



Kauno technologijos universitetas
Panevėžio technologijų ir verslo fakultetas

Mechatroninių įrenginių diagnostikos sistemos tyrimas

Baigiamasis magistro studijų projektas

Dovydas Briedis

Projekto autorius

Lekt. dr. Donatas Pelenis

Vadovas

Panevėžys, 2023



Kauno technologijos universitetas
Panevėžio technologijų ir verslo fakultetas

Mechatroninių įrenginių diagnostikos sistemos tyrimas

Baigiamasis magistro studijų projektas

Valdymo technologijos (6211EX014)

Dovydas Briedis

Projekto autorius

Lekt. dr. Donatas Pelenis

Vadovas

Recenzentas / Recenzentė

Panevėžys, 2023



Kauno technologijos universitetas

Panevėžio technologijų ir verslo fakultetas

Dovydas Briedis

Mechatroninių įrenginių diagnostikos sistemos tyrimas

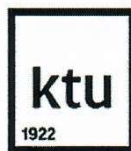
Akademinių sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektualinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Dovydas Briedis

Patvirtinta elektroniniu būdu



Kauno technologijos universitetas
Panevėžio technologijų ir verslo fakultetas

TVIRTINU
TVKC vadovė
Doc. dr. Nida Kvedaraitė

Baigiamojo magistro projekto užduotis

Diplomantui **Dovydui Briedžiui**

Baigiamojo projekto tema (lietuvių kalba)	Mechatroninių įrenginių diagnostikos sistemos tyrimas
--	---

Baigiamojo projekto tema (anglų kalba)	Research of Mechatronic Machines Diagnostic System
---	--

Patvirtinta 2023 m. balandžio 14 d. dekanų potvarkiu Nr. V25-13-11

Parengto baigiamojo projekto įkėlimo į Moodle aplinką terminas iki 2023 m. gegužės 29 d.

Duomenys, reikalavimai ir sąlygos baigiamajam projektui

Baigiamojo projekto užduotys / uždaviniai, kurie turi būti atskleisti projekte

1. Išnagrinėti mechatroninių mašinų diagnostikos metodus.
2. Išanalizuoti vaizdų ir laikinių signalų apdorojimo metodus bei priemones, padedančias klasifikuoti pramoninių valdiklių prievadus.
3. Sukurti mechatroniniuose mašinos integruotų pramoninių valdiklių diagnostikos algoritmo prototipą.
4. Ištirti vaizdų apdorojimo metodų veiksnumą, identifikuojant pramoninio valdiklio prievadų būsenas.
5. Ištirti mechatroninio įtaiso diagnostikos algoritmą kokybiniais ir kiekybiniais parametrais.

Vadovas	Lekt. dr. Donatas Pelėnis
	<i>(vadovo pareigos, vardas, pavardė, parašas)</i>
Užduotį gavau	Dovydas Briedis
	<i>(studento vardas, pavardė, parašas)</i>

2023 m. balandžio 20 d.

Briedis, Dovydas. Mechatroninių įrenginių diagnostikos sistemos tyrimas. Magistro studijų baigiamasis projektas / vadovas lekt. dr. Donatas Pelenis; Kauno technologijos universitetas, Panevėžio technologijų ir verslo fakultetas.

Studijų kryptis ir sritis: elektronikos inžinerija, technologijos mokslai.

Reikšminiai žodžiai: neuroninis tinklas, *YOLOv4*, *YOLOv4 Tiny*, LSTM, mechatroninė sistema, PLV, vaizdų atpažinimas, signalu prognozavimas, gedimų aptikimas.

Panevėžys, 2023. 66 p.

Santrauka

Šio baigiamojo magistro studijų projekto tikslas – ištirti mechatroninių įrenginių diagnostikos sistemą. Tikslui pasiekti iškelti tokie uždaviniai: išnagrinėti mechatroninių mašinų diagnostikos metodus; išanalizuoti vaizdų ir laikinų signalų apdorojimo metodus bei priemones, padedančias klasifikuoti pramoninių valdiklių prievadus; sukurti mechatroninėse mašinose integruotų pramoninių valdiklių diagnostikos algoritmo prototipą; ištirti vaizdų apdorojimo metodų veiksnumą, identifikuojant pramoninio valdiklio prievadų būsenas; ištirti mechatroninio įtaiso diagnostikos algoritmą kokybiniais ir kiekybiniais parametrais.

Projekte naudota programinė įranga – *MATLAB R2020b*.

Pirmame skyriuje aprašyti mechatroninių mašinų diagnostikos metodai, pagrįsti modeliais, signalais ir duomenimis. Pateikti skirtingų gamintojų PLV prievadai ir aprašytos problemos, susijusios su jų verčių nuskaitymu, naudojant vaizdų atpažinimą.

Antroje dalyje pristatyta tyrimo metodika ir sistema. Analizuotas objektas – „Siemens“ valdiklio valdomas batonų dėžių paletavimo įrenginys. Tirtas valdiklio prievadų nuskaitymo tikslumas ir greitaveika, naudojant *YOLOv4* ir *YOLOv4 Tiny* neuroninius tinklus; tinklų greitaveika ir tikslumas, esant skirtingoms aplinkos ir tinklų apmokymo sąlygoms. Tyrinėta vaizdo fiksavimo, aplinkos apšvietimo, neuroninio tinklo įvesties dydžio daroma įtaka aptikimo tikslumui ir greitaveikai.

Trečioje dalyje analizuotas duomenų prognozavimo LSTM neuroninis tinklas ir testuotas gedimų aptikimo algoritmas, sėkmingai nustatomi sistemos jutiklių gedimai. Nustatyta, kad apmokytas LSTM tinklas skirtas jutiklių gedimams aptikti ir nėra skirtas vykdyklių gedimų ar sistemos oro slėgio kritimo klaidoms nustatyti.

Ketvirtame skyriuje pateiktos sistemos naudojimo rekomendacijos ir pasiūlymai.

Gauti tokie atliktų tyrimų rezultatai:

1. Nustatyta, kad diagnostikos metodai yra skirstomi į analitinius ir aparattinius. Analitiniai diagnostikos metodai skirstomi į metodus, pagrįstus sistemos modeliu, sistemos signalais ir sistemos duomenimis. Aparatinės diagnostikos metodas paremtas įrangos dubliavimu, kur gedimai yra aptinkami lyginant dubliuotų įrenginių išduodamus signalus, o aptikus skirtumą traktuojamas sistemos gedimas.
2. Išanalizavus pramoninių valdiklių prievadus nustatyta, kad sistemos būsenoms atvaizduoti naudojamos šviesinės indikacijos, sumontuotos valdiklio korpuse. Nustatyta, kad turint iš anksto sudarytą sistemos įvykių žurnalą ir pasitelkiant neuroninius tinklus, galima aptikti sistemos gedimus, atkurti bei atvaizduoti sistemos vidinę logiką *Petri* tinklu.

3. Sukurtas mechatroninių mašinų integruotų pramoninių valdiklių diagnostikos algoritmo prototipas, galintis aptikti sistemos gedimus. Sukurtoji sistema tinkama tik jutiklių gedimas aptikti, vykdyklių gedimų nustatyti ji negali.
4. Eksperimentiškai nustatyta, kad kai vaizdo raiška 1920×1080 px, tolydaus apšvietimo atveju tikslumas siekia 93,1 proc., tuo tarpu netolydaus LED apšvietimo atveju – tik 70,2 proc. Tyrimų rezultatais patvirtinta, vaizdų apdorojimo sparta *YOLOv4 Tiny* neuroniniu tinklu siekia 49 kadrus per sekundę, naudojant GPU „RTX 4070“, ir 22 kadrus per sekundę, naudojant CPU „7600x“. Optimalus įvesties dydis greitaveikai užtikrinti, naudojant *YOLOv4* neuroninį tinklą siekia 352×352 px, o naudojant *YOLOv4 Tiny* – 806×806 px.
5. Tyrimo metu išsiaiškinta, kad gedimas yra aptinkamas per 300 ms nuo jutiklio signalo dingimo pradžios. Taip pat nustatyta, kad sistema veikia korektiškai, kai joje yra 10 valdiklio įėjimų ir 6 išėjimai.

Briedis, Dovydas. Research of Mechatronic Machines Diagnostic System. Master's Final Degree Project / supervisor lect. dr. Donatas Pelenis; Panevėžys Faculty of Technologies and Business, Kaunas University of Technology.

Study field and area (study field group): Electronics Engineering, Technology Sciences.

Keywords: Neural network, *YOLOv4*, *YOLOv4 Tiny*, LSTM, Mechatronic system, PLV, Image recognition, Signal prediction, Fault detection.

Panevėžys, 2023. 66 pages.

Summary

The aim of this Master's final project is to investigate the diagnostic system of mechatronic devices. To achieve this goal, the following tasks are set: to examine the diagnostic methods of mechatronic machines; to analyze the methods and tools for processing images and temporal signals that help to classify the ports of industrial controllers; to create a prototype of the diagnostic algorithm for industrial controllers integrated in mechatronic machines; to investigate the efficiency of image processing methods in identifying the states of the ports of an industrial controller; to investigate the diagnostic algorithm of the mechatronic device by qualitative and quantitative parameters.

The software used in the project is *MATLAB R2020b*.

The first chapter describes the diagnostic methods of mechatronic machines, based on models, signals, and data. Ports of PLCs from different manufacturers are presented and problems related to their state reading using image recognition are described.

The second part presents the research methodology and system. The analysed object is a palletizing device for bread boxes controlled by a Siemens controller. The accuracy and speed of reading controller ports using *YOLOv4* and *YOLOv4 Tiny* neural networks were examined; the performance and accuracy of networks under different environmental and network training conditions were evaluated. The impact of video capturing, environmental lighting, and the size of the neural network input on detection accuracy and performance were examined.

The third part analyses the LSTM neural network for data forecasting and tests the fault detection algorithm, successfully detecting system sensor faults. It was determined that the trained LSTM network is intended for detecting sensor faults and is not designed to detect actuator faults or system air pressure drop errors.

The fourth chapter provides recommendations and suggestions for the use of the system.

Conclusions and research results:

1. It has been determined that diagnostic methods are divided into analytical and hardware-based. Analytical diagnostic methods are divided into methods based on the system model, system signals, and system data. The hardware diagnostic method is based on equipment duplication, where faults are detected by comparing the signals issued by duplicated devices, and any detected difference is considered a system failure.
2. By analysing the ports of industrial controllers, it was found that light indications mounted in the controller are used to display system states. It was determined that having a pre-compiled system

event log and using neural networks, system failures can be detected, and the internal logic of the system can be restored and depicted with a *Petri* net.

3. A prototype of the diagnostic algorithm for industrial controllers integrated into mechatronic machines, capable of detecting system failures, was developed. The created system is suitable only for detecting sensor faults, it cannot detect actuator faults.
4. It was experimentally determined that when the video resolution is 1920×1080 px, the accuracy reaches 93.1% in the case of diffused lighting, while in the case of uneven LED lighting – only 70.2%. The research results confirmed that the speed of image processing with the *YOLOv4 Tiny* neural network reaches 49 frames per second using the GPU "RTX 4070", and 22 frames per second using the CPU "7600x". The optimal input size for performance using the *YOLOv4* neural network is 352×352 px, while using *YOLOv4 Tiny* – 806×806 px.
5. During the research, it was found that the fault is detected within 300 ms from the start of the sensor signal disappearance. It was also determined that the system works correctly when there are 10 controller inputs and 6 outputs in it.

Turinys

Lentelių sąrašas	10
Paveikslų sąrašas	11
Santrumpų sąrašas	13
Įvadas.....	14
1. Mokslinės literatūros analizė.....	15
1.1. Mechatroninių mašinų diagnostikos metodai.....	15
1.1.1. Įrangos dubliavimas.....	15
1.1.2. Modeliu pagrįsti metodai.....	16
1.1.3. Signalais pagrįsti metodai.....	17
1.1.4. Duomenimis pagrįsti metodai.....	18
1.2. Pramoninių valdiklių prievadų klasifikacija, pasitelkiant vaizdų ir laikinų signalų apdorojimo metodus.....	20
1.2.1. Vaizdų apdorojimo metodai	23
1.2.2. LED aptikimas naudojant kompiuterinę regą.....	24
1.2.3. LED aptikimas naudojant mašininį mokymąsi.....	26
1.2.4. Laikinių signalų apdorojimo metodai.....	29
2. PLV prievadų identifikavimo tyrimas, pasitelkiant neuroninius tinklus.....	33
2.1. Tyrimo metodika	33
2.2. Tiriamasis objektas.....	34
2.3. Vaizdų paruošimas tinklui apmokyti.....	35
2.4. Neuroninio tinklo apmokymo parametrų parinkimas.....	37
2.5. Vaizdo medžiagos rezoliucijos įtaka vaizdo atpažinimo tikslumui	39
2.6. Neuroninio tinklo įvesties dydžio įtaka tikslumui ir greitaveikai	41
2.6.1. Vaizdo plokštės įtaka neuroninio tinklo greitaveikai	42
3. Loginių sekų prognozavimas.....	44
3.1. LSTM tinklo apmokymas.....	46
3.2. LSTM tinklo testavimas	47
3.2.1. Jutiklio gedimo aptikimas.....	48
3.2.2. Kitų sistemos gedimų aptikimas.....	54
3.3. Mechatroninio įrenginio sistemos stebėjimo ir automatinio gedimo aptikimo žodinis algoritmas	57
4. Sistemos naudojimo rekomendacijos ir pasiūlymai	61
Išvados	62
Literatūros sąrašas	63

Lentelių sąrašas

1 lentelė. Klaidų aptikimo metodų palyginimas [31].....	20
2 lentelė. Eksperimento sąlygos [12]	28
3 lentelė. Eksperimento rezultatai lyginant <i>D2Net</i> su kitais moderniais algoritmais [12].....	28
4 lentelė. Klaidų palyginimas [11]	29
5 lentelė. Apmokymo parametrai	39
6 lentelė. Rezultatai, gauti naudojant 3840×2160 vaizdo rezoliuciją ir atviros spintos apšvietimą.....	40
7 lentelė. <i>YOLOv4</i> tinklo aptikimo tikslumo palyginimas, keičiant rezoliuciją ir apšvietimą.....	40

Paveikslų sąrašas

1 pav. Diagnostikos metodai [2]	15
2 pav. Daugumos balsavimo diagnostikos metodas [3]	16
3 pav. Neuroninis tinklas [9]	19
4 pav. „SIMATIC“ S7-1200 programuojamas loginis valdiklis (1 – sistemos būsenos indikacija, 2 – įėjimai, 3 – išėjimai, 4 – gaunamas signalas, 5 – negaunamas signalas) [sudaryta autoriaus]	21
5 pav. „OMRON SYSMAC“ CP1J programuojamas loginis valdiklis (1 – sistemos būsenos indikacija, 2 – įėjimai, 3 – išėjimai, 4 – gaunamas signalas, 5 – negaunamas signalas) [sudaryta autoriaus]	22
6 pav. „OMRON SYSMAC“ CJ1M programuojamas loginis valdiklis (1 – sistemos būsenos indikacija, 2 – įėjimai 3 – išėjimai) [sudaryta autoriaus]	22
7 pav. „Delta“ DVP20SX2 programuojamas loginis valdiklis (1 – sistemos būsenos indikacija, 2 – įėjimai, 3 – išėjimai) [sudaryta autoriaus]	23
8 pav. LED indikacijos aptikimo algoritmas 1-as [9]	24
9 pav. LED indikacijos aptikimo algoritmas 2 [9]	25
10 pav. a) keturios žymos (juodi apskritimai) vaizde; b) žymos po fono filtravimo ir priekinio plano kaukės pritaikymo; c) žymos po formos ir dydžio filtravimo; d) žymos po formos ir dydžio filtravimo [9]	25
11 pav. Algoritmų greitaveikos grafinis palyginimas [9]	26
12 pav. Automobilis su pritvirtinta LED matrica [12]	27
13 pav. Pasiūlyti LED matricos suskirstymo metodai: greitas kampų aptikimo algoritmas (angl. <i>fast</i>), matricos suskirstymas (angl. <i>split</i>), konvoliuciniai neuroniniai tinklai (angl. <i>convolutional neural networks</i> CNN) [12]	27
14 pav. Pasiūlytas LED matricos atpažinimo metodas A [12]	27
15 pav. Eksperimento sąlygos dieną (a ir b) ir naktį (c ir d) [12]	29
16 pav. Laikinių signalų apdorojimo metodo diagrama [13]	31
17 pav. Pirmoji sistemos valdymo programa C1 [12]	31
18 pav. Antroji sistemos valdymo programa C2 [13]	31
19 pav. Iš pirmosios programos išgautas <i>Petri</i> tinklas (žalia spalva parodo programos pradžią, raudona – pabaigą) [13]	32
20 pav. Iš antrosios programos išgautas <i>Petri</i> tinklas (žalia spalva parodo programos pradžią, raudona – pabaigą) [13]	32
21 pav. Tyrimo metu naudotas laikiklis [sudaryta autoriaus]	33
22 pav. PLV apšvietimas: a) natūralus patalpų apšvietimas; b) uždaro durys su papildomu LED apšvietimu; c) uždaro durys be papildomo apšvietimo [sudaryta autoriaus]	34
23 pav. Tirtas batono dėžių krautuvai [sudaryta autoriaus]	35
24 pav. Vaizdų apkarpymas: a) pradinis vaizdas; b) iškirptas PLV pagrindinis modulis; c) iškirptas PLV praplėtimų modulis [sudaryta autoriaus]	36
25 pav. Įrankiu „Image Labeler“ sužymėta nuotrauka [sudaryta autoriaus]	36
26 pav. Apmokymo greičio įtaka apmokyto tinklo objektų aptikimo tikslumui [36]	37
27 pav. Apmokymo metu naudotos vaizdų augmentacijos pavyzdys [sudaryta autoriaus]	38
28 pav. Aptikimo pavyzdys: 1 – teisingai aptikta indikacija; 2 – neteisingai aptikta indikacija, 3 – neaptikta indikacija [sudaryta autoriaus]	39
29 pav. Įvesties dydžio įtaka aptikimo tikslumui, naudojant <i>YOLOv4</i> tinklą [sudaryta autoriaus] ..	41

30 pav. Įvesties dydžio įtaka aptikimo tikslumui, naudojant <i>YOLOv4 Tiny</i> tinklą [sudaryta autoriaus]	41
31 pav. Įvesties dydžio įtaka aptikimo greitaveikai [sudaryta autoriaus]	42
32 pav. Kompiuterinės įrangos įtaka aptikimo greitaveikai naudojant <i>YOLOv4</i> neuroninį tinklą [sudaryta autoriaus]	43
33 pav. Kompiuterinės įrangos įtaka aptikimo greitaveikai naudojant <i>YOLOv4 Tiny</i> neuroninį tinklą [sudaryta autoriaus]	43
34 pav. Neuroninio tinklo, su ilgalaikę atmintimi, ląstelės modelis [39]	45
35 pav. Duomenys, naudoti LSTM tinklui apmokyti [sudaryta autoriaus]	46
36 pav. Apmokyto LSTM tinklo sugeneruoti išėjimai [sudaryta autoriaus]	47
37 pav. Sistemos jutiklio gedimas I0.1. „1“ pažymėtas jutiklio gedimo momentas (žalia spalva vaizduojama reali vertė, o raudona – prognozuojama) [sudaryta autoriaus]	48
38 pav. Skirtumas tarp prognozuojamos ir esamos verčių [sudaryta autoriaus]	49
39 pav. Vidutinis skirtumas tarp prognozuojamos ir esamos verčių [sudaryta autoriaus]	49
40 pav. Skirtumas tarp LSTM prognozuojamų ir realių verčių [sudaryta autoriaus]	49
41 pav. Skirtumas tarp LSTM prognozuojamų ir realių verčių (absoliutinės vertės) [sudaryta autoriaus]	50
42 pav. Vidutinis skirtumas tarp LSTM prognozuojamų ir realių verčių [sudaryta autoriaus]	50
43 pav. Vidutinis skirtumas tarp prognozuojamos ir esamos verčių (absoliučios vertės) [sudaryta autoriaus]	50
44 pav. Teisingai aptiktas jutiklio, kurio adresas I0.1, gedimas [sudaryta autoriaus]	51
45 pav. Jutiklio I0.4 gedimo aptikimo grafinis vaizdavimas (žalia spalva vaizduojama reali vertė, raudona spalva – prognozuojama) [sudaryta autoriaus]	51
46 pav. Skirtumas tarp prognozuojamos ir esamos verčių [sudaryta autoriaus]	52
47 pav. Vidutinis skirtumas tarp prognozuojamos ir esamos verčių [sudaryta autoriaus]	52
48 pav. Vidutinis skirtumas tarp prognozuojamos ir esamos verčių (absoliučios vertės) [sudaryta autoriaus]	52
49 pav. Skirtumas tarp LSTM prognozuojamų ir realių verčių [sudaryta autoriaus]	53
50 pav. Skirtumas tarp LSTM prognozuojamų ir realių verčių (absoliutinės vertės) [sudaryta autoriaus]	53
51 pav. Vidutinis skirtumas tarp LSTM prognozuojamų ir realių verčių [sudaryta autoriaus]	53
52 pav. Jutiklio, prijungto prie valdiklio įėjimo adreso I0.4, gedimo aptikimas [sudaryta autoriaus]	54
53 pav. Jutiklių loginių verčių kitimas, esant slėgio kritimui ir LSTM prognozuojamai vertei (žalia spalva vaizduojama reali vertė, raudona spalva – prognozuojama) [sudaryta autoriaus]	55
54 pav. Vidutinis skirtumas tarp prognozuojamos ir esamos verčių (absoliučios vertės) [sudaryta autoriaus]	56
55 pav. Sistemos gedimo aptikimas slėgio kritimo metu [sudaryta autoriaus]	57
56 pav. Užfiksuotas vaizdas [sudaryta autoriaus]	58
57 pav. Aptiktas QR kodas [sudaryta autoriaus]	58
58 pav. Iškirptas vaizdas [sudaryta autoriaus]	58
59 pav. Aptiktos valdiklio indikacijų loginės būsenos [sudaryta autoriaus]	59
60 pav. Duomenų išrikiavimo programos kodo iškarpa [sudaryta autoriaus]	59
61 pav. Valdiklio įėjimų / išėjimų būsenų atskyrimas [sudaryta autoriaus]	60
62 pav. Valdiklio loginės būsenos prieš išrikiavimą ir po [sudaryta autoriaus]	60

Santrumpų sąrašas

Santrumpos:

MATLAB – daugiaplatformė *MathWorks* programinė įranga, skirta įvairių mokslo šakų problemoms spręsti, ypač matematinėms;

ANN – dirbtinis neuroninis tinklas (angl. *artificial neural network*);

CNN – konvoliucinis neuroninis tinklas (angl. *convolution neural network*);

ML – mašininis mokymasis (angl. *machine learning*);

YOLOv4 – objektų aptikimui skirtas neuroninis tinklas (angl. *you only look once*);

YOLOv4 Tiny – supaprastinta *YOLOv4* versija;

CPU – centrinis procesoriaus blokas (angl. *central processing unit*);

GPU – grafinis procesoriaus blokas (angl. *graphics processing unit*);

PLV – programuojamas loginis valdiklis yra prietaisas, kuris priima jutiklių ir kitų įrenginių išduodamus signalus ir, remdamasis iš anksto parašyta programa, išduoda atitinkamus išėjimo signalus (angl. *programmable logic controller*, PLC);

LSTM – neuroninis tinklas, naudojantis ilgalaikės atminties ląstelę (angl. *long short-term memory*).

Išvadas

Gamybos procesų automatizacijos poreikis pramonėje auga dėl gaminamos produkcijos kokybės didinimo, darbo sąlygų gerinimo, gamybinių procesų efektyvinimo. Naujai įdiegti ar / ir modernizuojami technologiniai sprendimai užtikrina konkurencingą veiklą, aplinkai mažiau kenksmingą gamybą. Didėjant įvairios technologinės įrangos kiekiui, didėja ir jos gedimų tikimybė. Įrenginių gamintojai, saugodami intelektinę nuosavybę, apriboja prieigą prie programinės įrangos, valdančios įrengimą ar net technologinį procesą. Todėl įmonės techninio aptarnavimo personalui nustatyti pramoninių mašinų gedimus tampa sudėtinga ir brangia procedūra. Remiantis naujais technologiniais dirbtinių neuroninių tinklų pasiekimais, galima identifikuoti mechatroninių mašinų veikimo algoritmo diagramas, į pagalbą pasitelkiant pasikartojančių signalų sekų nustatymo bei mašininio mokymosi metodus. Automatinė gedimų nustatymo įranga, dėl savo inžineriškai paprastai taikytino būdo, gali sumažinti dėl gedimų sukeltas prietaisų prastovas.

Darbo aktualumas. Mechatroninių sistemų valdiklių programos yra kaip juodoji dėžė: jų vidinė veikimo logika ne visą laiką yra prieinama, o ir turint prieigą prie programos sunku perprasti veikimo algoritmą. Dėl šios priežasties sunku diagnozuoti sistemose įvykusius gedimus.

Tyrimų objektas – mechatroninių įrenginių diagnostikos sistema.

Projekto tikslas – ištirti mechatroninių įrenginių diagnostikos sistemą.

Darbo uždaviniai

1. Išnagrinėti mechatroninių mašinų diagnostikos metodus.
2. Išanalizuoti vaizdų ir laikinų signalų apdorojimo metodus bei priemones, padedančias klasifikuoti pramoninių valdiklių prievadus.
3. Sukurti mechatroninėse mašinose integruotų pramoninių valdiklių diagnostikos algoritmo prototipą.
4. Ištirti vaizdų apdorojimo metodų veiksnumą, identifikuojant pramoninio valdiklio prievadų būsenas.
5. Ištirti mechatroninio įtaiso diagnostikos algoritmą kokybiniais ir kiekybiniais parametrais.

Tyrimo metodai: mokslinės literatūros analizė, sistemų modeliavimas, eksperimentai.

Autoriaus publikuotų straipsnių bibliografinis sąrašas:

1. BRIEDIS, Dovydas; PELENIS, Donatas. Mechatroninių įrenginių diagnostikos algoritmo taikomo pramoniniuose valdikliuose tyrimas // Technologijų ir verslo aktualijos – 2022: studentų mokslinių darbų konferencijos pranešimų medžiaga, Lietuva, Panevėžys, 2022 m. lapkričio 25 d. / Kauno technologijos universiteto Panevėžio technologijų ir verslo fakultetas. Kaunas: Kauno technologijos universitetas. ISSN 2538-8045.

Autoriaus konferencijose skaityti pranešimai:

1. Pranešimas tema „Mechatroninių įrenginių diagnostikos algoritmo taikomo pramoniniuose valdikliuose kūrimas“ 22-oje studentų mokslinių darbų konferencijoje „Technologijų ir verslo aktualijos – 2022“. Panevėžys: Kauno technologijos universitetas, 2022 lapkričio 25 d.

