

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
FINANSŲ IR DRAUDIMO MATEMATIKOS BAKALAURO STUDIJŲ
PROGRAMA

Julius Čereška

**Sisteminės rizikos modeliavimas elektros perdavimo
tinkluose**

**Modeling of systemic risk in power transmission
networks**

Bakalauro baigiamasis darbas

Darbo vadovas

dr. doc. Martynas Manstavičius

VILNIUS
2020

Santrauka

Darbe nagrinėjama Lietuvos elektros energijos perdavimo tinklų struktūra ir vertinama jos sisteminė rizika. Analizuojamas tinklų stabilumas, ieškomi kritiniai perdavimo taškai. Šis darbas remiasi grafų teorijos modeliais ir algoritmais. Darbe yra kuriamas perdavimo tinklų energijos paskirstymo simuliacijos modelis, kuris remiasi „Network simplex“ mažiausių kaštų tėkmės paskirstymo metodika. Modelyje taip pat naudojama apkrovos numetimo strategija, jeigu paklausa ir pasiūlos galimybės nepasieks ekvivalentumo būsenos. Šio modelio pagalba simuliuojama sisteminės rizikos įtaką energijos perdavimo tinklams ir energijos paklausos patenkinimui.

Raktiniai žodžiai: Elektros energijos perdavimo tinklai, sisteminė rizika, grafų teorija, Network simplex metodika, energijos tėkmės simuliacija, Monte Karlo metodas

Summary

In this bachelor thesis we are analyzing the structure of Lithuanian electricity transmission network and its systemic risk. Network stability is tested with various different conditions and critical transmission points are determined. This work is based on graph theory models and algorithms. We are also building a "Network simplex" minimum cost flow allocation methodology based model for distributing energy flow in the model graph. This model is also capable of using load shedding strategy if the demand does not meet the supply with given restrictions. The model is then used to assess the systemic risk and loss of energy in the Lithuanian electricity transmission network.

Keywords: Electricity transmission networks, systemic risk, graph theory, Network simplex method, Monte Carlo method

TURINYS

IVADAS	4
1. LIETUVOS ENERGIJOS PERDAVIMO TINKLŲ ANALITINĖ APŽVALGA	6
1.1. Lietuvos energetiniai tinklai	6
1.2. Lietuvos energijos perdavimo tinklų analizė remiantis grafų teorija	7
1.3. Apibendrinimas	10
2. LIETUVOS ELEKTROS ENERGIJOS PERDAVIMO TINKLŲ MODELIS IR SISTEMINĖS RIZIKOS MODELIAVIMAS	12
2.1. Duomenų apžvalga ir agregavimas	12
2.2. Modelio metodika	15
2.2.1. Duomenų surinkimas	15
2.2.2. Nutraukiamos atitinkamos linijos	15
2.2.3. Duomenų perskirstymas grafo viršūnėms	15
2.2.4. Energijos deficito arba pertekliaus paskirstymas	17
2.2.5. Perdavimo tinklų talpos nustatymas	17
2.2.6. Minimalių kaštų srauto sprendinys	19
2.2.7. Duomenų kaupimas	21
2.3. Modelio rezultatai	21
2.3.1. Baziniai modelio rezultatai	21
2.3.2. Pavienių perdavimo linijų pašalinimas iš grafo	22
2.3.3. Perdavimo linijų lūžių tikimybės įtaka energijos paskirstymo modeliui	24
REZULTATAI IR IŠVADOS	30
PRIEDAI	31
1 priedas. „Facebook Prophet“ modelio prognozuojamos reikšmės	32
2 priedas. Lietuvos energijos eksporto ir importo duomenys	33
3 priedas. Simuliacinio modelio funkcijos	34

Įvadas

Elektros energija šiais laikais yra neatsiejama kasdienio gyvenimo dalis. Dauguma žmonių net nepagalvoja, koks sudėtingas ir sunkus yra elektros tinklų valdymas ir į kiek daug faktorių turima atsižvelgti, kad elektra patektų į jų namus. Ši keliauja beveik šviesos greičiu, taigi, nuo elektros pagaminamo jėgainėje iki sunaudojimo vartotojo elektroniniuose prietaisuose praeina milisekundės. Tačiau šis procesas yra ypač sudėtingas ir prižiūrimas kiekvienoje tinklo dalies sistemoje. Mažiausi elektros srovių nesuvaldyti pokyčiai gali nutraukti elektros tiekimą miestams ar net šalims. Todėl ypač svarbu suprasti energijos tinklų sisteminę riziką ir, ją įsivertinus, teisingai pasiruošti numatomoms pasekmėms. Eksperimentais ir turimais duomenimis nustatant perdavimo tinklo pažeidimo tikimybę, galima būtų įsivertinti, kokią energijos paklausos dalį sugebės patenkinti Lietuvos energetinis tinklas ir kokių priemonių reikėtų imtis, kad ši žala būtų kuo mažesnė. Taigi, šio darbo tikslas yra įsivertinti sisteminės rizikos poveikį Lietuvos energijos perdavimo tinklų modeliui.

Šis darbas parengtas remiantis įvairiomis analitinėmis ir modeliavimo metodikomis, aprašytomis „Risk analysis of critical loading and blackouts with cascading events“ [2] straipsnių rinkinyje. Šis rinkinys pateikia 12 skirtingų, bet su panašia tema susijusių straipsnių. Juose pateiktos metodikos (kaip veikia energijos perdavimo tinklai) bei metrikos (matuojančios energijos kiekio praradimus ir linijų nutrūkimo tikimybes) yra taikomos šiame darbe. Energijos tinklų sisteminė rizika šiame darbe yra apibrėžiama, kaip netikėtas energijos tiekimo nutrūkimas. Panašiai kaip bankų sektoriuje, kai kuris nors bankas bankrutuoja ar atsisako skolinti kitiems, visa bankinė sistema patiria šoką ir jaučia to pasekmes. Lygiai taip pat ir energijos perdavimo tinklai patiria tam tikrų sunkumų, kai kuri nors užsienio šalis nevykdo savo energijos tiekimo įsipareigojimų, perdavimo linija tampa pažeista, vėjo jėgainės tampa ypač aktyvios ar staigiai pakinta elektros rinkos kaina. Iš esmės visa tai modeliuojant galima simplifikuoti į energijos perdavimo linijos pašalinimą iš perdavimo tinklų modelio. Pavyzdžiui, tam tikros užsienio šalies tikimybė nutraukti energijos tiekimą gali būti traktuojama kaip energijos perdavimo tinklų su ta šalimi nutraukimo tikimybė.

Šiame darbe bus nagrinėjama Lietuvos elektros energijos perdavimo tinklų sisteminė rizika ir jos poveikis. Tinklų sistemą susivedus į grafą, visi skaičiavimai bus atliekami šiame teoretiniame perdavimo tinklų modelyje. Pradžioje bus nagrinėjami Lietuvos elektros perdavimo tinklai, jų išsidėstymas ir struktūra. Įvertinsime teoriškai prasčiausiai aprūpintus miestus ir rizikingiausias elektros perdavimo jungtis. Naudodami grafų teorijos žinias, apžvelgsime Lietuvos tinklų sistemą makro lygmenyje ir ieškosime, kurie miestai turi daugiausia galimybių apsirūpinti energija iš skirtingų energijos šaltinių bei kurios perdavimo linijos atskelia šalį į nesujungtų tinklų sistemą. Ši dalis remsis kelių, takų ir trasų analize bei tiltų ir nesujungtųjų pogrifių apžvalga.

Antroje dalyje bus apžvelgti Lietuvos sunaudojamos energijos duomenys ir kuriamas Lietuvos perdavimo tinklų energijos perskirstymą replikuojantis modelis. Duomenų apžvalgoje naudosime *Facebook* įmonės *Prophet* [13] modeliavimo metodiką, kad suprastume turimų laiko eilučių tendencijas ir periodiškumus. Modeliui sukurti naudosime grafų teorijos žinias ir algoritmus, pavyzdžiui – „*Network simplex*“, kartu su faktiniais duomenimis apie sunaudojamą energiją, eks-

portą ir importą bei linijų pralaidumus. Naudodami šį modelį nagrinėsime sisteminės rizikos žalą Lietuvos energijos paklausai. Bus atlikta jautrumo analizė nesuteiktos energijos kiekiui modelyje priklausomai nuo įvairių tikimybinių parametrų. Dėl nelygių energijos perdavimo tinklų svarbos ir tikimybių neužtikrintumo visos simuliacijos bus atliekamos Monte Karlo metodika, kad rezultatai būtų kuo mažiau paveikti variacijos. Šioje dalyje taip pat taikysime Markovo grandinių procesų metodiką, siekdami įsivertinti stabilumo būseną tarp veikiančių ir lūžusių perdavimo linijų.

Šiam darbui atlikti naudojama Python programavimo kalba. Duomenims apdoroti ir analizuoti naudota pandas biblioteka, o vizualizacijos buvo atliktos su matplotlib, seaborn ir plotly bibliotekomis. Funkcijoms, skaičiavimams ir modeliams, susijusiems su grafais, buvo pasitelktas NetworkX paketas, o statistiniams skaičiavimams ir įvertinimams – SciPy.

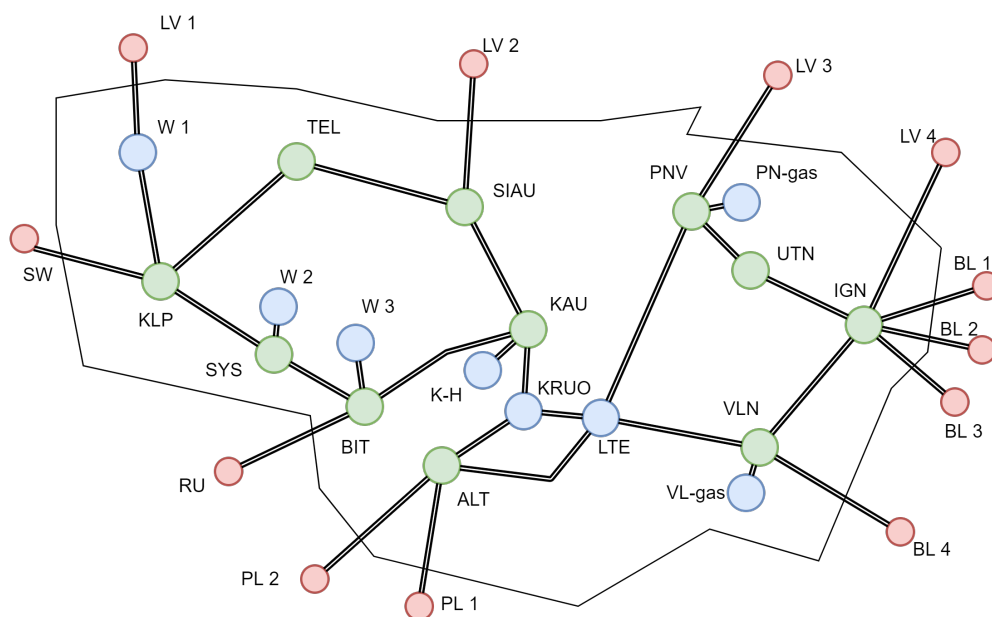
1. Lietuvos energijos perdavimo tinklų analitinė apžvalga

Šiame skyriuje apžvelgsime Lietuvos energijos perdavimo tinklų struktūrą, skaidysime tinklus į pografius ir bandysime nustatyti kritinius taškus. Šiam tikslui pasitelksime grafų teoriją ir neanalizuosime tikslių energijos mainų ir sunaudojimo duomenų skaičių.

1.1. Lietuvos energetiniai tinklai

Lietuvoje egzistuoja milijonai kabelių ir laidų, teikiančių elektros energiją gyventojams. Kiekvieno tokio laido nutrūkimas iš esmės paliktų ką nors be elektros iki gedimo pašalinimo. Svarbios instancijos, pavyzdžiui, ligoninės, nuo tokių netikėtų energijos tiekimo sutrikimų apsisaugo turėdamos dyzelinius generatorius, kurie gali laikinai aprūpinti pastatą elektra. Daugelis tokių laidų perneša nedidelius energijos kiekius ir jų išdėstymas bei jungtys yra svarbios tik bendrijoms, mikrolygyje.

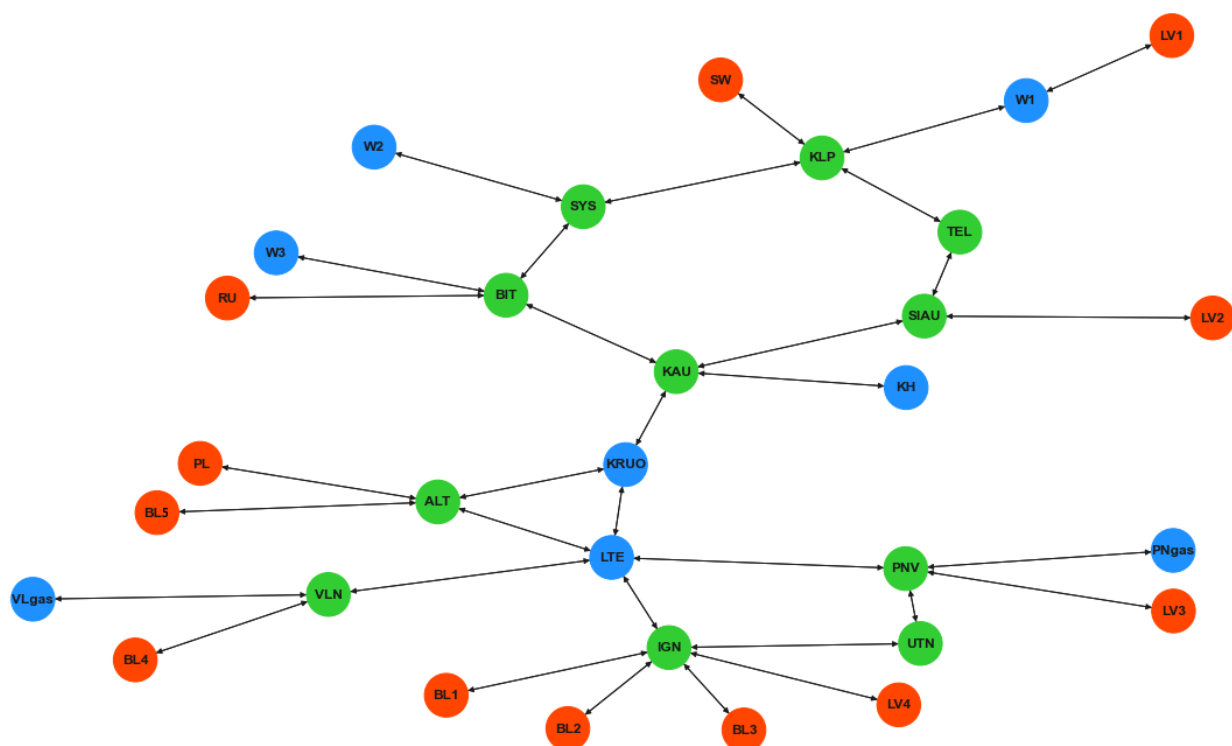
Šiame darbe apžvelgsime Lietuvos energijos tinklus makro lygmenyje. Tai reiškia, kad mūsų nagrinėjami tinklai bus ilgų atstumų ir aukštos įtampos. Lietuvos supaprastintame tokių perdavimo tinklų žemėlapyje egzistuoja trijų tipų stotys: šalies vidiniai generatoriai, užsienio generatoriai ir miestai – elektros energijos naudotojai. (1 pav. atitinkamai pažymėti spalvomis: mėlyni, raudoni ir žali.)



1 pav. Supaprastintas Lietuvos energijos tinklų tiekimo žemėlapis sugeneruotas remiantis www.entsoe.eu duomenimis

Nors ne visi raudonai pažymėti taškai yra elektros jėgainės, dėl paprastumo juos galime laikyti generatoriais, nes per juos, iš kažkurio užsienio generatoriaus, gali ateiti energija iki Lietuvos, tačiau mūsų šis pilnas kelias nedomina. Iš šio žemėlapio gana sunku pasakyti, kurios stotys ar perdavimo tinklai yra kritiniai. Tačiau transformavus šių tinklų išsidėstymą naudojant *Kamada–Kawai* algoritmą neorientuotiems grafams braižyti, iškart galime pastebėti, kuriuose taškuose gali kilti problemų, naudojant modelį [6].

1.2. Lietuvos energijos perdavimo tinklų analizė remiantis grafų teorija



2 pav. Kamada–Kawai algoritmu išbraižytas Lietuvos energijos perdavimo tinklų grafas

Grafe, pavaizduotame 2 pav., vis dar matomas Lietuvos miestų išsidėstymas teisinga tvarka (tik pasuktas 90° kampu). Tačiau dabar jau galime įžvelgti kritinius taškus, kaip Kruonio hidroakumuliacinė elektrinė ir Lietuvos elektrinė, atitinkamai pažymėtus *KRUO* ir *LTE*, kurių pašalinimas iš grafo atjungtų rytų ir vakarų Lietuvos regionus.

Tačiau abiejų šių energijos perdavimo taškų nutrūkimas yra ypač mažai tikėtinas, o nutrūkus tik vienam, elektros perdavimą galima perskirstyti per Alytaus pastotę. Kadangi mus domina vietos mūsų grafe, kurios sutrikdytų sklandžią Lietuvos energijos tėkmę, galime išskirti du pažeidimų tipus kurie yra svarbūs. Vieną jau aprašėme – tai energijos perskirstymo stoties išjungimas. Kitas – energijos tiekimo linijos nutrūkimas.

