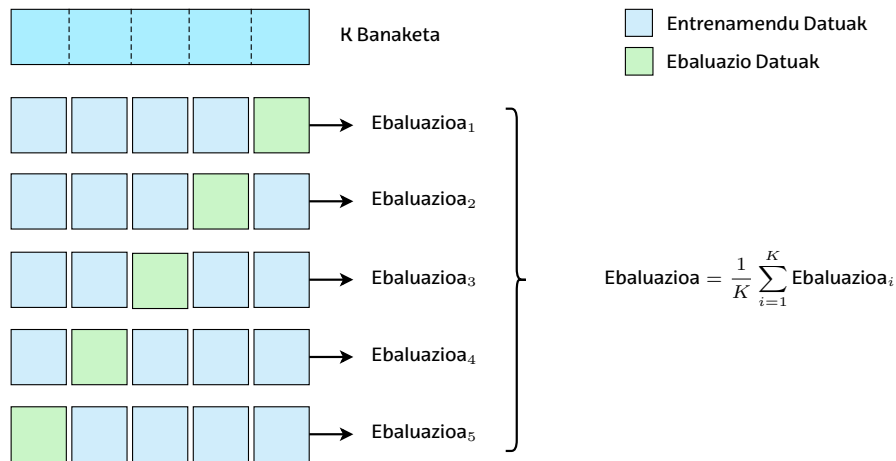
Pausu hauetan ML modeloen entrenamendua, aukeraketa eta hiperparametroen doitzea burutuko da. Detekzio-modeloa problema bitarra sailkatuko du, datuak egoera normalekoak edo akats egoerakoak diren sailkatuz. Horretarako, datuen F etiketaren balioa erabiliko da entrenamenduan.

8.4.1 Modeloen entrenamendua

ML modeloen entrenamenduan, datuak bi multzotan banatzen dira: entrenamendu datuak eta ebaluazio datuak. Modeloa datuen gehiengoarekin entrenatzen da, eta ostean, entrenamenduan ikusi ez dituen datuak erabiltzen dira modeloaren funtzionamendua ebaluatzeko. Modu honetan, modeloaren gaindoiketa ebaluatu daiteke. ML modeloak datu berdinekin entrenatu eta ebaluatuko balira, ebaluazio emaitza oso onak lortuko ziren beti.

Ordea, datuen klaseak orekatuak ez badaude, edo entrenamendu datu multzoak datuen patroia esanguratsuak barneratzen ez badituzte, datuen banaketa desberdinekin emaitza desberdinak lortuko dira.

Hau ekiditeko, *cross-validation* edo balidazio gurutzatu metodoa erabili da. Metodo honetan datuen entrenamendu eta ebaluazio banaketa desberdinak egiten dira. Ondoren, datu banaketa bakoitzarekin entrenamendua eta ebaluazioa burutzen da. Azkenik, banaketa guztiekin lortutako ebaluazio parametroen batezbestekoa ateratzen da (8.13 Irudia). Lan honetan 5 azpimultzoko balidazioa erabili da.



8.13 Irudia: *Cross-validation* metodoaren eskema.

8.4.2 Modeloaren aukeraketa

Lehen azaldu den moduan, lan honetan *shallow learning* ML modeloak erabiliko dira, datuen transformazioaren bitartez datuen klase desberdinak oso bereizgarriak izatea lortu delako. Aztertuko diren modeloak hurrengoak dira: RF, KNN eta *XGBoost*. Beste modeloak baztertu dira, datu-multzo handiekin entrenamendu denbora handiegia dituztelako.

Modeloen ebaluaziorako, zehaztasunaz gainera, parametro desberdinak erabili daitezke. Haien artean erabilienak *accuracy*, *precision*, *recall*, *F1 score* eta *Receiver Operating Characteristic Area Under Curve* (ROC AUC).

- **Accuracy:** Ondo sailkatutako datuen eta datu guztien arteko erlazioa.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8.3)$$

- **Precision:** Akats elektrikoek datuak zuzen sailkatzeko gaitasuna.

$$Precision = \frac{TP}{TP + FP} \quad (8.4)$$

- **Recall:** Akats elektrikoek datuak oker ez sailkatzeko gaitasuna.

$$Recall = \frac{TP}{TP + FN} \quad (8.5)$$

- **F1 Score:** Akats elektrikoak datuak oker ez sailkatzeko gaitasuna.

$$Accuracy = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} \quad (8.6)$$

- **ROC AUC:** ROC kurbaren azpiko azalera adierazten du. ROC kurbak TPR eta FPR balioen erlazioa adierazten du.

Non *TP true positive* edo positibo zuzenak, *FP false positive* edo positibo okerrak, *TN true negative* edo negatibo zuzenak, *FN false negative* edo negatibo okerrak diren, *TPR true positive ratio* edo positibo zuzenen ratioa eta *FPR false positive ratio* edo positibo okerren ratioa diren.

Ebaluazio-metrikoa desberdinak aztertuz, akatsen detekziorako *recall* parametroa bereziki interesgarria izan daiteke, akatsik ez daudenean modeloak positibo okerrak ematen ez dituela egiaztatzeko. Hau da, positiboak soilik ematen dira akatsak daudenean.

8.14 Taula: Detekzio modelo desberdinen ebaluazio metrikak.

Modeloa	Accuracy	Precision	Recall	F1 Score	ROC AUC
RF	0,9992	1,0000	0,9989	0,9994	0,9996
KNN	0,9992	1,0000	0,9989	0,9994	0,9995
XGBoost	0,9993	1,0000	0,9989	0,9994	0,9996

Emaitzak aztertuz, nabaria da RF, KNN eta XGBoost modeloekin emaitza oso onak lortzen direla, haien artean oso antzekoak. Haietatik XGBoost modelo zehaztasun altuena lortu duena izan arren, hautatuko den modelo RF izango da, alde oso txikia da-goelako, eta hurrengo atalean ikusiko den moduan, sailkapen modelorako baita RF aukeratu delako. Modeloen entrenamendu eta ebaluaziorako erabili den kodea III. Eranskinean aurkitzen da.

8.4.3 Modeloaren hiperparametroen doiketa

Modelo baten hiperparametroak entrenamendu fasetik kanpo finkatzen diren parametroak dira. Parametro hauek modeloaren araberakoak dira, eta RF modelo-

ren kasuan, hurrengoak dira orokorrean doitzen direnak: *n_estimators*, *max_features*, *max_depth*, *min_samples_split*, *min_samples_leaf* eta *bootstrap*.

Parametro hauen doiketa saiakuntza eta errorearen bitartez egiten da. Honetarako, metodo desberdinak erabili daitezke. Haien artean erabilienak *grid search*, *random search* eta *bayesian optimization* metodoak dira.

- **Grid search:** Metodo honen bitartez, hautatutako hiperparametroen eta finkatutako balio tarte guztien arteko konbinazio guztiak ebaluatu egiten dira. Hiperparametro asko edo balio tarte handiekin lan egiten denean kostu konputazional handia dauka.
- **Random search:** Metodo honetan, konbinazio guztiak probatu beharrean, ausaz hautatutako konbinazio desberdinak ebaluatzen dira. Honela, balio tarte handiagoak erabili daitezke, kostu konputazional gutxiagorekin.
- **Bayesian optimization:** Metodo honek hiperparametroen probabilitate modeloak erabiltzen ditu parametroen balio egokiak topatzeko. Hiperparametroen alaketek zehaztasunean daukaten eragina aztertuz, modu eraginkor baten bilatzen ditu parametroen balioak.

Lan honetan *random search* metodoa erabiltzea erabaki da, nagusiki, iterazio desberdinak paraleloan ebaluatzeko eskaintzen duen aukera eta duen kostu konputazional murriztuagoa dela eta.