1. Mokslinės literatūros analizė

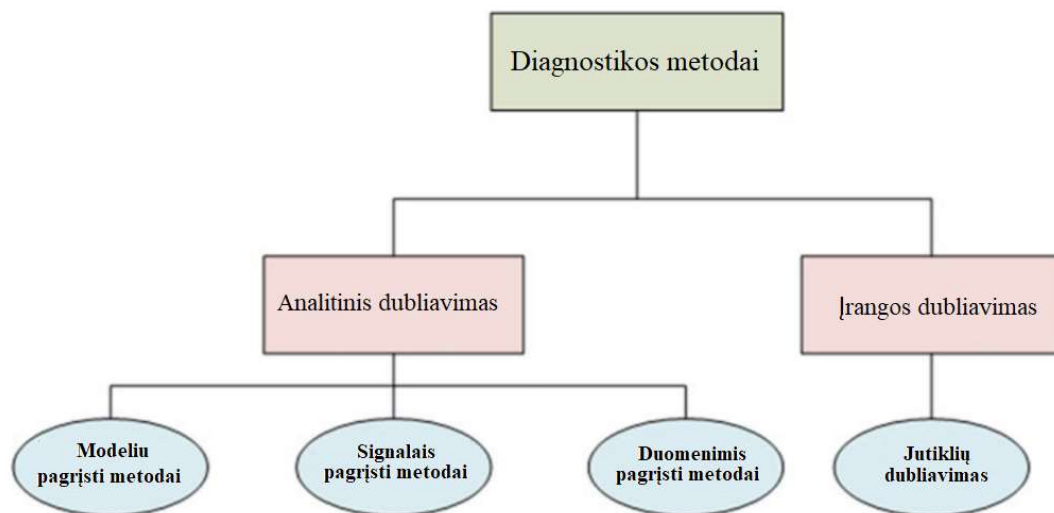
1.1. Mechatroninių mašinų diagnostikos metodai

Mechatronika – sinergetinė mechaninės inžinerijos, elektronikos, kompiuterių ir valdymo sistemų kombinacija [1].

Mechatroninės sistemos padidina gamybos efektyvumą ir našumą, todėl plačiai naudojamos daugelyje pramonės sričių. Didėjant sistemų komplikuotumui kartu didėja ir gedimų, sutrikdančių sistemos veikimą, tikimybė. Sistemose pasitaikančius gedimus galima išskirti į du tipus: veikimo gedimus (pvz., elektrinius, mechaninius, šiluminius, konstrukcijos defektus ir kt.) ir jutiklių gedimus (pvz., poslinkio, stiprinimo ar dreifo defektus). Toliau pateikiami ir analizuojami dažniausiai naudojami diagnostikos metodai [2].

Pramoninių mašinų gedimų nustatymų metodai klasifikuojami į tris sekas: gedimų aptikimas, gedimų izoliavimas ir gedimų nustatymas. Gedimų aptikimo seka – pirmoji ir pagrindinė gedimų diagnostikos užduotis, naudojama patikrinti, ar sistemoje nėra gedimo. Gedimų išskyrimo seka skirta nustatyti sugedusį įrengimo komponentą arba komponentų mazgą. Gedimų nustatymo seka – paskutinis diagnostikos žingsnis, kuriame nustatomas gedimo tipas ir priežastys [3]. Sprendžiant šias tris užduotis sukurti skirtingi stacionarių ir nestacionarių signalų diagnostikos metodai.

1 paveiksle pateikti dažniausiai taikomų mechatroninių sistemų gedimo nustatymo metodai.



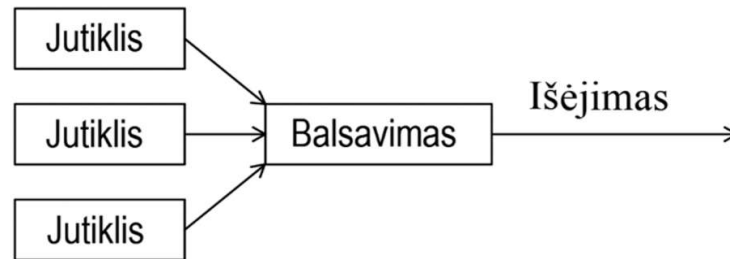
1 pav. Diagnostikos metodai [2]

Nors 1 paveiksle vaizdų apdorojimas nėra pateiktas kaip gedimo nustatymo metodas, bet jį galima pasitelkti kaip komunikacinės periferijos pakaitalą, nuskaitant mechatroninių mašinų jutiklių ir vykdyklių signalus. Nuskaityti signalai integruojami veikti per modelių pagrįstu diagnostikos metodu, dėl to galima sukurti gan efektyvią, ekonomiškai patrauklią mašinų gedimų diagnostikos sistemą.

1.1.1. Įrangos dubliavimas

Pagrindinė aparatinės įrangos rezervavimo idėja: naudoti kelis įrenginius tiems patiems signalams išmatuoti. Aparatinės įrangos dubliavimas, naudojant algebrinį ryšį tarp skirtingų sistemos kintamųjų,

yra susijęs su daugiau nei vieno jutiklio generuojamų pasikartojančių signalų palyginimu. Įrangos dubliavimo metu sistemos gedimas aptinkamas daugumos balsavimu (žr. 2 pav.). Daugumos balsavimo metu išėjimo vertės reikšmė būna nustatoma pagal daugumos vertę, o nesutapus bent vieno jutiklio vertei toje sistemos vietoje nustatomas gedimas. Šio metodo pranašumas yra jo patikimumas (gedimas nebūtų aptiktas tik tuo atveju, jeigu visi sudubliuoti įrenginiai sugestų vienu metu) ir paprastumas. Šio metodo trūkumas – dubliuojamos įrangos ir jos priežiūros kaina, kuri praktiškai padidėja tris kartus. Šio metodo naudojimą apriboja įrangos papildomai užimama vieta bei svoris. Todėl dažniausiai sistemose taikomas tik pagrindinių komponentų dubliavimas [3].



2 pav. Daugumos balsavimo diagnostikos metodas [3]

Įrangos dubliavimas taip pat yra pritaikomas ir mikroschemose.

Taikymo sritis: aviacijos ir kosmoso sistemos, atominės elektrinės, elektros perdavimo ir skirstymo įrenginiai, bankų ir finansų sistemos, pramonės valdymas ir automatizavimas bei kitos jautrios pramonės sritys paprastai naudoja įrangos dubliavimą. Tai sistemoms suteikia tam tikrą atsparumo laipsnį gedimams.

1.1.2. Modeliu pagrįsti metodai

Modeliu pagrįstiems metodams aprašyti nenaudojama papildoma aparatūrinė įranga, o tik matematinės formulės. Modeliais pagrįsti metodai palygina turimus faktinius sistemų matavimus su vertėmis, apskaičiuotomis sudarytame modelyje. Galima išskirti du modeliavimo metodus:

- deterministinis diagnostikos metodas;
- stochastinis diagnostikos metodas.

Deterministiniai diagnostikos metodai yra pagrįsti deterministiniu nagrinėjamos sistemos modeliavimu. Jie yra skirstomi į dvi sritis: pariteto erdvę ir diagnostiką, pagrįstą stebėtojais.

Pariteto erdvės principas: apskaičiuoti likutinę vertę, kuri yra skirtumas tarp apskaičiuotos ir faktiškai išmatuotos sistemos išėjimo vertės. Tikrinamas likutinės vertės nuoseklumas: be gedimų veikiančioje sistemoje apskaičiuotas likutis lygus nuliui. Tačiau jis nėra lygus nuliui kai sistemoje atsiranda modeliavimo klaidų, gedimų ar triukšmo. Pariteto erdvės metodas gali būti naudojamas laiko srities būsenos-erdvėje modelyje arba dažnio srities modelyje. Šis diagnostikos metodas taikomas tokiuose modeliuose, kaip nepilotuojamo orlaivio gedimams įvertinti [4].

Diagnostika pagrįsta stebėtojais. Stebėtojai, kurie pagrįsti sistemos būsenos-erdvėje vaizdu, atlieka pagrindinį vaidmenį deterministiniuose diagnostikos metoduose. Jie naudojami gedimams aptikti, atskirti ir nustatyti. Naudojant stebėtojai pagrįstus diagnostikos metodus svarbu tinkamai pasirinkti stebėtojo stiprinimą, kuris sukuria liekamuosius signalus, jautrius gedimams, bet nejautrius trikdžiams (modeliavimo klaidoms, triukšmui ir procesų trikdžiams). Įprastas metodas, naudojamas

nustatant stiprinimo stebėtojus, yra savosios struktūros priskyrimo metodas. Įvedami kairiojo ir dešiniojo savosios struktūros metodai, siekiant tiesiogiai panaikinti trikdžius. Pagrindinė diagnostika pagrįsta stebėtojais. Idėja yra ta, kad modeliuojamoje sistemoje nagrinėjami gedimai realizuojami kaip papildomos sistemos būsenos, taip įvertinant visuotinius išplėstinės būsenos vektorius (įskaitant gedimą ir pradinį signalus). Įvairių stebėtojų tipų derinys tapo svarbiu tyrinėjimo objektu ir yra naudojamas siekiant aptikti ir nustatyti gedimus sistemose [5].

Praktinis deterministiniu modeliu pagrįstas metodas nagrinėtas Zbigniewo veleno įtrūkimų diagnostikoje stebint veleno vibracijos nukrypimą nuo modelių apskaičiuotų verčių. Tyrimų metu nustatyta, kad taikant baigtinių elementų modeliavimą galima aptikti didesnius nei 2 proc. veleno įtrūkumus [18].

Stochastinės diagnostikos metodas – tai modeliavimo metodas, kur sistemos modeliavimas iš esmės yra atsitiktinis, o neapibrėžti veiksniai integruoti į modelį.

Dažniausiai naudojamas stochastinės diagnostikos metodas – likutinės vertės generavimas, naudojant Kalmano filtrą. Gedimai diagnozuojami atlikus statistinę likutinių verčių analizę. Atliekamas baltojo triukšmo, kovariacijos ir aritmetinio vidurkio tyrimas. Sukurta keletas statistinių priemonių, skirtų gedimų buvimui patikrinti, pavyzdžiui, didžiausios tikimybės metodas, apibendrintas tikimybės koeficientas ir kt. [19].

Kalmano filtras tyrinėtas Gilberto nuolatinių magnetų sinchroninių variklių pavarose, siekiant aptikti ir izoliuoti jutiklių gedimus [19]. Tyrimo metu įrodyta, kad taikant Kalmano filtrą galima aptikti variklio fazės srovės, dc-sąsajos (angl. *dc-link*) ir padėties jutiklių gedimus. Kalmano filtras taip pat nagrinėtas UAV (angl. *unmanned aerial vehicle*) dinamikai įvertinti esant jutiklio ar pavaros gedimams [21]. Naudojant Kalmano filtrą nustatomas gedimo tipas (jutiklis arba pava), tada atitinkamas jutiklis ar pava izoliuojami.

Parametrų įvertinimas yra vienas iš stochastinės diagnostikos metodų, kuriuo atsižvelgiama į dėl gedimo pasikeitusius sistemos parametrus (pavyzdžiui, tiesinių sistemų atveju, erdvės matrica erdvės būsenos atvaizdavime). Pagrindinė normalaus veikimo sistemos būklės struktūra yra žinoma ir lyginama su nuspėtųjų parametrų metodais, taip gaunant informaciją apie gedimą [5].

1.1.3. Signalais pagrįsti metodai

Signalais pagrįsti metodai gedimų diagnostikai naudoja išmatuotus sistemos signalus, kurie toliau yra lyginami su senesniais tos pačios sistemos signalų matavimais. Jie skirstomi į tris sritis: dažnio, laiko srities ir laiko dažnio metodai [22].

a) Laiko srities metodai.

Laiko srities metodai – tai pagrindiniai diagnostikos metodai, kurie yra pagrįsti išmatuotų signalų laiko srities charakteristikų išskyrimu. Labiausiai paplitę laiko srities metodai yra šie: vidurkio kvadratas, viršūnės reikšmės, absoliutinės reikšmės, kutozė ir kt. [22].

Stebint nuolatinį dinaminį procesą išgaunamos laiko srities ypatybės gedimų diagnostikai. Pavyzdžiui, [22] analizuojant išmatuotus vidutinių kvadratinių srovės charakteristikų pokyčius esant visiškai veikiančiai sistemai ir esant vieno ar dviejų tranzistorių trumpajam jungimui arba atvirai grandinei, buvo sukurtas galios keitiklių gedimų diagnostikos metodas, pritaikytas perjungto nenoro

varikliui (angl. *switched reluctance motor*). Pagal Chen [23], panaudojus išmatuotus variklio fazių sroves ir atitinkamus atskaitos signalus, buvo sukurtas realaus laiko algoritmas, leidžiantis aptikti maitinimo jungiklių atvirų grandinių gedimus keitikliuose, maitinančiuose kintamosios srovės variklius. Laiko sritimi pagrįstas klaidos aptikimo metodas naudotas DC-DC keitikliuose trumpam jungimui aptikti [15]. Nustatyta, kad taikant signalais pagrįstą metodą trumpas jungimas aptinkamas per 20 μ s, kai veikimo ciklas yra 40 proc., per 133 μ s, kai veikimo ciklas yra 20 proc., ir per 113 μ s, kai veikimo ciklas yra 80 proc. [16]

b) Dažnio srities metodai.

Dažnio srities metodai naudojami anomalijoms aptikti naudojant spektro analizės įrankius. Labiausiai paplitęs diagnostikos metodas – greitoji *Furjė* transformacija (angl. *fast Furje transformation*, GFT). Jos tikslas yra iš sistemos išmatuotos vibracijos išgauti gedimo indikaciją ir išanalizuoti dažnines charakteristikas ir defekto amplitudę naudojant spektro analizės metodus. Nors GFT yra labai svarbi stacionarių dažnio signalų analizės priemonė, tačiau ji negali iliustruoti nestacionarių signalų dažnio spektro pokyčių. Todėl GFT negali atskleisti informacijos apie gedimą nestacionariuose signaluose. Tam tikrais atvejais spektras yra neaiškus ir jame gali būti nepakankamas informacijos kiekis. Tokiais atvejais naudojamas kitas metodas, vadinamas cepstraline analize [22].

Dažnio srities metodas praktiškai pritaikytas indukcinį variklių statoriaus srovių spektro analizei. Naudojant šį metodą buvo sudaromas variklių tvarkingo veikimo statorių srovės parašas. Gedimo metu statoriaus srovės harmonikos pasikeičia ir taip indikuoja sistemos gedimą [25].

Mechaninės įrangos, tokios, kaip pavarų dėžės, būklei stebėti ir diagnostikai atlikti naudojamas vibracijos signalų analizės metodas. Mechaninių prietaisų darbo metu skleidžiami garsai nusako mašinos darbo būklę. Pan [26] aprašė kaip buvo naudojamas akustinio gedimo aptikimo metodas pavarų dėžės gedimams aptikti. Bandymu metu naudotas elektros variklis, kuris diržu buvo prijungtas prie pavarų dėžės, sudarytos iš dviejų krumpliaračių. Gedimui simuliuoti vienam krumpliaračiui panaikintas vienas dantis. Sistemos veikimo garsas buvo įrašytas trimis kryptiniais mikrofonais. Gautas garso įrašas yra perkeltas iš laiko srities į dažnio sritį naudojant *Furjė* transformaciją. [26]. Tyrimo metu įrodyta, kad naudojant mikrofonus galima užfiksuoti ir atpažinti, kaip pasikeičia garso charakteristikos sistemoje įvykus gedimui.

c) Laiko dažnio metodai.

Laiko dažnio metodai naudojami analizuoti nestacionarius signalus. Šiais metodais siekiama nustatyti signalų dažnines charakteristikas ir išgauti jų laiko variacijos ypatybes, kurios yra efektyvus diagnostikos įrankis. Labiausiai paplitę laiko dažnio metodai yra bangos ilgio transformacija, trumpojo laiko *Furjė* transformacija, *Hilberto* transformacija [22].

Trumpalaikis *Furjė* transformacijos (angl. *short-time Fourier transform*) metodas leidžia nustatyti vietinių sekcijų signalo dažnio turinį per tam tikrą laiką keičiantis signalui. Šis metodas plačiai taikomas induktorių, variklių statorių ir rotorų gedimams aptikti [27].

1.1.4. Duomenimis pagrįsti metodai

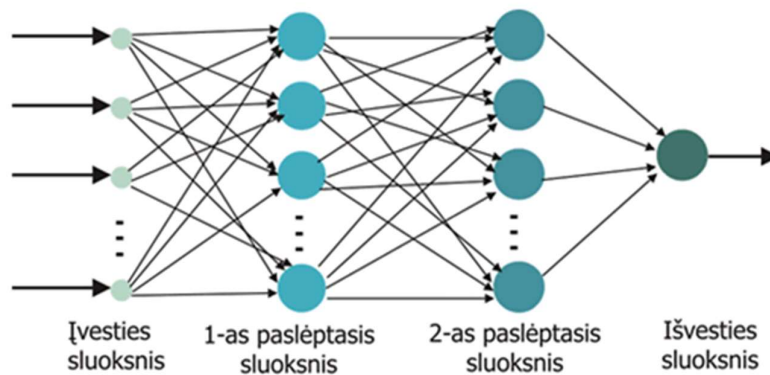
Duomenimis pagrįsti metodai nenaudoja matematinio modeliavimo. Jie pagrįsti likutine verte (skirtumas tarp išmatuotos ir prognozuotos vertės), gauta iš išmatuotų sistemos įėjimų ir išėjimų.

Labiausiai paplitę duomenimis pagrįsti diagnostikos metodai yra neraiškioji logika, neuroniniai tinklai ir klasterių analizė [8].

Neraiškioji logika (angl. *fuzzy logic*) – matematinis metodas, sprendžiantis įvairias modeliuojamas problemas, pagrįstas tik sistemos įėjimais ir išėjimais. Neraiškioji logika pateikia rezultatus rekomendacijų forma konkrečiam būsenų intervalui, naudodama žinių bazę apimančius duomenis. Tai leidžia apibrėžti tarpines reikšmes tarp įprastų vertinimų, pvz., „taip ar ne“, „tiesa ar klaidinga“ ir kt. Tokios sąvokos kaip „gana šilta“ arba „gana šalta“ gali būti suformuluotos matematiškai ir apdorotos kompiuteriais [7].

Fuzzy logika pritaikyta [28] nagrinėjant impulso pločio moduliavimo įtampos šaltinio keitiklius (angl. *pulsewidth modulation voltage-source inverter*, PWM-VSI). Nagrinėtas metodas gali aptikti ir rasti įvairių tipų atviros grandinės gedimus, įskaitant pavienius, daugybinius ir tarpinius gedimus. Sugedusius maitinimo jungiklius nustato praėjus mažiau nei 0,09 s po gedimo, o tai apytiksliai prilygsta dviem variklio fazės srovės pagrindiniams periodams. Kai variklis veikia normaliomis sąlygomis, eksperimentiniai rezultatai rodo, kad staigus apkrovos pokytis nesukelia klaidingos diagnozės, o tai garantuoja patikimą variklio pavaros veikimą [29].

Neuroniniai tinklai: biologinių neuronų įkvėpti tinklai, tinklai pritaikyti netiesinių sistemų diagnostikai. Neuroniniai tinklai sukuria sistemos įėjimų ir išėjimų susiejimų žemėlapi (žr. 3 pav.).



3 pav. Neuroninis tinklas [9]

Neuroniniai tinklai pasižymi įvairia architektūra. Plačiausiai naudojamas tiekimo į priekį tinklas, apmokytas atgalinio sklidimo būdu. Atgalinis sklidimas – sisteminis daugiasluoksnio dirbtinio neuroninio tinklo mokymo metodas. Atgalinio sklidimo algoritmas buvo pritaikytas daugeliui modelio atpažinimo problemų. Šios klasės neuroninio tinklo architektūra turi bendrą bruožą - visi sluoksnio neuronai yra sujungti su visais gretimų sluoksnių neurais per vienkryptes šakas. Tai reiškia, kad šakos gali transliuoti informaciją tik viena kryptimi, tai yra „kryptimi pirmyn“. Kiekviena šaka turi savus svorius, kurie koreguojami apmokymo metu. Apmokant tinklą atgalinio sklidimo algoritmu, gaunamas netiesinis įvesties ir išvesties kintamųjų atvaizdavimas. Taigi, atsižvelgiant į įvesties ir išvesties poras, tinklo svoriai koreguojami atgalinio sklidimo algoritmu, taip užfiksuojamas netiesinis ryšys [8].

Neuroninis tinklas sudarymas dviem veiksmis: mokymu ir testavimu. Mokymo metu neuroninis tinklas treniruojamas užfiksuoti ryšį tarp pasirinktų įėjimų ir išėjimų. Pasibaigus mokymui neuroniniai tinklai testuojami naudojant testavimo duomenų rinkinį, kuris nebuvo naudojamas

apmokymui. Baigus tinklo apmokymą ir testavimą, jie yra parengti gedimų aptikimui, skirtingomis darbo sąlygomis. Kuriant gedimų aptikimo modelį sprendžiami šie klausimai: įvesties ir išvesties kintamųjų pasirinkimas, duomenų generavimas, duomenų normalizavimas, tinklo struktūros pasirinkimas ir tinklo mokymas [8].

Darbe [29] neuroninis tinklas naudojamas ankstyviems statoriaus apvijos izoliacijos gedimams aptikti. Laiko srities statoriaus srovės signalo duomenys, gauti iš sveikų ir sugedusių indukcinio elektros variklių, buvo naudoti prižiūrimam neuroninio tinklo apmokymui. Eksperimento metu paaiškėjo, kad apmokius tinklą jis 94,73 proc. tikslumu aptinka pažeidimus. Kiranyaz aprašė kaip [30] 1-D konvoliuciniai neuroniniai tinklai (angl. *1-D convolutional neural networks*) naudoti modulinio daugiapakopio keitiklio jungiklio gedimas aptikti. Tyrimo metu pasiektas iki 0,1 s klaidos aptikimo laikas, iš viso aptikta 94,73 proc. klaidų ir tik 2,33 proc. klaidų buvo klaidingai teigiamos (angl. *false positive*). Heo aprašė [31] kaip neuroninis tinklas buvo lyginamas su kitais klaidų aptikimo metodais. Sistemoms palyginti buvo naudojamas *tennessee eastman process* [31]. Neuroninio tinklo klaidų aptikimo tikslumas buvo lyginamas su [33] DPCA (angl. *dynamic principal component analysis*), MPLS (angl. *modified partial least squares*), ICA (angl. *independent component analysis*) ir DBN (angl. *deep belief network*). Rezultatai pateikti 1 lentelėje.

1 lentelė. Klaidų aptikimo metodų palyginimas [31]

DPCA	MPLS	ICA	DBN(S)	DBN(G)	Neuroninis tinklas
89,13 %	96,32 %	94,83 %	61,47 %	86,65 %	97,73 %

Iš 1 lentelės matyti, kad neuroniniai tinklai tiksliausiai suklasifikavo gedimus – 97,73 proc. tikslumu.

1.2. Pramoninių valdiklių priedavų klasifikacija, pasitelkiant vaizdų ir laikinų signalų apdorojimo metodus

PLV priedavai yra skirti jungti įrenginį prie įvairių jutiklių, valdymo elementų, išėjimo įtaisų ir kitų komponentų. Priedavai gali būti klasifikuojami pagal keletą kriterijų:

- Funkcija. Įvesties (angl. *input*) priedavai skirti priimti signalus iš jutiklių ar kitų įtaisų, o išvesties (angl. *output*) priedavai skirti siųsti signalus valdymo elementams ar išėjimo įtaisams.
- Tipas. Priedavai gali būti skirti analoginiam arba skaitmeniniam signalų perdavimui. Analoginiai priedavai gali priimti arba siųsti įvairaus dydžio signalus, o skaitmeniniai priedavai priima arba siunčia tik tam tikrus būsenos signalus (pvz., įjungta / išjungta).
- Jungtis. Priedavai gali būti skirti įvairioms jungtims, tokioms kaip maitinimo, duomenų perdavimo arba specializuotoms jungtims (pvz., CAN, *Modbus*, *Ethernet/IP*).
- Gamintojas. PLC priedavai gali būti klasifikuojami pagal gamintoją, nes kiekvienas gamintojas gali turėti savo standartus ir sąsajas.

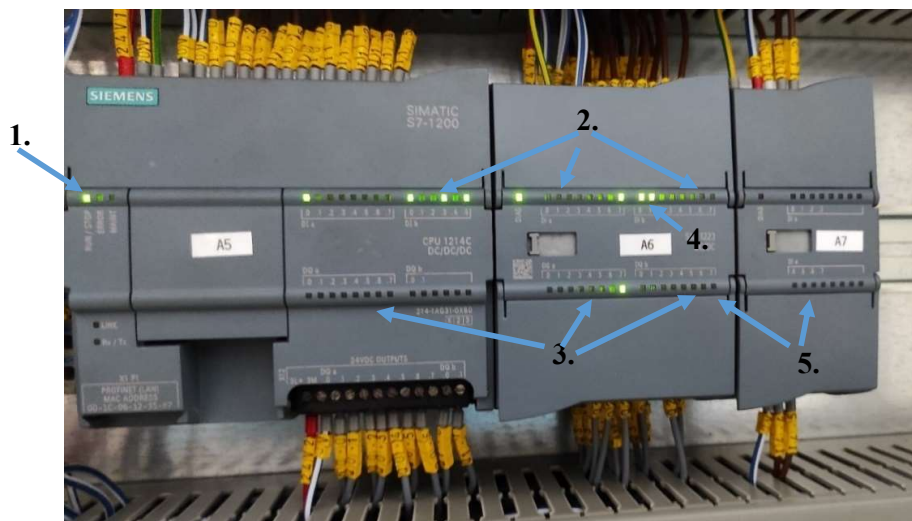
Svarbu paminėti, kad kiekvienas PLC modelis turi skirtingą priedavų konfigūraciją ir galimybes.

PLV yra plačiai naudojami automatizuotose sistemose dėl savo lankstumo ir patikimumo. PLV priedavų būsenų nuskaitymas yra svarbus veiksnys, leidžiantis stebėti ir valdyti įrenginius bei procesus. Yra keletas pagrindinių PLC priedavų būsenų nuskaitymo metodų:

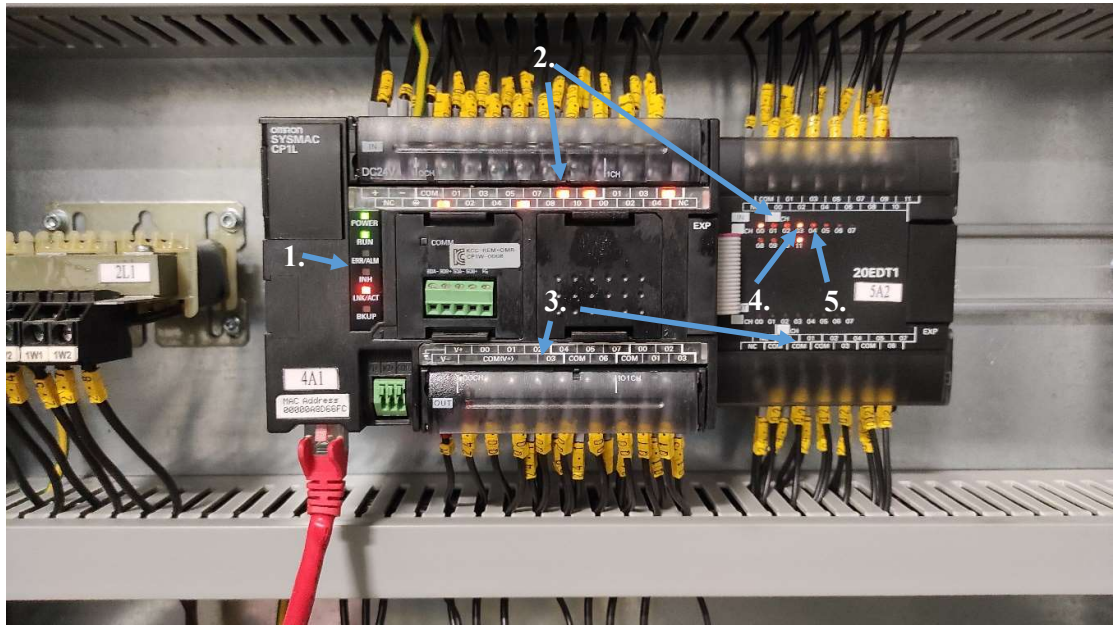
- Tiesioginis nuskaitymas. PLC prievadų būsenos nuskaitymas tiesiogiai iš valdiklio, naudojant specializuotas sąsajas ir protokolus, tokias kaip *Modbus*, *Profibus*, *Ethernet/IP* ir kt. Dažniausiai naudojami specializuoti komunikacijos moduliai, integruoti į PLC arba esantys atskiruose įrenginiuose.
- SCADA sistemos. Stebėjimo, kontrolės ir duomenų surinkimo (SCADA) sistemos naudojamos, kai reikia stebėti ir valdyti didelį skaičių PLC ir kitų įrenginių. SCADA sistemos komunikuoja su PLC per tinklus, kad gautų prievadų būsenų informaciją, kuri yra rodoma operatoriui arba naudojama automatizuotam valdymui.
- Skaitmeninių įvesties prievadų nuskaitymas. Skaitmeninės įvesties prievadai nuskaityti tik dvi būsenas: įjungtą (1) arba išjungtą (0). PLV prievadus galima nuskaityti tiesiogiai per fizines jungtis. Nuskaitymas gali būti atliekamas per nuoseklias arba paralelines jungtis.
- Vaizdo atpažinimas. Nors tai nėra įprastas būdas nuskaityti PLC prievadų būsenas, vaizdo atpažinimo technologijos gali būti naudojamos nuskaityti prievadų būsenas iš PLC operatoriaus panelių arba ekranų. Tai gali apimti naudojamus mašininio mokymosi algoritmus, tokius kaip *YOLO* arba kitus vaizdo analizės metodus, kad būtų galima nustatyti prievadų būsenų pokyčius.