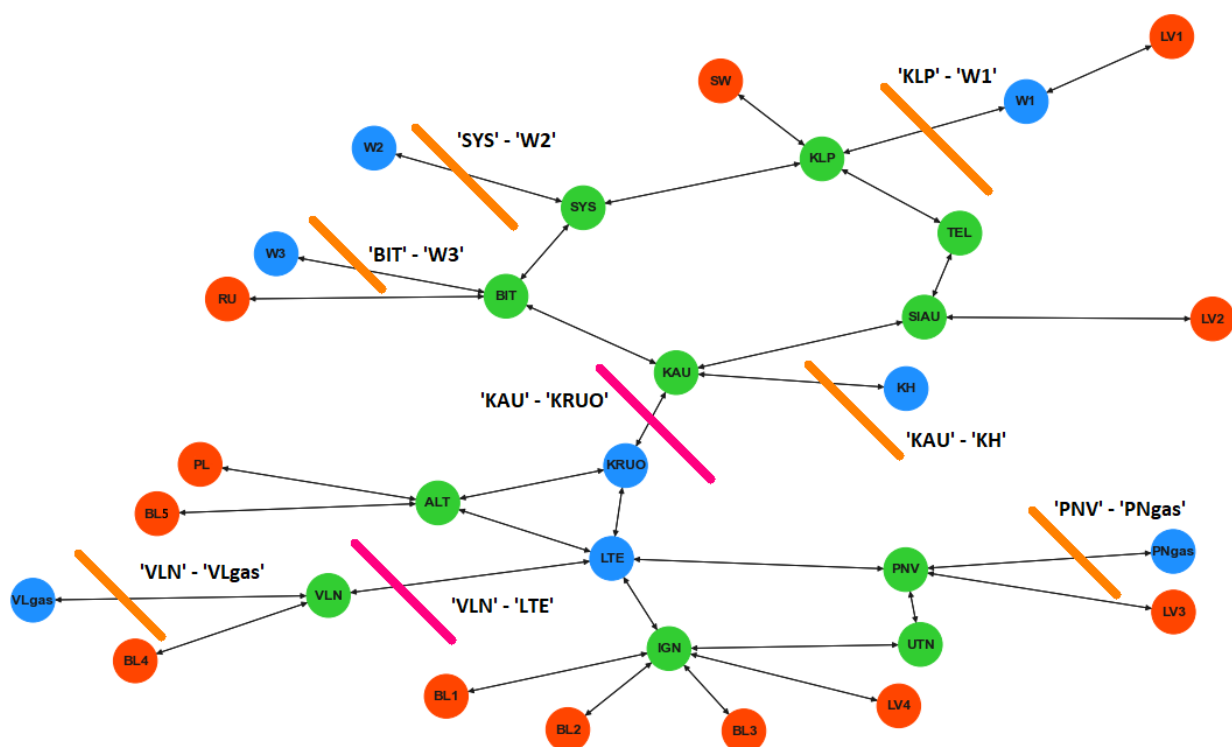
Pradėkime nuo dažniau pasitaikančių trukdžių – energijos perdavimo linijos nutrūkimo. Kaip ir anksčiau minėtame pavyzdyje, nutrūkus vienai tiekimo linijai, teoriškai galima naudoti kitą kelią perteikti energiją iš taško *A* į tašką *B*. Vėlgi naudojant 2 pav. grafą, galime matyti, jog *LTE* turi tiesioginę jungtį su Ignalinos miestu, tačiau jai nutrūkus, energija Ignaliną gali pasiekti keliu $LTE \rightarrow PNV \rightarrow UTN \rightarrow IGN$, taigi, šis linijos nutrūkimas iš esmės nėra labai svarbus. Kritinėmis linijomis vadinsime tokias perdavimo linijas, kurių pašalinimas iš grafo sukurtų du pografius, kurie nėra niekaip sujungti. Grafų teorijoje tokios briaunos vadinamos tiltais, o grafas, kuris turi nesujungtų briaunomis pografių, vadinamas nejungiu grafu [16].

Ieškodami tiltų šiame grafe, turime atsižvelgti ir į tai, kaip pasirinkome grafą apibrėžti. Kadangi nenagrinėjame energijos tinklų jungčių užsienio valstybėse, kiekviena mūsų pažymėta už-

sienio energijos generatoriais viršūnė bus tiltas. Šiuos atvejus atmetę, mes matome, jog grafe egzistuoja 8 tiltai.

Tiltai:	1 pagrafį sudaro:	2 pagrafį sudaro:
'KLP'-'W1'	18 generatorių ir 11 miestų	2 generatoriai ir 0 miestų
'PNV'-'PNgas'	19 generatorių ir 11 miestų	1 generatorius ir 0 miestų
'KAU'-'KRUO'	12 generatorių ir 5 miestai	8 generatoriai ir 6 miestai
'KAU'-'KH'	19 generatorių ir 11 miestų	1 generatorius ir 0 miestų
'SYS'-'W2'	19 generatorių ir 11 miestų	1 generatorius ir 0 miestų
'BIT'-'W3'	19 generatorių ir 11 miestų	1 generatorius ir 0 miestų
'VLN'-'LTE'	18 generatorių ir 10 miestų	2 generatoriai ir 1 miestas
'VLN'-'VLgas'	19 generatorių ir 11 miestų	1 generatorius ir 0 miestų

1 lentelė. Lietuvos elektros perdavimo tinklų vidinių viršūnių tiltai ir jų atskiriamų pagrafių kompozicija



3 pav. Linijų kirtimai, kurie skeltų grafą į dvi nesujungtas dalis

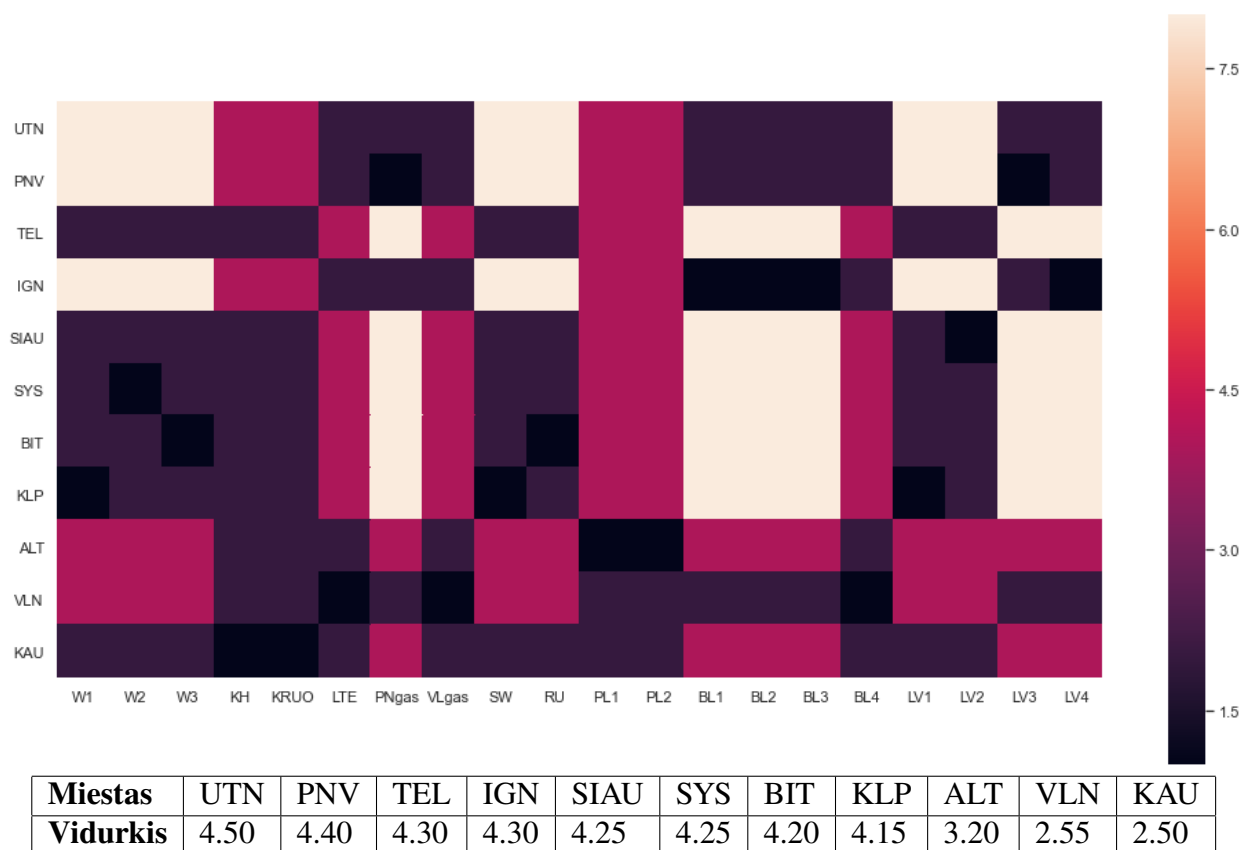
Iš aštuonių gautų tiltų, net šeši tiltai yra tarp miesto ir prie jo prijungto elektros generatoriaus. Jų giliau nagrinėti neverta, nes bendruoju atveju jie atjungia tik vieną ar kelis generatorius nuo Lietuvos energijos perdavimo tinklo. Taigi, radome dvi kritines linijas, kurių nutrūkimas galimai paveiktų mūsų tinklą. Toliau norėdami įsivertinti šių briaunų svorį tarpusavyje, galime, palyginti kokią santykį tarp energijos vartotojų ir energijos generatorių palieka kiekviename pagrafyje šių linijų eliminavimas.

	Energijos vartotojų ir generatorių santykis		Santykio skirtumas nuo vidurkio	
Tiltai:	1 pografo	2 pografo	1 pografo	2 pografo
(KAU, KRUE)	2.4	1.33	32%	–27%
(VLN, LTE)	1.8	2.00	–1%	10%

2 lentelė. Atskirtų pografių santykis tarp energijos vartotojų ir generatorių viršūnių

Nors abu tinklus laikome kritiniais, jie turi savitų problemų. Pirmuoju atveju, linijos *KAU–KRUE* nutrūkimas atneštų didelį disbalansą tarp generatorių ir miestų, dėl ko didesni kiekiai energijos turėtų būti gaminami ir perduodami per linijas toliau didinant riziką sugesti kitoms linijoms ir pradėti slenkantį miestų užtemimą (angl., *rolling blackout*). Antrasis *VLN–LTE* linijų nutrūkimas nesukurtų žymaus išbalansavimo tarp generatorių ir eikvojančių energiją miestų. Tačiau šis atvejis palieka Vilniaus miestą visiškai priklausomą nuo savo jėgainės tiekimo galimybių bei elektros importo iš Baltarusijos. Tai galimai sukeltų problemų paaštrėjus politiniams santykiams.

Toliau nagrinėjime Lietuvos energijos perdavimo tinklų grafo viršūnes ir bandykime teoriškai nusistatyti, kurie miestai turi didesnę riziką nukentėti, jei įvyktų neplanuotas energijos perdavimo linijos išjungimas. Tam taip pat naudosime grafų teoriją. Vienas svarbiausių klausimų kiekvienam miestui – kiek skirtingų būdų pasiekti energiją iš generatorių jis turi. Šį klausimą padeda atsakyti keliai, takai ir trasos. Žemiau pateiktame 4 pav. turime visų įmanomų kiekvieno miesto ir energijos šaltinio kelių kiekį.



4 pav. Kelių tarp miesto ir energijos generatorių šiluminis grafikas (angl., *heatmap*) išrikiuotas vidurkių mažėjimo seka ir miestų vidurkiai

Keliai – vienas iš faktorių, kuriais galima išmatuoti, keliais skirtingais būdais elektra gali

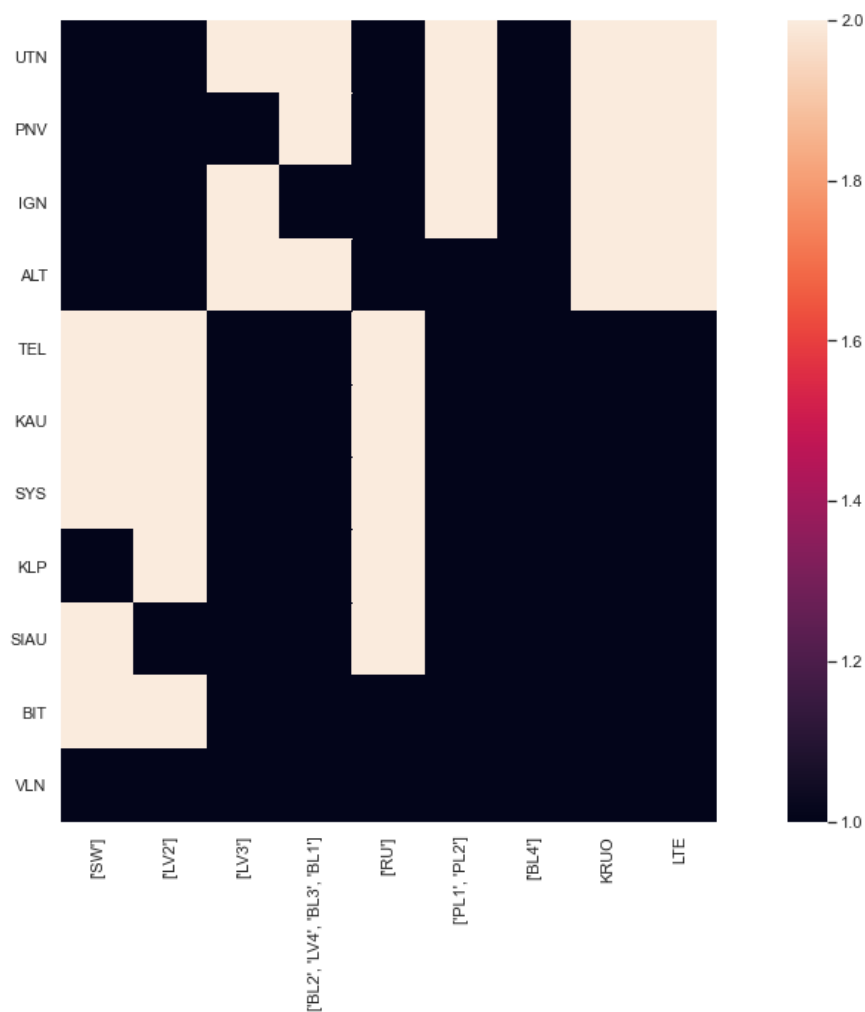
pasiiekti miestą iš generatoriaus. Iš 4 *pav.* grafiko ir vidurkių matome, kad Vilnius, Kaunas ir Alytus gerokai atsilieka nuo kitų miestų kelių iki energijos generatorių skaičiumi. Tai reiškia, kad šie miestai turi mažiausiai skirtingų galimybių elektrai patekti į juos. Tačiau negalime traktuoti, jog miestas, turintis daug kelių iki generatoriaus, yra saugus nuo elektros praradimo, kadangi visi šie keliai gali turėti vieną kritinę perdavimo liniją, kuriai nutrūkus, jie nebegalios. Taigi, šis grafikas labiau parodo, kurie miestai, esant pilnai veikiančioms perdavimo linijoms, yra labiausiai apsisaugoję nuo energijos trūkumo, kadangi esant vienos ar kitos linijos dideliame apkrovime, energiją galima perduoti kitu keliu. Pavyzdžiui, Utena turi 8 kelius iki *W1* vėjo jėgainės, Klaipėda, kuri yra tiesiogiai sujungta su šia jėgaine, turi tik vieną. Tokiu atveju, jeigu Klaipėdai nėra reikalo sunaudoti visos šios vėjo elektrinės generuojamos energijos, egzistuoja 8 skirtingi keliai, kuriais galima perduoti šią energiją Utenos miestui. Dėl didelio skirtingų kelių skaičiaus įmanoma rasti tokį kelią, kuris minimaliai apkraus kitas tiekimo linijas ir šios vėjo generuojamos energijos nepastovumas nesukels sunkumų. Tai būtų labai svarbu tuo atveju, jeigu vakaras būtų vėjuotas ir ši jėgainė gamintų labai didelį kiekį energijos. Vėjo energija yra pigi (yra buvę atvejų, kai ši buvo nemokama ar jos naudojimas net apmokamas valstybės [1]), bet labai nepastovi [15]. Taigi, visą miestą aprūpinti vien šia energija yra labai rizikinga, bet ją perskirstyti keliems miestams ir visą išnaudoti – ekonomiškai naudinga.

Iš kitos pusės pažvelgę į galimus perėjimus, kurie neturi nei vienos sutampančios briaunos tarp energijos generatorių ir miestų, matome visai kitą vaizdą. Tokie keliai grafų teorijoje vadinami trasomis [16]. Šių trasų skaičius iš esmės parodo, kaip saugiai miestas gali apsirūpinti energija. Miestas turintis 2 trasas visada turės priėjimą prie energijos jėgainės, jeigu įvyko tik vienos linijos nutrūkimas, kadangi pagal apibrėžimą šie keliai negali eiti per tas pačias perdavimo linijas. Vien iš grafo (žr., 2 *pav.*) matome, kad trasos gali būti tik iš *Kruonio* ir *Lietuvos energijos* elektros jėgainių, nes užsienio energijos perdavimo tinklai tiesiogiai sujungti viena jungtimi su kuriuo nors miestu. Taigi, nutrūkus šiai linijai, energija iš atitinkamo užsienio energijos tiekimo taško tampa neprieinama visai šaliai. Dėl šios priežasties atmesime tokias perdavimo linijas ir naujais užsienio generatoriais laikysime miestus į kuriuos tiesiogiai perduodama energija iš užsienio. Pavyzdžiui, Šiaulius šiuo atveju laikysime *LV2* energijos generatoriumi, o trasų skaičių tarp miesto, kuriam priskirtas generatorius, laikysime lygų vienetui. Taip pat atmesime vėjo energijos jėgaines ir Vilniaus, Kauno bei Panevėžio dujines elektrines, kadangi jos tiesiogiai sujungtos su atitinkamais miestais ir jų trasos būtų lygios 1 visiems miestams.