8.15 Taula: Detekzio modeloaren hiperparametro balio egokienak.

Parametroa	Balioa
<i>n_estimators</i>	500
<i>min_samples_split</i>	2
<i>min_samples_leaf</i>	2
<i>max_features</i>	sqrt
<i>max_depth</i>	None
<i>bootstrap</i>	False
Zehastasuna	0,999245

Hiperparametroen doiketa aurretik lortutako modeloaren zehaztasuna 0.999226 baliokoa da. Emaitzak aztertuta (8.15 Taula), ikusi daiteke modeloaren zehaztasuna ez dela ia hobetzen, aurretik zehaztasun oso altua lortu delako. *Random search* metodoaren bitartez, lortu da RF modeloak 500 $n_estimator$ edo DT erabiliz zehaztasuna pixka bat altuagoa dela. DT gehiagorekin, modeloaren entrenamendu eta exekuzio denbora handitzen da. Modeloaren $n_estimator$ parametroaren balio lehenetsia 100ekoa da, eta funtzionamendu optimo baterako balio hau nahikoa izan daiteke.

8.5 Sailkapen modelo

Sailkapen modelorako, detekzio modeloan jarraitu diren pausu berdinak jarraitu dira. Ordea, modelo honek bi klase baino gehiago sailkatzen ditu. Sailkapen problema hau modu desberdinetan sailkatu daiteke. Hasiera baten, datuen etiketak *multi-label* problema baten modura gorde dira, hau da, datu bakoitza klase bat baino gehiagotan egon daiteke (A, B, C edo G etiketen konbinazioko klasekoak izan ahal dira).

Lan honetan, problema *single-label multiclass* bezala ebaztea erabaki da, hau da, klase konbinazio bakoitzerako klase berri bat sortuz (AG, BG, CG, AB, etab.). Horretarako, datuetan etiketa zutabe berri bat sortu da, klase bakoitzari 0 - 9 tarteko balioak esleitzuz.

8.5.1 Modeloaren aukeraketa

Modelo eta ebaluazio parametro berdinak erabili dira sailkapen modelorako (8.16 Taula).

8.16 Taula: Sailkapen modelo desberdinen ebaluazio metrikak.

Modeloa	Accuracy	Precision	Recall	F1 Score	ROC AUC
RF	0,9956	0,9956	0,9956	0,9956	0,9999
KNN	0,9930	0,9930	0,9930	0,9930	0,9987
XGBoost	0,9866	0,9866	0,9866	0,9866	0,9998

Lehen aurreratu den moduan, RF modeloarekin lortzen da zehaztasun altuena. Beraz, erabaki da modelo hau erabiltzea akata-motak sailkatzeko. Ordea, egia da modelo guztiekin lortze den zehaztasuna oso altua dela, eta haietako edozein hautatzea zuzena izan daitekeela. Modelo desberdinen entrenamendu eta ebaluazioaren kodea IV. Eranskinean aurkitzen da.

8.5.2 Modeloaren hiperparametroen doiketa

Atal honetan detekzio modelorako jarraitu diren pausu berdinak eman dira.

8.17 Taula: Sailkapen modeloaren hiperparametro balio egokienak.

Parametroa	Balioa
<i>n_estimators</i>	500
<i>min_samples_split</i>	5
<i>min_samples_leaf</i>	1
<i>max_features</i>	sqrt
<i>max_depth</i>	None
<i>bootstrap</i>	False
Zehaztasuna	0,996275

Aurreko kasuan bezala, oso zehaztasun antzekoa lortzen da hiperparametroen doiketa burutu ondoren.

8.6 Emaitzen balioztatzea

Azkenik, IACC lanbide-institutu eta Valparaiso Unibertsitateko Informatika Ingeniaritza Eskolako ikertzaile eta irakasle baten lankidetzan, lortutako emaitzak balioztatu dira.

Datu-multzo berdina erabiliz, *deep learning* motako ML modelo bat erabili da, beste metodologia baten bitartez emaitzak konparatzeko. Erabilitako metodologia hurren-



goa da:

- **Datuen transformazioa:** Kasu honetan, datuak *continuous wavelet* transformatuarekin prozesatu dira, eskalograma baten bitartez transformatuen irudikapen grafiko bat lortuz. Datu-multzo osotik, 6000 irudi inguru lortu dira.
- **ML modeloa:** Aurretik entrenatutako sare konboluzionala (CNN) erabili da, *EfficientNetV2B3* sarea, aurreko pausuan lortutako irudiak sailkatzeko.

Metodologia honen bitartez, detekzioa %100 zehaztasunarekin burutu da, eta sailkapena %94 inguruko zehaztasunarekin. Beraz, lan honetan jarraitutako metodologia egokia izan dela ondorioztatu da.

9. Planifikazioa

Atal honetan lanean jarraitutako planifikazioa deskribatzen da. Lehenengoz, lana burutzeko finkatu diren zereginak azalduko dira. Ondoren, lanaren *Gantt* diagrama osatuko da.

9.1 Zereginen deskribapena

Planifikazioa, lanaren irismena jarraituz antolatu da, eta hurrengo bost faseetan banandu da:

1. Fasea: Formakuntza eta dokumentazioa

1. **Ataza:** Proiektuari buruzko formakuntza.

Lanaren lehenengo pausua GSGIH proiektuari buruzko formakuntza izan da. Ataza honetan, proiektuaren barnean burutu diren lan desberdinen ikasketa egin da, proiektuaren egoera ulertzeko.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *MATLAB/Simulink*.
- Iraupena: 6 aste (120h).

2. **Ataza:** Dokumentazioa.

Ondoren, lanean erabiliko diren teknologia desberdinen informazioa bilatu da, *IEEE Xplore* eta *ScienceDirect* bezalako erremintak erabiliz. Honekin batera, lanaren artearen egoera osatu da.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *Overleaf*.
- Iraupena: 8 aste (160h).

3. **Ataza:** *Machine Learning* tekniken inplementazioa ikasi.

Ataza honetan, *Machine Learning* teknikak ikasi eta inplementatzeko gaitasunak garatu dira. Inplementazioa *Python* programazio lengoaian egin da, *Visual Studio Code* kode editorean.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *Python*, *Visual Studio Code*.
- Iraupena: 4 aste (80h).

2. Fasea: Datuen sorkuntza

4. **Ataza:** Modelo elektrikoaren garapena.

Ataza honetan proiektuaren plataforma experimentalaren modelo elektrikoa garatu da, *Simulink* erabiliz.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *MATLAB/Simulink*.
- Iraupena: 2 aste (40h).

5. **Ataza:** Modeloaren simulazioak.

Garatutako modelo elektrikoarekin simulazioak egingo dira, ondoren modeloen entrenamendurako erabiliko diren datuak sortzeko.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *MATLAB/Simulink*.
- Iraupena: 3 aste (60h).

3. Fasea: Datuen prozesamendua

6. **Ataza:** Datuen aurreprozesamendua.

Lortutako datuen aurreprozesamendua egin da, datuei etiketak emanez, eta neurketa seinaleak eskalatuz.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *MATLAB/Simulink*.
- Iraupena: 1 aste (20h).

7. **Ataza:** Datuen transformazioa.

Ondoren, datuen transformazioa burutu da, *wavelet* transformatua erabiliz.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *Python, Visual Studio Code*.
- Iraupena: 2 aste (40h).

8. **Ataza:** Ezaugarrien erauzketa eta aukeraketa.

Behin datuak transformatuta, lortutako datu berrietatik ezaugarriak erauzi dira. Gero, ezaugarri esanguratsuenak hautatu dira.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *Python, Visual Studio Code*.
- Iraupena: 1 aste (20h).

4. Fasea: Modeloen entrenamendua eta ebaluazioa

9. **Ataza:** Detekzio-modeloaren entrenamendu eta ebaluazioa.

Detekzio-modelorako aukeratutako ezaugarriak erabiliz, modelo desberdinak entrenatu eta ebaluatu dira.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *Python, Visual Studio Code*.
- Iraupena: 2 aste (40h).

10. **Ataza:** Sailkapen-modeloaren entrenamendu eta ebaluazioa.