Darbe tiriami valdikliai, kurių diskretingiems įėjimams ir išėjimams atvaizduoti yra naudojamos LED indikacijos (žr. 4, 5, 6, 7 pav.). Paveiksluose vienetais pažymėtos valdiklio būsenų indikacijos, dvejetainiais pažymėti sistemų įėjimai, trejetainiais – sistemų išėjimai. Užsižiebusi LED indikacija parodo gaunamą / išduodamą signalą (loginis 1), užgesinta indikacija – negaunamą / neišduodamą signalą (loginis 0). Nuskaityti indikacijų kitimą (pasitelkus vaizdų atpažinimo metodus) laike, galima sudaryti tvarkingos sistemos veikimo duomenų bazę, kurią išanalizavus (laikinių signalų apdorojimo metodais) galima aptikti sistemos nukrypimą nuo tvarkingo veikimo.

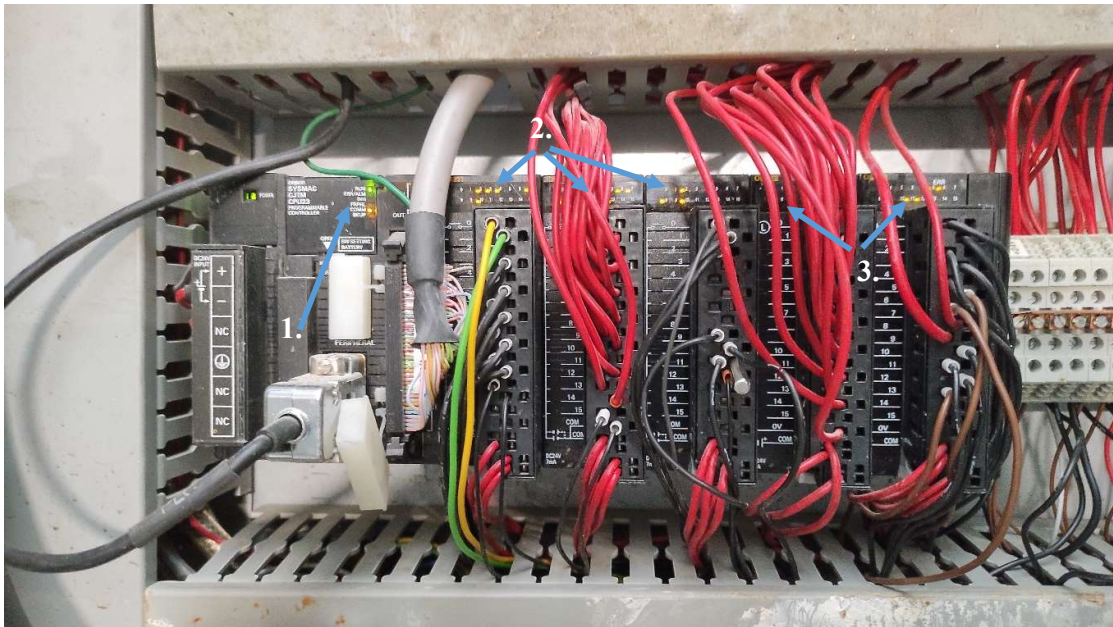
Pirma problema – LED prievadų šviečiančios LED indikacijos šviesos prasiskverbimas į aplinkines indikacijas. Tai galima matyti 4 ir 5 paveiksluose. Šalia šviečiančios LED indikacijos yra nešviečiančios („4“ pažymėta užsižiebusi indikacija, „5“ pažymėta neužsižiebusi indikacija, kuri dalinai atrodo kaip šviečianti). Antroji problema – instaliuoti laidai, dalinai uždengiantys indikacijas (žr. 6 pav.). Šios problemos sprendimas – perjungti laidus taip, kad jie neužstotų indikacijų.



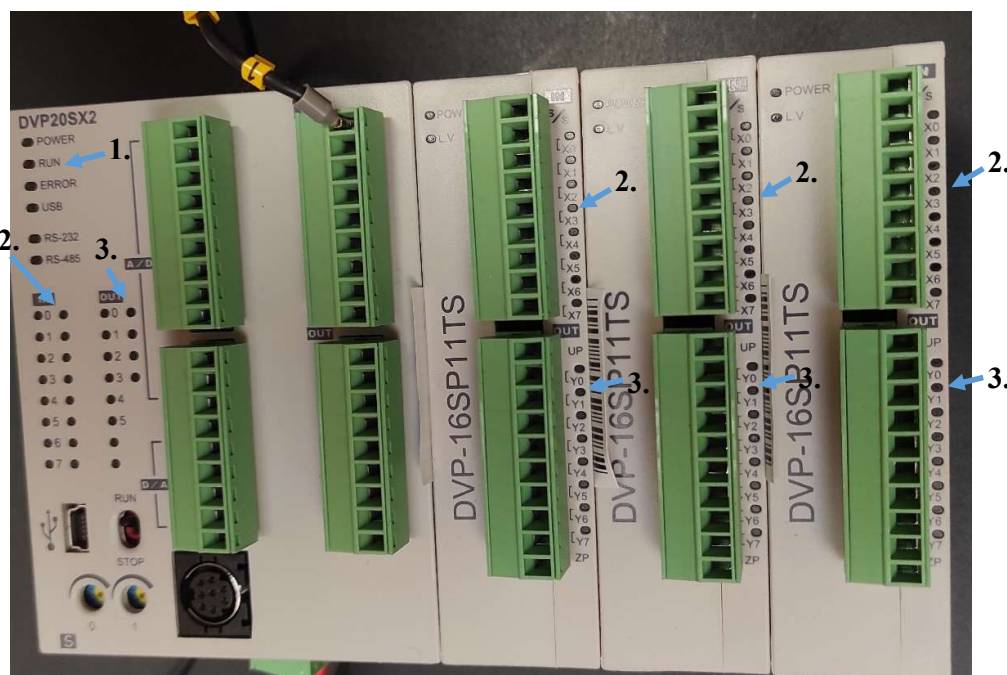
4 pav. „SIMATIC“ S7-1200 programuojamas loginis valdiklis (1 – sistemos būsenos indikacija, 2 – įėjimai, 3 – išėjimai, 4 – gaunamas signalas, 5 – negaunamas signalas) [sudaryta autoriaus]



5 pav. „OMRON SYSMAC“ CP1J programuojamas loginis valdiklis (1 – sistemos būsenos indikacija, 2 – įėjimai, 3 – išėjimai, 4 – gaunamas signalas, 5 – negaunamas signalas) [sudaryta autoriaus]



6 pav. „OMRON SYSMAC“ CJ1M programuojamas loginis valdiklis (1 – sistemos būsenos indikacija, 2 – įėjimai, 3 – išėjimai) [sudaryta autoriaus]



7 pav. „Delta“ DVP20SX2 programuojamas loginis valdiklis (1 – sistemos būsenos indikacija, 2 – įėjimai, 3 – išėjimai) [sudaryta autoriaus]

Tačiau tai padaryti praktiškai nėra lengva, nes dėl pasikeitusių laidų pravedimo vietų jie gali pasidaryti per trumpi, todėl reiktų pertvarkyti visus prie valdiklio prijungtus laidus.

1.2.1. Vaizdų apdorojimo metodai

Vaizdų apdorojimas – prasmingos informacijos ištraukimas iš vaizdų. Dažniausiai naudojamas informacijai iš skaitmeninių vaizdų išgauti. Vaizdo apdorojimo metu išgaunama brūkšniniam kode užkoduota informacija; žmogaus tapatybės patvirtinimas iš piršto antspaudu nuotraukos; pašalinių objektų aptikimas; pradingusių objektų aptikimas; augalų rūšies nustatymas iš jų lapo nuotraukos ir kt. Vaizdo atpažinimas susideda iš šių veiksmų: vaizdų importavimo, analizavimo, manipuliavimo ir vaizdo išvesties. Šiuo metu neegzistuoja universalus vaizdų apdorojimo metodas, tinkantis visoms galimoms vaizdų apdorojimo užduotims, todėl skirtingo tipo užduotims yra taikomi skirtingi vaizdų apdorojimo metodai. Toliau bus apžvelgiami vaizdų apdorojimo metodai, tinkantys LED būsenai nustatyti [13].

Vaizdų apdorojimą galima skirstyti į trys metodus: kompiuterinę regą (angl. *computer vision*), mašininis mokymasis (angl. *machine learning*) ir gilusis mokymasis (angl. *deep learning*).

Kompiuterinės regos atpažinimo metodas – tai vaizdų filtravimo, segmentavimo, savybių ištraukimo ir taisyklėmis pagrįsto klasifikavimo seka [13].

Vaizdo apdorojime taikomos operacijos:

- kraštų aptikimas (angl. *edge extraction*);
- linijų aptikimas (angl. *line extraction*);
- tekstūrų atskyrimas (angl. *texture discrimination*);
- formų atpažinimas (angl. *shape recognition*);
- objektų atpažinimas (angl. *object recognition*).

Po vaizdų apdorojimo yra naudojami matematiniai modeliai, kurie pagal gautus duomenis atpažįsta objektą. Pavyzdžiui, LED aptikimui galima naudoti kraštų ir formų aptikimo operacijas, o po to pritaikius tekstūros atskyrimo operaciją galima aptikti, ar LED šviečia, ar ne [13].

Mašiniame mokyme yra naudojami algoritmai, kurie apmokomi naudojant gerus bei blogus pavyzdžius. Šis metodas, lyginant su kompiuterinės regos metodu, yra paprastesnis, nes jam taikyti nereikalingos matematinės žinios, o papildomai naudojant gilųjį mokymosi metodą reikia tik mašininio mokymosi įrankių naudojimo žinių. Metodo trūkumas – visa mokymuisi naudojama medžiaga turi būti sužymima ir suklasifikuojama rankiniu metodu [13].

1.2.2. LED aptikimas naudojant kompiuterinę regą

Standartiškai naudojamas pilno kadro vaizdų atpažinimo algoritmas pateiktas 8 paveiksle.

Šio algoritmo trūkumas yra tas, kad kiekvieno gauto vaizdo pikselis yra dekoduojamas, o tai apima didelių trimačių duomenų apdorojimą. Tarkowski [9] pasiūlė efektyvesnį algoritmą LED indikacijoms aptikti (žr. 9 pav.).

10 paveiksle parodytas apdorojamas vaizdas. Siūlomas algoritmas sumažina duomenų, kuriuos reikia apdoroti dekodavimo proceso metu, kiekį. Pirmajame algoritmo etape taikomas fono modeliavimas, kuris nustato vaizdo sritis, kuriose šviesos diodų buvimas yra labiausiai tikėtinas (žr. 10 pav. dalis b). Tada likusiai vaizdo daliai yra taikomas papildomas dydžio ir formos filtras, pašalinantis neteisingai aptiktas šviesos diodų buvimo vietas, atsiradusias dėl pašalinių objektų judėjimo (žr. 10 pav. dalis c). Kitame žingsnyje apskaičiuojamas kiekvienos tikėtinės šviesos diodo buvimo vietos centroidas (žr. 10 pav. dalis d), kas leidžia sumažinti kiekvienos sritys dydį iki vieno pikselio. Šis procesas taip pat sumažina ir pikselių koordinačių, kurias reikia apdoroti antroje signalo demoduliavimo ir atpažinimo fazėje, skaičių. Dėl to bendras apdorojimo laikas trumpesnis, lyginant su viso kadro paieškos algoritmu [10].

Algoritmas 1. Viso kadro paieškos algoritmo pseudokodas:

kiekvienam pikseliui $p(x,y)$

kiekvienam kadrai n esančiame buferyje

Gauti intensyvumo vektorius $S_{x,y}$ laiko atžvilgiu

Apskaičiuoti slenkstinę reikšmę $r_{x,y}$

Slenkstis $S_{x,y}$ vektorius su $r_{x,y}$

Iššifruoti dvejetainį vektorių

Sutapatinti iššifruotus vektorius su žymomis

8 pav. LED indikacijos aptikimo algoritmas 1-as [9]

Algoritmams tarpusavyje palyginti naudotas kompiuteris su „Intel Core“ i3-2350M 2,3 GHz procesoriumi ir 8 GB RAM atmintimi. Algoritmai buvo įgyvendinti naudojant C++ programavimo kalbą su *OpenCV* vaizdo apdorojimo biblioteka. Jų palyginimui apdorojant vaizdus nebuvo naudojamas lygiagretus duomenų apdorojimas (angl. *parallel processing*). Bandymo rezultatai pateikti 11 paveiksle.

Bandymo metu nustatyta, kad naudojant antrąjį algoritmą vaizdų apdorojimas vyksta greičiau negu naudojant algoritmą 1. Esant mažos raiškos (0,13 MP) vaizdai antrasis algoritmas apdorojimą įvykdė

10,5 kartus greičiau nei pirmasis algoritmas. Didėjant apdorojamos medžiagos rezoliucijai algoritmų apdorojimo laikas didėja neproporcingai. Kai vaizdo rezoliucija yra 1,7 MP, antrasis algoritmas vaizdą apdoroja 250 kartų greičiau nei pirmasis algoritmas. Tarkowski įrodė, kad siūlomas algoritmas sumažina kompiuterinio skaičiavimo galią, reikalingą mirksintiems šviesos diodams aptikti [9].

Algoritmas 2. Siūlomas algoritmo pseudokodas:

-- Pirmas etapas:

kiekvienam kadrai n esančiame kadro buferyje

Fono analizė

Priekinio plano kaukės skaičiavimas

Regiono gavyba.

Formos ir dydžio filtravimas

Centroido skaičiavimas

--Antras etapas:

kiekvienam centroidui $p(x,y)$

Gauti intensyvumo vektorių $S_{x,y}$ laiko atžvilgiu

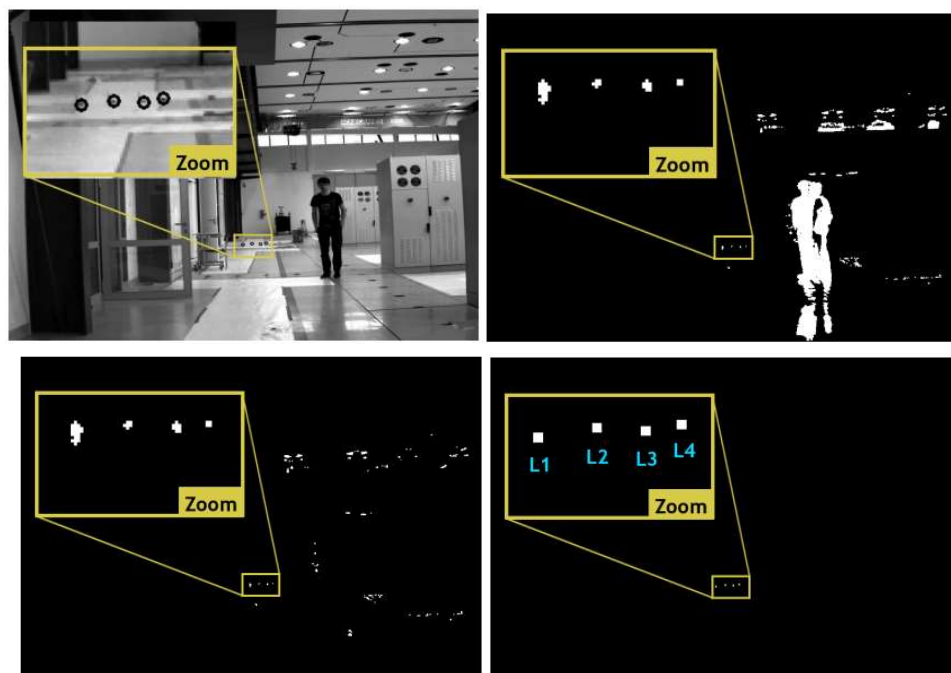
Apskaičiuokite slenkstinę reikšmę $r_{x,y}$

Slenkstis $S_{x,y}$ vektorius su $r_{x,y}$

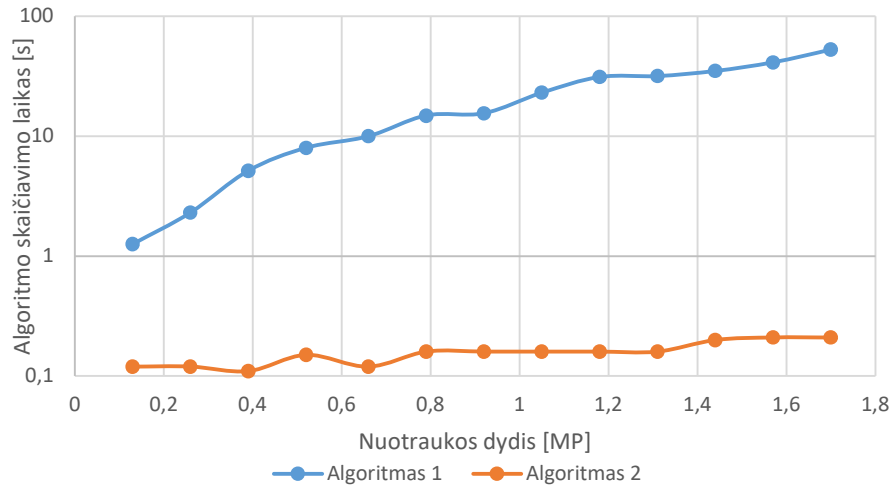
Iššifruoti dvejetainį vektorių

Suderinkite iššifruotus vektorius su žymomis

9 pav. LED indikacijos aptikimo algoritmas 2 [9]



10 pav. a) keturios žymos (juodi apskritimai) vaizde; b) žymos po fono filtravimo ir priekinio plano kaukės pritaikymo; c) žymos po formos ir dydžio filtravimo; d) žymos po formos ir dydžio filtravimo [9]



11 pav. Algoritmų greitimeikos grafinis palyginimas [9]

Pasiūlytas mirksinčių šviesos diodų aptikimo algoritmas, dėl sumažėjusios kompiuterinės skaičiuojamosios galios, gali būti įdiegtas įrenginiuose, turinčiuose didesnę nei 1 MP raiškos vaizdo kamerų jutiklius [9].

1.2.3. LED aptikimas naudojant mašininį mokymąsi

2021 m. Jilin universitete atliktas tyrimas, kurio metu buvo testuojamas duomenų perdavimas naudojant LED matricą kaip siųstuvą ir vaizdo kamerą kaip imtuvą. Šio tyrimo tikslas – išsiaiškinti, ar tokia sistema galėtų patikimai tarp važiuojančių automobilių perduoti informaciją, susijusią su iškilusiomis grėsmėmis kelyje. Komunikacijai naudota LED matrica pavaizduota pritvirtinta prie automobilio 12 paveiksle [12].

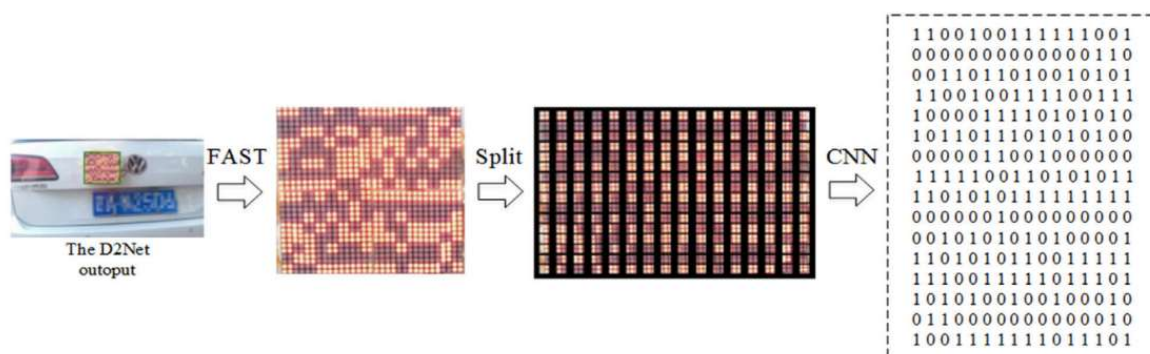
LED būsenų atpažinimą apsunkina lauko sąlygos: intensyvi saulės šviesa bei automobilio sukeliama judesio neryškumai. 1.2.2. punkte pademonstruotas LED aptikimo algoritmas šiuo atveju būtų neefektyvus, nes dėl saulės šviesos norimi aptikti LED ne visada būtų šviesiausi objektai nuotraukoje, o dėl judesių neryškumo nebūtų galima aptikti LED kontūrų. Šioms problemoms išspręsti buvo pasiūlyta naudoti realaus laiko objektų aptikimo algoritmo „*You Only Look Once*“ [17] 5-ąją versiją bei sukurtą LED segmentacijos atpažinimo metodą [11].

Tyrime naudojamas algoritmas buvo pavadintas *D2Net*. LED segmentacijai naudoti algoritmai pateikti 15 paveiksle. Pirmiausia naudotas greitas kampų aptikimo algoritmas, o po to gauta matrica yra suskirstyta į pavienius LED-us (suskirstyta į 256 skirtingus blokus). Galutinei informacijai nuskaityti pritaikytas konvoliucinio neuroninio tinklo algoritmas (angl. *convolutional neural networks*, CNN).

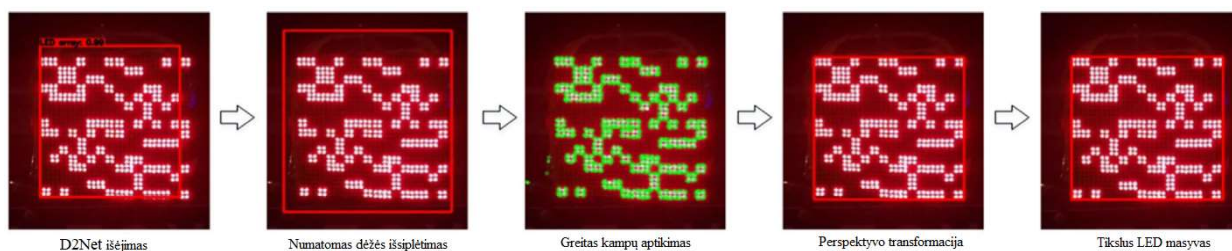
14 paveiksle parodyti LED matricos segmentavimo masyvo blokai. Padarytos LED masyvų nuotraukos rankiniu būdu palygintos su žinomomis vertėmis. Sužymėjus visas LED matricas, toliau apmokytas CNN modelis.



12 pav. Automobilis su pritvirtinta LED matrica [12]



13 pav. Pasiūlyti LED matricos suskirstymo metodai: greitas kampų aptikimo algoritmas (angl. *fast*), matricos suskirstymas (angl. *split*), konvoliuciniai neuroniniai tinklai (angl. *convolutional neural networks* CNN) [12]



14 pav. Pasiūlytas LED matricos atpažinimo metodas A [12]

D2Net algoritmo veikimui ištirti realiomis eismo sąlygomis surinkta testavimo medžiaga. Vaizdo įrašai daryti dienos bei nakties metu. Iš nufilmuotos medžiagos sugeneruota 6 000 neryškių bei ryškių nuotraukų ir surinkta 30 000 nuotraukų porų, iš jų 20 000 naudota *D2Net* tinklui apmokyti, o likusios poros – testavimo medžiaga.

Eksperimento sąlygos pateiktos 2 lentelėje, o 3 lentelėje – *D2Net* bei kitų modelių tyrimų rezultatai. 3 lentelėje naudoti žymėjimai:

- T – tikslumas;
- A – atmetimo dažnis;
- VA% – vidutinis tikslumas;
- Vid – rezultatų atitikimo lygis (modelio rezultato palyginimas su realia reikšme);
- S/T – signalo ir triukšmo santykis;
- SP – struktūriniai panašumai;
- Dydis – modelio dydis megabaitais;
- FPS – kameros užfiksuoti kadrai per sekundę.

2 lentelė. Eksperimento sąlygos [12]

Eksperimento parametrai		Vertė
LED matricos dydis		15 × 15 cm
LED iluminacija		350 lux (10 cm atstumu)
Dienos metu	Aplinkos viešumas	2940 lux
	Fotografavimo greitis	1/1000 s
	ISO	50
Nakties metu	Aplinkos viešumas	14 lux
	Fotografavimo greitis	1/320 s
	ISO	1600
Kameros fiksuojami kadrai per sekundę		240 fps
Kameros rezoliucija		1280 × 720
Kameros lęšių fokusavimo gylis		27 mm

3 lentelė. Eksperimento rezultatai lyginant *D2Net* su kitais moderniais algoritmais [12]

Modelis	T	A	VA%	Vid	S/T	SP	Dydis	FPS
SSD300 (VGG16)	0,801	0,859	82,23	0,69			92	58
<i>YOLOv3</i>	0,865	0,917	89,78	0,756			235	50
<i>YOLOv3-tiny</i>	0,639	0,765	71,28	0,598			33	235
<i>YOLOv4</i>	0,997	1,000	99,91	0,854			245	42
<i>YOLOv4-tiny</i>	0,708	0,836	85,34	0,609			23	217
<i>YOLOv5s</i>	0,868	0,922	89,81	0,763			27	112
<i>YOLOv5m</i>	0,953	0,965	96,52	0,817			84	79
<i>YOLOv5l</i>	0,997	0,998	99,83	0,842			192	55
<i>YOLOv5x</i>	0,998	1,000	99,99	0,912			367	35
<i>DeblurGAN-v2</i>	-	-	-	-	32,57	0,819	233	21
<i>D2Net</i>	0,996	0,998	99,81	0,833	31,70	0,810	206	39

15 paveiksle yra parodytos eksperimento sąlygos dienos ir nakties metu.

4 lentelėje palygintas skirtingų modelio bitų klaidų skaičius BER (angl. *bit error rate*). Jis gaunamas padalinus klaidingai perduotų bitų skaičių iš visų per tiriamo laiko intervalą perduotų bitų skaičiaus. 4 lentelėje susiliejusį nuotraukų rezultatai yra pateikti atskirai nuo nesusiliejusių.