Iš 5 *pav.* matome, kad nėra nei vieno miesto turinčio daugiau nei dvi trasas, kurios jungia jį ir generatorių. Taip pat prisiminę 4 *pav.* galime pastebėti, kad miestų išrikiavimas labai panašus: Utena, Ignalina, Panevėžys turi daugiausia kelių ir trasų, tuo tarpu Vilnius – mažiausiai.

1.3. Apibendrinimas

Taigi, iš šio skyriaus apžvalgos matome, kad didieji Lietuvos miestai turi mažiausiai skirtingų kelių ir trasų, kuriomis gali atitekti energija. Kita vertus visi jie turi tiesiogines jungtis su atitinkamais savo elektros generatoriais. Kaunas turi Kauno Algirdo Brazausko hidroelektrinę, Vilnius – Vilniaus šiluminę elektrinę, o Klaipėda – tiesioginę jungtį su Švedija, kuri yra ypač saugi,



Miestas	UTN	PNV	IGN	ALT	TEL	KAU	SYS	KLP	SIAU	BIT	VLN
Suma	14.0	13.0	13.0	12.0	12.0	12.0	12.0	11.0	11.0	11.0	9.0

5 pav. Trasų tarp miesto ir energijos generatorių šiluminis grafikas, išrikiuotas mažėjimo seka ir miestų trasų suma

nes per ją atiteka didžioji dalis Lietuvos sunaudojamos elektros. Grafuose rizikingiausiai atrodo perdavimo linijos – *KAU–KRUE* ir *VLN–LTE*, nes jos iš esmės padalina Lietuvos energijos perdavimo tinklus į dvi atskiras dalis. Pirmoji atskiria Lietuvos rytinės ir vakarinės dalies tinklus, antroji atskiria Vilnių nuo tinklų sistemos, palikdama miestą priklausomą nuo savo šiluminės elektrinės generuojamos energijos ir Baltarusijos tiekiamos energijos.

2. Lietuvos elektros energijos perdavimo tinklų modelis ir sisteminės rizikos modeliavimas

2.1. Duomenų apžvalga ir agregavimas

Šiame skyriuje trumpai apžvelgsime darbe naudojamus duomenis. Galėsime susidaryti aiškesnį vaizdą, ko galime tikėtis iš modelio dirbant su šiais duomenimis bei kokius periodus verta nagrinėti.

Visi pasitelkti duomenys yra traukti iš www.nordpool.com puslapio [5]. „Nord Pool Spot“ – tai elektros energijos pardavimų birža Šiaurės Europos regione, didžiausia tokio tipo organizacija Europoje. Lietuvai atstovauja įmonė „Litgrid“. Iš viso biržoje dalyvauja devynios šalys: Lietuva, Norvegija, Danija, Suomija, Švedija, Estija, Latvija, Jungtinė Karalystė ir Vokietija. Tačiau šiame modelyje naudosime tik su Lietuva susijusius duomenis.

Pirmasis duomenų paketas – Lietuvos energijos sunaudojimas. Čia galime rasti Lietuvos energijos suvartojimą pavalandžiui nuo 2017 metų. Pagal apibrėžimą šis rodiklis apskaičiuojamas taip:

$$Cons = Gen + Imp - Exp - Pump, \quad \text{čia}$$

Cons – energijos sunaudojimas,

Gen – vidinis šalies energijos sugeneruotas kiekis,

Imp – importuotas energijos kiekis,

Exp – eksportuotas energijos kiekis,

Pump – hidroakumuliacinės elektrinės saugyklos poreikiui patenkinti.

Iš agreguotų duomenų galime pastebėti, koks yra nuspėjamas elektros sunaudojimas šalyje ir kaip jis periodiškai pasiskirstęs. Naudodami Furjė transformacijos metodiką pagrįstą „Facebook Prophet“ modelį [13], galime įsivertinti energijos sunaudojamo kiekio skirtingus periodiškumus ir pagrindinį dreifą (angl., *trend*). Šis modelis išskirsto laiko eilutę į sudėtinę funkciją [12]:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \quad \text{čia} \tag{1}$$

t – laiko momentas, iš aibės \mathbb{T} ,

$y(t)$ – laiko eilutės reikšmė laiko momentu t ,

$g(t)$ – dreifo komponentė laiko momentu t ,

$s(t)$ – sezoniškumo komponentė laiko momentu t ,

$h(t)$ – šventinio laikotarpio dėmuo laiko momentu t ,

ϵ_t – baltasis triukšmas.

Šiuo atveju baltąjį triukšmą $\epsilon_t, t \in \mathbb{T}$ laikysime tokiu atsitiktiniu stochastiniu procesu, kurio egzis-

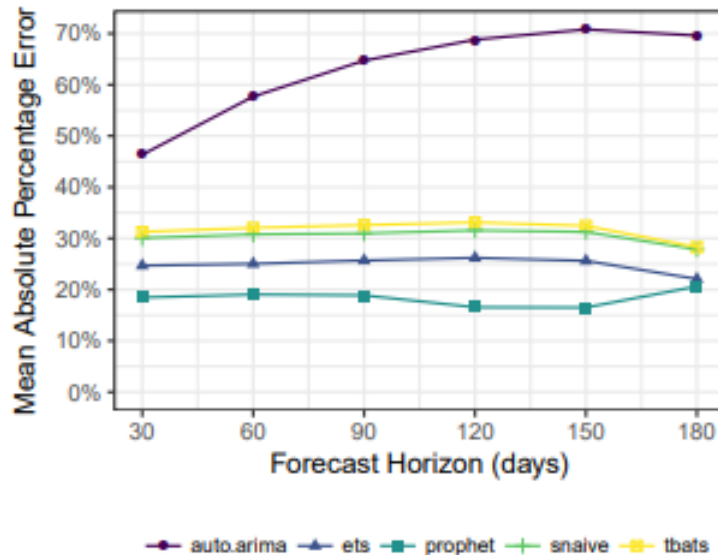
tavimo nepaaiškina visos kitos komponentės. Šis procesas turi savybes:

$$\mathbb{E} \epsilon_t = 0, \quad cov(\epsilon_t, \epsilon_s) = \begin{cases} \sigma^2 & \text{jei } t = s \\ 0 & \text{jei } t \neq s \end{cases} \quad (2)$$

Modelis gali būti adityvus ir multiplikatyvus. Antruoju atveju, (1) esanti lygtis būtų tų pačių komponentių sandauga. Adityvų modelį naudojame, jeigu sezoniniai ir šventiniai prieaugiai yra fiksuoti ir nepriklausomi nuo dreifo reikšmės dydžio. Tuo tarpu multiplikatyvų modelį naudojame, jeigu variacija dėl sezoniškumo ar švenčių didėja, didėjant dreifo reikšmėms. Mūsų atveju naudosime adityvų modelį, nes nagrinėjami duomenys yra pakankamai stacionarūs, neturi reikšmingai kintančios dreifo komponentės. Kaip jau buvo minėta, šis prognozavimo modelis remiasi Furjė transformacijomis nustatinėjant sezoniškumo komponentę $s(t)$. Laikykime P – numanomo periodiškumo laikotarpiu, pvz., $P = 7$, būtų naudojamas nustatinėjant savaitinį sezoniškumą, $P = 365.25$ – metinį sezoniškumą. Tuomet šią funkciją galima aproksimuoti:

$$s(t) = \sum_{n=1}^N \left(a_n \cos \left(\frac{2\pi nt}{P} \right) + b_n \sin \left(\frac{2\pi nt}{P} \right) \right), \quad (3)$$

čia a_n, b_n – svorio koeficientai, N – istoriniuose duomenyse turimų periodų kiekis.



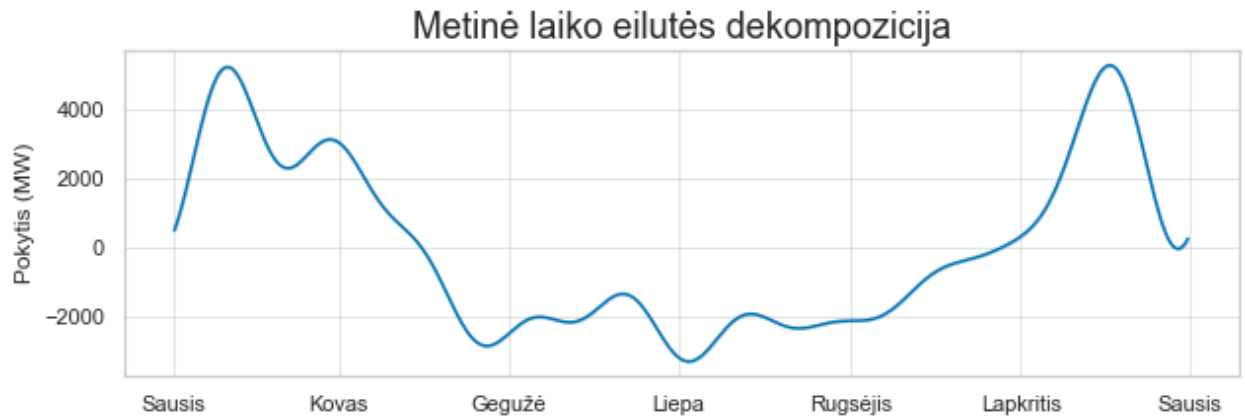
6 pav. [12] straipsnyje modelių nukrypimo nuo tikrųjų reikšmių vertinimas, x periodų į priekį.

Sugeneruoto *Prophet* modelio turimiems Lietuvos sunaudojamos energijos duomenims (žr., 19 pav.) prognozuojamas reikšmės įsivertiname su tikrosiomis reikšmėmis naudodami vidutinės

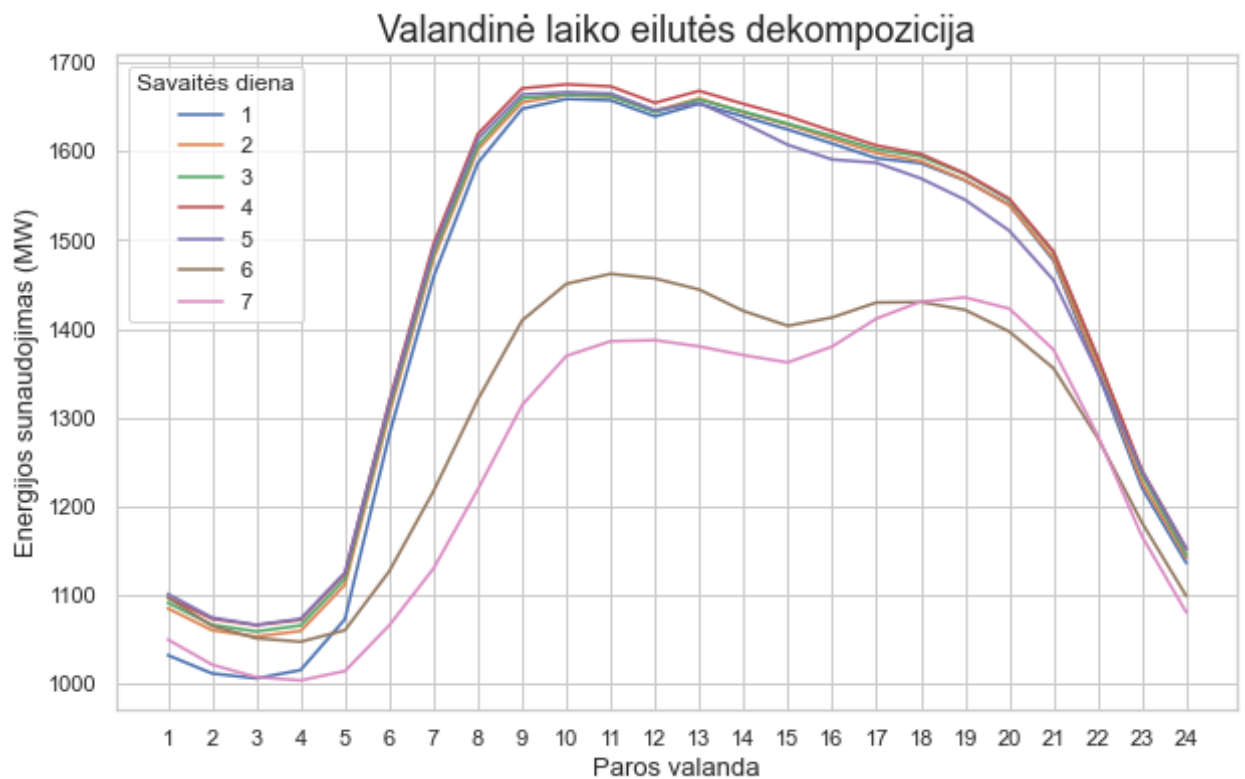
kvadratinės paklaidos tikslumo matą (angl., *root-mean square error*)

$$rmse = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2}.$$

Gauname paklaidą apie 1200 MW per dieną arba 50 MW per valandą ir ši patenka į 95% reikšminumo lygmens pasikliautiną intervalą. Taigi, galime teigti, kad šis modelis turėtų gana teisingai įvertinti periodiškumus, slypinčius po šia laiko eilute.



7 pav. Metinė Lietuvos energijos sunaudojimo laiko eilutės komponentė



8 pav. Valandinė ir savaitinė Lietuvos energijos sunaudojimo laiko eilutės komponentė

Iš 7 ir 8 pav. pavaizduotų modelio sezoniškumų dekompozicijos grafikų galime pastebėti, jog daugiausia energijos Lietuva sunaudoja gruodžio ir vasario mėnesiais, darbo dienomis nuo 8 iki 18

valandos. Taigi, vien iš trumpo duomenų apdorojimo ir analizės galime įsivertinti, kada Lietuvos energijos perdavimo tinklai patiria didžiausią apkrovą ir kokius periodus mūsų modeliui svarbiausia nagrinėti. Būtent dėl šios išvados mes naudosime valandinį duomenų periodą nuo 2019-12-01 iki 2020-02-17, kai Lietuvos energijos sunaudojamas kiekis yra didžiausias.

Antrasis šiame darbe naudojamas duomenų paketas – Lietuvos elektros energijos mainai su Latvija, Lenkija, Rusija, Baltarusija ir Švedija. Šie duomenys labiau chaotiški ir neturintys aiškaus periodiškumo. Iš importo / eksporto duomenų galime teigti, kad Lietuva importuoja nemažą kiekį energijos iš užsienio šalių, su vienu išimtiniu atveju – Lenkija, kuriai Lietuva dažniau eksportuoja energiją. Bendrai Lietuva vidutiniškai per dieną importuoja neto 25.9 tūkst. MW energijos. Taip pat pastebime, kad Lietuva pati negali patenkinti savo energijos vartojimo paklausos (žr., 20 pav.).

2.2. Modelio metodika

Šiame skyriuje aptarsime sisteminės rizikos modeliavimo metodiką. Modelis vertins kiekvieną laiko momentu Lietuvos elektros energijos perdavimo tinklų užimtumą, patenktą energijos paklausą ir paklausos priverstinį mažinimą. Žemiau pateiktas naudojamo modelio veikimo procesas ir jo žingsnių paaiškinimas (9 pav.).

2.2.1. Duomenų surinkimas

Lietuvos energijos perdavimo tinklų vertinimo procesas prasideda nuo duomenų surinkimo. Parenkamas atitinkamas periodas ir importuojami elektros sunaudojimo ir elektros mainų su užsienio šalimis duomenys. Šioje vietoje duomenys jau apdoroti ir sutvarkyti, taigi, jokių problemų nekyla.