Aurreko atazaren pausu berdina sailkapen-modelorako jarraitu dira.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *Python, Visual Studio Code*.

- Iraupena: 2 aste (40h).

11. **Ataza:** Emaitzen balioztatzea.

Lanean lortutako emaitzak beste metodologia bat jarraituz lortutako emaitzekin konparatu dira.

- Giza baliabideak: Ikaslea, MAL zuzendaria, ikertzailea.
- Baliabide teknikoak: Ordenagailua, *Python*, *Visual Studio Code*.
- Iraupena: 1 aste (20h).

5. Fasea: Idazketa

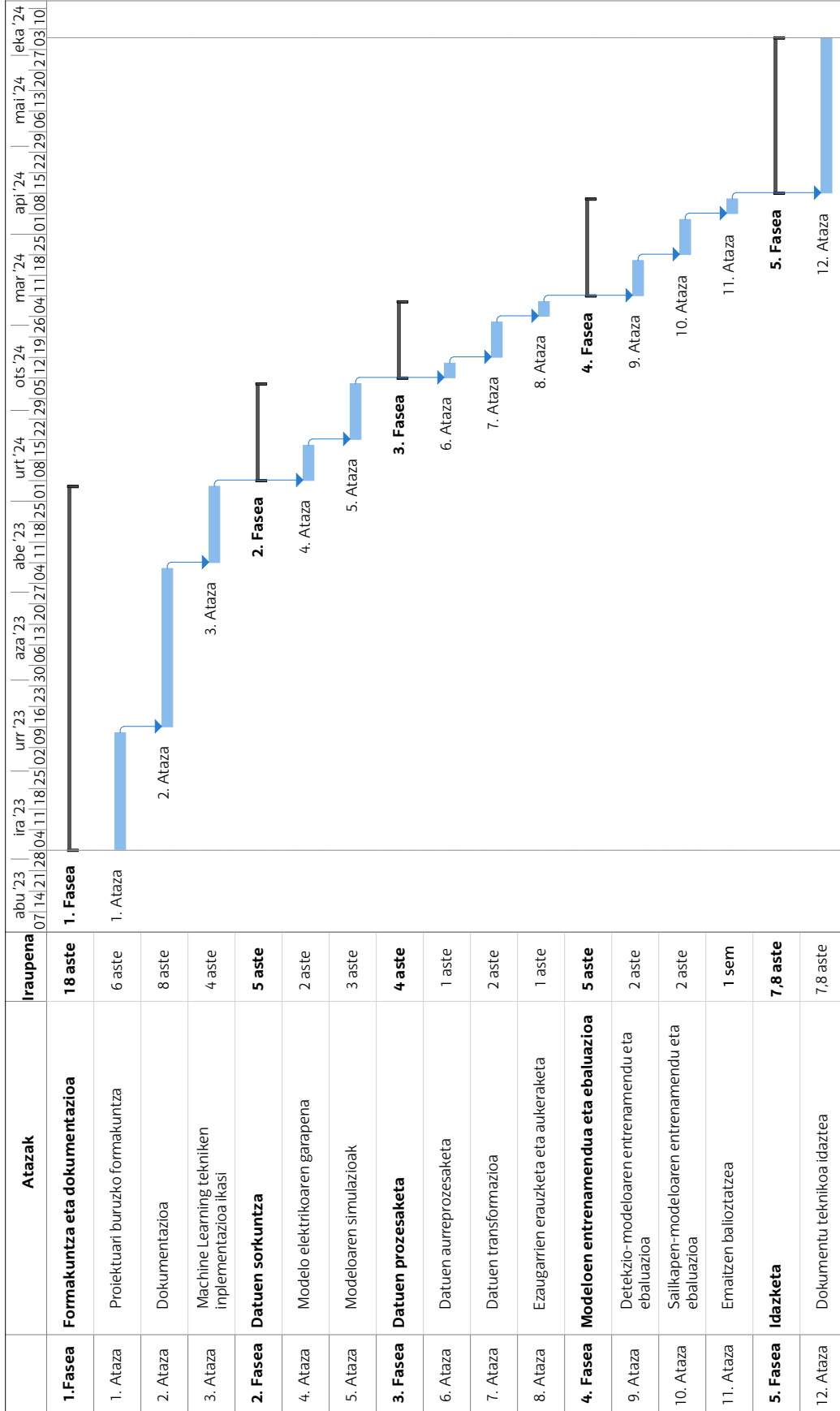
12. **Ataza:** Dokumentu teknikoa idaztea.

Behin lanaren emaitzak balioztatu direla, dokumentu teknikoa idatzi da.

- Giza baliabideak: Ikaslea, MAL zuzendaria.
- Baliabide teknikoak: Ordenagailua, *Overleaf*.
- Iraupena: 7,8 aste (156h).

9.2 **Gantt diagrama**

Behin zereginak eta haien baliabideak eta iraupena finkatuta, proiektuaren zereginak denboran izan duten kokapena ikusi ahal izateko *Gantt* diagrama osatu da (9.1 Iru-dia).



9.1 Irudia: Lanaren Gantt diagrama.

10. Alderdi ekonomikoak

Atal honetan, lanaren alderdi ekonomikoak landuko dira. Proiektuaren kostuak barne orduetan, amortizazioetan eta gastuetan banandu dira. Ostean, koste totala kalkulatu da.

10.1 Barne orduan

Barne orduetan, lanaren parte-hartzaileak egindako lan orduen kostua adierazten da (10.1 Taula).

- **Zuzendaria:** 100h inguruko lan orduak, 50,00 €/h orduko kostuarekin.
- **Ikaslea:** 796h inguru lan orduak, 6,66 €/h orduko kostuarekin.

10.1 Taula: 3.1 Taula: Barne orduak.

Kontzeptua	Orduko kostua (€/h)	Orduak (h)	Kostua (€)
Zuzendaria	50,00	100	5.000,00
Ikaslea	6,66	796	5.300,00
Guztira			10.300,00

10.2 Amortizazioak

Amortizazioak lanean zehar erabilitako tresnak, beste lan batzuetan erabili daitezkeenak, galdutako balioaren kostua adierazten du (10.2 Taula).

10.2 Taula: 3.2 Taula: Amortizazioak.

Kontzeptua	Kostua (€)	Bizi erab. (h)	Erabilera (h)	Kostua (€)
Ordenagailua	1.000,00	10.000	796	79,60
Matlab student	69,00	2.000	600	20,70
Guztira				100,30

10.3 Gastuak

Gastuak lan honetarako soilik erabiliko diren baliabideen kostuak adierazten ditu (10.3 Taula).

10.3 Taula: Taula 9.3: Gastuak.

Kontzeptua	Kostua (€/ud)	Ud	Guztira (€)
Bulego materiala	50	1	50
Guztira			50

10.4 Kostu totala

Ondoren, lan osoaren kostu totala adierazten da (10.4 Taula).

10.4 Taula: 3.4 Taula: Kostu totala.

Kontzeptua	Kostua (€)
Barne orduak	10.300,00
Amortizazioak	100,30
Gastuak	50,00
Azpi-totala	10.450,30
Kostu ez-zuzenak (%7)	731,52
Guztira	11.181,82

11. Ondorioak

Atal honetan, lanaren ondorioak eta etorkizunerako lan-ildoak laburbilduko dira.

Lan honetan, sistema elektriko baten akats elektrikoak detektatu eta sailkatzeko gai den Adimen Artifizialeko modelo bat garatu da. Lan hau GSGIH proiektuan garatutako plataforma esperimentaleko sistema batean oinarritu da, eta helburu nagusia akats elektrikoak modu zehatz eta azkar batean detektatzea eta sailkatzea izan da.

Hasieran finkatutako helburu guztiak bete egin dira. Alde batetik, akats elektrikoaren datu-multzo bat sortu da. Sortutako datu-multzoa sistema elektrikoaren hainbat gertakari kontuan hartzen ditu, hala nola akats-mota, akats-erresistentzia, akats-distantzia, HIF, desorekak eta karga-maila desberdinak. Emaitzak ikusita, ondorioztatu da ML teknikak egokiak direla sistema elektrikoaren gertakari desberdinekin detekzio eta sailkapen egoki bat egiteko.