15 pav. Eksperimento sąlygos dieną (a ir b) ir naktį (c ir d) [12]

4 lentelė. Klaidų palyginimas [11]

Modelis	BER		FPS
	Nesusiliejęs	Susiliejęs	
<i>YOLOv3-tiny</i> + Vidutinis slenkstis	0,649	0,928	202
<i>YOLOv3-tiny</i> + Centrinis slenkstis	0,635	0,927	202
<i>YOLOv3-tiny</i> + Segmentų atpažinimas	0,466	0,919	143
<i>YOLOv5x</i> + Vidutinis slenkstis	0,536	0,924	34
<i>YOLOv5x</i> + Centrinis slenkstis	0,526	0,923	34
<i>YOLOv5x</i> + Segmentų atpažinimas	0,268	0,912	34
Modelis	BER		FPS
	Nesusiliejęs	Susiliejęs	
<i>D2Net</i> + Vidutinis slenkstis	0,538	0,629	38
<i>D2Net</i> + Centrinis slenkstis	0,527	0,614	38
(Pasiūlytas algoritmas) <i>D2Net</i> + Segmentų atpažinimas	0,272	0,439	38

Iš gautų duomenų, matomų 3 ir 4 lentelėse, matyti, kad pasiūlytas *D2Net* modelis, lyginant su kitais modeliais, duomenis nuskaity dvigubai tiksliau, kai filmuojamoje medžiagoje vaizdas yra susiliejęs.

1.2.4. Laikinių signalų apdorojimo metodai

Sekų atpažinimas (angl. *pattern recognition*) – tai automatizuotas duomenų sekų ir dėšningumų atpažinimas, taikomas statistinei duomenų analizei, informacijos paieškai, signalams apdoroti, vaizdams analizuoti, kompiuterinei grafikai, bioinformatikai, duomenims suspausti ir mašiniam mokymuisi. Laiko sekų atpažinimas pradėtas vystyti statistikoje ir inžinerijoje. Mašininio mokymosi naudojimas pradėtas taikyti per paskutiniuosius du dešimtmečius. Laiko sekų atpažinimo sistemos dažniausiai mokomos iš pažymėtų „mokymo“ duomenų. Kai nėra pažymėtų duomenų, nežinomoms sekoms atrasti gali būti naudojamas duomenų kasimas (angl. *data mining*) [10].

Atpažinimas – tai etiketės priskyrimas nurodytai įvesties reikšmei. Statistikoje 1936 m. įvesta diskriminacinė analizė. Klasifikacija yra sekų atpažinimo pavyzdys, kuria bandoma priskirti kiekvieną įvesties reikšmę vienai iš tam tikrų klasių rinkinių (pavyzdžiui, siekiant nustatyti, ar duotas el. laiškas yra „spam“ ar „ne spam“). Sekos atpažinimas yra bendresnė problema, susijusi su kitų tipų išvestimis. Kiti pavyzdžiai yra regresija, pagal kurią kiekvienai įvesčiai priskiriama tikroji vertė; nuoseklusis žymėjimas, kuris kiekvienam reikšmių sekos nariui priskiria klasę (pvz., žodžio tipo žymėjimas, priskiriantis žodžio tipą kiekvienam įvesties sakinio žodžiui), ir analizavimas, kuris įvesties sakiniui priskiria analizės medį, apibūdinantį sakinio sintaksinę struktūrą [11].

Sekų atpažinimas skirstomas į kategorijas pagal sistemos išėjimų apmokymo būdą. Sekų atpažinimo apmokymas gali būti prižiūrimas (angl. *supervised learning*), neprižiūrimas (angl. *unsupervised learning*) ir pusiau prižiūrimas mokymasis (angl. *semi-supervised learning*). Prižiūrimo apmokymo metu yra pateikiamas apmokymo duomenų rinkinys, kurį sudaro egzempliorių rinkinys, kuriame iš anksto rankiniu būdu yra nurodomos teisingos išvestys. Mokymosi procedūra sukuria modelį, kuris bando pasiekti du kartais vienas kitam prieštarigus tikslus: kuo geriau prisitaikyti prie apmokymo duomenų ir kuo geriau apibendrinti naujus duomenis. Neprižiūrimas mokymasis naudoja apmokymo duomenis, kurie nebuvo iš anksto pažymėti rankiniu būdu. Mokymo metu yra bandoma aptikti esmines sistemos sąsajas, būdingas duotiems duomenims, pagal kuriuos vėliau galima nustatyti teisingą naujų duomenų išvesties vertę. Šių dviejų apmokymo būdų derinys yra pusiau prižiūrimas apmokymas. Apmokymo metu naudojamas pažymėtų ir nepažymėtų duomenų derinys (paprastai mažas sužymėtų duomenų rinkinys kartu su dideliu nesužymėtų duomenų kiekiu). Neprižiūrimo apmokymo atveju apmokymo duomenų gali ir nebūti [11].

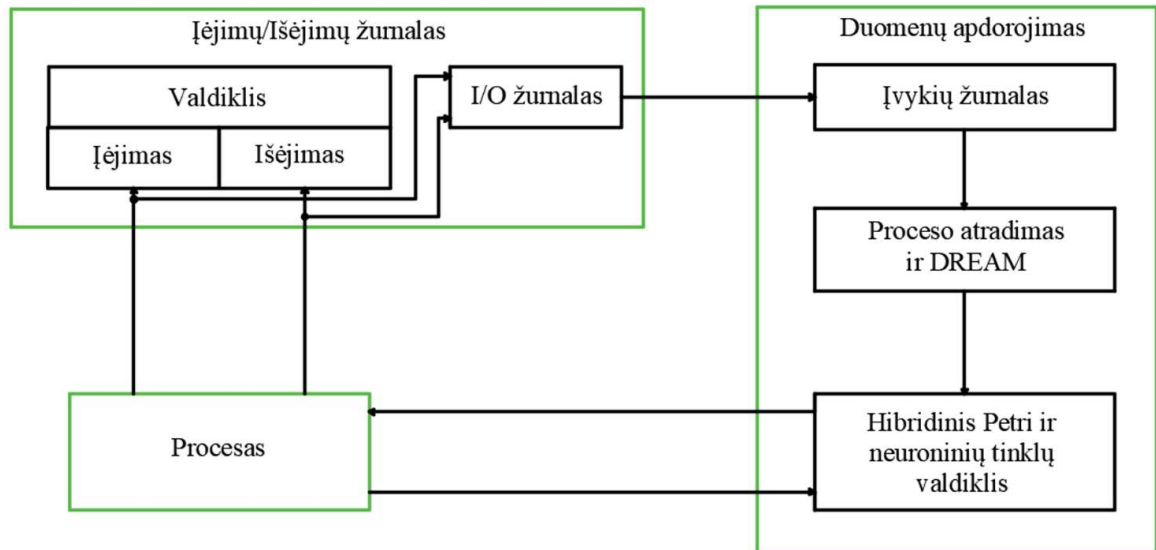
Praktinis duomenų apdorojimas buvo pademonstruotas 2019 metais tarptautinėje automatikos mokslo ir inžinerijos konferencijoje (angl. *International conference on Automation Science and Engineering*) [13]. Theis ir kiti pristatė būdą, kaip analizuojant PLC įėjimų ir išėjimų signalus atkurti vidinę valdiklio logiką ir ją atvaizduoti Petri tinklais (angl. *petri networks*). Šiam tikslui įgyvendinti pirmiausia yra nuskaitoma valdiklio įėjimo ir išėjimo binarinių verčių kaita laiko atžvilgiu. Gauti duomenys yra konvertuojami į įvykių žurnalą (angl. *event log*). Analizuojant įvykių žurnalą yra išgaunamas proceso modelis, kuris yra atvaizduojamas Petri tinklų. Laikinių signalų apdorojimo metodo diagrama yra pavaizduota 16 paveiksle.

Pasiūlyto metodo tyrimui buvo naudojamos skysčių talpos lygio valdymo sistemos, kurias galima rasti įvairiose pramonės srityse, tokiose kaip skysčių filtravimas, maisto perdirbimas ar vandens valymo įrenginiai. Sukurtą eksperimentinę aplinką sudaro modeliuojama gamyklos aplinka ir LLD (angl. *ladder logic diagram*) programa parašyta naudojant *OpenPLC* programinę įrangą. Talpos tūrio valdiniui (valdomos talpos tūris 100 galonų) buvo sukurti du scenarijai ir dvi skirtingos valdymo programos. Valdymo programos yra pavaizduotos 17 ir 18 paveiksluose [13].

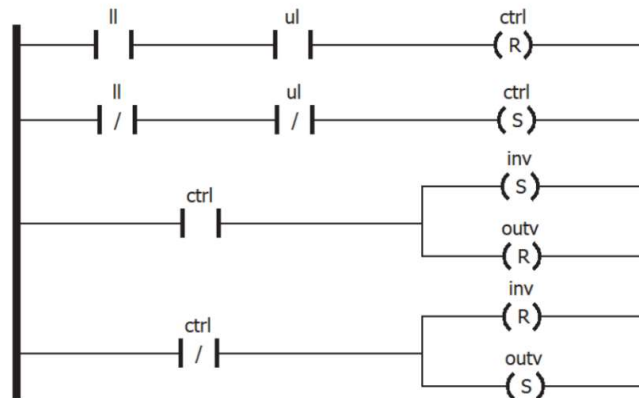
Programoje naudojamų trumpinių reikšmės:

- U1 – viršutinis talpos jutiklis (angl. *uper level*), kuris suveikia, kai sistemoje yra 90 galonų skysčio;
- M1 – vidurinis talpos jutiklis (angl. *midle level*), kuris suveikia, kai sistemoje yra 50 galonų skysčio;
- L1 – viršutinis talpos jutiklis (angl. *lover level*), kuris suveikia, kai sistemoje yra 10 galonų skysčio;
- Inv – padavimo į talpą vožtuvas (angl. *input valve*);

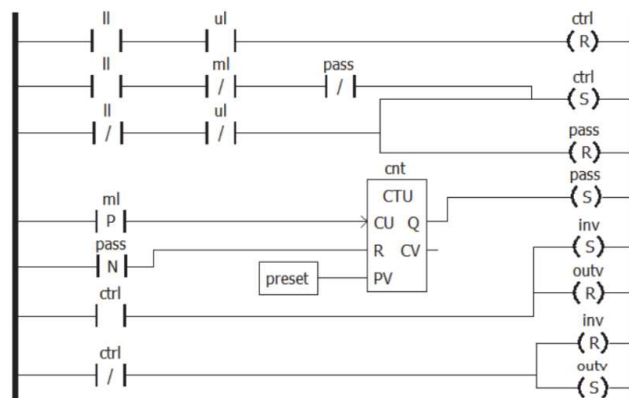
- Outv – išdavimo iš talpos vožtuvas (angl. *output valve*);
- Ctrl – vožtuvų valdymo atmintis;
- Pass – atmintis, ištuštinanti talpą po to, kai talpa yra pripildoma trečią kartą.



16 pav. Laikinių signalų apdorojimo metodo diagrama [13]

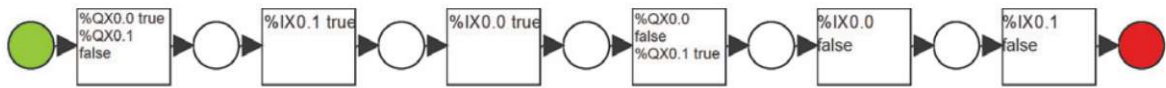


17 pav. Pirmoji sistemos valdymo programa C1 [12]

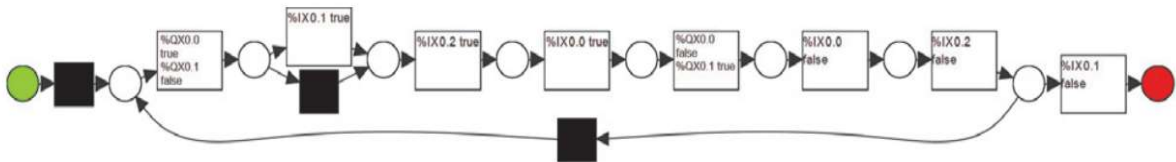


18 pav. Antroji sistemos valdymo programa C2 [13]

Naudojant 18 ir 19 paveiksluose pavaizduotas programas, sugeneruotas įėjimų ir išėjimų įvykių žurnalas, kurį pateikus į pasiūlytą algoritmą gaunami 19 ir 20 paveiksluose pavaizduoti *Petri* tinklai.



19 pav. Iš pirmosios programos išgautas *Petri* tinklas (žalia spalva parodo programos pradžią, raudona – pabaigą) [13]



20 pav. Iš antrosios programos išgautas *Petri* tinklas (žalia spalva parodo programos pradžią, raudona – pabaigą) [13]

Iš gautų rezultatų matyti, kad yra įmanoma atkurti valdiklio programos logiką, analizuojant valdiklio įėjimus ir išėjimus. Straipsnyje [12] nurodyta, kad tai pirma veikianti programos logikos atkūrimo sistema pasaulyje. Autoriai pateikia ir pagrindinį šio tyrimo trūkumą – mažas įėjimų ir išėjimų skaičius. Testuojamoje sistemoje buvo tik 3 įėjimai ir 2 išėjimai. Realiose sistemose įėjimų ir išėjimų skaičius didesnis. 4, 5 ir 6 paveiksluose yra pavaizduotos realios Lietuvos įmonėse veikiančios sistemos, kurios turi nuo 24 iki 48 įėjimų ir nuo 18 iki 32 išėjimų.

Tolesniems tyrimams bus naudojama kompiuterinė rega, nagrinėta punkte 1.2.2, skirta LED būsenų aptikimui ir įvykių žurnalo sudarymui. Toliau sudarytas įvykių žurnalas bus naudojamas neuroninio tinklo apmokymui. Apmokius tinklą bus testuojamas jo efektyvumas aptinkant sistemos klaidas.

2. PLV prievadų identifikavimo tyrimas, pasitelkiant neuroninius tinklus

Šiame skyriuje aptariamas tyrimo metodas, tirama sistema, duomenų rinkimas bei jų paruošimas neuroniniam tinklui apmokyti. Taip pat paaiškinti objektų atpažinimo įvertinimo kriterijai. Šie metodai ir būdai toliau bus naudojami eksperimentiniams tyrimams atlikti.

2.1. Tyrimo metodika

Pasirinkta tyrimo metodika – eksperimentinis tyrimas. Jo metu yra tiriamas pats regresinis neuroninis tinklas *YOLOv4* ir tinklui daromas aplinkos poveikis.

Eksperimentuose naudotą fizinę įrangą sudaro: išmanusis telefonas („Pocophone f1“, mikroschemų rinkinys „Snapdragon“ 845, 6 GB operatyvioji atmintis, 12 MP, f/1.9 kamera), telefono laikiklis, sumontuotas ant DIN bėgelio ir laikantis telefoną 17 cm atstumu nuo valdiklio (žr. 21 pav.), nešiojamas kompiuteris (procesorius I7-4710HQ, operatyvioji atmintis 16 GB, DDR3, 1600Mhz, vaizdo plokštė: GTX 850M, 4 GB) ir programuojamas loginis valdiklis: „Siemens“ S7-1214DC/DC/DC su papildomais praplėtimų moduliais 6ES7223-1BL32-0XB0.



21 pav. Tyrimo metu naudotas laikiklis [sudaryta autoriaus]

Eksperimentuose naudotą programinę įrangą sudaro: kompiuterio operacinė sistema *Windows 10*, vaizdams apdoroti naudota *MATLAB R2022a*, telefono operacinė sistema *Android 10* QKQ1.190828.002 ir vaizdo fiksavimo programa *Open Camera v1.50.1*.

Eksperimentui atlikti, naudojant skirtingus vaizdo fiksavimo nustatymus, nufilmuota 12 vaizdo įrašų. Vaizdui fiksuoti naudoti kintamieji: rezoliucija (3840×2160 , 1920×1080 , 1280×720 , 720×480), ir apšvietimas. Valdiklis apšviestas 3 būdais (žr. 22 pav.):

- elektros spintos durys atviros, šviesos šaltinis – natūralus patalpų apšvietimas;
- duris uždaros be papildomo apšvietimo;
- duris uždaros su papildomu telefone įmontuotu LED apšvietimu.



22 pav. PLV apšvietimas: a) natūralus patalpų apšvietimas; b) uždaros durys su papildomu LED apšvietimu; c) uždaros durys be papildomo apšvietimo [sudaryta autoriaus]

24 paveiksle matyti, kad šviečiančios indikacijos dalinai iluminouoja aplinkines indikacijas, tai gali sukelti sunkumų indikacijų būsenos vertėms nuskaityti.

2.2. Tiriamasis objektas

Tyrimo objektu pasirinktas batono dėžių krautuvas (žr. 23 pav.). Į krautuvą paduotos pavienės batonais užpildytos dėžės, kurios prietaiso viduje yra sukraunamos viena ant kitos. Pasiekus nustatytą dėžių krūvos aukštį, dėžės yra išduodamos į sukaupimo konvejerį.



23 pav. Tirtas batono dėžių krautuvas [sudaryta autoriaus]

Tokio tipo įrenginiai yra naudojami duonos kepimo fabrikuose, tokiuose kaip „Vilniaus duona“, „Lietuvos kepėjas“ ir „Latvijas Maiznieks“.

Atliekant vaizdų atpažinimo neuroninio tinklo apmokymą ir vaizdų atpažinimo bandymus, susidurta su problemomis.

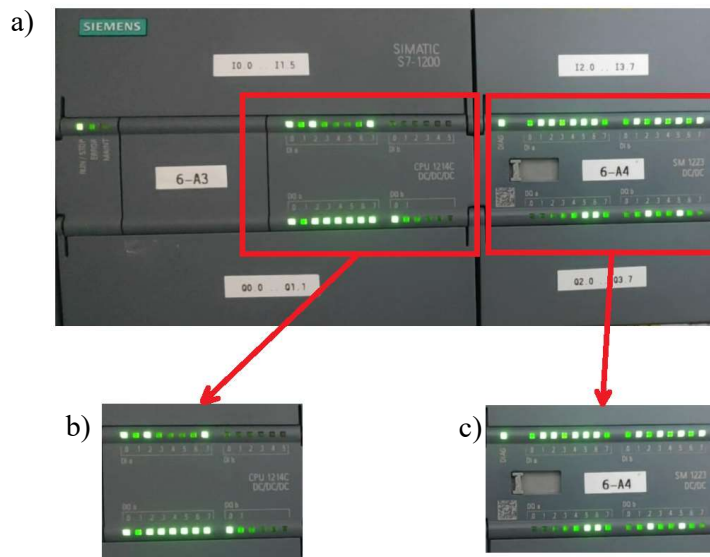
- Užsidegusios indikacijos apšviečia šalia esančias indikacijas, taip apsunkindamos indikacijos būsenos nustatymą (žr. pav. 28).
- Sistemos virpėjimas. Sistemos valdymo spinta yra sumontuota ant tos pačios konstrukcijos kartu su dėžių kėlimo cilindrais, dėl to kiekvieno judesio metu atsirandančios vibracijos sukelia vaizdo susiliejamą.
- Naudotas telefono kameros objektyvas nėra pritaikytas filmavimui plačiu kampu, dėl to filmuojamo vaizdo kraštuose vaizdas yra šiek tiek susiliejęs (žr. pav. 22).
- Duomenų žymėjimo neapibrėžtumas. Filmuotoje medžiagoje užfiksuojamas LED indikacijos perėjimas tarp užsidegimo / užgesimo būsenos. Tai sukelia neaiškumą, kaip tokia indikacija turėtų būti traktuojama.
- Ilgai trunkantis duomenų sužymėjimo procesas reikalauja nenutrūkstamo atidumo.

2.3. Vaizdų paruošimas tinklui apmokyti

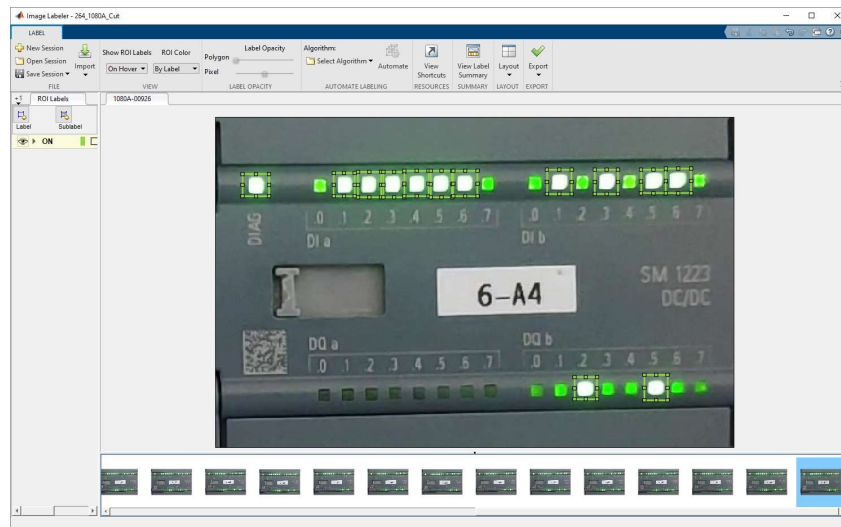
Paruošti 12 vaizdo įrašų (žr. poskyrį 2.1) yra išskaidomi į pavienes nuotraukas, naudojant programinę įrangą *Shotcut* ir taip sudarant 12 nuotraukų rinkinių. Iš kiekvieno rinkinio atsitiktiniu būdu atrinkta

po 20 nuotraukų. Atrinktos nuotraukos apkirptos ir padalintos į dvi dalis, nuotraukų padalijimas pavaizduotas 24 paveiksle.

Iškirptos pagrindinio valdiklio modulio nuotraukos panaudotos tinklui apmokyti (žr. 26 pav. b dalis), o valdiklio praplėtimo modulio nuotraukos – apmokytam tinklui testuoti (žr. 26 pav. c dalis). Mokomoji medžiaga (užsižiebusios LED indikacijos) yra sužymima, naudojant programinės įrangos *Matlab* įrankį „Image Labeler“. Visoms užsižiebusioms LED indikacijoms suteiktas „ON“ pavadinimas. Sužymėtos nuotraukos pavyzdys pateiktas 25 paveiksle.



24 pav. Vaizdų apkarpymas: a) pradinis vaizdas; b) iškirptas PLV pagrindinis modulis; c) iškirptas PLV praplėtimų modulis [sudaryta autoriaus]



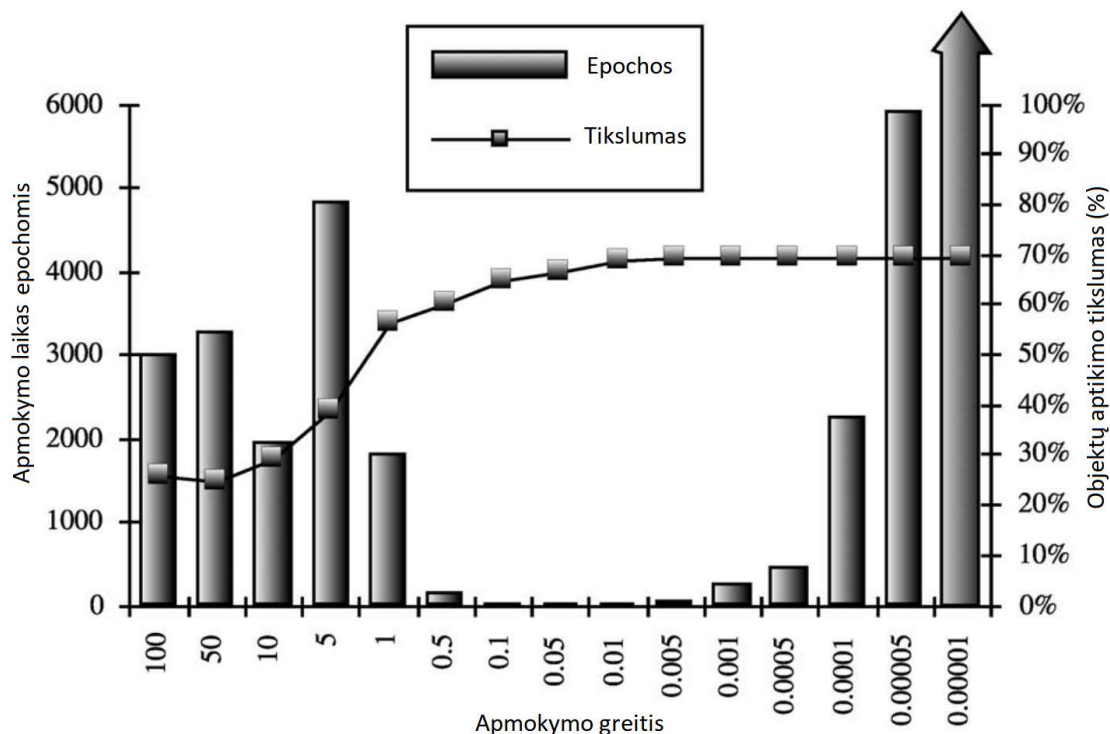
25 pav. Įrankiu „Image Labeler“ sužymėta nuotrauka [sudaryta autoriaus]

Tokiu būdu sudaromi 12 apmokymui naudojamų duomenų rinkinių.

2.4. Neuroninio tinklo apmokymo parametrų parinkimas

Parenkant neuroninio tinklo apmokymo parametrus, reikia kompromisuoti tarp duomenų rinkinio ypatumų, kompiuterinės įrangos pajėgumų ir norimų pasiekti rezultatų. Toliau pateikiamos mokslinės literatūros apžvalgos metu rastos parametrų parinkimo rekomendacijos, kurios yra naudojamos darbo atlikimui.

- Išnagrinėti duomenų rinkinio dydį, kokybę ir klasių pasiskirstymą. Geresnių rezultatų pasiekimui, didesniems duomenų rinkiniams reikia daugiau mokymosi iteracijų. Svarbu užtikrinti, kad duomenų rinkinys turėtų tikslias anotacijas ir subalansuotą klasių pasiskirstymą [34].
- Vaizdo rezoliucija: YOLOv4 gali būti mokomas įvairiose rezoliucijose, pvz., 416×416 , 608×608 arba 832×832 . Aukštesnė rezoliucija dažnai lemia geresnį tikslumą, bet reikalauja didesnių skaičiavimo išteklių ir ilgesnio mokymosi laikotarpio [34]. Toliau darbe bus tirama rezoliucijos įtaka aptikimo tikslumui ir nustatoma geriausia rezoliucija tyrinėjamam objektui ieškant balanso tarp aptikimo tikslumo ir greitaveikos.
- Iš anksto apmokytų svorio reikšmių naudojimas pagreitinant tinklo apmokymo procesą. Naudojant iš anksto apmokytą neuroninį tinklą ir jį permokant ant norimų aptikti objektų yra sutaupoma laiko, reikalingo pasiekti maksimalų objektų aptikimo tikslumą [35].
- Mokymosi greitį pagal Wilson [36] rekomenduojama rinktis tarp 0,5 ir 0,001 (žr. 26 pav.). Tyrimo metu pasirinkus per didelį apmokymo greitį apmokymo procesas yra nutraukiamas, nes aptiktų objektų koordinatės išeina iš nuotraukos ribų. Parinktas apmokymo greitis 0,01.