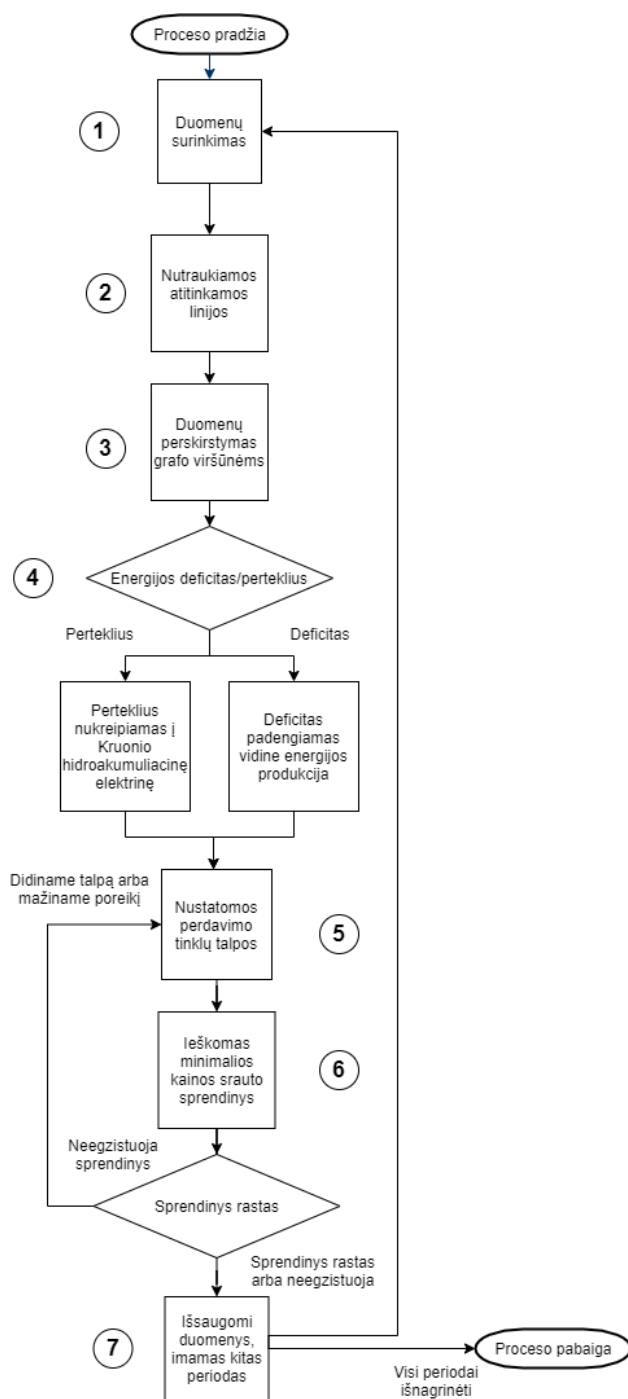
2.2.2. Nutraukiamos atitinkamos linijos

Toliau modelyje yra pašalinamos briaunos, kurios prieš tai buvusioje iteracijoje buvo pašalintos ir liko neatstatytos. Taip pat įvertinama tikimybė pašalinti naujas briaunas ir tikimybės pasekmė realizuojama. Šiuo metodu modelyje simuliuojamas perdavimo linijos gedimas. Šioje dalyje taip pat ieškome ar pašalintos briaunos grafe nesukūrė izoliuotų taškų, kurių briaunų laipsnis būtų lygus 0. Tokiu atveju egzistuoję tokia viršūnė, kuri neturėtų galimybės nei perduoti, nei priimti energijos, taigi, neegzistuoję sprendinys, kuris tenkina visas paklause. Tokias viršūnes eliminuojame iš grafo. Taigi, šioje dalyje apibrėžiamas grafas, jo viršūnės ir briaunos, ant kurių modelis operuos:

$$G = (V, E).$$

2.2.3. Duomenų perskirstymas grafo viršūnėms

Šioje modelio skiltyje vyksta turimų duomenų padalinimas atitinkamoms viršūnėms, jeigu jos egzistuoja. Užsienio mainų kiekis priskiriamas dirbtinai sukuriamoms motininėms viršūnėms, kurios sujungiamos su kiekviena savo dukterine viršūne. Tokiu atveju nebūtina tolygiai paskirstyti importuojamų ir eksportuojamų energijos srautų kiekvienai atitinkamos šalies perdavimo lini-



9 pav. Lietuvos energijos paskirstymo modelio proceso medis

jai. Viršūnė, kuri energiją atiduoda, vadinama šaltiniu ir jai priskiriama neigiama energijos kiekio reikšmė, o viršūnė, kuri energiją prisiima, vadinama kriaukle ir jai priskiriama reikšmė yra teigiama. Toliau Lietuvos sunaudojamas energijos kiekis proporcingai išdalijamas miestus reprezentuojančioms viršūnėms. Proporcija nustatyta remiantis miestų savivaldybėse gyvenančių populiacijų santykiu [10] (3 lent.).

Miestas:	UTN	IGN	SYS	BIT	TEL	ALT	PNV	SIAU	KLP	KAU	VLN
Procentinė dalis:	2.07	2.27	4.28	4.28	4.76	4.86	7.71	9.40	11.37	20.00	29.00

3 lentelė. Lietuvos populiacijos procentinis pasiskirstymas pagal miestus

Taigi, šioje dalyje nustatomi grafo viršūnių šaltiniai (s) ir kriauklės (t) bei joms priskiriamos reikšmės. Užtikrinama sąlyga:

$$s \in V, \quad t \in V, \quad d(s) \in \mathbb{R}^-, \quad d(t) \in \mathbb{R}^+, \quad (4)$$

čia $d(s)$ yra viršūnės s paklausos kiekis, \mathbb{R}^- yra neigiamų realiųjų skaičių aibė, o \mathbb{R}^+ – teigiamų.

2.2.4. Energijos deficito arba pertekliaus paskirstymas

Modelio veikimo principas paremtas balansu tarp elektros pasiūlos ir paklausos, t.y. visų grafo viršūnių atiduodamas energijos kiekis ir priimamas energijos kiekis turi sutapti. Šioje proceso dalyje būtent ir užtikrinamas šis faktas. Apskaičiavus elektros pasiūlos trūkumą, šis bus padengiamas Lietuvos vidinių elektros generatorių – Lietuvos elektrinės, vėjo ir kitų. Šis vidinės sugeneruojamos elektros kiekis negali viršyti 60% maksimalios jėgainės produkcijos lygio. Tokiu būdu bus simuliuojamas elektrinių nesugebėjimas staigiai išnaudoti viso savo pajėgumo. Dėl nepastovaus vėjo jėgainių generuojamo energijos kiekio, joms priskiriamas atsitiktinis generuojamos elektros energijos dydis $X \sim \mathcal{U}(0,100)$ t.y. X pasiskirstęs tolygiai intervale nuo 0 iki 100. Jeigu Lietuvos grafe egzistuotų elektros perteklius, jis bus nukreipiamas į Kruonio hidroakumuliacinę elektrinę energijos kaupimui. Tokiu būdu išlyginamas energijos tiekimo ir sunaudojimo kiekis ir užtikrinama pusiausvyra. Tuomet visų viršūnių paklausa ir pasiūla turi būti lygios, o kadangi paklausos kiekį žymime teigiama reikšme, o pasiūlos – neigiama, išpildome sąlygą:

$$\sum_{v \in V} d(v) = 0. \quad (5)$$

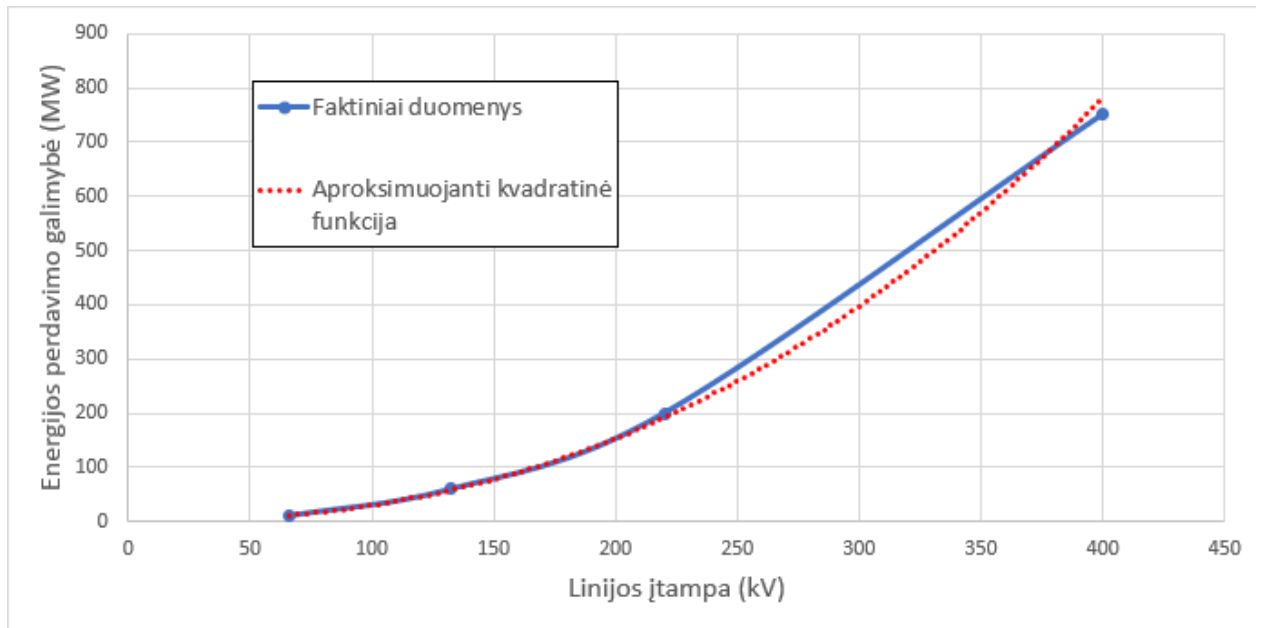
2.2.5. Perdavimo tinklų talpos nustatymas

Toliau modelyje turime nustatyti maksimalią talpą kiekvienai briaunai. Šiame proceso etape iš pradžių nustatomos ypač mažos talpos, kurios vėliau didinamos, jeigu neegzistuoja sprendinys tenkinantis visas sąlygas esant nustatytoms talpoms. Tokiu būdu išvengiama perdavimo tinklų neišnaudojimo problema, nes esant pakankamai didelei talpai ir sprendžiant kaštų optimizavimo uždavinį, reikia naudoti kuo mažiau perdavimo linijų. O iš esmės ribodami maksimalią talpą perdavimo linijose, mes versime energiją tekėti per kitas, neužimtas linijas, taip tolygiau apkraudami visą Lietuvos energijos perdavimo tinklą. Didžioji Lietuvos energijos perdavimo tinklų įtampa nurodoma 300–330 kV. Remiantis „Modern trends in electrical design of E.H.V. transmission line“ [11] straipsniu ir jame pateiktais duomenimis, galima suformuoti priklausomybės funkciją (žr., 10 pav.), su kuria galėsime įsivertinti nežinomas maksimalias talpas Lietuvos perdavimo linijoms. Radę duomenis aproksimuojančią funkciją,

$$y = 0.0006x^{2.3464}, \quad R^2 = 0.9994,$$

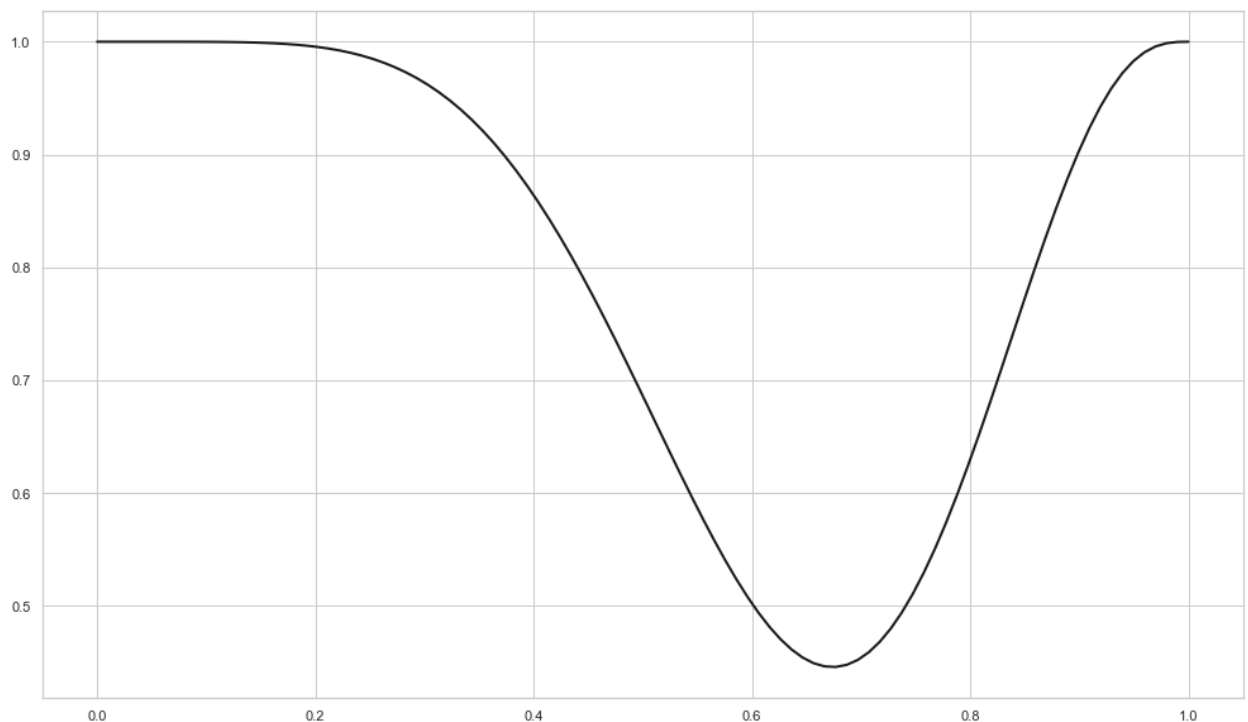
galime teigti, jog energijos perdavimo maksimalus kiekis 300–330 kV linijoms turėtų būti tarp 390 ir 490 MW. Taigi, šiame darbe visas perdavimo linijas, kurių talpos nežinome iš kitų šaltinių,

laikysime 440 MW talpos – vidurkiu rastų reikšmių.



10 pav. Linijos maksimalaus energijos perdavimo kiekio priklausomybės nuo įtampos grafikas

Alternatyvus metodas išspręsti šią tolygaus tinklo apkrovimo problemą būtų nustatyti netiesinę kaštų funkciją kiekvienai linijai.



11 pav. Normalizuotas Beta $\mathcal{B}(\alpha, \beta)$ skirstinio tankio funkcijos atspindys $1 - \frac{x^{\alpha-1}(1-x)^{\beta-1}}{5\mathcal{B}(\alpha, \beta)}$ su parametrais $\alpha = 7.2$ ir $\beta = 4$

Norint kuo tolygiau apkrauti perdavimo tinklus, kaštų funkcijos turėtų būti panašios į tikimybių skirstinius, kuomet didžioji dalis reikšmių grupuojasi aplink tam tikrą rėžį. Vienas iš logiškiausiai atrodančių pasirinkimų būtų Beta skirstinys. Parinkus atitinkamus parametrus ir transforma-

vus funkciją, kaip pavyzdžiui 11 pav., galima kontroliuoti bendrą energijos tėkmės pasiskirstymą grafe. Būtent ši netiesinė funkcija simuliuotų optimalų tinklų apkrovimą, nes skatintų kiekvieną perdavimo liniją apsikrauti maždaug 70% savo pajėgumo. Tai taip pat simuliuotų ekonomišką ne-naudingumą išlaikyti ir operuoti liniją, kuria teka mažas energijos kiekis. Iš kitos pusės taip pat būtų bandoma nesiekti pilno linijos talpos užpildymo, kadangi tokie perdavimo tinklai būtų labiau linkę sutrikti [14].

Taigi, šioje proceso dalyje apibrėžiamas maksimalus galimas energijos perdavimo kiekis kiekvienoje perdavimo linijoje. Šis dydis turi būti teigiamas. Užtikrinama sąlyga:

$$(v,u) \in E, \quad c(u,v) \in \mathbb{R}^+. \quad (6)$$

2.2.6. Minimalių kaštų srauto sprendinys

Šioje dalyje bandoma spręsti mažiausių kaštų tėkmės paskirstymo problema su turimais energijos pasiūlos ir paklausos kiekio bei linijų talpos parametrais. Tai iš esmės yra tiesinio programavimo uždavinys. Matematiškai uždavinį galima apsibrėžti taip:

$$\min_f \sum_{(v,u) \in E} \alpha(v,u) f(v,u), \quad (7)$$

Su sąlygomis:

$$f(v,u) \leq c(v,u), \quad \forall (v,u) \in E, \quad (8)$$

$$\sum_{u \in V} f(u,v) - \sum_{u \in V} f(v,u) = d(v) \quad \forall v \in V, \quad (9)$$

$$\sum_{v \in V} d(v) = 0, \quad (10)$$

$$f(v,u) \geq 0 \quad \forall (v,u) \in E, \quad (11)$$

čia

$d(v)$ – reikiamas energijos kiekis, $d(v) < 0$ energiją generuojanti viršūnė, $d(v) > 0$ energiją naudojanti viršūnė,

$f(u,v)$ – energijos srautas absoliučiu didumu tarp viršūnių,

$\alpha(u,v)$ – energijos tėkmės vieneto kaštai tarp viršūnių,

$c(u,v)$ – briaunos talpa tarp viršūnių.

Sąlyga (10) išpildoma 2.2.4 skyrelyje aprašytoje proceso dalyje. Tuomet (9) lygtį galime pasiversti

į lygčių sistemą

$$\begin{cases} f(v_1, v_1) + f(v_1, v_2) + \dots + f(v_1, v_n) = d(v_1) \\ f(v_2, v_1) + f(v_2, v_2) + \dots + f(v_2, v_n) = d(v_2) \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \ddots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ f(v_n, v_1) + f(v_n, v_2) + \dots + f(v_n, v_n) = d(v_n) \end{cases},$$

įsivedus papildomas sąlygas,

$$\begin{aligned} f(v, v) &= 0 \quad \forall v \in V, \\ f(v, u) &= 0, \quad \text{kai } (v, u) \notin E. \end{aligned} \tag{12}$$

Tokią lygčių sistemą galima išspręsti Gauso–Žordano eliminacijos metodu [8], jeigu sprendinys arba, dažnesniu atveju, begalinė sprendinių aibė egzistuoja. Jeigu egzistuoja tik vienas sprendinys, jį ir laikysime optimaliu tėkmių paskirstymo vektoriumi. O kai leistinių sprendinių aibė turi ne vieną elementą, gauname sprendinių vektorių

$$F = (f(v_1, v_1), f(v_1, v_2), \dots, f(v_n, v_n)),$$

su tam tikrais laisvaisiais nariais. Šių vektorių skaliariškai sudauginame su kaštų konstantų vektoriumi

$$A = (\alpha(v_1, v_1), \alpha(v_1, v_2), \dots, \alpha(v_n, v_n)).$$

Tuomet sprendžiamo optimizavimo (7) uždavinį naudodami simplekso metodą. Ieškome minimalios kainos kintant laisviesiems F vektorių nariams:

$$\min_f \sum_{i=1}^n F \cdot A = \min(\alpha(v_1, v_1)f(v_1, v_1) + \alpha(v_1, v_2)f(v_1, v_2) + \dots + \alpha(v_n, v_n)f(v_n, v_n))$$

su sąlygomis (8) ir (11) (13)

$$f(v, u) \leq c(v, u) \quad \forall (v, u) \in E$$

$$f(v, u) \geq 0 \quad \forall (v, u) \in E$$

Taip gauname optimalų Lietuvos perdavimo tinklų tėkmių pasiskirstymo vektorių F .