Beste alde batetik, ezaugarriak erauzteko datuen transformazioa egin da. Honen birtartez, datu-multzo konplexu batean *shallow learning* modeloak erabili ahal izan dira zehaztasun oso altuekin. Beraz, ondorioztatu da datuen transformazio egoki batekin sistema elektrikoaren gertaera desberdinak eta akats elektrikoak ondo bereiztea posiblea dela.

Azkenik, ML modelo desberdinak erabili eta ebaluatu dira. Ordea, aurreko pausuko transformazioa eta gero, aztertu diren modelo guztiekin zehaztasun oso altuak lortu dira, detekzioan bai sailkapenean. Ez dira modelo gehiago aztertu, sortutako datu-multzoa oso handia denez, entrenamendu denborak altuegiak behar dituztelako. Ondorioz, aztertu da edozein modelo baliagarria izan daitekeela emaitza onak lortzeko. Ordea, beste *shallow learning* modelo batzuk erabiltzeko, datu-multzoa murriztu beharko da.

Gainera, lortutako emaitzak datu-multzo berdinean frogatu den metodologia desberdinarekin lortutako emaitzekin konparatu dira. Beste metodologian, aurretik entre-

natutako *EfficientNetV2B3* sare konboluzionala erabili da, hau da, *deep learning* modelo bat. Lan honetan, detekzioan emaitza antzekoak, baina sailkapenean emaitza hobekak lortu dira. Hemen, ikusi daiteke *deep learning* metodoekin zehaztasun hain onak lortzeko entrenamendu esfortzu handiagoak egin behar direla, baina aurretik entrenatutako modeloak erabiltzea aukera ona izan daitekeela entrenamendu denbora murriztu batekin emaitza antzekoak lortzeko.

11.1 Etorkizunerako lan-ildo posibleak

Lan honek ML tekniken bidez akats elektrikoaren detekzioa eta sailkapena bideragarria dela ondorioztatzen du. Ordea, lana simulazio datuekin burutu denez, metodologiak duen aplikagarritasun industrialaz aztertzea falta da. Hau lortzeko, etorkizunerako planteatzen diren aukerak hiru dira:

- Alde batetik, datu errealak erabiltzen dituen metodologia bat inplementatu daiteke, plataforma esperimentalarekin lortutako datuak erabiliz. Akats elektrikoak HIF motakoak izan beharko dira,
- Beste alde batetik, simulazio-datuekin lortutako modeloak datu erreal murriztza goekin balioztatzea nahikoa izan daiteke aplikagarritasun industrialaz frogatzeko.
- Azkenik, simulazioetan erabilitako sistema elektriko konplexuago egin daiteke, datu errealen ezaugarriak erreplikatu ahal izateko.

Bestalde, etorkizunean *deep learning* modelo desberdinak inplementatzea eta frogatzea proposatzen da. Horrela, datuen transformazio sinpleagoen erabilera aztertu ahalko da, modeloen bidez ezaugarriek zuzenean erauziz.

Erreferentziak

- [1] S. Bayhan, H. Abu-Rub, J. I. Leon, S. Vazquez, and L. G. Franquelo, "Power electronic converters and control techniques in ac microgrids," in *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, 2017, pp. 6179–6186.
- [2] United Nation Department of Economic and Social Affairs, *Informe de los Objetivos de Desarrollo Sostenible 2023: Edición especial*, 2023rd ed. United Nations, 2023.
- [3] International Energy Agency, "Number of people without access to electricity by region, 2010-2023," 2023, IEA, Paris, Licence: CC BY 4.0. [Online]. Available: <https://www.iea.org/data-and-statistics/charts/number-of-people-without-access-to-electricity-by-region-2010-2023>
- [4] Y. Bansal and R. Sodhi, "Microgrid fault detection methods: Reviews, issues and future trends," in *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, 2018, pp. 401–406.
- [5] S. Parhizi, H. Lotfi, A. Khodaei, and S. Bahramirad, "State of the art in research on microgrids: A review," *IEEE Access*, vol. 3, pp. 890–925, 2015.
- [6] G. Peter, A. A. Stonier, A. Sherine, and S. Iderus, "A review about zig zag and double zig zag connections in transformer," in *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2022, pp. 1701–1704.
- [7] J. Vernieri, M. Barbieri, and P. Arnera, "Consequence of an unbalanced supplying condition on a distribution transformer," in *2001 IEEE Porto Power Tech Proceedings (Cat. No.01EX502)*, vol. 4, 2001, pp. 6 pp. vol.4–.
- [8] P. P.-D. Xose M. Lopez-Fernandez, C. Alvarez-Mariño and J. Turowski, "Rnm2d_0 fast stray losses hazard evaluation on transformer tank wall & cover due to zero

- sequence," in *Advanced Research Workshop on Transformers*, vol. 3rd International Advanced Research Workshop on Transformers, 2010, pp. 338–343.
- [9] E. Sorrentino Ramírez, "Determinación de las impedancias homopolares de un transformador trifásico de tres columnas con conexión ynynd a partir de sus dimensiones geométricas," Ph.D. dissertation, UC3M, 2014.
- [10] P. P.-D. Xose M. Lopez-Fernandez, C. Alvarez-Mariño and J. Turowski, "Analysis of a three-limb core power transformer under earth fault," in *ICEM*, vol. 16th International Conference on Electrical Machines, 2004, pp. 338–343.
- [11] N. Tleis, *Power Systems Modelling and Fault Analysis: Theory and Practice*, ser. Newnes Power Engineering Series. Elsevier Science, 2007.
- [12] G. Buigues Beraza, "Metodología para la detección y localización de faltas en redes de distribución con puesta a tierra activa," Ph.D. dissertation, UPV/EHU, 2011.
- [13] A. Raza, A. Benrabah, T. Alquthami, and M. Akmal, "A review of fault diagnosing methods in power transmission systems," *Applied Sciences*, vol. 10, no. 4, 2020. [Online]. Available: <https://www.mdpi.com/2076-3417/10/4/1312>
- [14] J. C. J. Theron, A. Pal, and A. Varghese, "Tutorial on high impedance fault detection," in *2018 71st Annual Conference for Protective Relay Engineers (CPRE)*, 2018, pp. 1–23.
- [15] Y. Bansal and R. Sodhi, "Microgrid fault detection methods: Reviews, issues and future trends," in *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, 2018, pp. 401–406.
- [16] A. A. Memon and K. Kauhaniemi, "A critical review of ac microgrid protection issues and available solutions," *Electric Power Systems Research*, vol. 129, pp. 23–31, 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378779615002163>
- [17] F.Noor, R. Arumugam, and M. Vaziri, "Unintentional islanding and comparison of prevention techniques," in *Proceedings of the 37th Annual North American Power Symposium*, 2005, pp. 90–96.