26 pav. Apmokymo greičio įtaka apmokyto tinklo objektų aptikimo tikslumui [36]

- Mini partijos dydžio parinkimas (angl. *mini batch size*). Parenkamas pagal turimos atminties dydį. Didesnis partijos dydis lemia geresnį modelio apmokymą, bet tai reikalauja daugiau atminties [36]. Dėl ribotos atminties dydžio pasirinkta naudoti partijos dydį 1.
- Mokymosi iteracijų skaičius nustatomas atsižvelgiant į duomenų rinkinio dydį ir norimą modelio veikimą. Pasirinkus per didelę apmokymo iteracijų dydį galima susidurti su permokymo problema, tai yra apmokytas tinklas teisingai aptiks objektus tik kai tiriamas vaizdas bus identiškas naudotam apmokymo metu. Pagal paveiksle pateiktus duomenis pasirinktas apmokymo iteracijų skaičius yra 70 [36].
- Taikyti duomenų augmentaciją. Taikant duomenų augmentacijos metodus, tokius kaip vaizdo pasukimas, apvertimas ir spalvų koregavimas, dirbtinai padidinamas duomenų rinkinys, pagerinamos modelio objekto aptikimo galimybės, tai padeda išvengti tinklo permokymo [37]. Apmokymo metu naudotos vaizdo augmentacijos pavyzdys pateikiamas 27 paveiksle.



27 pav. Apmokymo metu naudotos vaizdų augmentacijos pavyzdys [sudaryta autoriaus]

- Nuostolių funkcija. *YOLOv4* naudoja Kryžminio-entropijos nuostolio, IoU (sąjungos su sankirta) nuostolio ir objektyvumo nuostolio derinį. Svarbu užtikrinti, kad šios nuostolių funkcijos būtų tinkamai apsvarstytos, siekiant pasiekti optimalų lokalizacijos, klasifikavimo ir objektyvumo balansą [38].
- Modelio architektūra. Reikalingas eksperimentavimas su skirtingais *YOLOv4* variantais (pvz., *YOLOv4-tiny*), atsižvelgiant į norimą tikslumo ir skaičiavimo efektyvumo pusiausvyrą. Mažesni modeliai paprastai yra greitesni, tačiau mažiau tikslūs, o didesni modeliai dažnai pasižymi geresniu veikimu, bet reikalauja didesnių skaičiavimo išteklių [35].

Neuroninių tinklų apmokymui naudojami parametrai pateikiami 5 lentelėje.

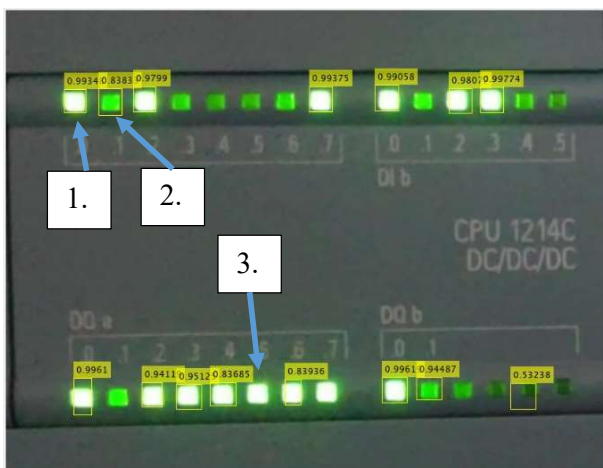
5 lentelė. Apmokymo parametrai

Parametras	Parametro vertė
Gradiento nykimo faktorius (angl. <i>gradient decay factor</i>)	0,9
Kvadratinis gradiento nykimo faktorius (angl. <i>squared gradient decay factor</i>)	0,999
Pradinis mokymosi rodiklis (angl. <i>initial learn rate</i>)	0,01
Pradinio mokymosi tvarkaraštis (angl. <i>learn rate schedule</i>)	Jokio
Mini partijos dydis (angl. <i>mini batch size</i>)	1
L2 reguliavimas (angl. <i>L2 regularization</i>)	0.0005
Apmokymo epochų skaičius (angl. <i>max epochs</i>)	70
Partijos normalizavimo statistika (angl. <i>batch normalization statistics</i>)	Judantis
Tarpinis tinklo verčių išsaugojimas (angl. <i>dispatch in background</i>)	Ne
Ivesties normalizavimo atstatymas (angl. <i>reset input normalization</i>)	Ne
Nuotraukų sumaišymas (angl. <i>shuffle</i>)	Kiekviena epocha
Apmokymo rezultatų pateikimo dažnumas (angl. <i>verbose frequency</i>)	20
Vykdomo aplinka (angl. <i>execution environment</i>)	CPU

Tinklui apmokyti yra naudotas procesorius, o ne vaizdo plokštė, nes turimos vaizdo plokštės atminties nepakanka tokiam tinklui apmokyti (turimos vaizdo plokštės atmintis 4 GB, o reikalinga ne mažiau kaip 6 GB). Pasinaudojus 5 lentelėje pateiktais nustatymais, vieno tinklo apmokymas vidutiniškai užtruko 4 valandas. Apmokymo metu sistema naudojo iki 12,5 GB RAM atminties.

2.5. Vaizdo medžiagos rezoliucijos įtaka vaizdo atpažinimo tikslumui

28 paveiksle pateiktas aptikto vaizdo pavyzdys, kur „1“ pažymėta teisingai aptikta indikacija, „2“ – neteisingai aptikta indikacija, „3“ – neaptikta indikacija. Tyrimo metu gauti rezultatai pateikiami 6 ir 7 lentelėse.



28 pav. Aptikimo pavyzdys: 1 – teisingai aptikta indikacija; 2 – neteisingai aptikta indikacija, 3 – neaptikta indikacija [sudaryta autoriaus]

6 lentelė. Rezultatai, gauti naudojant 3840×2160 vaizdo rezoliuciją ir atviros spintos apšvietimą

Nuotraukos Nr.	Užsižiebusių indikacijų skaičius	Teisingai aptiktų indikacijų skaičius	Neaptiktų indikacijų skaičius	Klaidingai aptiktų indikacijų skaičius
1	14	14	0	0
2	13	13	0	0
3	15	15	0	1
4	11	11	0	0
5	14	14	0	1
6	17	17	0	0
7	12	12	0	0
8	12	12	0	0
9	15	15	0	0
10	12	12	0	0
Suma	135	135	0	2

6 lentelėje pateiktas duomenų rinkimo pavyzdys. Kiekvienam neuroninio tinklo apmokymui sudaroma tokio tipo lentelė. Stulpelyje „Užsižiebusių indikacijų skaičius“ užrašomas realus užsižiebusių indikacijų skaičius. Stulpelyje „Teisingai aptiktų indikacijų skaičius“ surašomas teisingai aptiktų indikacijų skaičius. Stulpelyje „Neaptiktų indikacijų skaičius“ surašomas užsižiebusių indikacijų skaičius, kurių neuroninis tinklas neaptiko. Stulpelyje „Klaidingai aptiktų indikacijų skaičius“ pateiktas neteisingai pažymėtų indikacijų skaičius, tai yra atvejai, kai neuroninis tinklas „manė“ aptikęs užsižiebusią indikaciją, nors iš tikrųjų toje vietoje indikacija buvo užgesusi. Surinkti duomenys užfiksuoti 7 lentelėje. Joje pateikiami neuroninių tinklų aptikimo tikslumo ir klaidingai teigiamai aptiktų indikacijų duomenys, išreikšti procentais. Aptikimo tikslumas skaičiuojamas kaip visų teisingai aptiktų užsižiebusių indikacijų skaičius, padalintas iš realaus užsižiebusių indikacijų skaičiaus. Klaidingai aptiktų indikacijų skaičius gaunamas padalinus klaidingai aptiktų indikacijų skaičių iš visų aptiktų indikacijų skaičiaus.

7 lentelė. YOLOv4 tinklo aptikimo tikslumo palyginimas, keičiant rezoliuciją ir apšvietimą

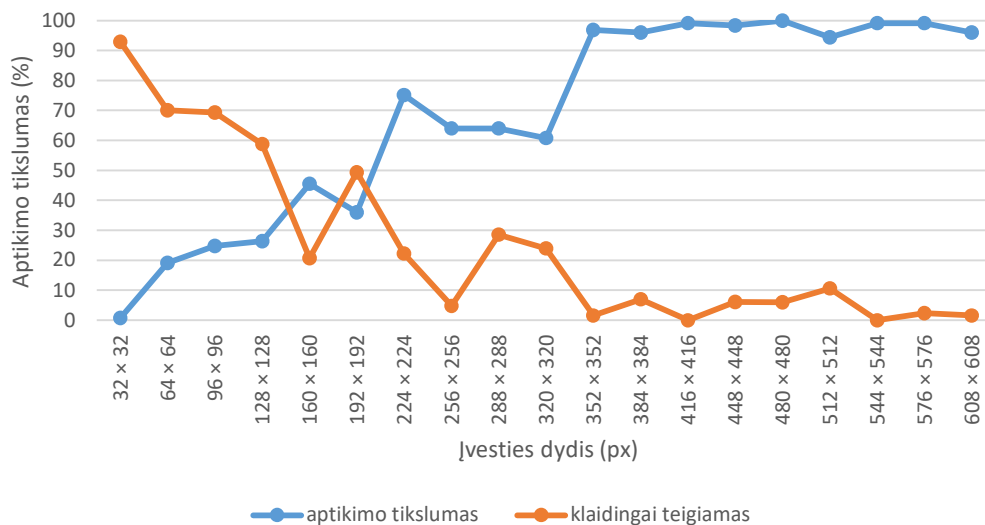
Rezoliucija, px	Apšvietimas					
	Atviras		Uždaras		Uždaras su LED	
	Aptikimo tikslumas, %	Klaidingai teigiamas, %	Aptikimo tikslumas, %	Klaidingai teigiamas, %	Aptikimo tikslumas, %	Klaidingai teigiamas, %
3840×2160	100	1,5	93,9	0,0	96	1,6
1920×1080	98	41	97,5	2,5	70,2	22,8
1280×720	61,8	39,0	95,0	2	97	17
720×480	45	9	85,0	5	62	53

Iš 7 lentelės matyti, kad didėjant vaizdo medžiagos rezoliucijai aptikimo tikslumas didėja. Didžiausias tikslumas pasiektas naudojant atvirą apšvietimą ir esant 3840×2160 rezoliucijai. Prie mažesnių rezoliucijų (480×720) didžiausias tikslumas (85 proc.) pasiektas esant uždarai elektros spintai be papildomo apšvietimo.

2.6. Neuroninio tinklo įvesties dydžio įtaka tikslumui ir greitimeikai

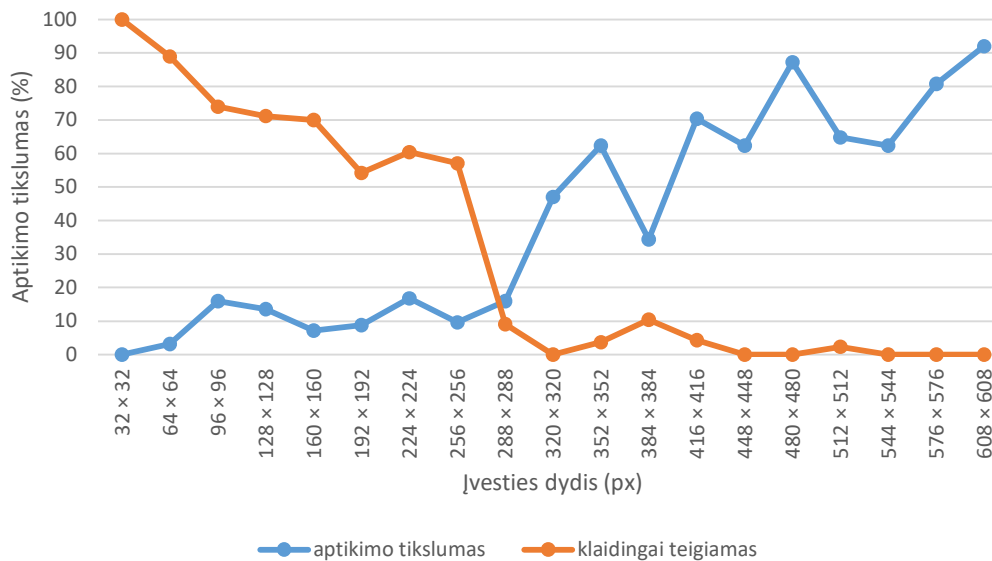
Naudojant 3840×2160 rezoliucijos nuotraukas su LED apšvietimu, apmokyti 19 *YOLOv4* ir 19 *YOLOv4Tiny* neuroniniai tinklai, kurie vienas nuo kito skyrėsi įvesties dydžiu. Jis buvo keičiamas nuo 32×32 iki 608×608 žingsniu 32. Rezultatai pateikti 29 ir 30 paveiksluose. Vidutinis apdorojimo laikas apskaičiuotas, apdorojus 100 nuotraukų ir išvedus vienos nuotraukos apdorojimo laiko aritmetinį vidurkį.

29 paveiksle pateikta aptikimo tikslumo priklausomybė nuo tinklo įvesties dydžio, naudojant *YOLOv4* neuroninį tinklą.



29 pav. Įvesties dydžio įtaka aptikimo tikslumui, naudojant *YOLOv4* tinklą [sudaryta autoriaus]

30 paveiksle pateikta aptikimo tikslumo priklausomybė nuo tinklo įvesties dydžio, naudojant *YOLOv4 Tiny* neuroninį tinklą.



30 pav. Įvesties dydžio įtaka aptikimo tikslumui, naudojant *YOLOv4 Tiny* tinklą [sudaryta autoriaus]

31 paveiksle pateikta aptikimo greitaveikos priklausomybė nuo tinklo įvesties dydžio, naudojant *YOLOv4* ir *YOLOv4 Tiny* neuroninius tinklus.



31 pav. Įvesties dydžio įtaka aptikimo greitaveikai [sudaryta autoriaus]

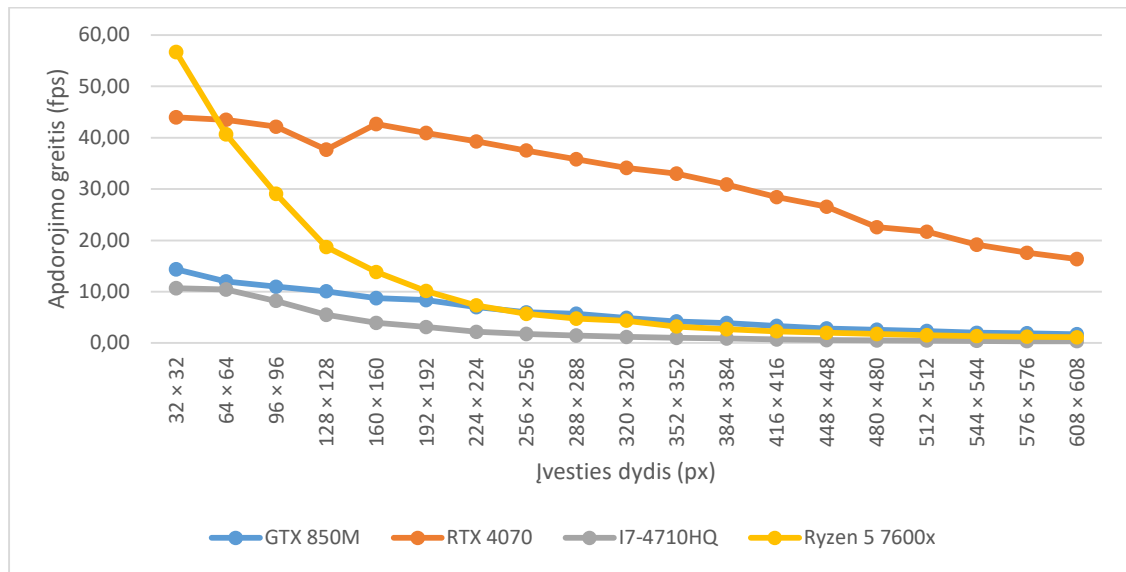
Iš 29 paveikslo matyti, kad įvesties dydžiui pasiekus 352×352 vertę, gautas 96,6 proc. aptikimo tikslumas ir 1,6 proc. klaidingų aptikimų tikslumas. Toliau didėjant įvesties dydžiui aptikimo tikslumas keičiasi tarp 94 ir 100 procentų, o neteisingi aptikimai kinta 0–10,6 proc. ribose. 31 paveiksle matyti, kad didėjant įvesties dydžiui vienos nuotraukos apdorojimo laikas didėja nuo 70 ms, esant įvesties dydžiui 32×32 , iki 595 ms, esant įvesties dydžiui 608×608 . Kai įvesties dydis siekia 352×352 , vidutinis apdorojimo laikas – 238 ms. Nustatyta, kad optimaliausias laiko ir tikslumo atžvilgiu įvesties dydis – 352×352 .

Palyginus *YOLOv4* ir *YOLOv4 Tiny* neuroninius tinklus, pastebėta, kad *YOLOv4* turi didesnę aptikimo tikslumą (99,2 proc.) nei *YOLOv4 Tiny* (92 proc.), tačiau *YOLOv4 Tiny* yra greitesnis ir pasiekia 2,5 karto didesnę greitaveiką, esant panašiam aptikimo tikslumui. Tai yra lyginama *YOLOv4* aptikimo greitaveikai ir taiklumas esant 352×352 įvesties dydžiui su *YOLOv4 Tiny* 608×608 įvesties dydžiu. Iš to išeina, kad turimomis eksperimento sąlygomis yra geriau naudoti *YOLOv4 Tiny* neuroninį tinklą dėl jo 2,5 didesnės greitaveikos.

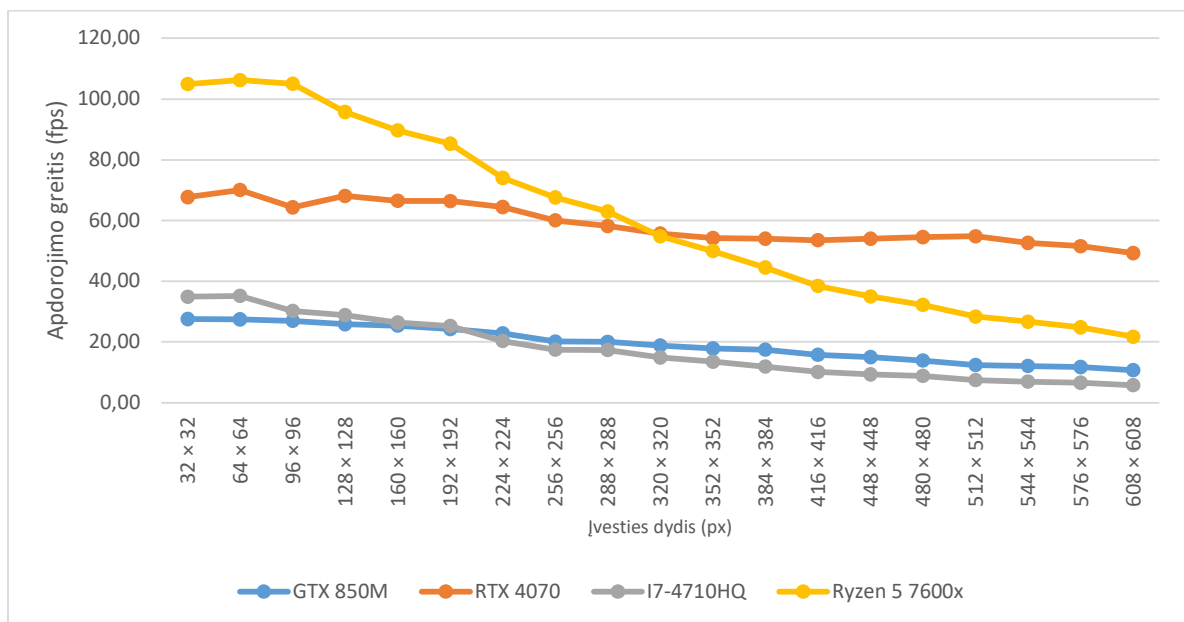
2.6.1. Vaizdo plokštės įtaka neuroninio tinklo greitaveikai

Kadangi skyriuje 2.6 nebuvo pasiektas duomenų apdorojimas realiuoju laiku, toliau bus tiriama skirtingų GPU ir CPU komponentų įtaka *YOLOv4* ir *YOLOv4 Tiny* vaizdo aptikimo greitaveikai.

Tiriami komponentai: CPU I7-4710hq ir „Ryzen“ 7 7600x. GPU „nvidia“ GTX 850m, „nvidia“ RTX 4070. Tyrimo rezultatai pateikiami 32 ir 33 paveiksluose.



32 pav. Kompiuterinės įrangos įtaka aptikimo greitimeikai naudojant *YOLOv4* neuroninį tinklą [sudaryta autoriaus]



33 pav. Kompiuterinės įrangos įtaka aptikimo greitimeikai naudojant *YOLOv4 Tiny* neuroninį tinklą [sudaryta autoriaus]

Iš gautų duomenų matome (žr. 32 pav.), kad realaus laiko vaizdo atpažinimas, naudojant *YOLOv4* tinklą (33 kadrus per sekundę esant 352×352 įvesties dydžiui), pasiekiamas tik naudojant vaizdo plokštę RTX 4070. 33 paveiksle matyti, kad realaus laiko vaizdo atpažinimas naudojant neuroninį tinklą *YOLOv4 Tiny* pasiekiamas ne tik naudojant vaizdo plokštę RTX 4070, bet ir naudojant procesorių „Ryzen“ 5 7600x (vaizdo plokštė pasiekė 49 kadrų per sekundę apdorojimo greitimeiką, o procesorius – 22 kadrus per sekundę). Tai reiškia, kad ekonominiu požiūriu, diegiant tokią sistemą, galima sutaupyti lėšų nusipirkus tik procesorių, o ne vaizdo plokštę (RTX 4070 darbo rašymo metu kainuoja 700 EUR).

3. Loginių sekų prognozavimas

Loginių sekų prognozavimas yra esminis procesas, padedantis suprasti ir numatyti įvairių sistemų, įskaitant mechatronines, elgseną ilginiui. Šis procesas apima duomenų analizę, modeliavimą ir prognozavimą remiantis loginiais (arba diskrečiais) duomenimis, gaunamais iš mechatroninių sistemų komponentų, tokių kaip jutikliai, valdikliai, mechanizmai ir pan. Loginių sekų prognozavimas gali būti pritaikytas mechatroninių sistemų gedimų diagnozavimui, kad būtų galima nustatyti potencialius gedimus prieš jų įvykimą, leidžiant laiku imtis prevencinių priemonių, sumažinti neigiamą poveikį gamybos procesui ir užtikrinti didesnę sistemos veikimo patikimumą [39].

Loginių sekų prognozavimo taikymas mechatroninių sistemų gedimų diagnozavimui apima šiuos etapus:

1. Duomenų rinkimas. Renkami loginiai duomenys iš mechatroninės sistemos komponentų, pavyzdžiui, jutiklių, valdiklių ir mechanizmų. Duomenys gali apimti sistemos būseną, veikimo parametrus, veikimo istoriją ir kitą informaciją, padedančią prognozuoti gedimus [39].
2. Duomenų analizė. Statistiniais metodais, laiko eilučių analize ar mašininio mokymosi metodais analizuojami surinkti loginiai duomenys, nustatant svarbius modelius ar tendencijas, rodančias galimus gedimus [39].
3. Modeliavimas. Remiantis surinktais duomenimis ir atlikta analize, sukuriami modeliai, padedantys prognozuoti mechatroninės sistemos elgseną laikui bėgant ir nustatyti potencialius gedimus [39].
4. Prognozavimas. Taikomi sukurti modeliai loginių sekų raidai prognozuoti ir galimiems mechatroninės sistemos gedimams nustatyti [39].
5. Pritaikymas. Pagal prognozavimo rezultatus priimami sprendimai dėl tolesnių veiksmų, tokių kaip priežiūros planavimas, atsarginių įrenginių įjungimas, sistemos optimizavimas ar komponentų keitimas. Be to, atsižvelgiant į prognozuojamus gedimus, galima parengti techninio personalo mokymo programas [39].

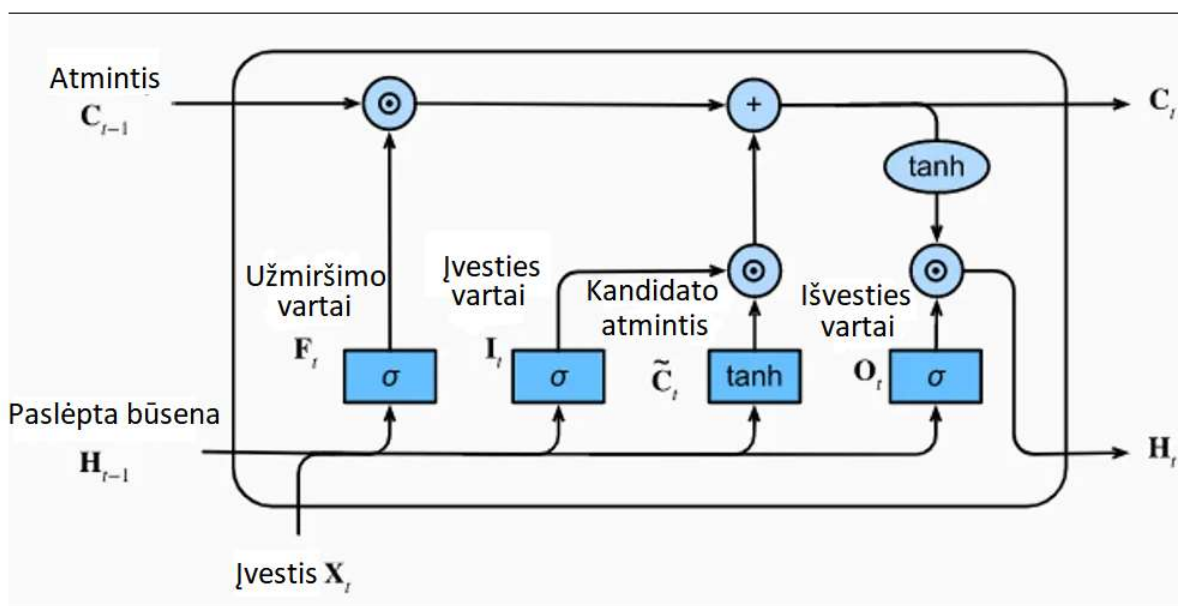
LSTM (ilgalaikių atminties tarpinių sluoksnių) neuroniniai tinklai yra specialieji rekurentinio neuroninio tinklo (RNN) architektūra, sukurta siekiant geriau išmokyti priklausomybes tarp ilgalaikių sekvencijų duomenų. LSTM tinklai buvo išrasti 1997 m. Sepp Hochreiter ir Jürgen Schmidhuber, kadangi jie puikiai tinka sprendžiant sekvencijos į sekvenciją, laiko sekų prognozavimo ir teksto generavimo problemas [39].

LSTM tinklų efektyvumas yra susijęs su jų gebėjimu ilgą laiką saugoti informaciją, dėl ko jie gali atpažinti ilgalaikes priklausomybes tarp duomenų. Be to, jie gali apdoroti didelius kiekius duomenų ir dirbti su didelio ilgio sekų informacija, kas yra svarbu mechatroninių sistemų gedimų prognozavime.

Pagrindinis skirtumas tarp LSTM tinklų ir paprastų RNN yra tas, kad LSTM naudoja specialias atminties ląsteles, kurios leidžia jiems išlaikyti ilgalaikę informaciją ir atlikti sudėtingesnes sąveikas tarp įvesties ir išvesties duomenų [39].

LSTM tinklai veikia su trimis pagrindiniais elementais (žr. 34 pav.).