Jeigu sprendinys neegzistuoja, t.y. tikslo funkcija nėra aprėžta leistinių sprendinių aibėje, procesas grįžta į 2.2.5 skyriuje aprašytą dalį ir didina talpos parametrus iki kol atsiranda sprendinys arba priskiriama maksimali talpa kiekvienai briaunai. Jeigu tokiu momentu sprendinys vis tiek neegzistuoja, pradedamas vadinamasis „apkrovos numetimas“ (angl., *Load shedding*) kuomet mažinama kiekvienos energiją prisiimančios viršūnės paklausa. Šis sumažintos energijos paklausos kiekis yra vienas iš svarbių indikatorių vertinant sisteminę riziką ir jos atnešamą žalą.

2.2.7. Duomenų kaupimas

Kiekvieno periodo metu išsaugomi modelio duomenys ir parametrai, kad galėtume įsivertinti sisteminę riziką ir jos kitimą Lietuvos elektros perdavimo tinkluose. Pagrindiniai duomenys, kuriuos naudosime nagrinėjimui:

- **Grafas.** Saugosime duomenis apie grafą, jo turimas briaunas ir viršūnes;
- **Modelis.** Saugosime minimalių kaštų srauto sprendinį, linijų apkrovą;
- **Sunaudota energija.** Saugosime sunaudotą energijos kiekį ir pradinį pateiktą energijos kiekį;
- **Nutrūkusios linijos.** Saugosime informaciją apie nutrūkusias perdavimo linijas;
- **Tikimybių parametrai.** Saugosime parametrų vertes.

2.3. Modelio rezultatai

Šiame skyriuje nagrinėsime įvairių parametrų simuliacijos rezultatus ir bandysime įsivertinti sisteminę riziką.

2.3.1. Baziniai modelio rezultatai

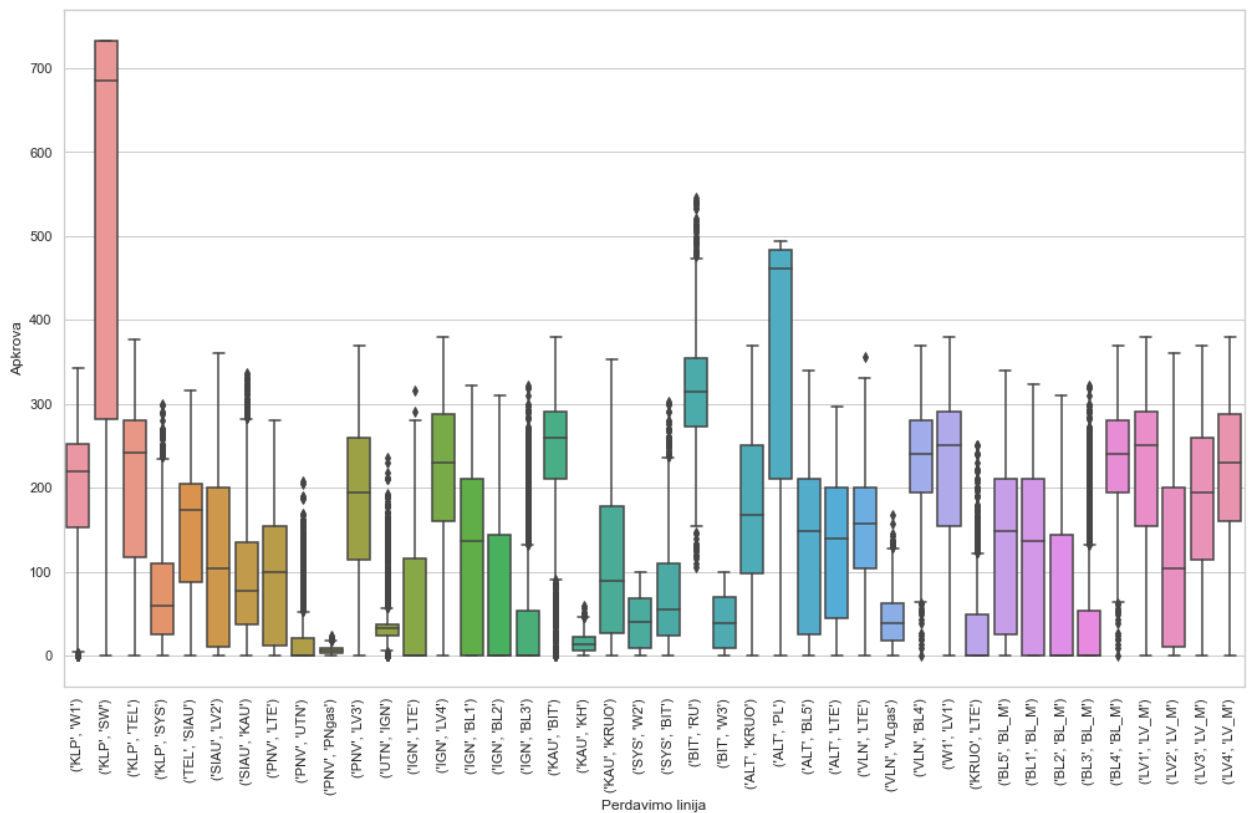
Šioje dalyje įsivertinsime bazinius Lietuvos perdavimo tinklų rodiklius. Baziniu modeliu vadinsime tokį energijos tėkmės modelį, kuriame nei viena energijos perdavimo linija nėra nutrūkus ir negali nutrūkti.

Iš 12 pav. matome tris išskirtines perdavimo linijas – *KLP–SW*, kuri reprezentuoja povandeninę „NordBalt“ jungtį tarp Švedijos ir Lietuvos, *BIT–RU*, kuri atitinka jungtį tarp Rusijos Kaliningrado srities ir Lietuvos bei *ALT–PL* – „LitPol link“ jungtis tarp Lietuvos ir Lenkijos. Šios perdavimo linijos vienintelės viršija mūsų nustatytą 440 MW maksimalios energijos perdavimo talpos ribą. Tačiau „NordBalt“ turi nustatytą maksimalią perdavimo ribą 700 MW, remiantis „Litgrid“ publikacija [3], o Rusijos perdavimo tinklo maksimalią talpą nustatysime didesne, remdamiesi importo / eksporto duomenimis iš „Nordpool“. Lenkijos – Lietuvos jungtis „LitPol link“ taip pat turi patvirtintą 500 MW talpą, remiantis energijos ministerijos duomenimis [9]. Visos kitos Lietuvos perdavimo tinklų linijos neviršija 400 MW, o 75% atvejų, patenka į žemesnę nei 300 MW apkrovos ribą. Šie skaičiai rodo, kad modelis pakankamai gerai pastatytas ir jo veikimas su visomis mūsų darytomis prielaidomis neprieštarauja realiems duomenims.

Perdavimo linijos	KAU-BIT	KLP-TEL	ALT-KRUO	VLN-LTE	TEL-SIAU
Išnaudota talpa (%)	55.33	46.20	38.20	34.74	33.98

4 lentelė. 5–ių didžiausių energijos judėjimą patiriančių Lietuvos energijos perdavimo tinklų talpos išnaudojimas procentais

Kadangi mus domina tik Lietuvos vidinės energijos perdavimo linijos, atmeskime visus elektros judėjimus vykstančius tarp užsienio šalių ir jų perdavimo jungčių su Lietuva. Randame, kad



12 pav. Lietuvos perdavimo tinklų linijų apkrovos „boxplot“ tipo grafikas

net labiausiai užimtoms perdavimo linijoms vidutiniškai neviršija 56% galimos energijos perdavimo talpos.

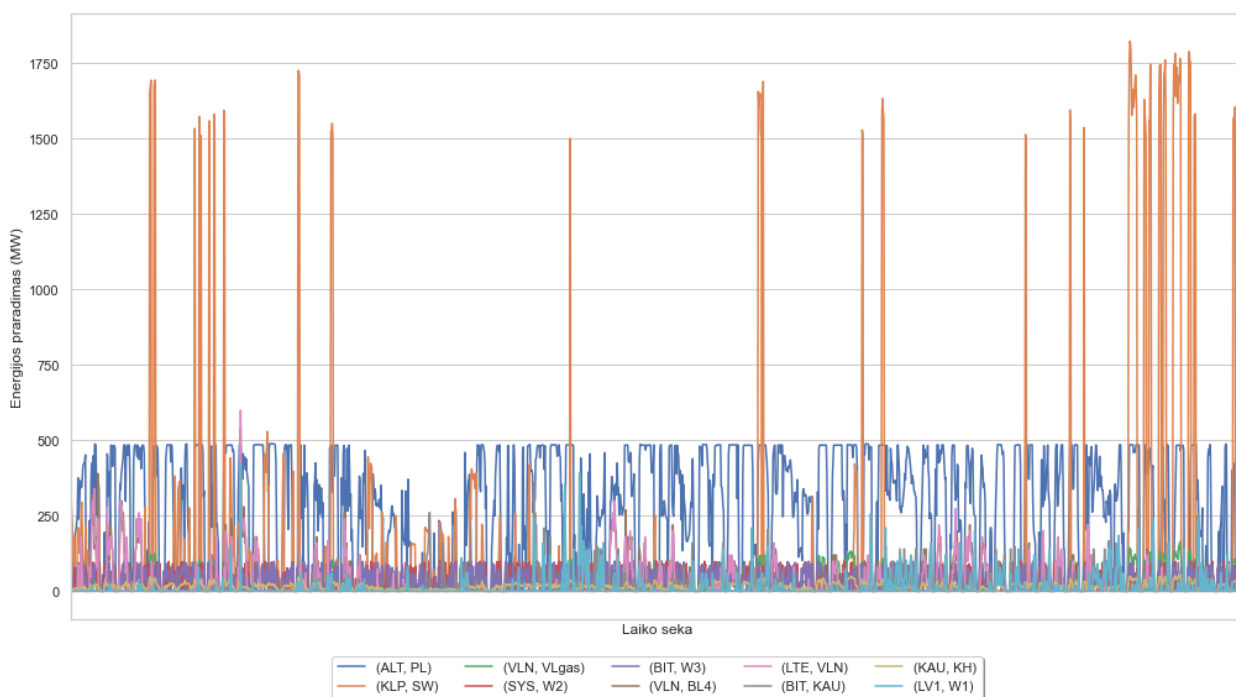
Šiuo baziniu modeliu Lietuvos perdavimo tinkluose neįvyko nei vienas apkrovos numetimas, taigi, Lietuvos energijos paklausa buvo išpildyta 100%.

2.3.2. Pavienių perdavimo linijų pašalinimas iš grafo

Šioje dalyje įsivertinsime vienos perdavimo linijos panaikinimo efektą Lietuvos perdavimo tinklų modelyje. Priešingai nei buvo manyta, nei vienos perdavimo linijos, kurios abiejų viršūnių laipsniai $\deg(u), \deg(v) > 1$, nutūkimas nesukėlė totalaus Lietuvos aptemdymo. Tai reiškia, kad tik linijos, kurios tiesiogiai jungia energijos generatorių su vienu ir vieninteliu miestu, atneša reikšmingus energijos nuostolius. Logiška, kad Klaipėdos–Švedijos jungties gedimas sukuria didžiausią priverstinį apkrovos numetimą. Maži elektros apkrovos numetimai, jeigu kelia problemų, gali būti koreguojami slenkančiais užtemimais, kuomet palaipsniui atjunginėjamas energijos tiekimas ne kritiniams ir mažesnio aktyvumo regionams, kol energijos paklausa ir pasiūla išsilygina [4]. Iš 13 pav. matome, kad didžioji dalis energijos apkrovos numetimo vidutinių reikšmių neviršija 100 MW, tačiau „NordBalt“ jungties sugedimas priveda iki tokių atvejų, kai numetamas beveik visas Lietuvos sunaudojamos energijos kiekis per valandą. Tokie dideli energijos pasiūlos kitimai neabejotinai privestų šalį į visišką elektros užtemdymą. Aiškesnį vaizdą matome 5 lentelėje. Kaliningrado srities jungtis taip pat turi atvejų, kuomet jos gedimas visiškai sugriautų Lietuvos energijos perdavimo tinklus. Visų kitų jungčių praradimas, jeigu nutūkimas būtų suvaldytas teisingai,

paliktų ne daugiau kaip trečdalį šalies be energijos. Iš kitos pusės, matome, jog Alytaus–Lenkijos perdavimo linijos gedimas sukelia nuoseklius energijos nuostolius. Tokie pastovūs energijos netekimai galėtų būti subalansuoti padidinus išperkamos energijos kiekį iš kitų šalių ir paleidžiant dirbti vidines šalies jėgaines pilnu pajėgumu. Verta pastebėti, kad beveik visos problemas keliančios linijos jungia energijos generatorių su miestu. Tačiau yra ir viena išsiskirianti linija tarp Bitėnų ir Kauno, kurios nutraukimas irgi smarkiai sutrikdytų tinklų veiklą.

Taigi, šioje dalyje nustatėme svarbiausius Lietuvos perdavimo tinklus, kurių nutrūkimas sukeltų ne tik energijos tiekimo trikdžius, bet galimai ir visišką Lietuvos energijos tiekimo nutrūkimą. Toliau nagrinėjamuose modelio variantuose šių linijų neįtrauksime į galimybę sugesti, nes jos pačios iš savęs turi didelę tikimybę sugriauti visą Lietuvos energijos perdavimo tinklą.



13 pav. Energijos apkrovos numetimo kiekis, nutraukus perdavimo linijas (10 didžiausių vidurkių turinčios linijos)

Perdavimo Linija	Maksimalus
(KLP, SW)	1823.0
(BIT, RU)	1586.0
(LTE, VLN)	600.0
(KLP, W1)	600.0
(KAU, KRUE)	600.0
(SIAU, LV2)	600.0
(VLN, BL4)	540.0
(ALT, PL)	489.0
(LV1, W1)	420.0
(BIT, KAU)	392.0

Perdavimo Linija	Vidutinis
(ALT, PL)	264.5
(KLP, SW)	107.8
(VLN, VLgas)	41.9
(SYS, W2)	41.0
(BIT, W3)	39.9
(VLN, BL4)	21.7
(LTE, VLN)	21.7
(BIT, KAU)	14.9
(KAU, KH)	14.8
(LV1, W1)	12.8

5 lentelė. 10 Svarbiausių perdavimo linijų energijos numetamo kiekio atžvilgiu

2.3.3. Perdavimo linijų lūžių tikimybės įtaka energijos paskirstymo modeliui

Šioje dalyje ieškosime kertinių tikimybių parametrų, su kuriais mūsų modelyje atsirastų reikšmingi energijos nuostoliai. Pirmasis parametras – linijos nutrūkimo tikimybė modelyje p_1 . Iš 2.3.2 skyrelio rezultatų matome, kad pavieniai linijų, kurios nesijungia su užsienio šalimis, nutrūkimai didelės įtakos neturėjo. Šiuo atveju modelyje galės būti daugiau nei viena nutrūkusi linija. Kad išvengtume neišsprendžiamų kraštutinumų, kurie atsiranda modelio grafui tapus nejungiu, nagrinėsime tik pografį su didesniu kiekiu viršūnių. Žinoma, perdavimo tinklų linijos pačios nepasitaiko, dėl to mes įsivesime dar vieną tikimybinį parametą – nutrauktos linijos tikimybę grįžti į modelį p_2 . Šis parametras simuliuos nutrūkusią liniją taisymo laikotarpį. Pasirinkau šį procesą vertinti tikimybėje, o ne fiksuotu laiku, kad atsirastų tam tikra variacija, kuri atspindėtų realių darbų trukmės neužtikrintumą. Dėl nemažo kiekio tikimybinių matų ir neužtikrintumo dirbant su jais, modelio vertinamą periodą trumpinsime, tačiau simuliaciją su kiekvienu parametru vertinsime daugiau kartų. Taip tikimasi sumažinti atsitiktinių ir mažai tikėtinų reikšmių įtaką.