- [18] A. Supannon and P. Jirapong, "Recloser-fuse coordination tool for distributed generation installed capacity enhancement," in *2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, 2015, pp. 1–6.
- [19] M. Jamil, S. K. Sharma, and R. Singh, "Fault detection and classification in electrical power transmission system using artificial neural network," *SpringerPlus*, vol. 4, no. 1, p. 334, 2015. [Online]. Available: <https://doi.org/10.1186/s40064-015-1080-x>
- [20] E. B. M. Tayeb and O. A. A. A. Rhim, "Transmission line faults detection, classification and location using artificial neural network," in *2011 International Conference & Utility Exhibition on Power and Energy Systems: Issues and Prospects for Asia (ICUE)*, 2011, pp. 1–5.
- [21] B. K. Sahoo, S. Pradhan, B. K. Panigrahi, B. Biswal, N. C. Patel, and S. Das, "Fault detection in electrical power transmission system using artificial neural network," in *2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)*, 2020, pp. 1–4.
- [22] P. K. Ray, B. K. Panigrahi, P. K. Rout, A. Mohanty, and H. Dubey, "Detection of faults in power system using wavelet transform and independent component analysis," *CoRR*, vol. abs/1609.08650, 2016. [Online]. Available: <http://arxiv.org/abs/1609.08650>
- [23] L. Sun, L. Muguira, J. Jiménez, J. Lázaro, and W. Yong, "High performance platform to detect faults in the smart grid by artificial intelligence inference," *IEEE Transactions on Smart Grid*, vol. 15, no. 1, pp. 504–512, 2024.
- [24] M. Wadi and W. Elmasry, "An anomaly-based technique for fault detection in power system networks," in *2021 International Conference on Electric Power Engineering – Palestine (ICEPE- P)*, 2021, pp. 1–6.
- [25] T. V. Addison Howard, Sohier Dane, "Vsb power line fault detection," 2018. [Online]. Available: <https://kaggle.com/competitions/vsb-power-line-fault-detection>
- [26] J. V. Souza, G. Lopes, J. C. M. Vieira, and E. Asada, "High impedance fault detection in distribution systems: An approach based on fourier transform and artificial neural

- networks," in *2020 Workshop on Communication Networks and Power Systems (WCNPS)*, 2020, pp. 1–6.
- [27] A. Y Imaz and G. Bayrak, "Real-time disturbance detection using stft method in microgrids," *Academic Perspective Procedia*, vol. 2, pp. 1115–1121, 11 2019.
- [28] N. Zhang and M. Kezunovic, "Transmission line boundary protection using wavelet transform and neural network," *IEEE Transactions on Power Delivery*, vol. 22, no. 2, pp. 859–869, 2007.
- [29] K. Sekar, N. K. Mohanty, and A. K. Sahoo, "High impedance fault detection using wavelet transform," in *2018 Technologies for Smart-City Energy Security and Power (ICSESP)*, 2018, pp. 1–6.
- [30] V. Torres, H. F. Ruiz-Paredes, R. V. Venegas, Z. S. Ramírez, and G. J. P. Torres, "Hif analysis using discrete wavelet transform," in *2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, 2017, pp. 1–6.
- [31] T. Tiwari, T. Tiwari, and S. Tiwari, "How artificial intelligence, machine learning and deep learning are radically different?" *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 8, p. 1, 03 2018.
- [32] A. Abraham Iorkaa, M. Barma, and H. Muazu, "Machine learning techniques, methods and algorithms: Conceptual and practical insights," *International Journal of Engineering Research and Applications*, vol. 11, pp. 55–64, 08 2021.
- [33] V. Gupta, V. K. Mishra, P. Singhal, and A. Kumar, "An overview of supervised machine learning algorithm," in *2022 11th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 2022, pp. 87–92.
- [34] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, p. 160, Mar 2021. [Online]. Available: <https://doi.org/10.1007/s42979-021-00592-x>
- [35] K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 1255–1260.

- [36] D. Martens, B. Baesens, and T. Van Gestel, "Decompositional rule extraction from support vector machines by active learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 2, pp. 178–191, 2009.
- [37] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: <https://doi.org/10.1023/A:1010933404324>
- [38] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S002200009791504X>
- [39] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, S. Dasgupta and D. McAllester, Eds., vol. 28, no. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1310–1318. [Online]. Available: <https://proceedings.mlr.press/v28/pascanu13.html>
- [40] C. Xu, J. Shen, X. Du, and F. Zhang, "An intrusion detection system using a deep neural network with gated recurrent units," *IEEE Access*, vol. 6, pp. 48 697–48 707, 2018.
- [41] The MathWorks, Inc., "Mathworks." [Online]. Available: <https://www.mathworks.com>
- [42] A. Delavari, I. Kamwa, and P. Brunelle, "Simscape power systems benchmarks for education and research in power grid dynamics and control," in *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, 2018, pp. 1–5.
- [43] PowerWorld Corporation, "Powerworld." [Online]. Available: <https://www.powerworld.com>
- [44] S. Syafaruddin and S. Latief, "Lesson learned from power system design with powerworld simulator," in *2018 Conference on Power Engineering and Renewable Energy (ICPERE)*, 2018, pp. 1–6.

- [45] DlgSILENT GmbH, "Powerfactory." [Online]. Available: <https://www.digsilent.de>
- [46] M. Stifter, R. Schwalbe, F. Andrén, and T. Strasser, "Steady-state co-simulation with powerfactory," in *2013 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, 2013, pp. 1–6.
- [47] I. Stan in and A. Jovi , "An overview and comparison of free python libraries for data mining and big data analysis," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MI-PRO)*, 2019, pp. 977–982.
- [48] J. R. Barr, "Machine learning, a tutorial with r," in *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*, 2018, pp. 120–121.
- [49] B. Boehmke and B. Greenwell, *Hands-On Machine Learning with R*, ser. Chapman & Hall/CRC The R Series. CRC Press, 2019.
- [50] A. Ghatak, *Deep Learning with R*. Springer Nature Singapore, 2019.
- [51] A. Emanuel, D. Cyganski, J. Orr, S. Shiller, and E. Gulachenski, "High impedance fault arcing on sandy soil in 15 kv distribution feeders: contributions to the evaluation of the low frequency spectrum," *IEEE Transactions on Power Delivery*, vol. 5, no. 2, pp. 676–686, 1990.
- [52] A. R. Sedighi and M. R. Haghifam, "Simulation of high impedance ground fault in electrical power distribution systems," in *2010 International Conference on Power System Technology*, 2010, pp. 1–7.
- [53] M. Misiti, Y. Misiti, G. Oppenheim, and J.-M. Poggi, "Wavelet Toolbox User's Guide," in *Mathworks*, 1996. [Online]. Available: https://www.mathworks.com/help/pdf_doc/wavelet/wavelet_ug.pdf
- [54] D. Gogolewski, "Influence of the edge effect on the wavelet analysis process," *Measurement*, vol. 152, p. 107314, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0263224119311789>

I. Eranskina. Datuen sorkuntza.

```

1  %% PARAMETERS
2  % Electrical parameters
3  f = 50; % Frequency (Hz)
4  Pn = 10000; % Nominal power (VA)
5  V12 = 400; % Line-to-line voltage (Vph-ph rms)
6  R1 = 0.32; % Resistance (Ohm)
7  L1 = 2.72e-3; % Inductance (H)
8  R2 = 0.191; % Resistance (Ohm)
9  L2 = 2.84e-3; % Inductance (H)
10 Rm = 3.11e4; % Magnetizing resistance (Ohm)
11 Lm = 0.102; % Magnetizing inductance (H)
12 L0 = 0.8 * (L1 + L2); % Zero-sequence inductance (H)
13
14 % Line parameters
15 Rline = [0.2232, 0.4464] / 10; % Resistance per unit
    length (Ohm)
16 Lline = [1.42e-3, 5.34e-3] / 10; % Inductance per unit
    length (H)
17 Cline = [0.672e-6, 0.448e-6] / 10 * 2; % Capacitance per unit
    length (F)
18
19 % Fault and simulation parameters
20 L = 10; % Line length (km)
21 Lf = 5; % Fault distance (km)
22 Rf = 0.25; % Fault resistance (Ohm)
23 Sim_sample = 1e6; % Simulation sample rate (Hz)
24 Sample = 1e4; % Data sample rate (Hz)
25 Fault_time = 0.04; % Fault initiation time (s)
26 Stop_time = 0.08; % Simulation stop time (s)
27 Data_n = round((Stop_time) * Sample); % Number of data points
28
29 % High impedance fault (HIF) parameters
30 Rp_values = [1000, 2900, 3500; 900, 2800, 2500; 1500, 9000,
    6500];
31 Rn_values = [1050, 3000, 3550; 905, 2850, 2550; 1505, 9600,
    7000];
32 Vp_values = [4000, 8000, 7500; 1800, 3000, 4000; 9000, 8000,
    11000];
33 Vn_values = [4500, 8100, 7600; 1900, 3500, 4500; 10000, 9000,
    11050];
34 c = 0.02; % Scaling factor
35
36 % Dataset model
37 model = 'dataset_model';
38 model_HIF = 'dataset_model_HIF';
  