1. Įvesties vartai (angl. *input gate*). Šis mazgas kontroliuoja, kokia informacija turi būti įtraukta į atminties ląstelę. Įgyvendinimo mazgas naudoja sigmoidinę aktyvacijos funkciją, kad nuspręstų, kuri informacija yra svarbi ir turi būti saugoma atminties ląstelėje [39].
2. Pamiršimo vartai (angl. *forget gate*). Šis mazgas nustato, kuri informacija turėtų būti pašalinta iš atminties ląstelės, siekiant išlaikyti tik svarbiausią informaciją. Pamiršimo mazgas taip pat naudoja sigmoidinę aktyvacijos funkciją, kad nuspręstų, kuri informacija yra pasenusi ir gali būti pašalinta [39].
3. Išvesties vartai (angl. *output gate*). Galiausiai išvesties mazgas nustato, kokia informacija turėtų būti perduota į sekančią atminties ląstelę arba pateikta kaip išvestis. Išvesties mazgas naudoja sigmoidinę aktyvacijos funkciją, kad nuspręstų, kuri informacija yra svarbi ir turėtų būti perduota tolesniam apdorojimui [39].



34 pav. Neuroninio tinklo, su ilgalaikė atmintimi, ląstelės modelis [39]

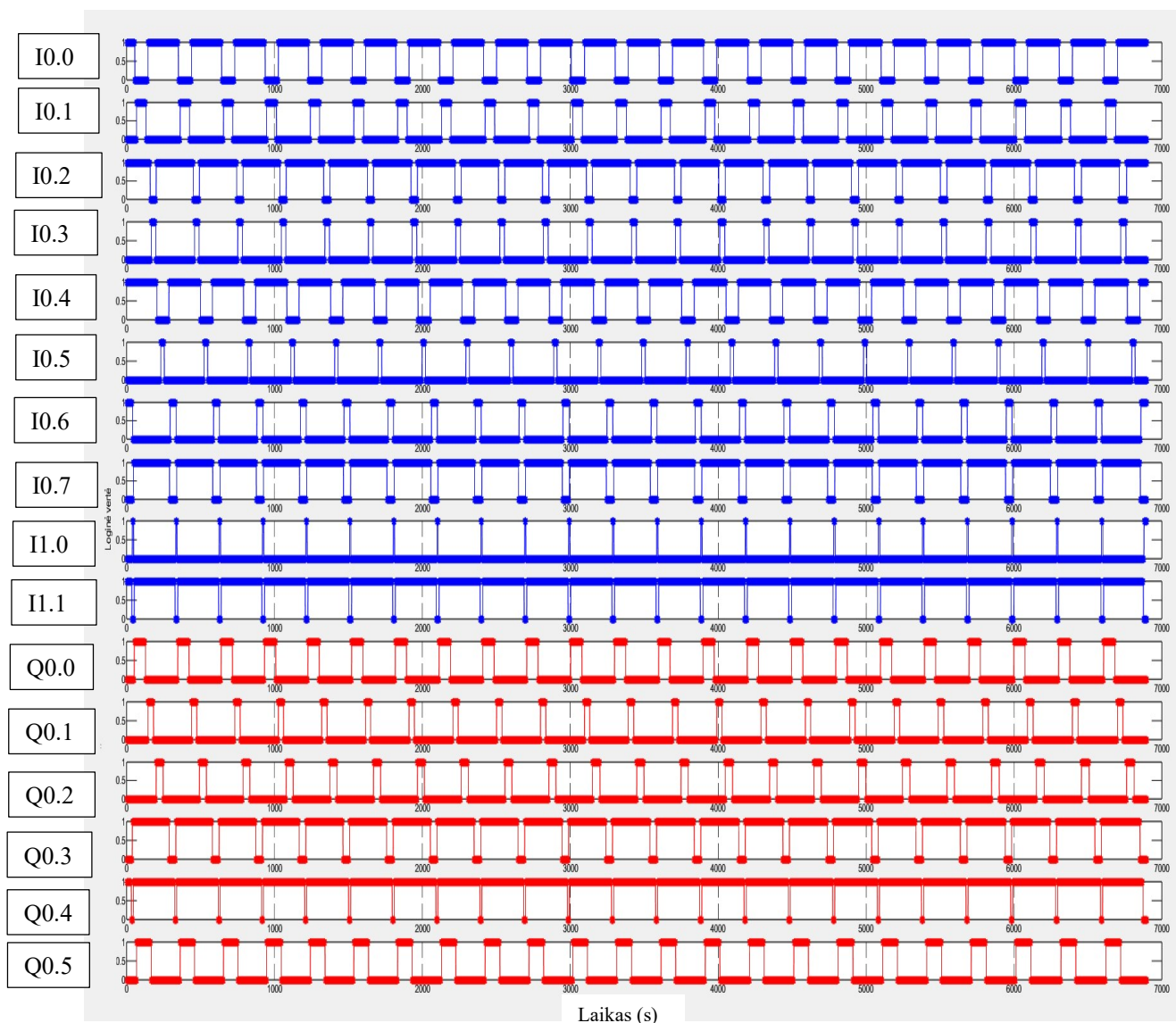
Šios specialiosios atminties ląstelės leidžia LSTM tinklams efektyviai apmokyti ilgas sekvenčias ir išlaikyti ilgalaikę informaciją, kas yra svarbu daugeliui uždavinių, tokių kaip teksto generavimas, vertimas, laiko eilučių prognozavimas ir kalbėjimas. Dėl šios priežasties LSTM tinklai tapo populiariu pasirinkimu sprendžiant tokio pobūdžio problemas [39].

LSTM tinklai taip pat turi keletą modifikacijų ir variacijų, tokias kaip paleidžiamieji (angl. *Peephole*) LSTM, kurie leidžia mazgams „žvilgtelėti“ į atminties ląstelių būseną, kai sprendžiama, ar atnaujinti informaciją. Kitas populiarus variantas yra GRU (žaliosios ląstelės atnaujinimo) tinklai, kurie yra supaprastinta LSTM architektūra, sukurta siekiant sumažinti skaičiavimų sudėtingumą, tačiau išlaikant daugumą ilgalaikės atminties savybių [40].

Išanalizavus prognozavimo rezultatus, gaunamą informaciją galima naudoti siekiant optimizuoti mechatroninių sistemų priežiūrą, planuoti prevencines priemones ir gerinti gamybos procesų efektyvumą. Tai padeda suprasti sistemos veikimo principus, taip galimai atskleidžiant naujus sprendimus, kaip sumažinti gedimų tikimybę ir pagerinti mechatroninių sistemų patikimumą.

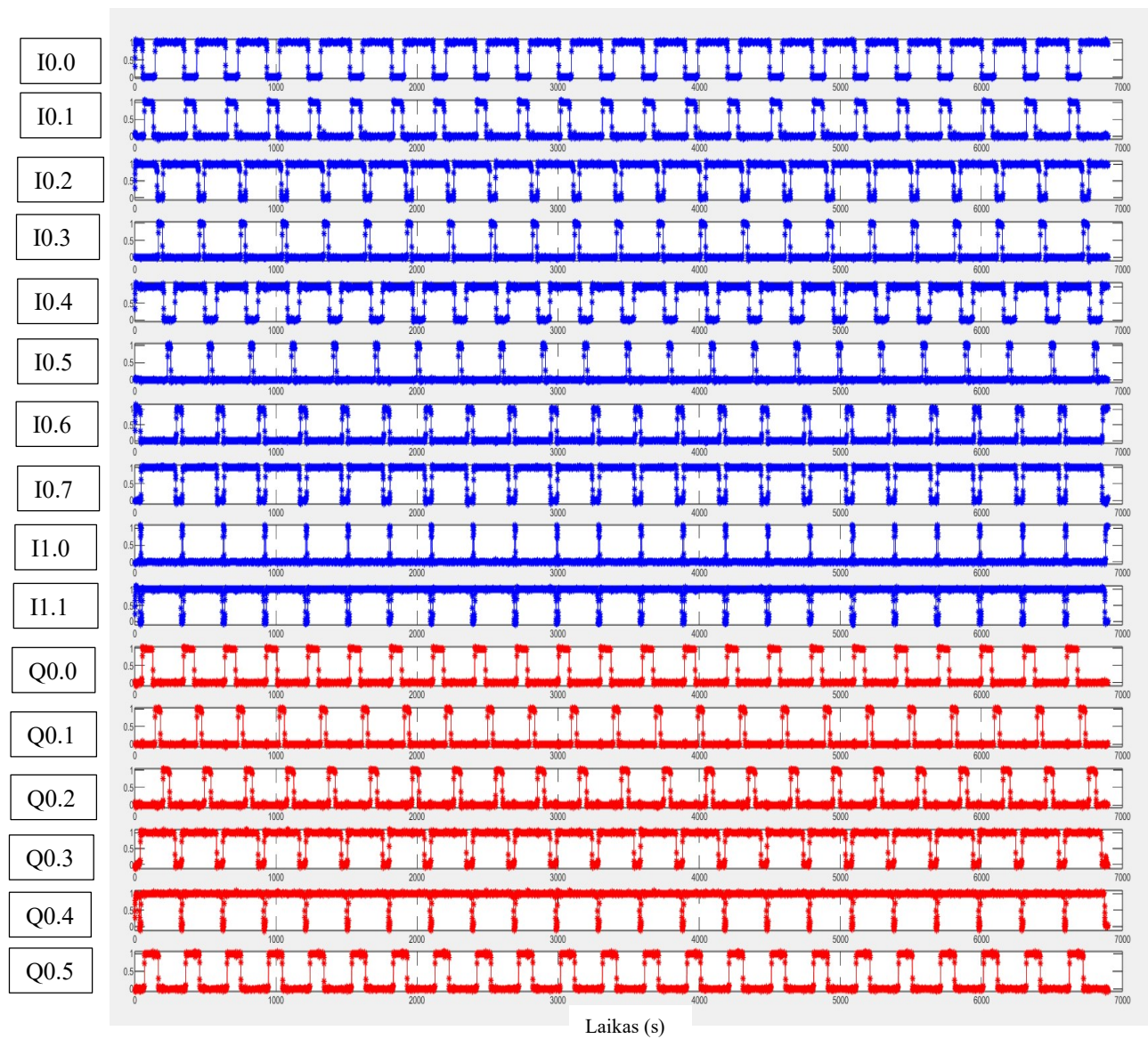
3.1. LSTM tinklo apmokymas

Sistemai apmokyti nufilmuotas 4 minučių trukmės gerai veikiančios sistemos vaizdas. Kadangi LSTM nagrinėja ne tik esamas vertes, bet ir jų kitimą laike, apmokymui reikalingas daugiau nei vienas judesio ciklas. Šiuo atveju sistemai apmokyti užfiksuoti dvidešimt trys teisingi judesių ciklai. Filmuota medžiaga apdorota naudojantis 2-jame šio projekto skyriuje apmokytu neuroniniu tinklu ir sudarius įėjimų ir išėjimų masyvą. 35 paveiksle pateikti apmokymui naudoti duomenys. Mėlynai pažymėti sistemos įėjimai, o raudonai – išėjimai.



35 pav. Duomenys, naudoti LSTM tinklui apmokyti [sudaryta autoriaus]

LSTM tinklui apmokyti visi užfiksuoti įėjimai ir išėjimai vienu metu yra paduodami į tinklą, kaip jo įėjimas ir išėjimas, tai yra tinklas prognozuoja pats save. 36 paveiksle pavaizduotas gautas apmokyto tinklo išėjimas, kai į tinklą yra paduoti 35 paveiksle pavaizduoti duomenys.



36 pav. Apmokyto LSTM tinklo sugeneruoti išėjimai [sudaryta autoriaus]

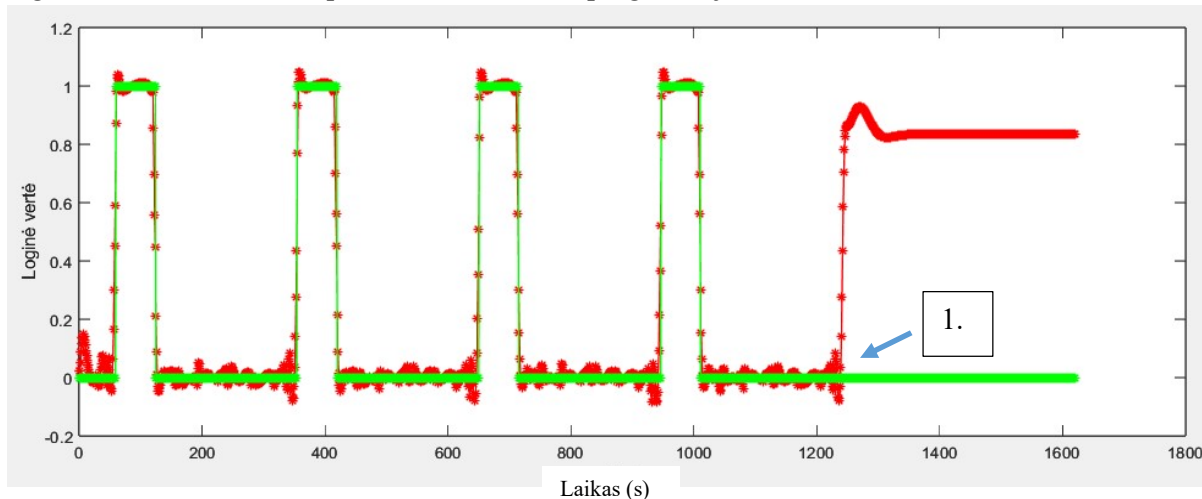
Iš 34 paveikslo matyti, kad LSTM tinklo išduodamos vertės yra „fuzzy“, tai yra, nors sistemos įėjimai yra loginiai „0“ ir „1“, sistema išduoda apytiksles vertes tarp 0 ir 1.

3.2. LSTM tinklo testavimas

LSTM tinklui testuoti nufilmuoti 5 vaizdo įrašai. Pirmajame bandyme buvo atjungtas prie valdiklio įėjimo adreso I0.1 prijungtas jutiklis, antrajame bandyme – prie valdiklio įėjimo adreso I0.4 prijungtas jutiklis, trečiajame bandyme atjungtas prie valdiklio išėjimo Q0.0 prijungtas vožtuvas, ketvirtajame bandyme atjungtas prie valdiklio išėjimo Q0.2 prijungtas vožtuvas. Penktajame vaizdo įrašė vyksta oro slėgio kritimas, t. y. išjungus kompresorių, sistemos oro slėgis krenta iki tol, kol sistema pradeda strigti.

3.2.1. Jutiklio gedimo aptikimas

Paveiksle yra pateiktas pirmasis testuotas gedimas. 37 paveiksle žalia spalva pavaizduota jutiklio loginė būseną, o raudona spalva – LSTM tinklo prognozuojama vertė.

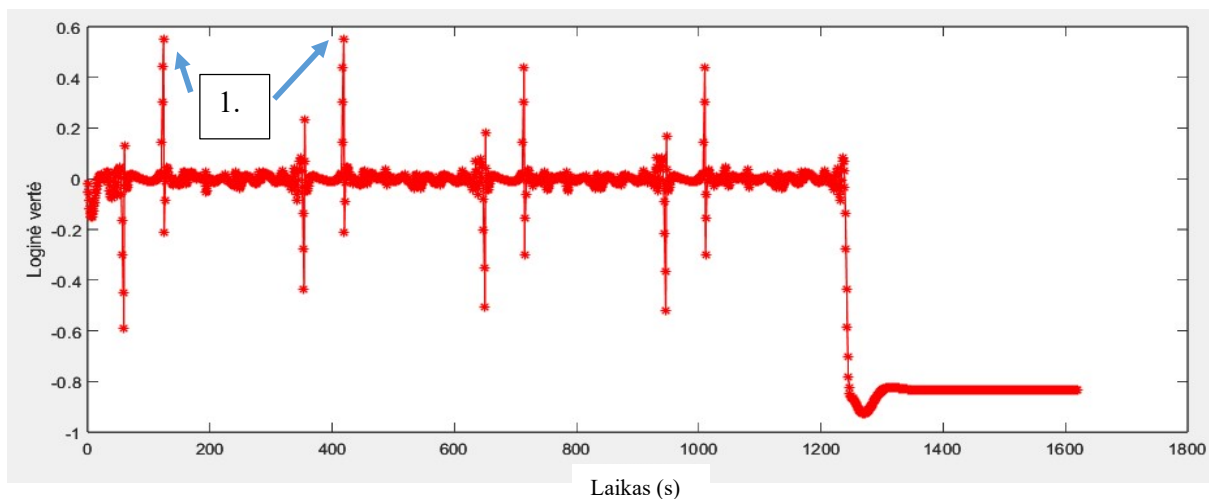


37 pav. Sistemos jutiklio gedimas IO.1. „1“ pažymėtas jutiklio gedimo momentas (žalia spalva vaizduojama reali vertė, o raudona – prognozuojama) [sudaryta autoriaus]

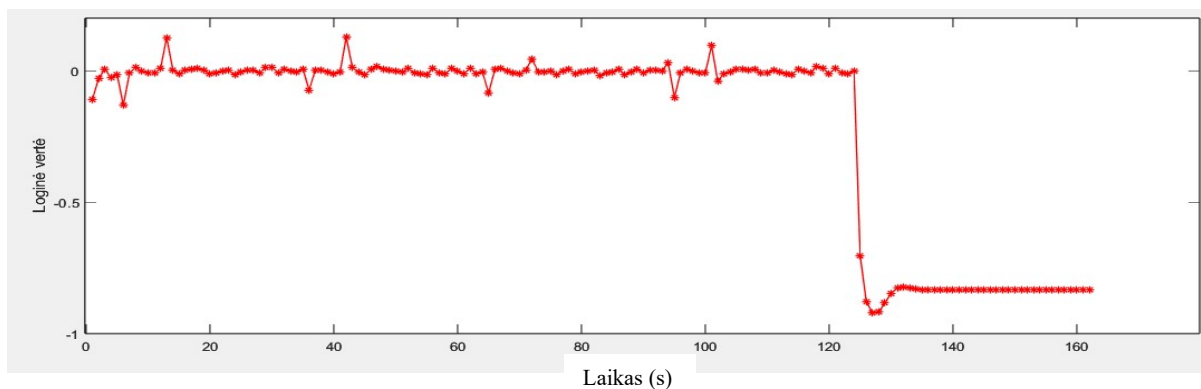
Iš 37 paveikslo matyti, kad įvykus jutiklio gedimui (paveiksle pažymėta „1“), LSTM tinklas prognozuoja, kad jutiklio vertė neturėtų būti loginis „0“ (prognozuojama vertė 0,43). Nors vizualiai gedimas ir yra matomas, bet to neužtenka, kad sistema automatiškai jį aptiktų. 37 paveiksle pateiktas skirtumas tarp esamų ir tinklo prognozuojamų verčių.

Iš 38 paveikslo „1“ pažymėtos vietos matyti, kad automatiniam klaidos aptikimui neužtenka nustatyti padidėjusį skirtumą tarp prognozuojamos ir realios verčių, nes LSTM tinklo prognozuojamos vertės kinta lėčiau nei reali vertė, t. y. jutiklio vertė iš loginio „0“ į loginį „1“ pereina per vieną užfiksuotą kadrą, kai tuo tarpu prognozuojamos vertės kitimas užtrunka apie 2–4 vaizdo kadrus. Pereinamojo proceso netikslumui panaikinti panaudotas paklaidos verčių vidurkio skaičiavimas. Tai yra sistemos paklaidos yra sudedamos kas 10 kadrų ir padalinamos iš 10, taip sumažinant pereinamojo proceso daroma paklaidą. Gautas rezultatas pateiktas 39 paveiksle.

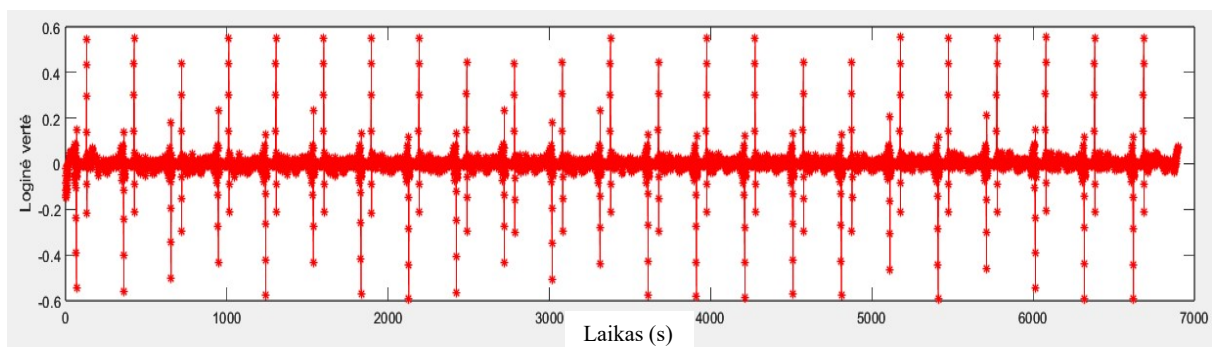
Toliau norint aptikti gedimo momentą, reikia apibrėžti svertinę vertę, kurią viršijus bus priimta, kad įvyko gedimas. Tai atliekama analizuojant pradinį apmokymo duomenis ir iš jų išskaičiuojant leistiną paklaidą, kuri yra gaunama prognozuojant sistemos būseną LSTM tinklu. 40, 41 ir 42 paveiksluose pateikti grafikai, pagal kuriuos buvo apskaičiuota leistina paklaida.



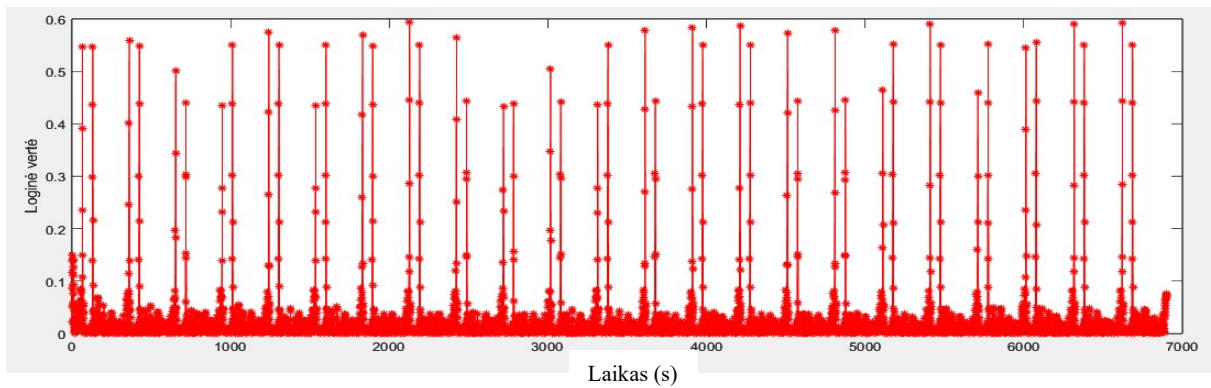
38 pav. Skirtumas tarp prognozuojamos ir esamos verčių [sudaryta autoriaus]



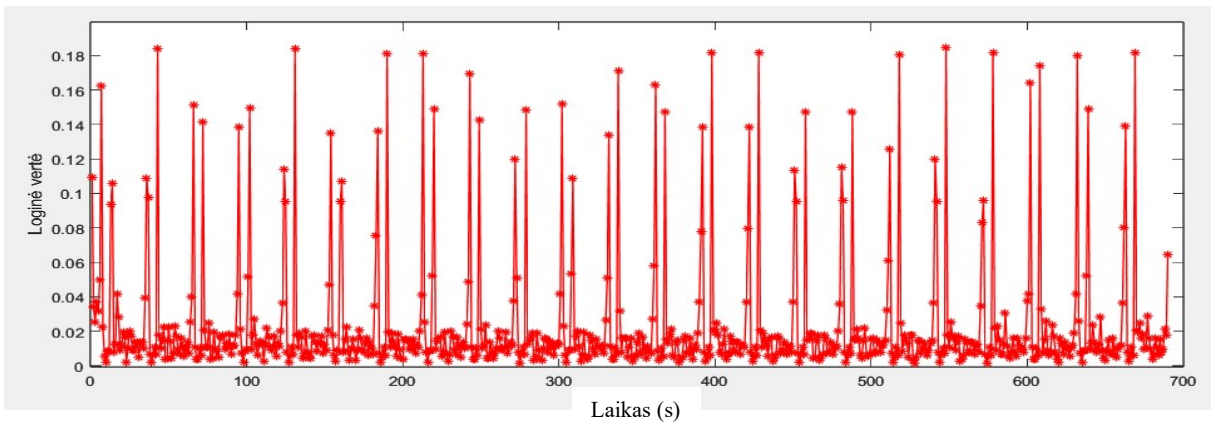
39 pav. Vidutinis skirtumas tarp prognozuojamos ir esamos verčių [sudaryta autoriaus]



40 pav. Skirtumas tarp LSTM prognozuojamų ir realių verčių [sudaryta autoriaus]

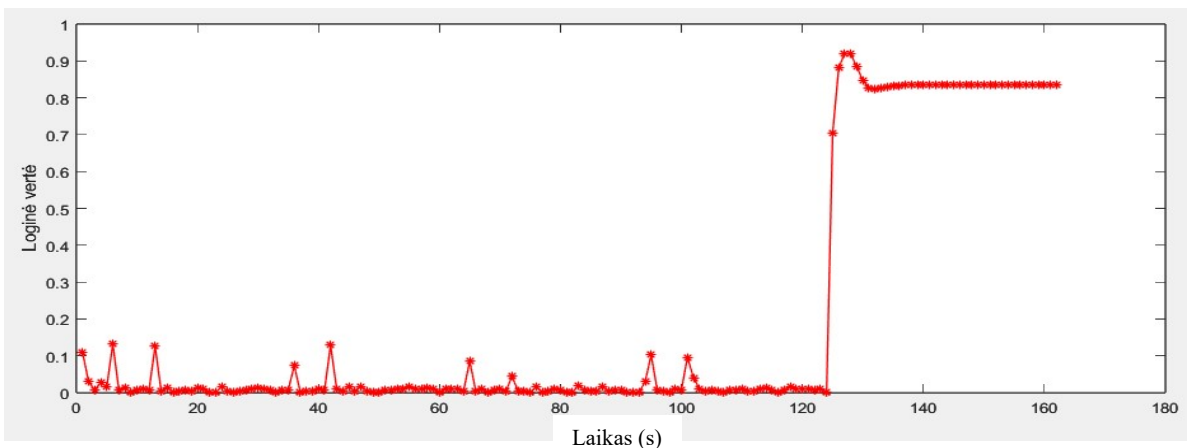


41 pav. Skirtumas tarp LSTM prognozuojamų ir realių verčių (absoliutinės vertės) [sudaryta autoriaus]



42 pav. Vidutinis skirtumas tarp LSTM prognozuojamų ir realių verčių [sudaryta autoriaus]

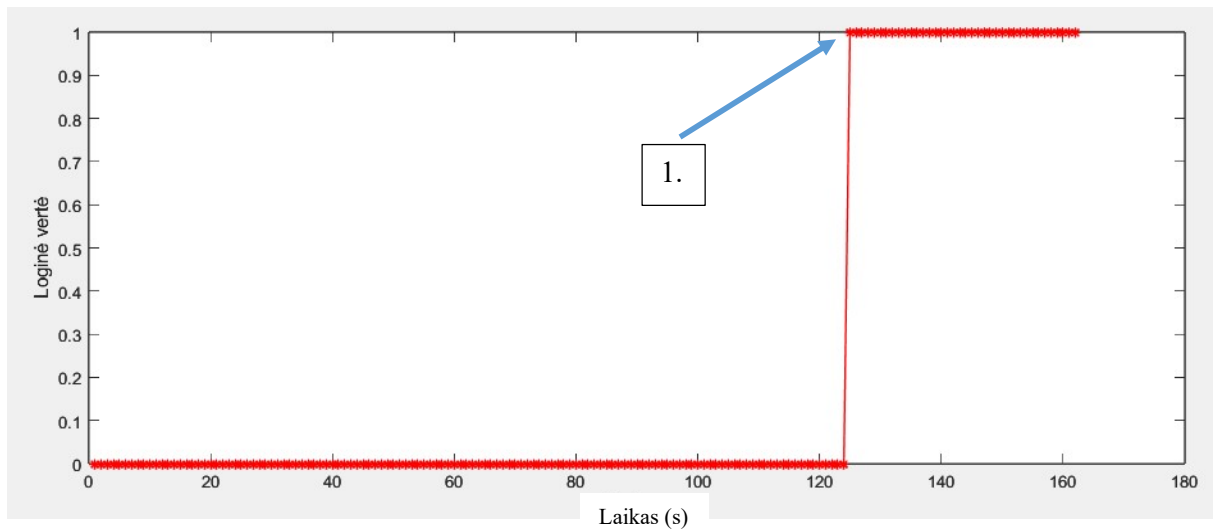
40 paveiksle pavaizduotas skirtumas tarp LSTM praruožuojamos ir realios verčių. 41 paveiksle yra pavaizduotos absoliučiosios skirtumo vertės. 42 paveiksle (taip pat kaip ir 39 pav.) yra apskaičiuojami paklaidų verčių vidurkiai 10 kadrų intervale. Iš 41 paveikslo nustatyta, kad leistina paklaida arba klaidos aptikimo svertinė riba negali būti mažesnė nei 0,181. Svertinė ribos vertė padauginama iš atsargos koeficiento 1,5 ir gaunama naujoji vertė 0,2715. Kadangi leistina paklaida buvo apskaičiuota kaip absoliuti vertė, 39 paveiksle pavaizduotos vertės yra taip pat perskaičiuojamos kaip absoliučios vertės, rezultatai pateikti 43 paveiksle.