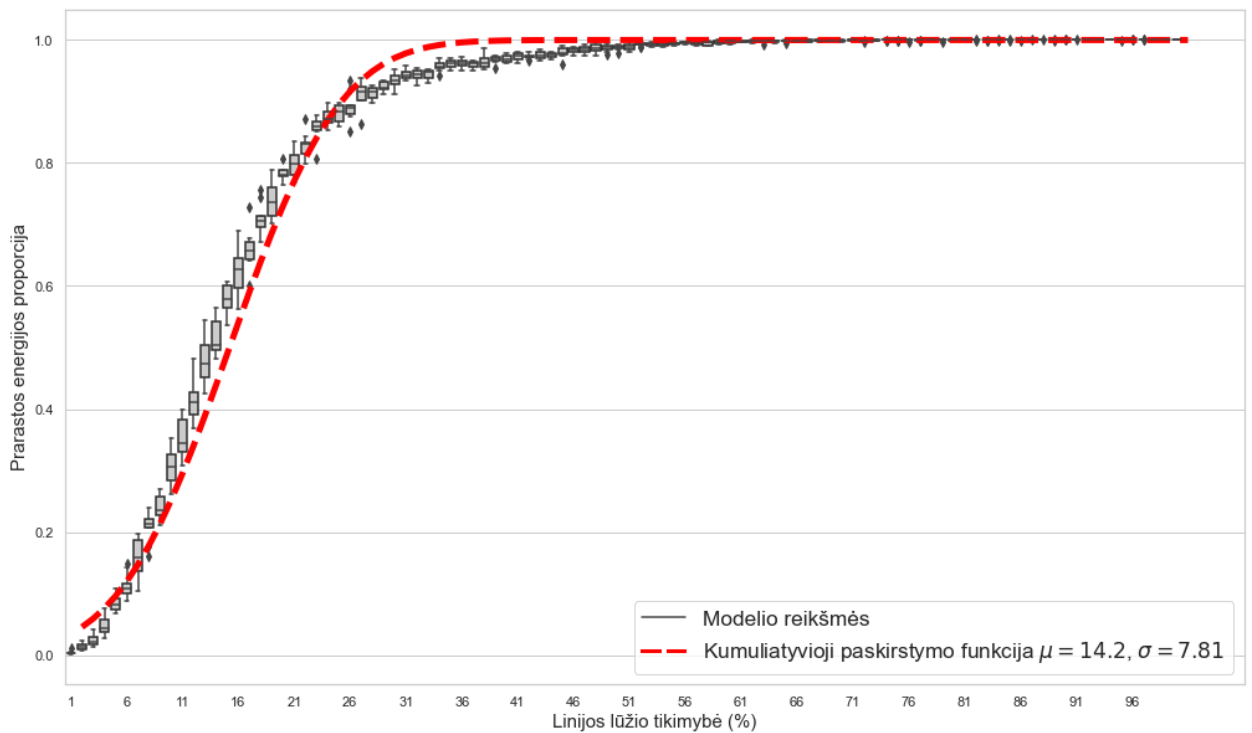
Pradžioje įsivertinsime modelį su fiksuotu parametru $p_2 := 1$, o kiekvienos linijos nutrūkimo tikimybės laikysime nepriklausomomis ir lygiomis p_1 . Taigi, kiekvieną valandą mūsų modelis atsistato į pirminę padėtį su visomis apibrėžtomis ir veikiančiomis perdavimo linijomis. 14 pav. matome pavaizduotus simuliacijos $p_1 = [0, 0.1, \dots, 0.9, 1]$, replikuotos $n = 20$ kartų rezultatus. Reikšmių pasiskirstymas ištis varijuoja nemažai, tačiau priklausomybė tarp linijos lūžių tikimybės ir prarastos energijos yra aiškiai matoma. Vos su 15% lūžio tikimybėje daugiau nei pusė Lietuvos liktų be energijos, o jeigu tokie lūžiai būtų prastai suvaldyti ir energija reikiamai neperskirstyta, galimai visa Lietuva netektų energijos. Tuo tarpu lūžio tikimybė 50% ir didesnė paliktų šalį visiškai be galimybės apsirūpinti elektros tiekimu.

Naudodami mažiausios paklaidos metodiką, galime surasti aproksimuojančią funkciją mūsų modelio sugeneruotoms reikšmėms. Apskaičiuojame, kad normaliojo skirstinio su parametrais $\mu = 14.78$ ir $\sigma = 8.71$ kumuliatyvioji pasiskirstymo funkcija gana tiksliai replikuoja mūsų duomenis. Tai užtikrina ir determinacijos koeficiento (14) reikšmė $R^2 = 0.993$.

$$\begin{aligned} R^2 &= 1 - \frac{SS_{res}}{SS_{tot}}, \quad \text{čia} \\ SS_{res} &= \sum_i (y_i - f_i)^2, \\ SS_{tot} &= \sum_i (y_i - \bar{y})^2, \quad \text{o} \end{aligned} \tag{14}$$

y_i – gautos vertės, \bar{y} – gautų verčių vidurkis, o f_i – aproksimuojančios funkcijos reikšmės.

Turint omenyje, jog mūsų modelis neužtikrina pačio tiksliausio energijos perskirstymo varianto, ši aproksimuojanti kreivė, rodydama mažesnius energijos praradimus intervale $p_1 \in (0.11, 0.26)$, galimai labiau atitiktų tikro Lietuvos perdavimo tinklų energijos praradimų kiekius. Dėl savo tolydumo, ši pasiskirstymo funkcija padeda mums įsivertinti prarandamos energijos tempo kitimą, kylant linijų lūžių tikimybėms. Funkciją išvestinės būtent ir parodo šį rodiklį. Toliau



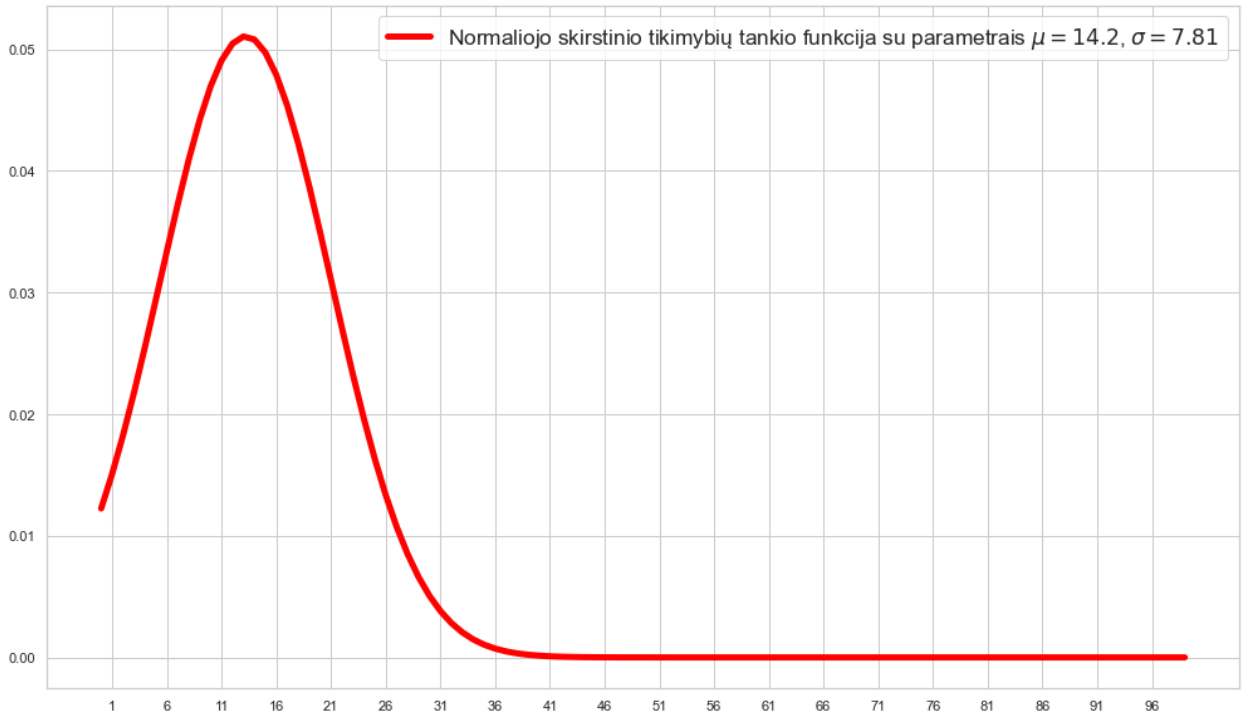
14 pav. Simuliuojamo modelio prarastos energijos proporcijos nuo linijos nutrūkimo tikimybės p_1 priklausomybės grafikas; normaliojo skirstinio kumuliatyvioji paskirstymo funkcija su parametrais $\mu = 14.2$ ir $\sigma = 7.81$

naudosimės tikimybių teorijos faktu, kuris teigia, kad kumuliatyvioji pasiskirstymo funkcija lygi pasiskirstymo funkcijos tankio integralui:

$$F(x) = \int_{-\infty}^x f(x)dx.$$

Rezultatus aproksimavome normaliuoju skirstiniu taigi, nagrinėsime jo išvestinę – to pačio skirstinio tankio funkciją. Pavaizdavę ją 15 pav. matome, kaip kiekvienas papildomas linijos nutrūkimo procentas atneša vis didesnę prarandamą energijos proporciją iki 13-ojo procento. Pirmasis energijos nutrūkimo procentas atneša šiek tiek daugiau nei 1% energijos praradimo, tuo tarpu papildomas vieno procento prieaugis linijos nutrūkimo tikimybei nuo 12% iki 13%, atneša 5% energijos praradimą Lietuvos perdavimo tinklų modeliui.

Toliau vertinsime, kokią įtaką prarandamos energijos kiekiui atneša parametro p_2 - perdavimo linijos sutaisymo tikimybės kitimas. Kaip jau buvo minėta, šis tikimybinių parametras atspindės lūžusios linijos atsigavimo tikimybę, kad ir kokia jos nutrūkimo priežastis būtų buvusi. Ar tai elektros inžinierių galimybė sutaisyti nutrūkusią ar pažeistą perdavimo liniją, ar elektros rinkos kainos atsistatymas į normalią būseną, ar vėjo jėgainių grįžimas į generuojančią stadiją. Tai, jog šis parametras nėra fiksuotas laiko tarpas per kurį regeneruojama linija, o tikimybė linijai tapti veikiančia laiko momentu, mums atveria galimybę išprognuoti būsimas reikšmes. Tai galime padaryti panaudoję Markovo grandinių stacionariu paskirstymu. Sakysime, kad procesas yra Markovo grandinė (tiksliau – diskrečiojo laiko Markovo grandinė), jei, bet kuriam natūraliajam skaičiui n ir bet



15 pav. Normaliojo skirstinio su parametrais $\mu = 14.2$ ir $\sigma = 7.81$ tankio funkcija

kuriems $k, j_0, j_1, \dots, j_{n-2}, j \in E$, teisingos lygybės

$$\begin{aligned} P(X(n) = k | X(0) = j_0, X(1) = j_1, \dots, X(n-2) = j_{n-2}, X(n-1) = j) \\ = P(X(n) = k | X(n-1) = j). \end{aligned} \quad (15)$$

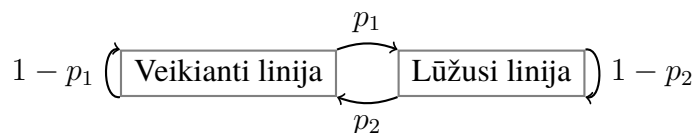
čia, $X = \{X(t), t \in \mathbb{T}\}$ – atsitiktinis procesas apibrėžtas tikimybinėje erdvėje $(\Omega, \mathcal{F}, \mathbb{P})$ ir įgyjantis reikšmes iš mačios erdvės (E, \mathcal{E}) . Tuomet (15) lygybės dešinėje esančią tikimybę vadinsime *perėjimo iš j -osios būsenos į k -ąją būseną momentu n tikimybe* ir žymėsime $p_{jk}^{(n)}$. O matricą

$$\pi^{(n)} = \begin{pmatrix} p_{11}^{(n)} & p_{12}^{(n)} & \cdots \\ p_{21}^{(n)} & p_{22}^{(n)} & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix}, \quad (16)$$

vadinsime *perėjimo matrica* [7]. Kadangi mūsų tikimybės linijoms trūkti ar tapti sveikomis nepriklauso nuo laiko momento n , o tik nuo jų būsenos, tai $p_{jk}^{(n)} = p_{jk}$, $\forall n \in \mathbb{T}$. Tokią Markovo grandinę vadinsime *homogenine*. Markovo grandinės stacionariuoju skirstiniu arba pusiausvyros būseną vadinsime tokių reikšmių rinkinį P , kurio Markovo perėjimo matrica nepakeičia, t.y.

$$P = P\pi. \quad (17)$$

Mūsų atveju kiekviena perdavimo linija turi dvi būsenas: veikianti ir lūžusi. Tuomet tikimybė p_1 simbolizuoja tvarkingos linijos perėjimą į sugedusią, o p_2 – sugedusios linijos perėjimą į tvarkingą tikimybę. O pasilikimo esamoje būsenoje tikimybės atitinkamai $1 - p_1$ ir $1 - p_2$, veikiančiai linijai ir lūžusiai. Taigi, galime susidaryti Markovo grandinės procesą (16 pav.).



16 pav. Markovo grandinės procesas perdavimo tinklų linijoms

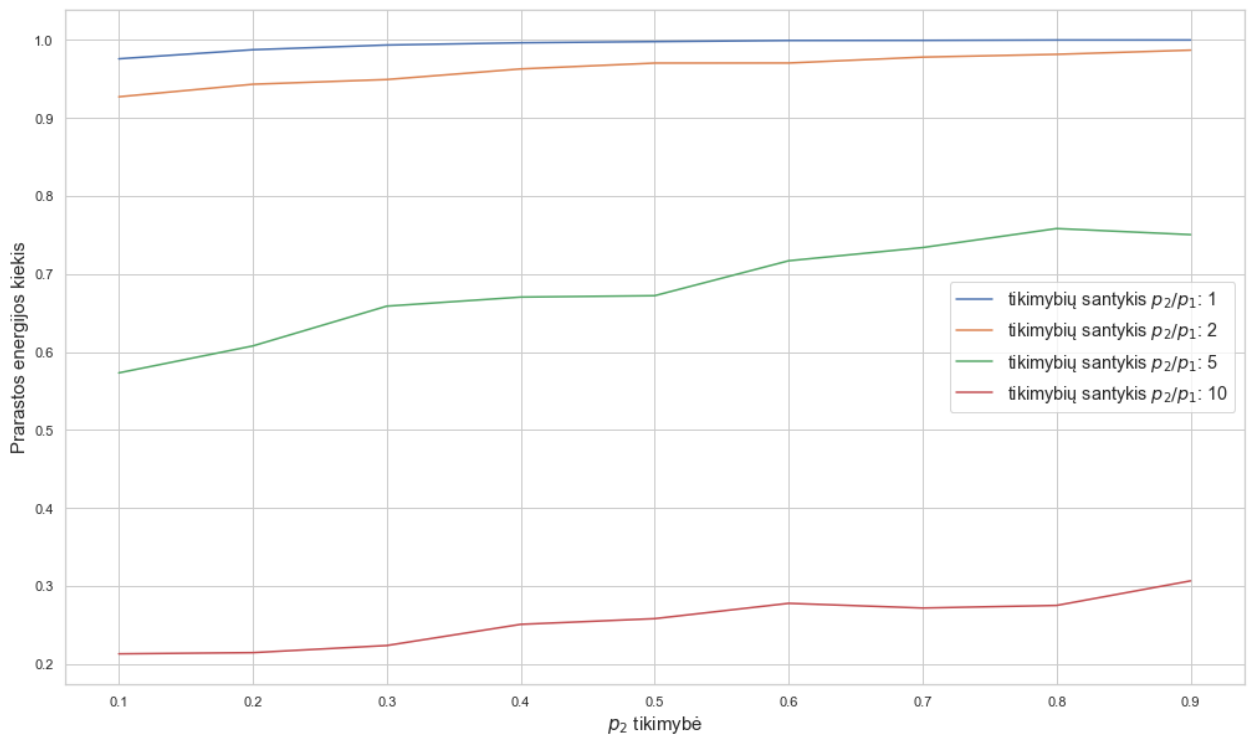
Tuomet, žinodami abu rodiklius, mes galime rasti stacionarią poziciją, t.y. kiek lūžusių ir sveikų perdavimo linijų bus grafe po tam tikro laikotarpio. Laikykite kintamąjį s_1 – sveikų linijų proporcija, o s_2 – nutrauktų. Tuomet išsprendę lygtį:

$$\begin{aligned}
 s_1 + s_2 &= 1, \\
 s_2 &= s_2(1 - p_2) + s_1 p_1, \quad \text{ir} \\
 s_1 &= s_1(1 - p_1) + s_2 p_2, \quad \text{pagal (17)} \implies \\
 \frac{p_2}{p_1} &= \frac{s_1}{s_2}, \quad \text{kai } p_1 \neq 0.
 \end{aligned} \tag{18}$$