```

```

39
40 %% DATASET UNBALANCED
41 % Load system model
42 load_system(model);
43 dataset_name = 'dataset_def';
44
45 % Check if dataset already exists
46 if isfile(fullfile('Dataset', [dataset_name, '.csv']))
47     error(['Dataset name: ', dataset_name, '.csv', ' is in use'
48         ])
49 end
50 % Initialize CSV file with headers
51 data_tag = ["Ia", "Ib", "Ic", "Va", "Vb", "Vc", "F", "A", "B",
52     "C", "G", "L", "Rf", "Unbalance", "HIF"];
53 writematrix(data_tag, fullfile('Dataset', [dataset_name, '.csv'
54     ]));
55 % Define parameter sets
56 FaultPhase_set = [1 0 0 1; 0 1 0 1; 0 0 1 1; 1 1 0 0; 0 1 1 0;
57     1 0 1 0; 1 1 0 1; 0 1 1 1; 1 0 1 1; 1 1 1 0];
58 Rf_set = [0.01, 0.25, 2.5, 5, 10, 15];
59 L_set = 1:9;
60 P_set = (0.2:0.2:1) * Pn;
61 P_set_unb = (0.1:0.1:1) * Pn;
62 % Generate dataset
63 for FaultPhase = FaultPhase_set'
64     for Lf = L_set
65         for Rf = Rf_set
66             % Set fault parameters in the model
67             set_param([model, '/Three-Phase Fault'], 'FaultA',
68                 FaultPhase(1))
69             set_param([model, '/Three-Phase Fault'], 'FaultB',
70                 FaultPhase(2))
71             set_param([model, '/Three-Phase Fault'], 'FaultC',
72                 FaultPhase(3))
73             set_param([model, '/Three-Phase Fault'], '
74                 GroundFault', FaultPhase(4))
75
76             % Balanced condition
77             [Pa, Pb, Pc] = deal(P_set(randi(length(P_set))));
78             out = sim(model);
79             data = [out.data, repmat(FaultPhase', Data_n, 1),
80                 repmat(Lf, Data_n, 1), repmat(Rf, Data_n, 1),
81                 zeros(Data_n, 1), zeros(Data_n, 1)];
82             writematrix(data, fullfile('Dataset', [dataset_name
83                 , '.csv']), 'WriteMode', 'append');
84
85             % Unbalanced condition
86             Pa = P_set_unb(randi(length(P_set_unb)));
87             Pb = P_set_unb(randi(length(P_set_unb)));
  
```



```

80     Pc = P_set_unb(randi(length(P_set_unb)));
81     Unbalance = abs(max([(Pa - Pb) / Pn, (Pa - Pc) / Pn
      , (Pc - Pb) / Pn], [], "ComparisonMethod", 'abs'
      )));
82     out = sim(model);
83     data = [out.data, repmat(FaultPhase', Data_n, 1),
      repmat(Lf, Data_n, 1), repmat(Rf, Data_n, 1),
      repmat(Unbalance, Data_n, 1), zeros(Data_n, 1)];
84     writematrix(data, fullfile('Dataset', [dataset_name
      , '.csv']), 'WriteMode', 'append');
85     end
86   end
87 end
88
89 %% DATASET UNBALANCED HIF
90 % Load HIF system model
91 load_system(model_HIF);
92 dataset_name = 'dataset_def_HIF';
93
94 % Check if dataset already exists
95 if isfile(fullfile('Dataset', [dataset_name, '.csv']))
96     error(['Dataset name: ', dataset_name, '.csv', ' is in use'
97         ])
98 end
99 % Initialize CSV file with headers
100 writematrix(data_tag, fullfile('Dataset', [dataset_name, '.csv'
101     ]));
102 % Generate HIF dataset
103 for FaultPhase = FaultPhase_set'
104     for Lf = L_set
105         for i = 1:3
106             Rp = Rp_values(i, :) * c;
107             Rn = Rn_values(i, :) * c;
108             Vp = Vp_values(i, :) * c;
109             Vn = Vn_values(i, :) * c;
110
111             % Set HIF parameters in the model
112             set_param([model_HIF, '/HIF'], 'FaultA', FaultPhase
113                 (1))
114             set_param([model_HIF, '/HIF'], 'FaultB', FaultPhase
115                 (2))
116             set_param([model_HIF, '/HIF'], 'FaultC', FaultPhase
117                 (3))
118             set_param([model_HIF, '/HIF'], 'GroundFault',
119                 FaultPhase(4))
120
121             % Balanced condition
122             [Pa, Pb, Pc] = deal(P_set(randi(length(P_set))));
123             out = sim(model_HIF);
124             data = [out.data, repmat(FaultPhase', Data_n, 1),
  
```

```
121         repmat(Lf, Data_n, 1), repmat(i, Data_n, 1),  
           zeros(Data_n, 1), ones(Data_n, 1)];  
writematrix(data, fullfile('Dataset', [dataset_name  
    , '.csv']), 'WriteMode', 'append');  
122  
123     % Unbalanced condition  
124     Pa = P_set_unb(randi(length(P_set_unb)));  
125     Pb = P_set_unb(randi(length(P_set_unb)));  
126     Pc = P_set_unb(randi(length(P_set_unb)));  
127     Unbalance = abs(max([(Pa - Pb) / Pn, (Pa - Pc) / Pn  
        , (Pc - Pb) / Pn], [], "ComparisonMethod", 'abs'  
        ));  
128     out = sim(model_HIF);  
129     data = [out.data, repmat(FaultPhase', Data_n, 1),  
           repmat(Lf, Data_n, 1), repmat(i, Data_n, 1),  
           repmat(Unbalance, Data_n, 1), ones(Data_n, 1)];  
130     writematrix(data, fullfile('Dataset', [dataset_name  
        , '.csv']), 'WriteMode', 'append');  
131     end  
132 end  
133 end
```

II. Eranskina. Datuen transformazioa.

```

1 import numpy as np
2 import pywt
3 import pandas as pd
4 from scipy.stats import entropy
5
6 # Read the dataset
7 df = pd.read_csv("Def/dataset_def_ALL.csv")
8
9 # Define constants
10 window_size = 200
11 wavelet = 'db4'
12 level = 2
13 simulation_size = 800
14 columns = ['Ia', 'Ib', 'Ic', 'Va', 'Vb', 'Vc']
15 label_columns = ['F', 'A', 'B', 'C', 'G']
16
17 # Define the functions to apply
18 def rms(x):
19     return np.sqrt(np.mean(np.square(x)))
20
21 def mean_derivative(x):
22     return np.mean(np.gradient(x))
23
24 def percentile_25(x):
25     return np.percentile(x, 25)
26
27 def percentile_75(x):
28     return np.percentile(x, 75)
29
30 # List of functions to apply
31 functions = [np.mean, np.max, np.var, np.median,
32             percentile_25, percentile_75, rms, mean_derivative]
33
34 # Define a function to process the dataset with a given
35 # function
36 def process_function(func, func_name):
37     # Initialize the features dictionary with empty lists for
38     # each feature
39     features = {f'{column}_d1': [] for column in columns}
40     features.update({f'{column}_d2': [] for column in columns})
41     features.update({f'{column}_a2': [] for column in columns})
42     features.update({column: [] for column in label_columns})
43
44     # Define the number of simulations