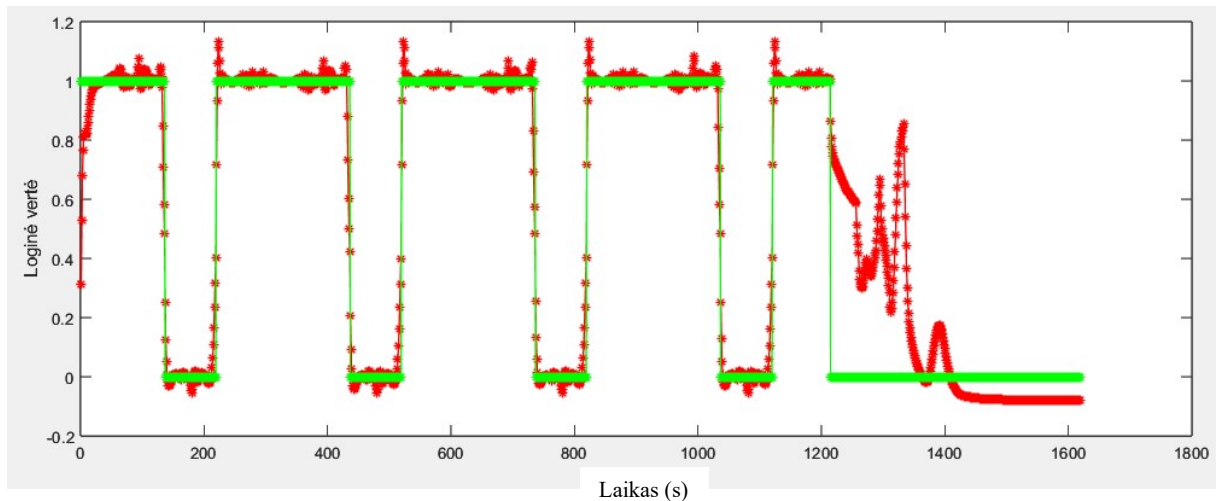
43 pav. Vidutinis skirtumas tarp prognozuojamos ir esamos verčių (absoliučios vertės) [sudaryta autoriaus]

Pritaikius 42 paveiksle apskaičiuotą svertinę vertę, 43 paveiksle pateikti duomenys yra perskaičiuojami: visoms vertėms, viršijančioms svertinės vertės priskiriant „1“, o neviršijančioms – „0“.

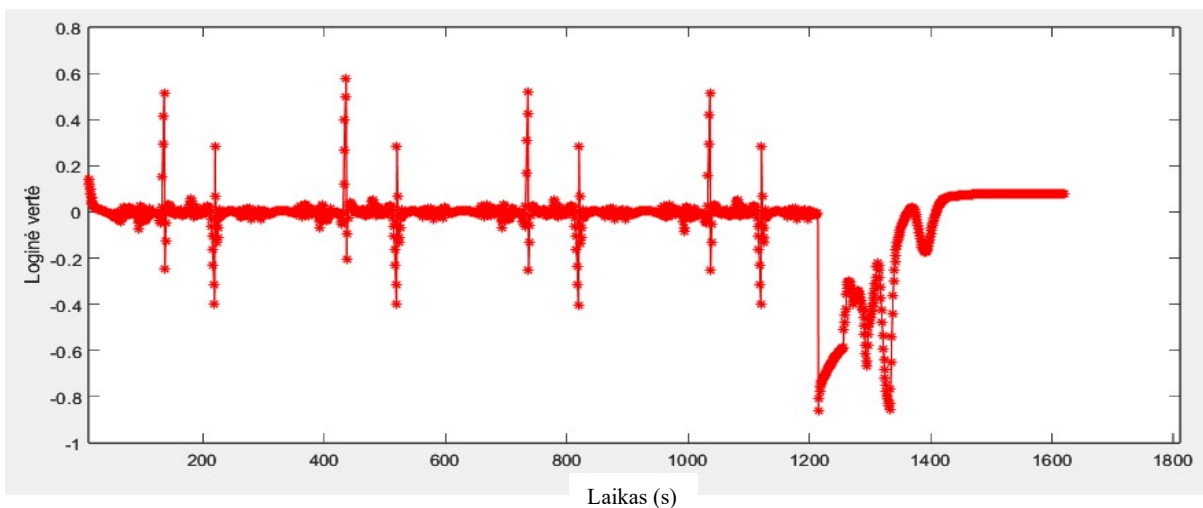
44 paveiksle atvaizduotas gedimo aptikimas, t. y. loginė vertė (pažymėta „1“), kuri pereidama iš „0“ į „1“ nusako sistemos gedimo momentą. Kol loginė vertė yra lygi „1“ laikoma, kad sistemoje yra gedimas. Norint aptikti visus galimus sistemos jutiklių gedimus, skaičiavimai atlikti su visais sistemoje esančiais jutikliais, t. y. kiekvienam jutikliui apskaičiuotos svertinės vertės ir po to jos pritaikytos gedimui aptikti. 45–52 paveiksluose yra pateiktas grafinis skaičiavimo vaizdavimas aptinkant jutiklio, prijungto prie valdiklio įėjimo adresu I0.4, gedimą.



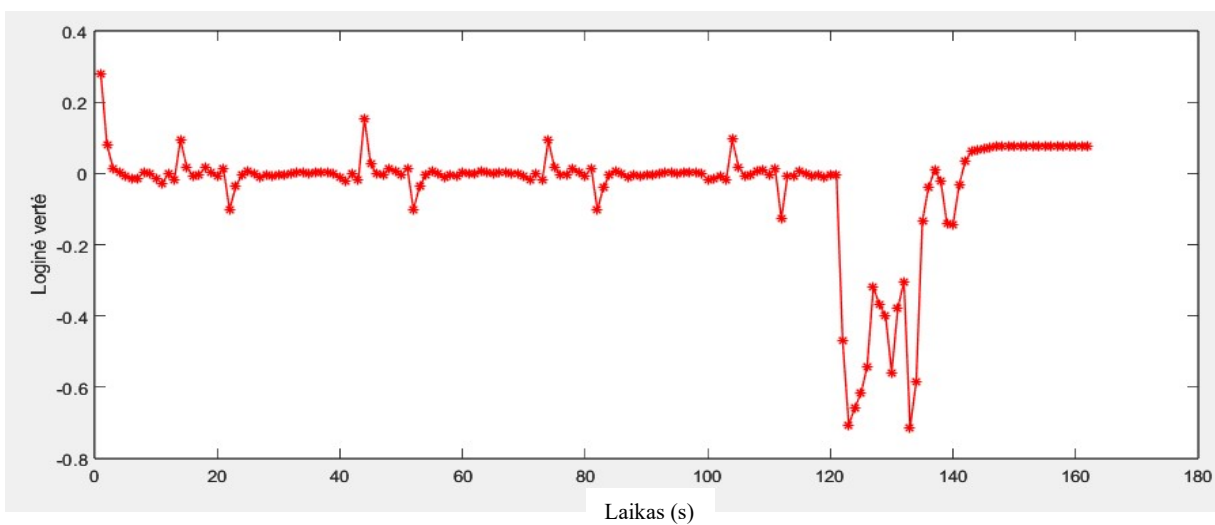
44 pav. Teisingai aptiktas jutiklio, kurio adresas I0.1, gedimas [sudaryta autoriaus]



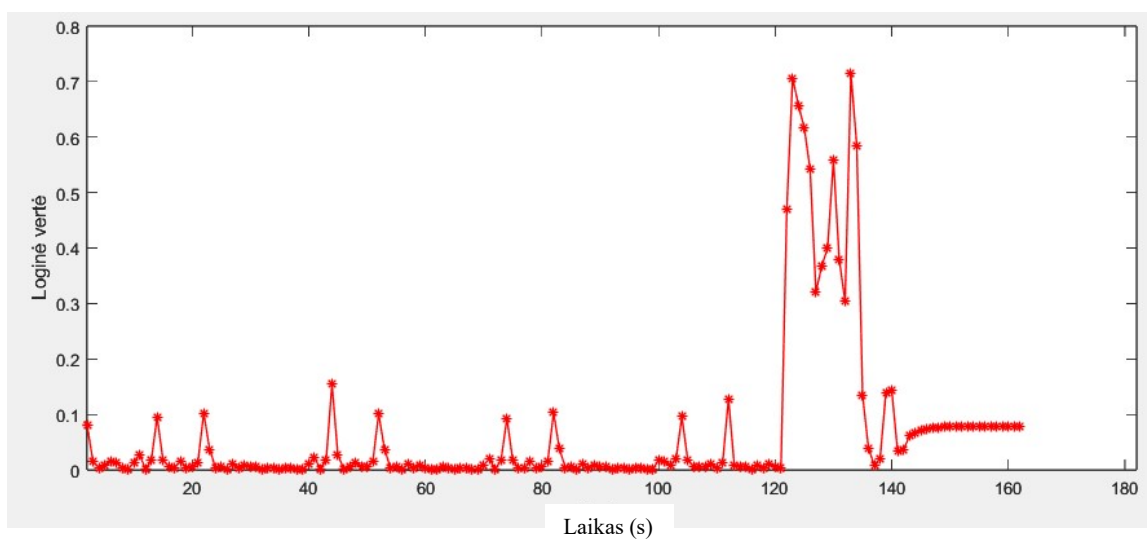
45 pav. Jutiklio I0.4 gedimo aptikimo grafinis vaizdavimas (žalia spalva vaizduojama reali vertė, raudona spalva – prognozuojama) [sudaryta autoriaus]



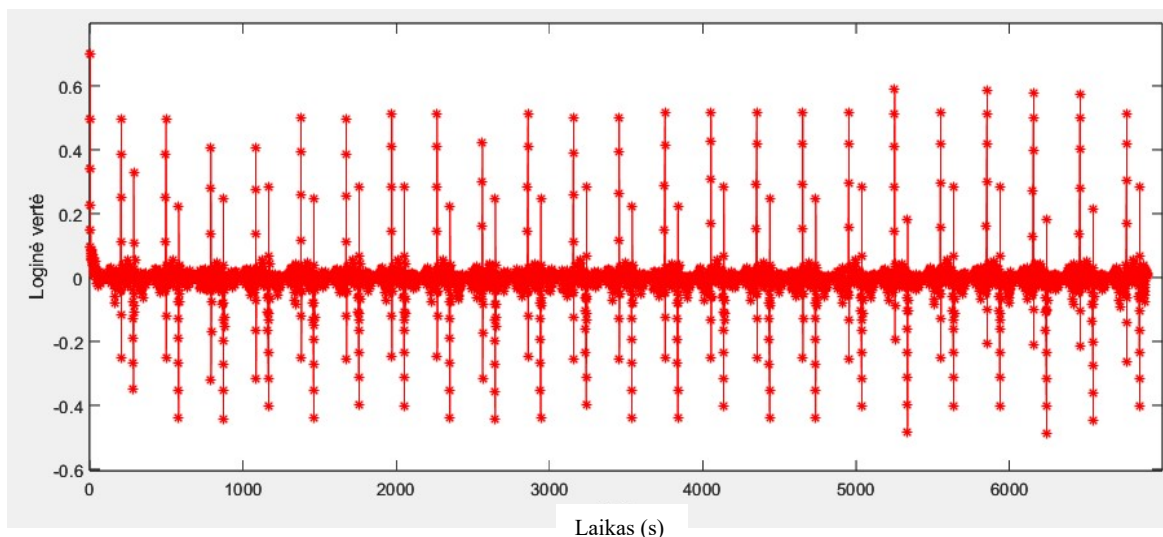
46 pav. Skirtumas tarp prognozuojamos ir esamos verčių [sudaryta autoriaus]



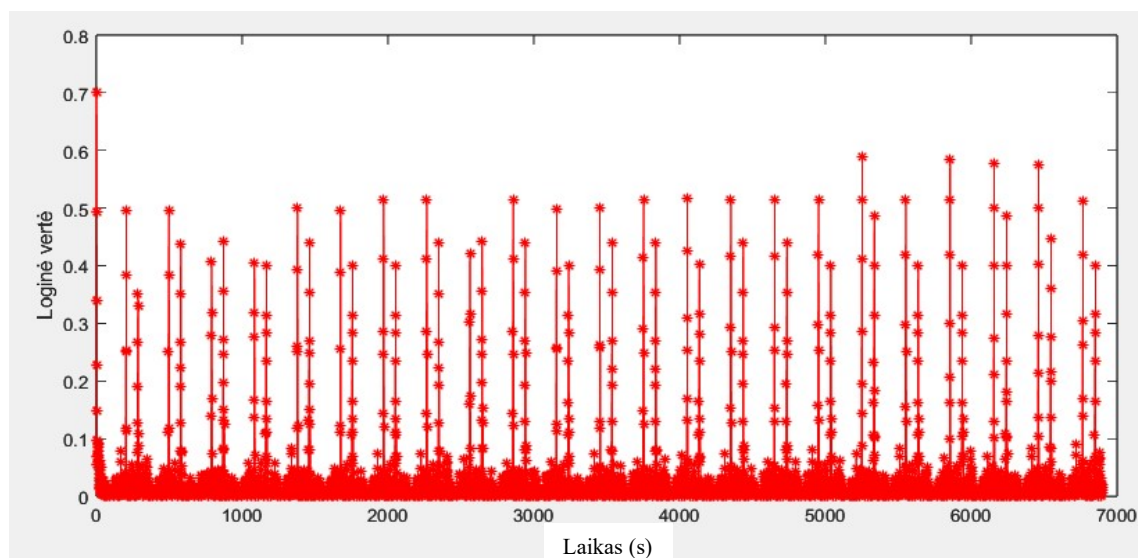
47 pav. Vidutinis skirtumas tarp prognozuojamos ir esamos verčių [sudaryta autoriaus]



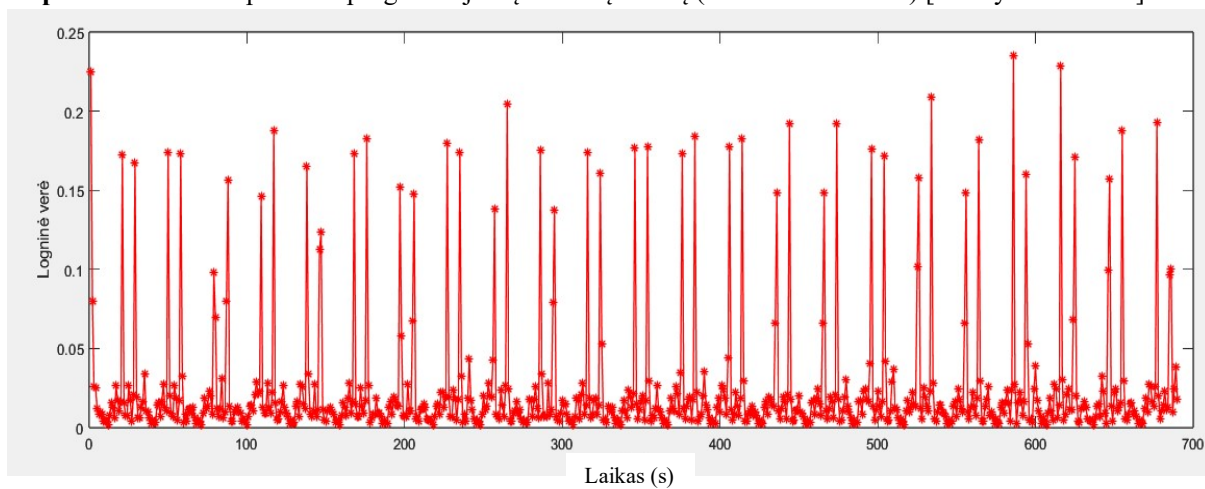
48 pav. Vidutinis skirtumas tarp prognozuojamos ir esamos verčių (absoliučiosios vertės) [sudaryta autoriaus]



49 pav. Skirtumas tarp LSTM prognozuojamų ir realių verčių [sudaryta autoriaus]

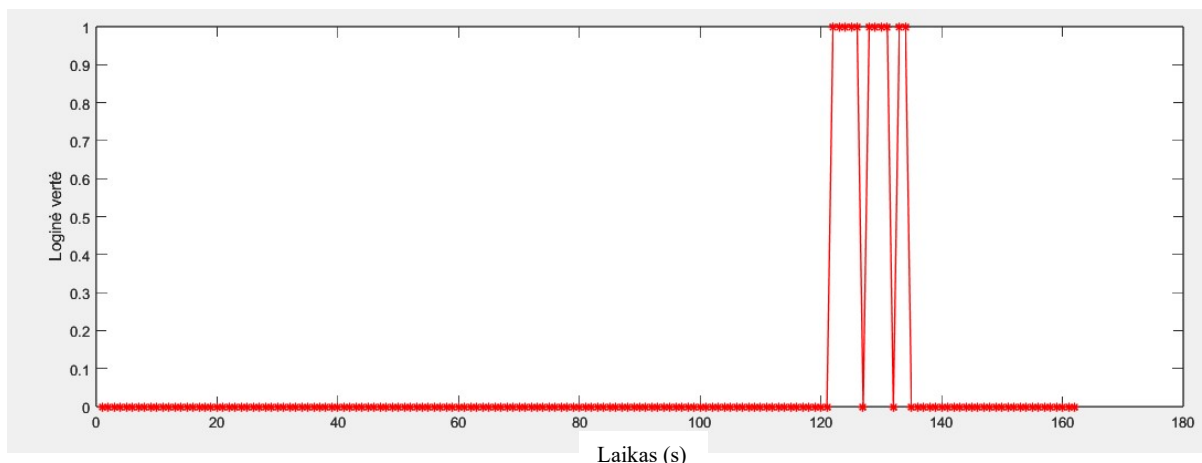


50 pav. Skirtumas tarp LSTM prognozuojamų ir realių verčių (absoliutinės vertės) [sudaryta autoriaus]



51 pav. Vidutinis skirtumas tarp LSTM prognozuojamų ir realių verčių [sudaryta autoriaus]

51 paveiksle pavaizduoti duomenys parodo, kad svertinė klaidos aptikimo vertė, jutiklio, prijungto prie valdiklio įėjimo adreso I0.4, negali būti mažesnė nei 0,235. Svertinė ribos vertė dauginama iš atsargos koeficiento 1,5 ir gaunama naujoji vertė 0,3525. 52 paveiksle pateiktas jutiklio gedimo aptikimas, naudojant apskaičiuotą gedimo svertinę vertę.



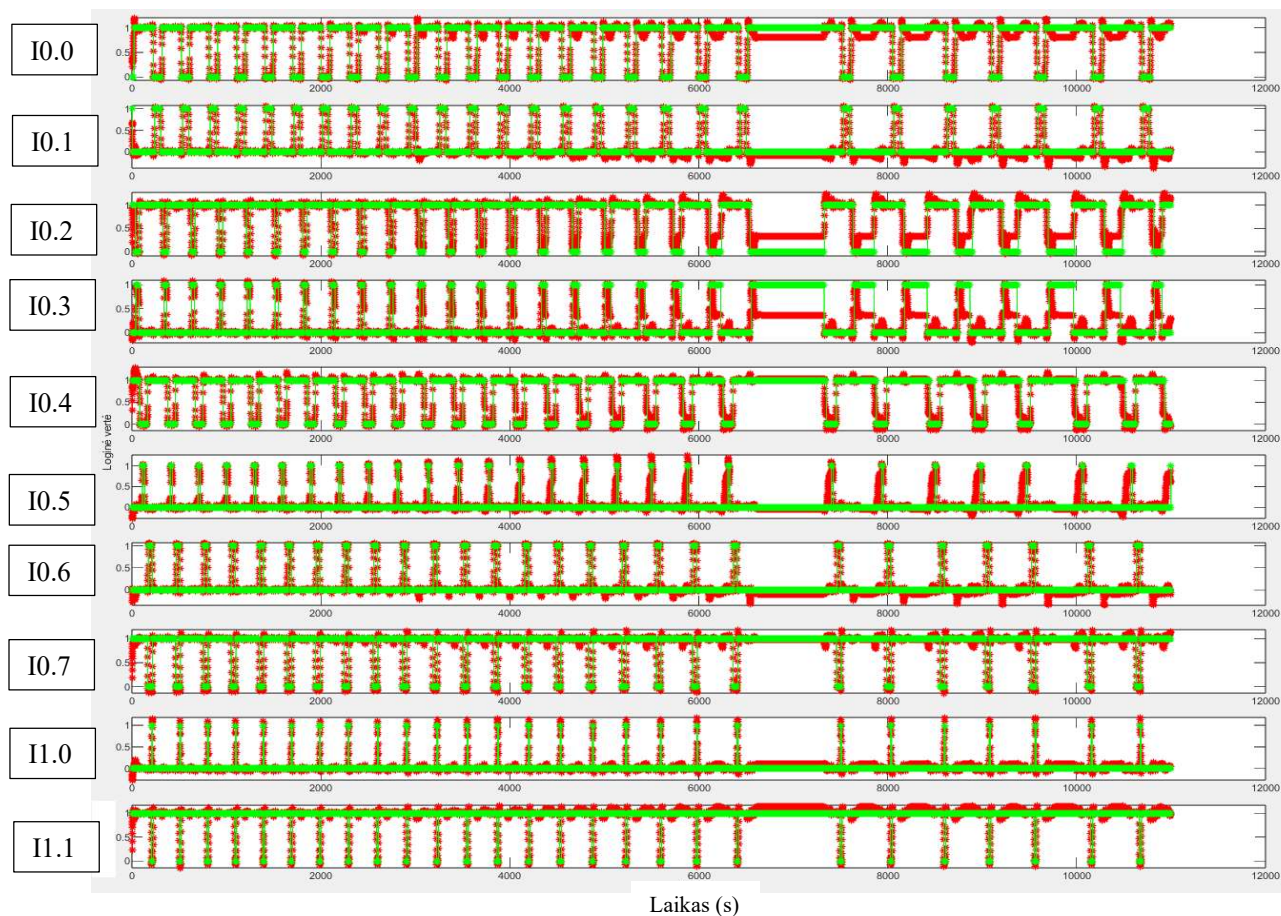
52 pav. Jutiklio, prijungto prie valdiklio įėjimo adreso I0.4, gedimo aptikimas [sudaryta autoriaus]

51 paveiksle matyti, kad jutiklio gedimo momentas yra teisingai aptiktas, tačiau praėjus 4 sekundėms nuo gedimo aptikimo pradžios, LSTM tinklas pradeda prognozuoti, kad jutiklis nėra sugedęs, nors sistema šiuo momentu yra užstrigusi dėl jutiklio gedimo.

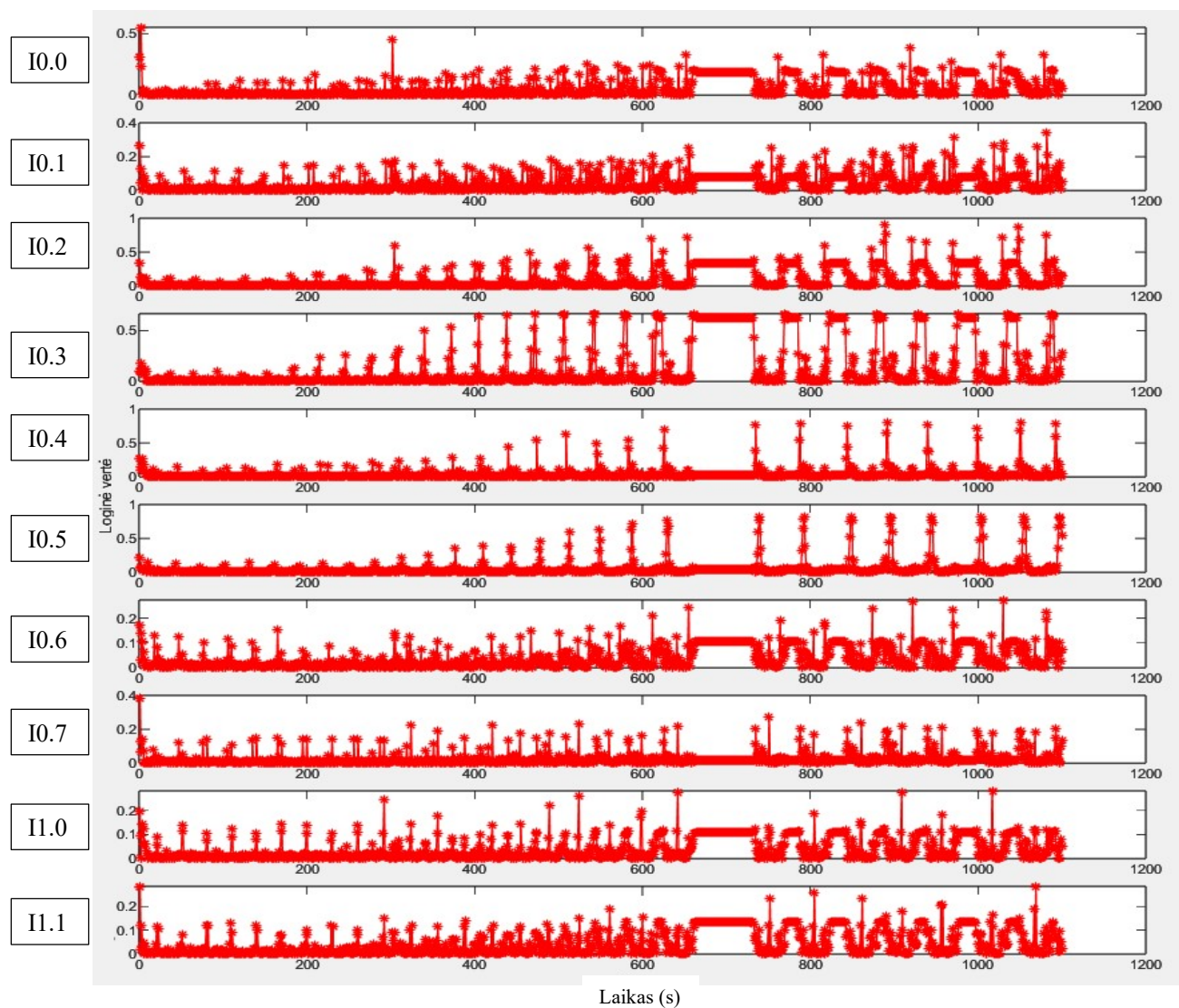
3.2.2. Kitų sistemos gedimų aptikimas

Vykdyklių gedimo aptikimas, naudojant 3.2.1 skyriuje pateiktą metodiką, yra neįmanomas. Tai yra, valdiklio išėjimo indikacija užsižiebia nepriklausomai nuo to, ar toliau prijungtas prietaisas yra veikiantis ar ne. Bandant aptikti vožtuvo, prijungto prie valdiklio išėjimo adresu Q0.0, gedimą, realybėje nustatomas jutiklio, prijungto prie valdiklio įėjimo adresu I0.0, gedimas, kadangi vožtuvas valdo cilindą, kurio pradinės padėties jutiklis yra prijungtas prie valdiklio adreso I0.0. Galima daryti išvadą, kad, nors gedimo vieta yra neteisingai aptikta, tačiau sistema geba teisingai aptikti sistemos vietą, kurioje yra gedimas.