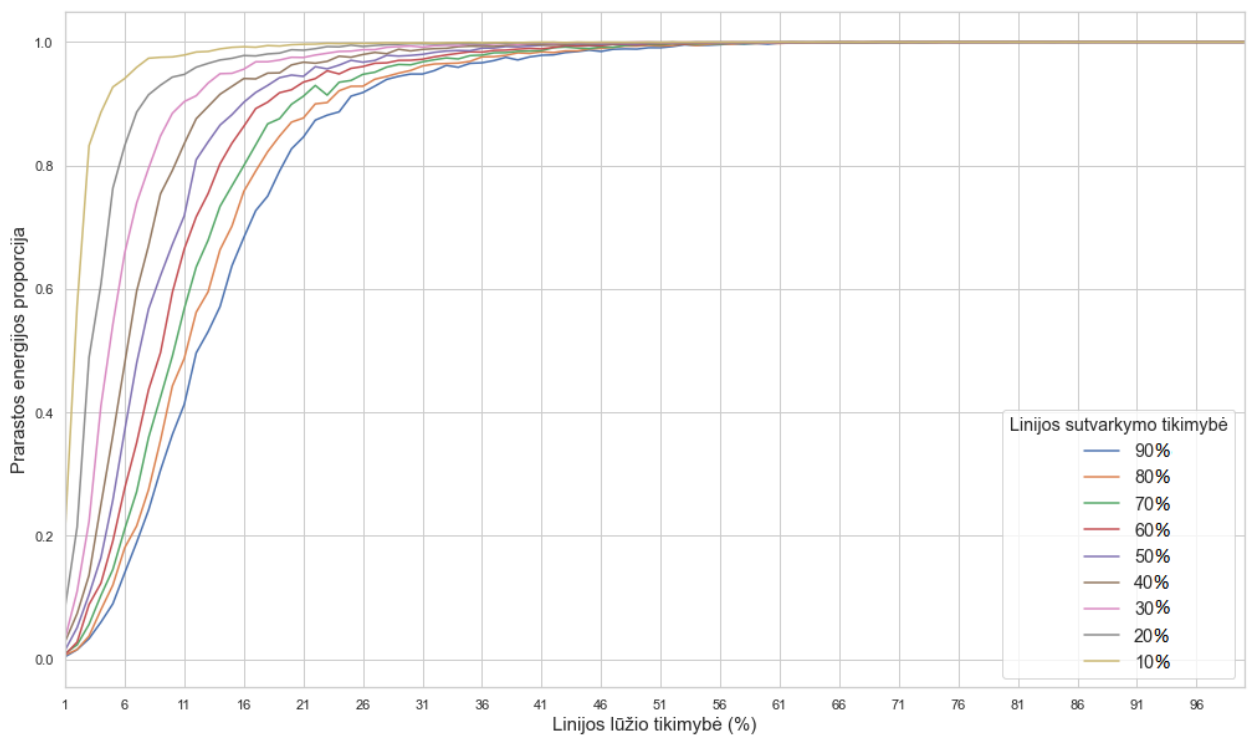
Gauname, jog sveikų ir lūžusių linijų santykis tiesiogiai proporcingas linijos sutaisymo ir linijos sugedimo proporcijai. Toliau naudodami prielaidą, kad nutrūkusių perdavimo linijų kiekis proporcingai nurodo prarastos energijos kiekį: $s_2 \sim l_e$, kur l_e – prarastos energijos proporcija, galime bandyti nuspėti būsimas reikšmes. Taip pat jeigu $s_2 \sim l_e$, tuomet ir $\frac{p_2}{p_1} \sim l_e$, tokiu atveju gauname priklausomybę, tarp visų s_1, s_2, p_1, p_2 kintamųjų. Taigi, mums pakaktų žinoti du iš jų ir galėtume teoriškai pasakyti likusiųjų reikšmių vertes. Realiam gyvenime tokie įvertiniai yra ypač naudingi, nes dažnu atveju visų parametrų nustatyti neįmanoma. Jeigu žinotume nutrūkusių linijų kiekį ir jų taisymo laiką, galėtume apytiksliai įsivertinti, kokio dydžio sisteminę riziką patyrė Lietuvos perdavimo tinklai. Žinoma realybėje energijos tinklų sutrikimai yra sprendžiami kaip įmanoma greitesniu tempu ir energijos praradimai retai viršija kelias valandas. Per tokį trumpą laiką tarpą perdavimo tinklai nespėtų įgyti Markovo grandinės pusiausvyros. Taip pat esant ribotam ir nedideliui linijų kiekiui, proporcijos ne visada galės būti įgyvendintos. Tokį rezultatą gauname ir iš modelio simuliacijos. Dėl sutrumpinto laikotarpio, kuriame simuliuojamas Lietuvos perdavimo tinklų energijos perskirstymo modelis, šis neturi laiko pasiekti ekvivalentumo būsenos. Jeigu Markovo grandinės stacionarus pasiskirstymas būtų pasiektas, o prielaida $s_1 \sim l_e$ teisinga, 17 pav. kreivės turėtų būti nepriklausomos nuo kintamųjų ir įgytų panašias reikšmes. Tačiau iš simuliuotų rezultatų matome, jog ši priklausomybė egzistuoja ir yra didėjanti augant tikimybėms p_1 ir p_2 .

Bendriniai šio modelio grafikai 18 pav. panašūs į prieš tai nagrinėtą 14 pav. grafiką su reikšmėmis $p_2 := 1$ ir $p_1 \in [0.01, 1]$. Skirtumai tarp šių visų energijos prarandamo kiekio grafų – jų statusas. Kaip ir tikėtasi, kuo mažesnė energijos perdavimo linijos atsitiesimo tikimybė, tuo didesniu tempu, kintant tikimybei p_1 , prarandamas energijos kiekis.

Taigi, šiame skyriuje iš simuliacijų rezultatų pamatėme, jog mažos nutrūkimo tikimybės, atneša mažus, bet suvaldomus ir greit koreguojamus energijos praradimus. Tačiau labai svarbu šias lūžių tikimybės eliminuoti arba stabdyti jų augimą, nes kuo labiau šios tikimybės auga, tuo didesnį kiekį energijos Lietuvos tinklų modelis praranda. Taigi, norint išvengti visiško energijos pradi-



17 pav. Simuliuojamo modelio prarastos energijos proporcijos priklausomybė nuo santykio $\frac{p_2}{p_1}$



18 pav. Simuliuojamo modelio prarastos energijos proporcijos nuo linijos nutrūkimo tikimybių p_1 ir p_2 priklausomybės grafikas

mo, labai svarbu ne tik užkirsti kelią kitų linijų gedimams, kaip matėme 14 pav., bet ir užtikrinti greitą lūžusių energijos perdavimo tinklų atstatymą. Taip pat radome, jog trumpalaikiai perdavimo linijų sutrikimai (iki 7 d.), yra nepakankamai ilgi, kad įsigaliotų Markovo grandinių pusiausvyra. Taigi, vietoj stabilaus ir vienodo energijos prarandamo kiekio, kuris teoriškai turėtų įsivyrėti,

iš modeliavimo rezultatų matome, jog kuo nepastovesnės simuliuojamo Lietuvos perdavimo tinklų sistemos, tuo didesnis energijos kiekis prarandamas. Iš kitos pusės pastebėjome, kad pavienių energijos perdavimo linijų praradimai simuliaciniame modelyje atneša reikšmingos žalos tik tuo atveju, kai šie jungia miestą su energijos generatoriumi. Svarbiausi, žinoma, tie tinklai, kurie sujungti su didžiausią energiją tiekiančiomis viršūnėmis. Lietuvos atveju tai būtų „NordBalt“ jungtis su Švedija, „LitPol“ jungtis su Lenkija ir, retkarčiais, jungtis su Rusijos Kaliningrado sritimi. Remdamiesi šia sisteminės rizikos modeliavimo informacija, Lietuvos perdavimo tinklų specialistai, įsivertinę kiekvienos jungties lūžio tikimybę, priklausomą nuo išorinių faktorių: didelių karščių, stiprių vėjų ar užsienio šalių įsipareigojimų nevykdymų, galėtų išsiskaičiuoti, kokios dalies energijos Lietuva galimai neteks. Tuomet turėtų galimybę įspėti atitinkamas jėgaines, kad šios būtų pilnai pasiruošusios didinti pajėgumus, jeigu šie lūžiai įvyktų.

Rezultatai ir išvados

Sisteminės rizikos modeliavimo darbe buvo atliktos dvi rizikos analizės: viena remiantis grafų teorijos metodika, kita – kuriant simuliacinį perdavimo tinklų modelį. Pirmojoje dalyje buvo apžvelgtas Lietuvos energijos perdavimo tinklų grafas ir sudarytos pirminės, grafų teorija paremtos, išvados apie tinklų stabilumą ir svarbą bei miestų energijos apsirūpinimą. Radome tinklų jungtis kurių lūžiai skeltų perdavimo tinklų sistemos grafą į du nejungius pografius. Taip pat buvo įvertinta miestų galimybė priimti energiją, atsižvelgiant į visus skirtingus būdus, kuriais ši juos gali pasiekti.

Antrojoje dalyje buvo apžvelgti Lietuvos energijos sunaudojimo ir eksporto, importo duomenys, nustatytas jų periodiškumas ir įvertintas energijos paklausą prognozuojančio Furjė transformacija paremto modelio tikslumas. Buvo rasti periodai kuriuose energijos paklausa didžiausia ir remiantis šiais duomenimis toliau vykdoma analizė. Darbe taip pat sukurtas Lietuvos energijos paskirstymo perdavimo tinklais modelis. Šis modelis optimaliausiu būdu paskirstydavo energiją ir turėjo energijos numetimo funkciją, kuri leidžia rasti paskirstymo sprendinį, sumažinus energijos paklausos kiekį, jeigu pirminiu būdu toks sprendinys neegzistuoja. Su šiuo modeliu buvo atliktos simuliacijos, kuriomis buvo nustatyti svarbiausi perdavimo tinklai ir jų sutrikimo energetinė žala. Taip pat buvo įvertintas sisteminės rizikos poveikis šalies energijos pasiūlos kiekiui. Gautas priklausomybės sąryšis tarp sisteminės rizikos ir neaprupintos energijos paklausos Lietuvos modelyje. Šiai priklausomybei taip pat rasta aproksimuojanti funkcija, kuri padėjo įsivertinti kiekvieno papildomo linijos trūkimo procento atneštą žalą modeliui. Taip pat buvo išbandytas Markovo grandinių stabilizavimosi sprendinys, tačiau praktiniai duomenys parodė, jog per trumpą periodą sisteminės rizikos įvertiniai nepasiekia šio ekvivalentumo.

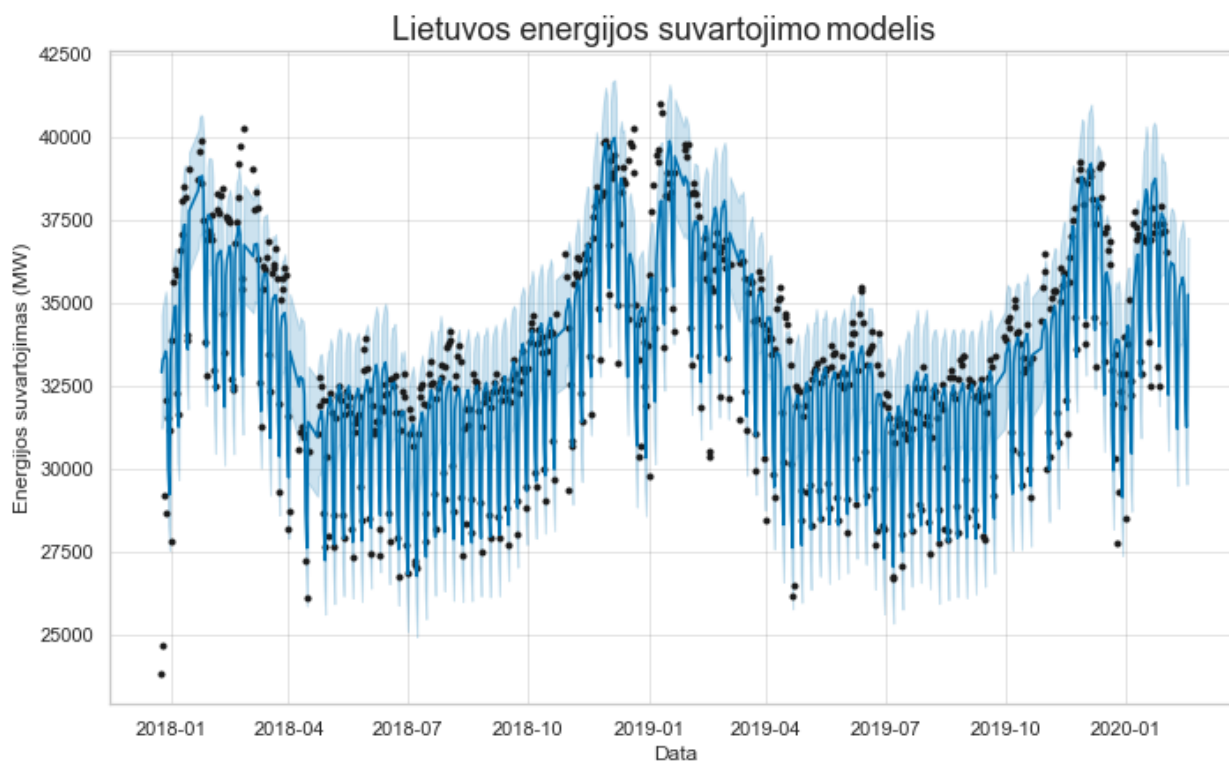
Šį darbą tęsiant būtų įdomu išbandyti kitokią energijos paskirstymo perdavimo tinklais metodiką, kuri perdavimo kaštus priskirtų pagal netiesinę funkciją, kaip pvz., 11 pav.. Taip pat naudojantis Furjė transformacijos modelio ar kitu laiko eilučių prognozėms skirtu modeliu (*LSTM*, *SARIMA*) išprognuotomis reikšmėmis simuliuoti nupirktą energijos kiekį. Šį kiekį tuomet įsivertinti su realia paklausa ir ieškoti optimaliausio energijos paskirstymo varianto, kuris nesukeltų didelės sisteminės rizikos, pvz., kad kiekvienas miestas turėtų tam tikrą dalį patenkintos energijos paklausos, o linijos perdavimo talpa neviršytų tam tikro kiekio.

Literatūra

- [1] M. J. Coren. Germany had so much renewable energy on sunday that it had to pay people to use electricity. 2016. URL: <https://qz.com/680661/germany-had-so-much-renewable-energy-on-sunday-that-it-had-to-pay-people-to-use-electricity/> (tikrinta 2020-04-13).
- [2] I. Dobson ir B. A. Carreras. Risk analysis of critical loading and blackouts with cascading events. *Consortium for Electric Reliability Tech. Solutions (CERTS)*, 2005.
- [3] L. electricity transmission system operator - Litgrid. Ten-year 330 kv and 110 kv network development plan 2013–2022 of lithuanian power system:26, 2013.
- [4] D. energy. Rolling blackouts. 2020. URL: <https://www.directenergy.com/learning-center/energy-efficiency/rolling-blackouts> (tikrinta 2020-04-13).
- [5] N. group. Market data for nordpool group. 2020. URL: <https://www.nordpoolgroup.com/Market-data1/#/nordic/map> (tikrinta 2020-04-13).
- [6] T. Kamada, S. Kawai ir k.t. An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15, 1989.
- [7] K. Kubilius. Stochastiniai matematiniai modeliai:49–57. DOI: <http://www.techmat.vgtu.lt/konspektai/stochastiniai.pdf>.
- [8] U. of Babylon. Gauss-jordan elimination method:1–5. DOI: http://www.uobabylon.edu.iq/eprints/publication_11_24675_1652.pdf.
- [9] M. of Energy of the Republic of Lithuania. Electricity link litpol link. 2020. URL: <https://enmin.lrv.lt/en/strategic-projects/electricity-sector/electricity-link-litpol-link> (tikrinta 2020-04-13).
- [10] L. oficialiosios statistikos rengėjai. Lietuvos gyventojai (2019 m. leidimas). 2019. URL: <https://osp.stat.gov.lt/lietuvos-gyventojai/lietuvos-gyventojai-2019/salies-gyventojai/gyventoju-skaicius-ir-sudetis> (tikrinta 2020-04-13).
- [11] V. B. S. M. Takalkar. Modern trends in electrical design of e.h.v. transmission line:9. DOI: <http://www.iitk.ac.in/npsc/Papers/NPSC1998/p2.pdf>.
- [12] S. J. Taylor ir B. Letham. Forecasting at scale, 2017. DOI: 10.7287/peerj.preprints.3190. URL: <https://peerj.com/preprints/3190.pdf>.
- [13] F. C. D. S. team. Fb pophet. URL: <https://facebook.github.io/prophet/>.
- [14] Wikipedia contributors. Electric power transmission — Wikipedia, the free encyclopedia, 2020. URL: https://en.wikipedia.org/wiki/Electric_power_transmission#Capacity. [Online; accessed 6-May-2020].
- [15] R. Wilson. How reliable is wind power? 2015. URL: <https://www.weforum.org/agenda/2015/01/how-reliable-is-wind-power/> (tikrinta 2020-04-13).
- [16] V. Zacharovas. Grafų teorija:1–83, 2018.