```

```

42 num_simulations = len(df[columns[0]]) // simulation_size
43
44 # Loop over the simulations
45 for i in range(num_simulations):
46     # Loop over the windows within the current simulation
47     for j in range(simulation_size - window_size + 1):
48         # Loop over each column
49         for column in columns:
50             # Get the current window of data
51             window = df[column].iloc[i * simulation_size +
52                                     j : i * simulation_size + j + window_size]
53
54             # Apply the wavelet transform with 2 levels of
55             # decomposition
56             coeffs = pywt.wavedec(window, wavelet,
57                                   mode='smooth', level=level)
58
59             # Compute the feature for each level
60             features[f'{column}_d1'].append(func(coeffs[2]))
61             # detail coefficients at level 1
62             features[f'{column}_d2'].append(func(coeffs[1]))
63             # detail coefficients at level 2
64             features[f'{column}_a2'].append(func(coeffs[0]))
65             # approximation coefficients at level 2
66
67             # Add the labels for the current window
68             for label_column in label_columns:
69                 features[label_column].append(df[label_column].iloc[i
70                                         * simulation_size + j + window_size - 1])
71
72 # Convert the dictionary of features to a DataFrame
73 df_features = pd.DataFrame(features)
74
75 # Save the DataFrame to a CSV file
76 df_features.to_csv(f'Def/Iterations/labeled_features_ALL_4_2_{func_name}
77                    index=False)
78
79 # Apply each function sequentially
80 for func in functions:
81     func_name = func.__name__
82     process_function(func, func_name)
  
```

III. Eranskina. Detekzio modeloak.

```

1 import os
2 import pandas as pd
3 import numpy as np
4 from sklearn.model_selection import StratifiedKFold
5 from sklearn.metrics import accuracy_score,
6     classification_report, confusion_matrix, roc_auc_score,
7     precision_score, recall_score, f1_score
8 from xgboost import XGBClassifier
9 from sklearn.ensemble import RandomForestClassifier
10 from sklearn.neighbors import KNeighborsClassifier
11 from sklearn.svm import SVC
12 from sklearn.linear_model import LogisticRegression
13 from sklearn.neural_network import MLPClassifier
14 from sklearn.preprocessing import StandardScaler
15 from imblearn.over_sampling import SMOTE
16
17 # Path to the CSV file
18 file_path = os.path.join('C:/Users/jonre/OneDrive - UPV
19     EHU/23-24/TFM/python/pythonProject/Def/Test',
20     'labeled_features_ALL_4_2_std.csv')
21
22 # Initialize a dictionary to store results
23 results = {}
24
25 # Classifier choices
26 classifier_choices = ['randomforest']
27
28 # Function to perform cross-validation with multiple metrics
29 def evaluate_model_with_cv(X, y, model, cv=5):
30     skf = StratifiedKFold(n_splits=cv, shuffle=True,
31         random_state=42)
32     accuracy_scores = []
33     roc_auc_scores = []
34     precision_scores = []
35     recall_scores = []
36     f1_scores = []
37     classification_reports = []
38     confusion_matrices = []
39
40     for train_index, test_index in skf.split(X, y):
41         X_train, X_test = X.iloc[train_index],
42             X.iloc[test_index]
43         y_train, y_test = y.iloc[train_index],
44             y.iloc[test_index]
45
46     # Initialize and fit the scaler to the training data
  
```

```

40     scaler = StandardScaler()
41     X_train_scaled = scaler.fit_transform(X_train)
42     X_test_scaled = scaler.transform(X_test)
43
44     model.fit(X_train_scaled, y_train)
45     y_pred = model.predict(X_test_scaled)
46
47     accuracy_scores.append(accuracy_score(y_test, y_pred))
48     roc_auc_scores.append(roc_auc_score(y_test,
49     model.predict_proba(X_test_scaled)[: , 1]))
50     precision_scores.append(precision_score(y_test,
51     y_pred, zero_division=1))
52     recall_scores.append(recall_score(y_test, y_pred,
53     zero_division=1))
54     f1_scores.append(f1_score(y_test, y_pred,
55     zero_division=1))
56     classification_reports.append(classification_report(y_test,
57     y_pred, output_dict=True, zero_division=1))
58     confusion_matrices.append(confusion_matrix(y_test,
59     y_pred))
60
61     return {
62         'accuracy': (np.mean(accuracy_scores),
63         np.std(accuracy_scores)),
64         'roc_auc': (np.mean(roc_auc_scores),
65         np.std(roc_auc_scores)),
66         'precision': (np.mean(precision_scores),
67         np.std(precision_scores)),
68         'recall': (np.mean(recall_scores),
69         np.std(recall_scores)),
70         'f1': (np.mean(f1_scores), np.std(f1_scores)),
71         'classification_reports': classification_reports,
72         'confusion_matrices': confusion_matrices
73     }
74
75 # Function to choose the model
76 def get_model(choice):
77     if choice == 'xgboost':
78         return XGBClassifier(use_label_encoder=False,
79         eval_metric='logloss')
80     elif choice == 'randomforest':
81         return RandomForestClassifier()
82     elif choice == 'knn':
83         return KNeighborsClassifier()
84     elif choice == 'svm':
85         return SVC(kernel='linear', probability=True) # SVM
86         needs probability=True for ROC AUC
87     elif choice == 'logistic':
88         return LogisticRegression(max_iter=200)
89     elif choice == 'mlp':
90         return MLPClassifier(max_iter=500)
91     else:

```

```

80         raise ValueError("Unsupported model choice. Use
            'xgboost', 'randomforest', 'knn', 'svm',
            'logistic', or 'mlp'.")
81
82 # Load the CSV file
83 df = pd.read_csv(file_path)
84
85 # Select the features
86 selected_features = ['Ia_d2', 'Ia_a2', 'Ib_d2', 'Ib_a2',
            'Ic_d2', 'Ic_a2']
87
88 X = df[selected_features]
89 y = df['F']
90
91 for classifier_choice in classifier_choices:
92     # Get the model based on the choice
93     model = get_model(classifier_choice)
94
95     # Evaluate the model using cross-validation
96     metrics = evaluate_model_with_cv(X, y, model, cv=5)
97
98     # Store the results
99     results[classifier_choice] = metrics
100    print(f'{classifier_choice}: Mean Accuracy =
            {metrics["accuracy"][0]:.10f}, Std =
            {metrics["accuracy"][1]:.4f}')
101    print(f'{classifier_choice}: Mean ROC AUC =
            {metrics["roc_auc"][0]:.4f}, Std =
            {metrics["roc_auc"][1]:.4f}')
102    print(f'{classifier_choice}: Mean Precision =
            {metrics["precision"][0]:.4f}, Std =
            {metrics["precision"][1]:.4f}')
103    print(f'{classifier_choice}: Mean Recall =
            {metrics["recall"][0]:.4f}, Std =
            {metrics["recall"][1]:.4f}')
104    print(f'{classifier_choice}: Mean F1 Score =
            {metrics["f1"][0]:.4f}, Std = {metrics["f1"][1]:.4f}')
105
106 # Find the best model based on accuracy (or another metric if
            preferred)
107 best_model = max(results, key=lambda k:
            results[k]['accuracy'][0])
108 print(f'\nBest model: {best_model} with mean accuracy
            {results[best_model]["accuracy"][0]:.10f} and std
            {results[best_model]["accuracy"][1]:.4f}')
109 print(f'Mean ROC AUC =
            {results[best_model]["roc_auc"][0]:.4f}, Std =
            {results[best_model]["roc_auc"][1]:.4f}')
110 print(f'Mean Precision =
            {results[best_model]["precision"][0]:.4f}, Std =
            {results[best_model]["precision"][1]:.4f}')
111 print(f'Mean Recall = {results[best_model]["recall"][0]:.4f},

```



```
Std = {results[best_model]["recall"][1]:.4f}')  
112 print(f'Mean F1 Score = {results[best_model]["f1"][0]:.4f},  
Std = {results[best_model]["f1"][1]:.4f}')
```


IV. Eranskina. Sailkapen modeloak.

```

1 import os
2 import pandas as pd
3 import numpy as np
4 from sklearn.model_selection import StratifiedKFold
5 from sklearn.metrics import accuracy_score,
6     classification_report, confusion_matrix, roc_auc_score,
7     precision_score, recall_score, f1_score
8 from xgboost import XGBClassifier
9 from sklearn.ensemble import RandomForestClassifier
10 from sklearn.neighbors import KNeighborsClassifier
11 from sklearn.svm import SVC
12 from sklearn.linear_model import LogisticRegression
13 from sklearn.neural_network import MLPClassifier
14 from sklearn.preprocessing import StandardScaler
15 from imblearn.over_sampling import SMOTE
16
17 # Directory containing the CSV files
18 file_path = os.path.join('C:/Users/jonre/OneDrive - UPV
19     EHU/23-24/TFM/python/pythonProject/Def/Test',
20     'labeled_features_ALL_4_2_std.csv')
21
22 # Initialize a dictionary to store results
23 results = {}
24
25 # Classifier choices
26 classifier_choices = ['randomforest']
27
28 # Function to perform cross-validation with multiple metrics
29 for multiclass classification
30 def evaluate_model_with_cv(X, y, model, cv=5):
31     skf = StratifiedKFold(n_splits=cv, shuffle=True,
32         random_state=42)
33     accuracy_scores = []
34     roc_auc_scores = []
35     precision_scores = []
36     recall_scores = []
37     f1_scores = []
38     classification_reports = []
39     confusion_matrices = []
40
41     for train_index, test_index in skf.split(X, y):
42         X_train, X_test = X.iloc[train_index],
43             X.iloc[test_index]
44         y_train, y_test = y.iloc[train_index],
45             y.iloc[test_index]

```

```

39 # Initialize and fit the scaler to the training data
40 scaler = StandardScaler()
41 X_train_scaled = scaler.fit_transform(X_train)
42 X_test_scaled = scaler.transform(X_test)
43
44 model.fit(X_train_scaled, y_train)
45 y_pred = model.predict(X_test_scaled)
46
47 accuracy_scores.append(accuracy_score(y_test, y_pred))
48 roc_auc_scores.append(roc_auc_score(y_test,
49   model.predict_proba(X_test_scaled),
50   average='macro', multi_class='ovr'))
51 precision_scores.append(precision_score(y_test,
52   y_pred, average='macro', zero_division=1))
53 recall_scores.append(recall_score(y_test, y_pred,
54   average='macro', zero_division=1))
55 f1_scores.append(f1_score(y_test, y_pred,
56   average='macro', zero_division=1))
57 classification_reports.append(classification_report(y_test,
58   y_pred, output_dict=True, zero_division=1))
59 confusion_matrices.append(confusion_matrix(y_test,
60   y_pred))
61
62 return {
63   'accuracy': (np.mean(accuracy_scores),
64     np.std(accuracy_scores)),
65   'roc_auc': (np.mean(roc_auc_scores),
66     np.std(roc_auc_scores)),
67   'precision': (np.mean(precision_scores),
68     np.std(precision_scores)),
69   'recall': (np.mean(recall_scores),
70     np.std(recall_scores)),
71   'f1': (np.mean(f1_scores), np.std(f1_scores)),
72   'classification_reports': classification_reports,
73   'confusion_matrices': confusion_matrices
74 }
75
76 # Function to choose the model
77 def get_model(choice):
78   if choice == 'xgboost':
79     return XGBClassifier(use_label_encoder=False,
80       eval_metric='mlogloss')
81   elif choice == 'randomforest':
82     return RandomForestClassifier(**{'n_estimators': 500,
83       'min_samples_split': 5, 'min_samples_leaf': 1,
84       'max_features': 'sqrt', 'max_depth': None,
85       'bootstrap': False})
86   elif choice == 'knn':
87     return KNeighborsClassifier()
88   elif choice == 'logistic':
89     return LogisticRegression(max_iter=200)
90   elif choice == 'mlp':

```

```

76     return MLPClassifier(hidden_layer_sizes=(6,),
77                           max_iter=500)
78   elif classifier_choice == 'svm':
79     model = SVC(kernel='linear', probability=True)
80   else:
81     raise ValueError("Unsupported model choice. Use
82                       'xgboost', 'randomforest', 'knn', 'svm',
83                       'logistic', or 'mlp'.")
84
85 df = pd.read_csv(file_path)
86
87 # Filter rows where 'F' equals 1
88 df = df[df['F'] == 1]
89
90 # Define class conditions and corresponding labels
91 class_conditions = [
92     (df['A'] == 1) & (df['B'] == 0) & (df['C'] == 0),
93     (df['A'] == 0) & (df['B'] == 1) & (df['C'] == 0),
94     (df['A'] == 0) & (df['B'] == 0) & (df['C'] == 1),
95     (df['A'] == 1) & (df['B'] == 1) & (df['C'] == 0) &
96     (df['G'] == 0),
97     (df['A'] == 1) & (df['B'] == 0) & (df['C'] == 1) &
98     (df['G'] == 0),
99     (df['A'] == 0) & (df['B'] == 1) & (df['C'] == 1) &
100    (df['G'] == 0),
101    (df['A'] == 1) & (df['B'] == 1) & (df['C'] == 0) &
102    (df['G'] == 1),
103    (df['A'] == 1) & (df['B'] == 0) & (df['C'] == 1) &
104    (df['G'] == 1),
105    (df['A'] == 0) & (df['B'] == 1) & (df['C'] == 1) &
106    (df['G'] == 1),
107    (df['A'] == 1) & (df['B'] == 1) & (df['C'] == 1) &
108    (df['G'] == 0)
109 ]
110 class_labels = range(10) # 0, 1, 2, ..., 9
111
112 # Create a new column for the class labels
113 df['class'] = np.select(class_conditions, class_labels,
114                        default=np.nan)
115
116 # Select the features
117 selected_features = ['Ia_d1', 'Ib_d1', 'Ic_d1', 'Va_d1',
118                    'Vb_d1', 'Vc_d1', 'Ia_a2', 'Ib_a2', 'Ic_a2']
119
120 X = df[selected_features]
121 y = df['class']
122
123 for classifier_choice in classifier_choices:
124     # Get the model based on the choice
125     model = get_model(classifier_choice)
126
127     # Evaluate the model using cross-validation
  
```

```
116 metrics = evaluate_model_with_cv(X, y, model, cv=5)
117
118 # Store the results
119 results[classifier_choice] = metrics
120 print(f'{classifier_choice}: Mean Accuracy =
    {metrics["accuracy"][0]:.10f}, Std =
    {metrics["accuracy"][1]:.4f}')
121 print(f'{classifier_choice}: Mean ROC AUC =
    {metrics["roc_auc"][0]:.4f}, Std =
    {metrics["roc_auc"][1]:.4f}')
122 print(f'{classifier_choice}: Mean Precision =
    {metrics["precision"][0]:.4f}, Std =
    {metrics["precision"][1]:.4f}')
123 print(f'{classifier_choice}: Mean Recall =
    {metrics["recall"][0]:.4f}, Std =
    {metrics["recall"][1]:.4f}')
124 print(f'{classifier_choice}: Mean F1 Score =
    {metrics["f1"][0]:.4f}, Std = {metrics["f1"][1]:.4f}')
125
126 # Find the best model based on accuracy (or another metric if
    preferred)
127 best_model = max(results, key=lambda k:
    results[k]['accuracy'][0])
128 print(f'\nBest model: {best_model} with mean accuracy
    {results[best_model]["accuracy"][0]:.10f} and std
    {results[best_model]["accuracy"][1]:.4f}')
129 print(f'Mean ROC AUC =
    {results[best_model]["roc_auc"][0]:.4f}, Std =
    {results[best_model]["roc_auc"][1]:.4f}')
130 print(f'Mean Precision =
    {results[best_model]["precision"][0]:.4f}, Std =
    {results[best_model]["precision"][1]:.4f}')
131 print(f'Mean Recall = {results[best_model]["recall"][0]:.4f},
    Std = {results[best_model]["recall"][1]:.4f}')
132 print(f'Mean F1 Score = {results[best_model]["f1"][0]:.4f},
    Std = {results[best_model]["f1"][1]:.4f}')
```