Priedas nr. 1

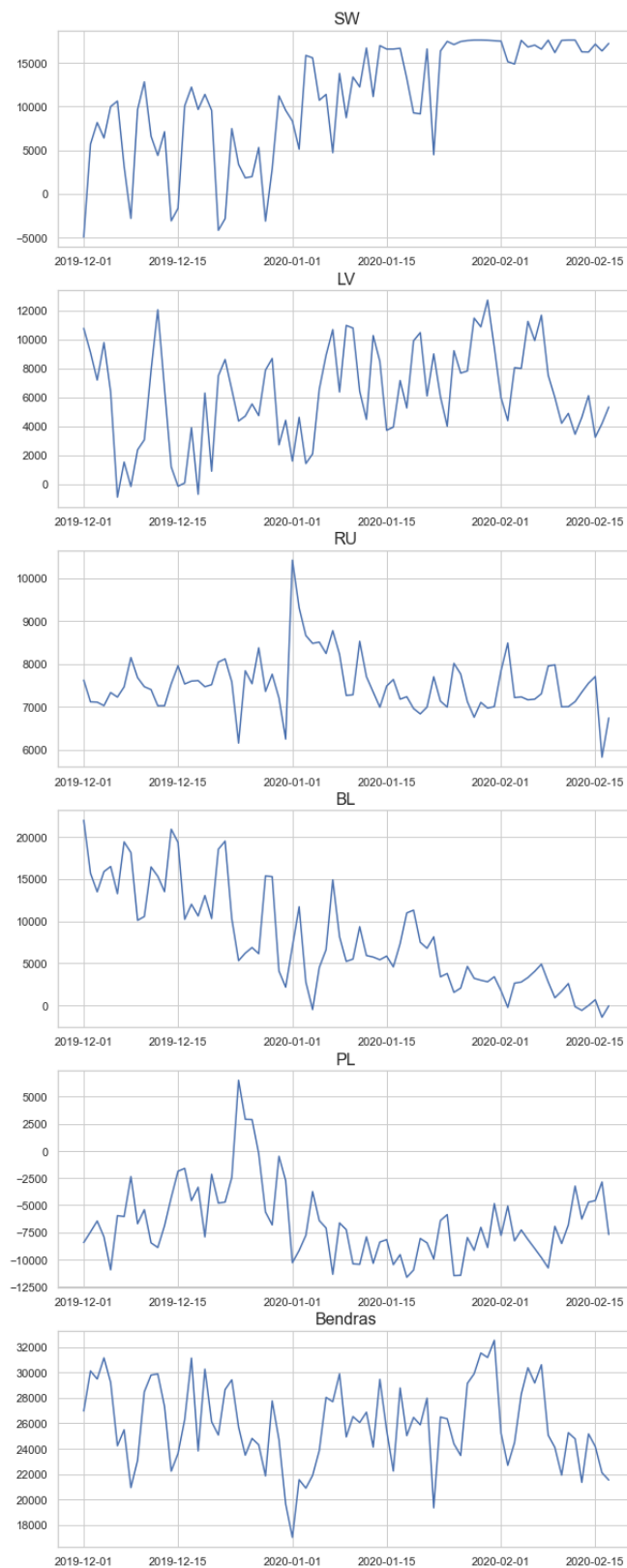
„Facebook Prophet“ modelio prognozuojamos reikšmės



19 pav. „Facebook Prophet“ modelio prognozuotos reikšmės (mėlyna linija) ir faktinės reikšmės (juodi taškai)

Priedas nr. 2

Lietuvos energijos eksporto ir importo duomenys



20 pav. Lietuvos energijos mainų (MW) su užsienio šalimis grafikas. Neigiamos reikšmės - eksportas, teigiamos - importas

Priedas nr. 3

Simuliacinio modelio funkcijos

Visą darbo kodą su analizėmis ir duomenimis galite gauti parašius tiesiogiai darbo autoriui

```
def Assign_capacity(G, line_1):  
  
    '''Funkcija priskirianti grafo G linijoms atalp line_1'''  
  
    for line in G.edges:  
        G.add_edge(line[0], line[1], capacity = line_1 )  
        G.add_edge(line[1], line[0], capacity = line_1 )  
        if line == ('KLP', 'SW') or line == ('SW', 'KLP'):  
            G.add_edge('SW' , 'KLP', capacity = 750 )  
            G.add_edge('KLP', 'SW', capacity = 750 )  
        if line == ('RU', 'BIT') or line == ('BIT', 'RU'):  
            G.add_edge('RU' , 'BIT', capacity = 750 )  
            G.add_edge('BIT', 'RU', capacity = 750 )  
        if line == ('PL', 'ALT') or line == ('ALT', 'PL'):  
            G.add_edge('PL' , 'ALT', capacity = 750 )  
            G.add_edge('ALT', 'PL', capacity = 750 )  
  
    return G
```

```
def Return_flow_dict(Flowsai):  
  
    '''Sukuria Dict tipo pasiskirstyma, kur liniju pora yra raktas, is min  
        cost flow Dict tipo'''  
  
    Dict_a = {}  
    for i in Flowsai:  
        for a in Flowsai[i]:  
            if Flowsai[i][a] != 0:  
                Dict_a[(i,a)] = Flowsai[i][a]  
  
    return Dict_a
```

```
def bound_range(*args, **kwargs):  
  
    '''Range funkcija, kuri ima paskutini argumenta 0, jeigu jis butu buves ne  
        0'''  
  
    unbound_range = np.arange(*args, **kwargs)  
    if unbound_range.min() > 0:  
        return np.append(unbound_range, 0)  
    return unbound_range
```

```
def Foreign_dict(LV, RU, SW, PL, BL):
```

```

'''Transformuoja Importo/Eksporto duomenis ir sudeda juos i Dict'''

dictas = {
    'LV_M' : {'demand' : Decimal(LV *(-1))},
    'RU' : {'demand' : Decimal(RU *(-1))},
    'SW' : {'demand' : Decimal(SW *(-1))},
    'PL' : {'demand' : Decimal(PL *(-1))},
    'BL_M' : {'demand' : Decimal(BL *(-1))}
}

return dictas

```

```

def Demand_dict(Demand):

    '''Paskirsto elektros poreikio duomenis po miestus'''

    # nuo maziausiai svarbaus miesto iki labiausiai:
    # https://osp.stat.gov.lt/lietuvas-gyventojai/lietuvas-gyventojai-2019/
    # salies-gyventojai/gyventoju-
    # skaicius-ir-sudetis

    Miestai = {
        'VLN' : Decimal('0.2900'),
        'KAU' : Decimal('0.2000'),
        'KLP' : Decimal('0.1137'),
        'SIAU' : Decimal('0.0940'),
        'PNV' : Decimal('0.0771'),
        'ALT' : Decimal('0.0486'),
        'TEL' : Decimal('0.0476'),
        'UTN' : Decimal('0.0207'),
        'IGN' : Decimal('0.0227'),
        'SYS' : Decimal('0.0428'),
        'BIT' : Decimal('0.0428')
    }# svarbumo eile

    Miestai_rev = dict(sorted(Miestai.items(), key=lambda x: x[1]))

    M_dict = {}
    for i in Miestai_rev:
        M_dict[i] = {'demand' : Decimal(Demand)*Miestai[i]}

    return M_dict

```

```

def Distribute_Demand(G, Demand_dictas, Foreign_dict):

    '''Paskirsto energijos poreiki ir islygina susidariusi deficita'''

    M = G.copy()
    nx.set_node_attributes(M, Demand_dictas)
    init_dem = Decimal(sum(nx.get_node_attributes(M, 'demand').values()))
    nx.set_node_attributes(M, Foreign_dict)

```

```

deficitas = sum(nx.get_node_attributes(M, 'demand').values())

if deficitas <= 0 :
    if not nx.get_node_attributes(G, 'demand').get('KRU0'):
        prad_k = 0
    else:
        prad_k = nx.get_node_attributes(G, 'demand').get('KRU0')
    nx.set_node_attributes(M, {'KRU0' : {'demand' :prad_k -1*deficitas }})

if deficitas > 0 and deficitas <= 1500:

    Internal_supply = Internal_generation(deficitas)
    nx.set_node_attributes(M, Internal_supply)

chekas = Decimal(sum(nx.get_node_attributes(M, 'demand').values()))
if chekas != 0:
    nx.set_node_attributes(M, Demand_dict(init_dem - chekas))

return M

```

```

def Internal_generation(deficitas):

    '''Paskirsto reikiamo energijos pagaminimo kieki po lietuvos generatorius
    '''

    w1 = Decimal(f"{-100 * random.random():.0f}")
    w2 = Decimal(f"{-100 * random.random():.0f}")
    w3 = Decimal(f"{-100 * random.random():.0f}")
    if -1*(w1+w2+w3) >= deficitas :
        w1,w2,w3 = 0, 0, 0

    likutis = deficitas - -1*(w1+w2+w3)
    # reikia perskirstyti proporcingai, net jeigu nuluzta vienas is situ

    dictas = {
        'W1' : {'demand' : w1} ,
        'W2' : {'demand' : w2} ,
        'W3' : {'demand' : w3} ,
        'KRU0' : {'demand' : Decimal('-0.37')*likutis } ,
        'LTE' : {'demand' : Decimal('-0.42')*likutis } ,
        'KH' : {'demand' : Decimal('-0.05')*likutis } ,
        'PNgas' : {'demand' : Decimal('-0.02')*likutis } ,
        'VLgas' : {'demand' : Decimal('-0.14')*likutis } ,
    }
    return dictas

```

```

def Draw_solution(Distributed_demand):

    '''Funkcija, kuri nupiesia grafa su Return_folw_dict tipo duomenimis'''

```

```

A = Distributed_demand
nx.draw(A, pos = nx.kamada_kawai_layout(G), node_size = 1400)
lab = {}
for i in A.nodes:
    for k in nx.get_node_attributes(A, 'demand'):
        if i ==k:
            lab[i] = str(i) + " " + str(round(nx.get_node_attributes(A, '
                                                    demand')[k],1))

            break
    lab[i] = str(i) + " 0"
lab
nx.draw_networkx_labels(A, pos=nx.kamada_kawai_layout(G), font_size=12,
                        font_color='k', font_family='sans-
                        serif', font_weight='bold', alpha=1
                        .0, labels = lab)
nx.draw_networkx_edge_labels(A,pos=nx.kamada_kawai_layout(G), edge_labels=
                        Return_flow_dict(nx.min_cost_flow(A
                        )))

plt.show()

```

```

def Min_cost_flow_model(G):

    '''Maziausiu kastu pasiskirstymo modelis'''

    M = nx.min_cost_flow(G)

    return M

```

```

def Initialize_Model(G, Demand, LV, RU, SW, PL, BL, prob, broken_edge_list):

    '''Modelio iniciavimas'''

    C = G.copy()

    for broken in broken_edge_list: # break edges that are not fixed
        try:
            C.remove_edge(broken[0], broken[1])
        except:
            pass
        try:
            C.remove_edge(broken[1], broken[0])
        except:
            pass
    try:
        list_of_edges = list(C.edges())
        list_of_edges.remove(('KLP', 'SW'))
        list_of_edges.remove(('SW', 'KLP'))
        list_of_edges.remove(('BIT', 'RU'))

```

```

list_of_edges.remove(('RU', 'BIT'))
list_of_edges.remove(('PL', 'ALT'))
list_of_edges.remove(('ALT', 'PL'))
return_broken_edges = []
for broken_edge in list_of_edges:
    if random.random() <= prob:          # break edge with percentage
                                           given by prob
        return_broken_edges.append(broken_edge)
        try:
            C.remove_edge(broken_edge[0], broken_edge[1])
        except:
            pass
        try:
            C.remove_edge(broken_edge[1], broken_edge[0])
        except:
            pass
except:
    pass

izoliuoti = list(nx.isolates(C))
try:
    for iso in izoliuoti:
        C.remove_node(iso)
except:
    pass

if not nx.is_connected(C.to_undirected()):
    # get a list of unconnected networks
    sub_graphs = list(nx.connected_component_subgraphs(C.to_undirected()))
    main_graph = sub_graphs[0]
    # find the largest network in that list
    for sg in sub_graphs:
        if len(sg.nodes()) > len(main_graph.nodes()):
            main_graph = sg

    C = main_graph.to_directed()

Ci = C.copy()
foreign = Foreign_dict(LV, RU, SW, PL, BL) # distribute foreign
                                           electricity distribution to a dict
dem = Demand_dict(Demand)
C = Distribute_Demand(C, dem, foreign)

for i in range(10,450,10): # try capacity increase
    C = Assign_capacity(C, i)
    try:

```

```

        M = Min_cost_flow_model(C)
        break
    except:
        pass

try:
    M
except NameError:
    Ci = Assign_capacity(Ci, 440)
    kaunter = 10
    LV_j = LV
    RU_j = RU
    SW_j = SW
    PL_j = PL
    BL_j = BL
    for j in np.arange(Decimal(Demand)-10, Decimal(0), Decimal(-10)): #
        try demand reduce

        LV_j = 0 if abs(LV_j) <= 11 else LV_j - np.sign(LV)*kaunter
        RU_j = 0 if abs(RU_j) <= 11 else RU_j - np.sign(RU)*kaunter
        SW_j = 0 if abs(SW_j) <= 11 else SW_j - np.sign(SW)*kaunter
        PL_j = 0 if abs(PL_j) <= 11 else PL_j - np.sign(PL)*kaunter
        BL_j = 0 if abs(BL_j) <= 11 else BL_j - np.sign(BL)*kaunter

        foreign = Foreign_dict(LV_j, RU_j, SW_j, PL_j, BL_j) # distribute
            foreign electricity
            distribution to a dict
        dem = Demand_dict(j) # distribute demand for each city

        Cj = Ci.copy()
        Cj = Distribute_Demand(Cj, dem, foreign)
        try:
            M = Min_cost_flow_model(Cj)
            C = Cj
            break
        except:
            pass

try:
    M
except NameError:
    M=[]

```



```

C=Cj

try:
    l = broken_edge_list+return_broken_edges
except:
    l = broken_edge_list

return C, M, l

```

```

def edge_usage(G):

    '''Grazina esamas jungtis tarp virsuniu'''

    C = nx.DiGraph.to_undirected(G)

    return list(C.edges)

```

```

def flip(test_list):

    '''Isrikiuoja duota masyva atvirkstine tvarka'''

    return [(sub[1], sub[0]) for sub in test_list]

```

```

def append_results(df, modelis):

    '''Sudeda duomenis i DataFrame tipo struktura, uztikrina, kad ta pati
        linija tik skirtinga tekmes puse
        butu vienas irasas'''

    df1 = pd.DataFrame.from_dict(data = Return_flow_dict(modelis), orient = '
        index')
    df2 = pd.DataFrame.from_dict(data = Return_flow_dict(modelis), orient = '
        index')

    df2.index = flip(df2.index)
    df = df.merge(df1, how = 'left', left_index = True, right_index = True )
    df = df.merge(df2, how = 'left', left_index = True, right_index = True )
    df['sum']= df.sum(axis = 1)
    df = df['sum'].to_frame()
    return df

```

```

def Animate_solution(df):

    '''Suanimuoja linijos apkrovos duomenis'''

    adf = df.iloc[:,0:-2].T
    adf['Data'] = adf.index
    anim_df = adf.melt(id_vars = 'Data', var_name = 'Perdavimo linija',
        value_name = 'Apkrova')
    anim_df['Data'] = anim_df['Data'].astype(str)

```

```

anim_df['Perdavimo linija'] = anim_df['Perdavimo linija'].astype(str)

fig = px.bar(anim_df, x="Perdavimo linija", y="Apkrova", color="Perdavimo linija",
             animation_frame="Data", animation_group="Perdavimo linija",
             range_y=[0,750])

fig.show()

```

```

# Duomenų padavimas į modeli ir rezultatus saugojimas
prob = 0
taisymas = 1
luzusios_virsunes = []
list_of_graphs = []
list_of_broken_edges = []
list_of_load_shed = []
list_of_orig_demand = []
# Catch data from model
df = pd.DataFrame(index = edge_usage(G))
dfx = pd.DataFrame(index = edge_usage(G))

prob_dict = {}
# grab each value from data
draw_dict = {}
break_dict = {}
draw_dict = {}
for draw in range(0,20):
    for prob in np.arange(0.01,1.01,0.01):
        #initialize for each prob
        luzusios_virsunes = []
        list_of_graphs = []
        list_of_broken_edges = []
        list_of_load_shed = []
        list_of_orig_demand = []
        # Catch data from model
        df = pd.DataFrame(index = edge_usage(G))
        dfx = pd.DataFrame(index = edge_usage(G))
        for e in Demandas.index[:7]:
            print(f'Taisymo proc: {taisymas}, Draw: {draw}, Prob: {prob}, day
                  = {e}', end='\r')

            for s in Demandas.columns:
                temp_list = []
                for l_v in luzusios_virsunes:
                    if random.random() >= taisymas:
                        temp_list.append(l_v) # lieka nepataisytas
                luzusios_virsunes = temp_list.copy()

            sw=Decimal(SW[s][e])

```

```

lv=Decimal(LV[s][e])
bl=Decimal(BL[s][e])
ru=Decimal(RU[s][e])
pl=Decimal(PL[s][e])
dem=Decimal(Demandas[s][e]) # load them up and run the model

Grafas, modelis, briaunos = Initialize_Model(G, dem, lv, ru,
                                              sw, pl, bl, prob,
                                              luzusios_virsunes ) #
                                              get data from the model


luzusios_virsunes = briaunos.copy()
list_of_graphs.append(Grafas)
list_of_broken_edges.append(briaunos)


df = df.merge(append_results(dfx, modelis), how = 'left',
              left_index = True,
              right_index = True)
df.columns = [*df.columns[:-1], e+timedelta(hours=s)]

innit_demand = dem - sum( x for x in [lv,bl,ru,pl,sw] if x <0
                          )
end_demand = sum( x for x in nx.get_node_attributes(Grafas, '
                                                    demand').values() if x>
                  0 )

temp_val = innit_demand - end_demand
list_of_load_shed.append(temp_val)
prob_dict[prob] = {}
prob_dict[prob]['grafas'] = list_of_graphs.copy()
prob_dict[prob]['df'] = df.copy()
prob_dict[prob]['shed'] = list_of_load_shed.copy()
prob_dict[prob]['edges'] = list_of_broken_edges.copy()

draw_dict[draw] = prob_dict

with open('break_prob_1.obj', 'wb') as f:
    pickle.dump(break_dict, f)

```