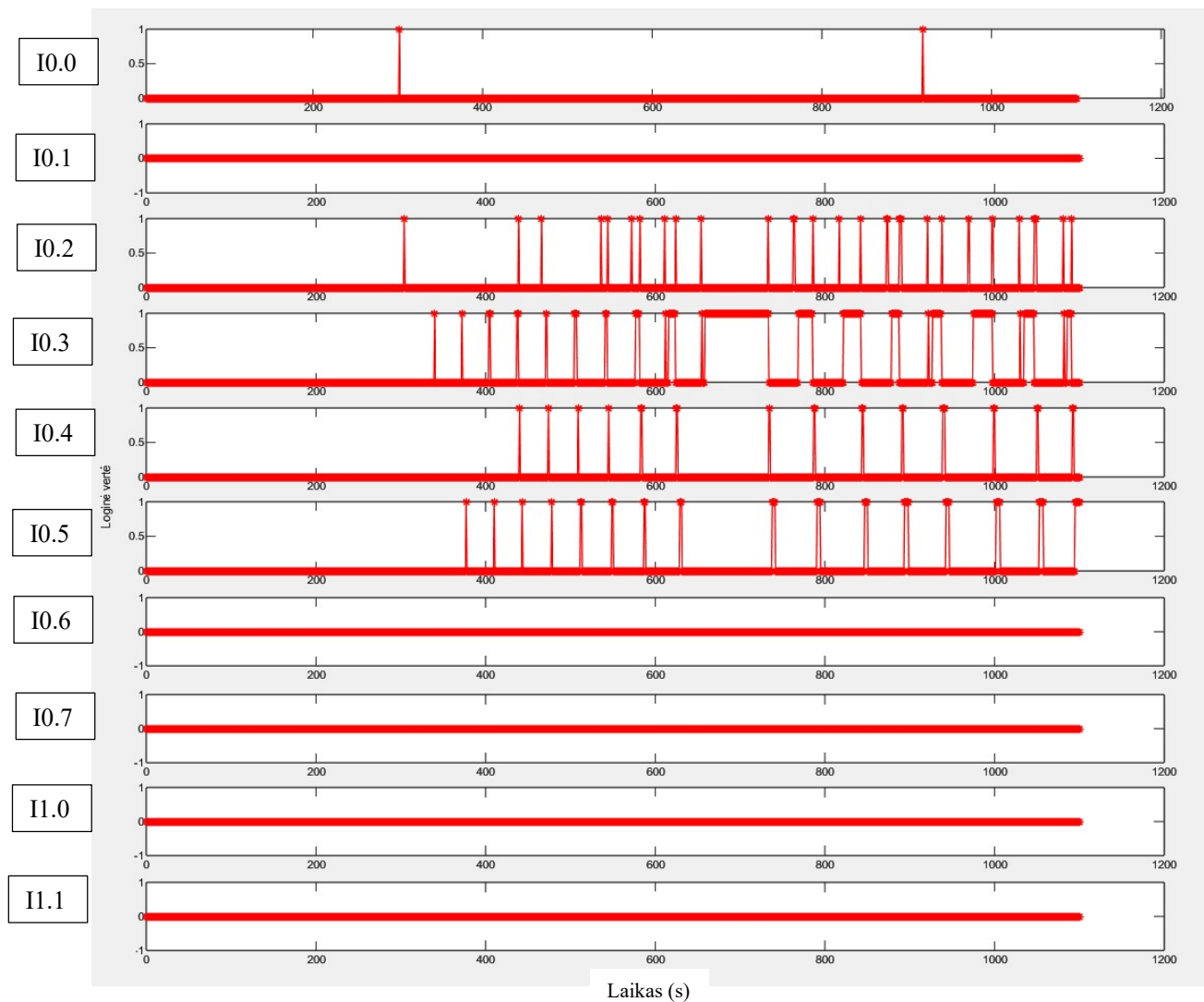
53, 54 ir 55 paveiksluose pateikti rezultatai gauti nagrinėjant sistemos gedimą, kai sumažėja oro slėgis. 53 paveiksle žalia spalva pateiktos sistemos tikrosios vertės, raudona spalva – prognozuojamos vertės. 54 paveiksle pateiktas skirtumas tarp prognozuojamų ir realių sistemos būsenos verčių. 55 paveiksle pavaizduoti aptikti sistemos gedimai, skaičiavimai atlikti tuo pačių principu kaip ir 3.2.1 skyriuje.



53 pav. Jutiklių loginių verčių kitimas, esant slėgio kritimui ir LSTM prognozuojamai vertei (žalia spalva vaizduojama reali vertė, raudona spalva – prognozuojama) [sudaryta autoriaus]



54 pav. Vidutinis skirtumas tarp prognozuojamos ir esamos verčių (absoliučios vertės) [sudaryta autoriaus]



55 pav. Sistemos gedimo aptikimas slėgio kritimo metu [sudaryta autoriaus]

Iš 55 paveikslo matyti, kad gedimai yra aptinkami keliuose jutikliuose vienu metu, tačiau iš gautų duomenų sunku nustatyti gedimo priežastį. Dėl to naudojant LSTM tinklą galima aptikti sistemos neteisingą veikimą, bet, analizuojant gautus rezultatus, negalima nustatyti tikslios gedimo priežasties, t. y. slėgio kritimo.

Teigtina, kad LSTM tinklas gali būti naudojamas jutiklių gedimams aptikti, bet, esant kitokio tipo, tiksli jų priežastis nenustatoma, t. y. gedimas aptinkamas, o jo tiksli vieta – ne.

3.3. Mechatroninio įrenginio sistemos stebėjimo ir automatinio gedimo aptikimo žodinis algoritmas

Sistemos gedimo aptikimo žodinis algoritmas.

1. Įjungiamą sistemą, pradedamas vaizdo fiksavimas. Pereinama į 2 veiksmą.
2. Vaizdo kamera užfiksuotas kadras (žr. 56 pav.) perduodamas į kompiuterį. Pereinama į 3 veiksmą.
3. Aptinkamas QR kodas. Toliau jis dešifruojamas, nustatant QR kodo žinutę ir koordinatas (žr. 57 pav.). Pereinama į 4 veiksmą.

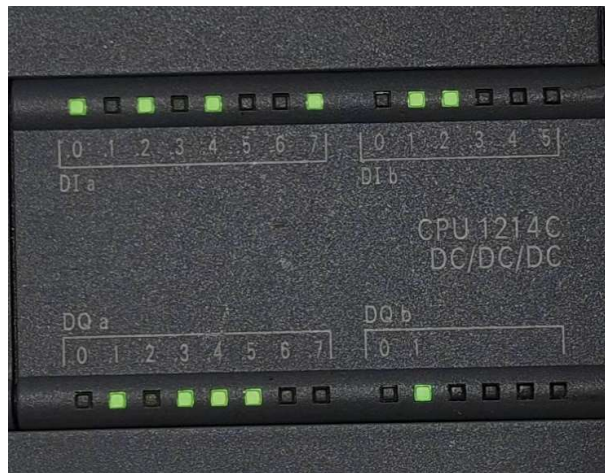
4. Naudojant QR kodo koordinates apskaičiuojamos vaizdo apkarpymo koordinatės ir atliekamas apkarpymas (žr. 58 pav.). Pereinama į 5 veiksmą.



56 pav. Užfiksuotas vaizdas [sudaryta autoriaus]



57 pav. Aptiktas QR kodas [sudaryta autoriaus]



58 pav. Iškirptas vaizdas [sudaryta autoriaus]

5. Iškirpta nuotrauka paduodama į neuroninį tinklą, valdiklio įėjimo / išėjimo loginėms būsenoms aptikti (žr. 59 pav.). Pereinama į 6 veiksmą.

6. Apskaičiuojamas aptiktų indikacijų y koordinačių aritmetinis vidurkis (žr. 60 pav.). Sudaromi sistemos įėjimų ir išėjimų masyvai. Įėjimų masyvui priskiriami visi aptikti objektai, kurių y koordinatė yra mažesnė už apskaičiuotą vidurkį (žr. 61 pav.). Išėjimų masyvui priskiriami visi aptikti objektai, kurių y koordinatė yra didesnė už apskaičiuotą vidurkį (žr. 61 pav.). Pereinama į 7 veiksmą.
7. Lyginami sudarytų įėjimų ir išėjimų masyvų dydžiai su QR kode užfiksuota verte. Jeigu vertės sutampa, pereinama į 8 veiksmą. Jeigu vertės nesutampa, fiksuojamas neteisingai nuskaitytas vaizdas, pereinama į 2 veiksmą.



59 pav. Aptiktos valdiklio indikacijų loginės būsenos [sudaryta autoriaus]

```
X_koordinates_vid_reiksme = mean(Aptiktos_indikaciju_koordinates(:,2));
Iejimu_NR = 0;
Isejimu_Nr = 0;
for i = 1:numel(Aptiktos_indikaciju_koordinates)
    if Aptiktos_indikaciju_koordinates(i,2) < X_koordinates_vid_reiksme
        Iejimu_NR = Iejimu_NR + 1;
        Iejimu_aptikimo_pasitikejimas(Iejimu_NR) = Aptiktu_indikaciju_pasitikejimas(i);
        Iejimu_aptikimo_koordinates(Iejimu_NR,:) = Aptiktu_indikaciju_koordinates(i,:);
        Iejimu_aptikimo_busena(Iejimu_NR) = Aptiktu_indikaciju_busenos(i);
    else
        Isejimu_Nr = Isejimu_Nr + 1;
        Isejimu_aptikimo_pasitikejimas(Isejimu_Nr) = Aptiktu_indikaciju_pasitikejimas(i);
        Isejimu_koordinates(Isejimu_Nr,:) = Aptiktu_indikaciju_koordinates(i,:);
        Isejimu_aptikta_busena(Isejimu_Nr) = Aptiktu_indikaciju_busenos(i);
    end
end
[~, a_order] = sort(Iejimu_aptikimo_koordinates(:,1));
Iejimu_aptikimo_pasitikejimas = Iejimu_aptikimo_pasitikejimas(a_order,:);
Iejimu_aptikimo_koordinates = Iejimu_aptikimo_koordinates(a_order,:);
Iejimu_aptikimo_busena = Iejimu_aptikimo_busena(a_order,:);

[~, a_order] = sort(Isejimu_koordinates(:,1));
Isejimu_aptikimo_pasitikejimas = Isejimu_aptikimo_pasitikejimas(a_order,:);
Isejimu_koordinates = Isejimu_koordinates(a_order,:);
Isejimu_aptikta_busena = Isejimu_aptikta_busena(a_order,:);
```

60 pav. Duomenų išrikiavimo programos kodo iškarpa [sudaryta autoriaus]



61 pav. Valdiklio įėjimų / išėjimų būsenų atskyrimas (kairėje sistemos įėjimai, dešinėje išėjimai) [sudaryta autoriaus]

8. Įėjimų ir išėjimų masyvai išrikiuojami didėjimo tvarka pagal x koordinatę (žr. 62 pav.). Masyvų vertės išsaugomos *Microsoft Excel* faile. Pereinama į 9 veiksmą.
9. Nuskaitymos visos *Microsoft Excel* faile esančios vertės. Nuskaitytos vertės perduodamos į LSTM tinklą sistemos teisingoms būsenoms prognozuoti. Pereinama į 10 veiksmą.
10. Iš realių valdiklio įėjimų / išėjimų būsenų atėmus LSTM tinklo prognozuojamą vertę sudaromas paklaidų masyvas. Pereinama į 1 veiksmą.
11. Tarp prognozuojamos ir esamos verčių, žingsniu kas 10, apskaičiuojama vidutinė paklaida. Pereinama į 12 veiksmą.
12. Lyginami apskaičiuoti paklaidos vidurkiai su svertinėmis klaidos fiksavimo vertėmis. Jeigu bent viena vertė viršijo svertinę vertę, pereinama į 13 veiksmą, jeigu nė viena vertė neviršijo svertinės paklaidos, pereinama į 2 veiksmą.



62 pav. Valdiklio loginės būsenos prieš išrikiavimą ir po [sudaryta autoriaus]

13. Tikrinama, kurio jutiklio prognozuojama vertės paklaida pirmiausia viršijo svirtinę vertę, pagal tai fiksuojamas jutiklio gedimas. Pereinama į 2 veiksmą.

4. Sistemos naudojimo rekomendacijos ir pasiūlymai

Sistemos apmokymui reikalingi korektiškai veikiančios sistemos duomenys. Tyrimo metu algoritmas buvo apmokytas fiksuojant 23 teisingus sistemos judesio ciklus. Dėl sistemų dydžio ir įvairovės, reikiamų duomenų kiekis LSTM tinklui apmokyti yra sunkiai nustatomas, todėl rekomenduojama fiksuoti kuo daugiau praktiškai įmanomų duomenų. Šis procesas atliekamas per kelias dienas, sudarant valdiklio įėjimų / išėjimų žurnalą. Svarbu užtikrinti, kad duomenų rinkimo metu sistemoje nebūtų įvykų gedimų, kitaip sistema gali būti neteisingai apmokyta.

Bet kokio sisteminio pakeitimo atveju būtinas sistemos permokymas, tai yra, reikia sudaryti naują, teisingai veikiančios sistemos valdiklio įėjimų / išėjimų žurnalą.

Sukurtas ir ištirtas diagnostikos algoritmas buvo išbandytas tik vienoje mechatroninėje sistemoje, kurioje buvo 10 valdiklio įėjimų ir 6 išėjimai. Dėl šios priežasties sistema šiuo metu negali būti naudojama komerciniais tikslais. Norint visapusiškai suprasti sistemos galimybes, reikėtų ją papildomai išbandyti įvairiose mechatroninėse sistemose, kuriose yra didesnis valdiklio įėjimų ir išėjimų skaičius.

Išvados

1. Nustatyta, kad diagnostikos metodai yra skirstomi į analitinius ir aparatinčius. Analitiniai diagnostikos metodai skirstomi į metodus, pagrįstus sistemos modeliu, sistemos signalais ir sistemos duomenimis. Aparatinės diagnostikos metodas paremtas įrangos dubliavimu, kur gedimai yra aptinkami lyginant dubliuotų įrenginių išduodamus signalus, o aptikus skirtumą traktuojamas sistemos gedimas.
2. Išanalizavus pramoninių valdiklių prievadus nustatyta, kad sistemos būsenoms atvaizduoti naudojamos šviesinės indikacijos, sumontuotos valdiklio korpuse. Nustatyta, kad turint iš anksto sudarytą sistemos įvykių žurnalą ir pasitelkiant neuroninius tinklus, galima aptikti sistemos gedimus, atkurti bei atvaizduoti sistemos vidinę logiką *Petri* tinklu.
3. Sukurtas mechatroninių mašinų integruotų pramoninių valdiklių diagnostikos algoritmo prototipas, galintis aptikti sistemos gedimus. Sukurtoji sistema tinkama tik jutiklių gedimas aptikti, vykdyklių gedimų nustatyti ji negali.
4. Eksperimentiškai nustatyta, kad kai vaizdo raiška 1920×1080 px, tolydaus apšvietimo atveju tikslumas siekia 93,1 proc., tuo tarpu netolydaus LED apšvietimo atveju – tik 70,2 proc. Tyrimų rezultatais patvirtinta, vaizdų apdorojimo sparta *YOLOv4 Tiny* neuroniniu tinklu siekia 49 kadrus per sekundę, naudojant GPU „RTX 4070“, ir 22 kadrus per sekundę, naudojant CPU „7600x“. Optimalus įvesties dydis greitaveikai užtikrinti, naudojant *YOLOv4* neuroninį tinklą siekia 352×352 px, o naudojant *YOLOv4 Tiny* – 806×806 px.
5. Tyrimo metu išsiaiškinta, kad gedimas yra aptinkamas per 300 ms nuo jutiklio signalo dingimo pradžios. Taip pat nustatyta, kad sistema veikia korektiškai, kai joje yra 10 valdiklio įėjimų ir 6 išėjimai.

Literatūros sąrašas

1. *Visuotinė Lietuvių enciklopedija 2022* [interaktyvus]. 2022 [žiūrėta 2022-03-20]. Prieiga per: <https://www.vle.lt/straipsnis/mechatronika/>.
2. DERBEL, S. ir kt. Diagnosis methods for mechatronic systems. *International Conference Mechatronics: Mechatronics 4.0* [interaktyvus]. 2019, 43–55 [žiūrėta 2022-03-25]. Prieiga per: https://link.springer.com/chapter/10.1007/978-3-030-46729-6_5.
3. GAO, Z. ir kt. A survey of fault diagnosis and fault-tolerant techniques – part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics* [interaktyvus]. 2015, 62(6) 3757–3767 [žiūrėta 2022-03-25]. Prieiga per: <https://ieeexplore.ieee.org/document/7069265>.
4. WEIXIN, H. ir kt. Fault estimation for a quadrotor unmanned aerial vehicle by integrating the parity space approach with recursive least squares. *Journal of Aerospace Engineering* [interaktyvus]. 2017, 783–796 [žiūrėta 2022-03-05]. Prieiga per: <https://journals.sagepub.com/doi/full/10.1177/0954410017691794>.
5. ZHANG, K. ir kt. Adaptive observer-based fast fault estimation. *International Journal of Control Automation and Systems* [interaktyvus] 2008, 320–326 [žiūrėta 2022-03-25]. Prieiga per: https://www.researchgate.net/publication/228342405_Adaptive_Observer-based_Fast_Fault_Estimation.
6. BACHIR, S. ir kt. Diagnosis by parameter estimation of stator and rotor faults occurring in induction machines. *IEEE Transactions on Industrial Electronics* [interaktyvus]. 2006, 963–973 [žiūrėta 2022-03-25]. Prieiga per: <https://ieeexplore.ieee.org/document/1637838>.
7. QUIROGA, J. ir kt. Fuzzy logic based fault detection of PMSM stator winding short under load fluctuation using negative sequence analysis. *2008 American Control Conference* [interaktyvus]. 2008, 4262–4267 [žiūrėta 2022-03-25]. Prieiga per: <https://ieeexplore.ieee.org/abstract/document/4587163>.
8. RAJAKARUNAKARAN, S ir kt. Artificial neural network approach for fault detection in rotary system. *Applied Soft Computing* [interaktyvus]. 2006, 8 740–748 [žiūrėta 2022-03-25]. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S1568494607000580>.
9. TARKOWSKI, M. ir kt. Efficient algorithm for blinking LED detection dedicated to embedded systems equipped with high performance cameras. *International Conference on Computer as a Tool (EUROCON)* [interaktyvus]. 2015 [žiūrėta 2022-03-25]. Prieiga per: https://www.researchgate.net/publication/308846394_Efficient_algorithm_for_blinking_LED_detection_dedicated_to_embedded_systems_equipped_with_high_performance_cameras.
10. HOWARD, W. Pattern recognition and machine learning. *Kybernetes* [interaktyvus]. 2015, 36(2) 275–275 [žiūrėta 2022-04-08]. Prieiga per: <https://www.emerald.com/insight/content/doi/10.1108/03684920710743466/full/html>.
11. BRAGA-NETO, U. *Fundamentals of Pattern Recognition and Machine Learning* [interaktyvus]. 2020, 1–10, ISBN 978-3-030-27655-3 [žiūrėta 2022-04-08]. Prieiga per: <https://link.springer.com/content/pdf/10.1007/978-3-030-27656-0.pdf>.
12. SUN, X. ir kt. An LED detection and recognition method based on deep learning in vehicle optical camera communication. *The 5th Conference on Information and Knowledge Technology*

- [interaktyvus]. 2013, 9, 80897–80905 [žiūrėta 2022-05-01]. Prieiga per: <https://ieeexplore.ieee.org/document/9444430>.
13. THEIS, J. ir kt. Process mining of programmable logic controllers: input/output event Llogs, *International Symposium on Computer, Consumer and Control* [interaktyvus]. 2019, 216–221 [žiūrėta 2022-05-05]. Prieiga per: <https://ieeexplore.ieee.org/abstract/document/8842900/figures#figures>.
 14. SONKA, M. ir kiti. Image processing, analysis, and machine vision (4th Edition). *CI Engineering*. ISBN-13: 978-1-133-59360-7, ISBN: 1-133-59360-7.
 15. SHAHBAZI, M. ir kt.. Open and short-circuit switch fault diagnosis for nonisolated DC–DC converters using field programmable gate array. *IEEE Transactions on Industrial Electronics* [interaktyvus]. 2013, 60(9), 4136–4146 [žiūrėta 2022-05-07]. Prieiga per: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6329431>.
 16. JOKSIMOVIĆ, G. ir kt. Stator-current spectrum signature of healthy cage rotor induction machines. *IEEE Transactions on Industrial Electronics* [interaktyvus]. 2013, 60(9), 4025–4033 [žiūrėta 2022-05-08]. Prieiga per: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6399591>.
 17. JUAN, D. Understanding of object detection based on CNN family and YOLO. *Journal of Physics*. [interaktyvus]. 2018 [žiūrėta 2022-05-18]. Prieiga per: <https://iopscience.iop.org/article/10.1088/1742-6596/1004/1/012029/meta>.
 18. KULESZA, Z. ir kt. Robust fault detection filter using linear matrix inequalities' approach for shaft crack diagnosis. *Journal of Vibration and Control* [interaktyvus]. 2012, 19(9) 1421–1440 [žiūrėta 2022-05-19]. Prieiga per: <https://journals.sagepub.com/doi/full/10.1177/1077546312447838>.
 19. MEHRA, R., PESCHON, J. An innovation approach to fault detection and diagnosis in dynamic systems. *Automatica* [interaktyvus]. 1971, 7(5) 637–640 [žiūrėta 2022-05-19]. Prieiga per: <https://www.sciencedirect.com/science/article/abs/pii/0005109871900288>.
 20. GILBERT, F. ir kt. A sensor fault detection and isolation method in interior permanent-magnet synchronous motor drives based on an extended kalman filter. *IEEE Transactions on Industrial Electronics* [interaktyvus]. 2013 60(8), 3485–3495 [žiūrėta 2022-05-19]. Prieiga per: <https://ieeexplore.ieee.org/document/6428675>.
 21. HAJIYE, C., SOKEN, H. Robust adaptive kalman filter for estimation of UAV dynamics in the presence of sensor/actuator faults. *Aerospace Science and Technology* [interaktyvus]. 2013, 28(1), 376–383 [žiūrėta 2022-05-21]. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S1270963812002027>.
 22. ISERMANN, R. Model-based fault-detection and diagnosis – status and applications. *Automatic Control in Aerospace* [interaktyvus]. 2004, 6, 49–60 [žiūrėta 2022-05-21]. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S1367578805000052#aep-section-id45>.
 23. CHEN, H., LU, S. Fault diagnosis digital method for power transistors in power converters of switched reluctance motors. *IEEE Transactions on Industrial Electronics* [interaktyvus]. 2013, 60(2), 749–763 [žiūrėta 2022-05-21]. Prieiga per: https://www.researchgate.net/publication/260541825_Fault_Diagnosis_Digital_Method_for_Power_Transistors_in_Power_Converters_of_Switched_Reluctance_Motors.

24. A new algorithm for real-time multiple open-circuit fault diagnosis in voltage-fed PWM motor drives by the reference current errors. *IEEE Transactions on Industrial Electronics* [interaktyvus]. 2013, 60(8), 3496–3505 [žiūrėta 2022-05-21]. Prieiga per: <https://www.researchgate.net/publication/233808513>.
25. NANDI, S., TOLIYAT, H., LI, X. Condition monitoring and fault diagnosis of electric motors. *IEEE Trans. Energy Convers* [interaktyvus]. 2005, 20(4), 719–729 [žiūrėta 2022-05-21]. Prieiga per: <https://ieeexplore.ieee.org/document/1546063>.
26. PAN, N. ir kt. Combined failure acoustical diagnosis based on improved frequency domain blind deconvolution. *Journal of Physics: Conference Series* [interaktyvus]. 2012, 364 [žiūrėta 2022-05-21]. Prieiga per: <https://iopscience.iop.org/article/10.1088/1742-6596/364/1/012078>.
27. NANDI, S. ir kt. Detection of eccentricity faults in induction machines based on nameplate parameters. *IEEE Transactions on Industrial Electronics* [interaktyvus]. 2011, 58(5), 1673–1683 [žiūrėta 2022-05-21]. Prieiga per: <https://www.researchgate.net/publication/224151583>.
28. YAN, H. ir kt. PWM-VSI fault diagnosis for a PMSM drive based on the fuzzy logic approach. *IEEE Transactions on Power Electronics* [interaktyvus]. 2019, 34(1), 759–768 [žiūrėta 2022-05-21]. Prieiga per: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8310954>.
29. VERMA, A. ir kt. An efficient neural-network model for real-time fault detection in industrial machine. *Neural Computing and Applications* [interaktyvus]. 2019, (33) 1297–1310 [žiūrėta 2022-05-20]. Prieiga per: <https://link.springer.com/article/10.1007/s00521-020-05033-z>.
30. KIRANYAZ, R. ir kt. Real-time fault detection and identification for MMC using 1-D convolutional neural networks. *IEEE Transactions on Industrial Electronics* [interaktyvus]. 2019, 66(11), 8760–8771 [žiūrėta 2022-05-21]. Prieiga per: <https://ieeexplore.ieee.org/abstract/document/8353773>.
31. HEO, S., JAY, H. Fault detection and classification using artificial neural networks. *International Federation of Automatic Control* [interaktyvus]. 2018, 51(18), 470–475 [žiūrėta 2022-05-22]. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S2405896318320664>.
32. CHEN, X. Tennessee eastman simulation dataset. *IEEE Dataport* [interaktyvus]. 2019 [žiūrėta 2022-05-22]. Prieiga per: <https://dx.doi.org/10.21227/4519-z502>.
33. YEN, S. ir kt. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control* [interaktyvus]. 2012, 22(9) 1567–1581 [žiūrėta 2022-05-22]. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0959152412001503>.
34. BOCHKOVSKIY, A., WANG, C. Y., IR LIAO, H. Y. M. YOLOv4: optimal speed and accuracy of object detection. *arXiv preprint* [interaktyvus]. 2004, 10934 [žiūrėta 2023-04-22]. Prieiga per: <https://arxiv.org/abs/2004.10934>.
35. DENG, J., DONG, W., SOCHER, R., LI, L. J., LI, K., FEI-FEI, L. ImageNet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition* [interaktyvus]. 2009, 248–255 [žiūrėta 2023-04-22]. Prieiga per: <https://ieeexplore.ieee.org/abstract/document/5206848>.
36. WILSON, D. R. IR MARTINEZ, T., R. The general inefficiency of batch training for gradient descent learning. *Neural Networks*. 2003, 16(10), 1429–1451 [žiūrėta 2022-05-22]. Prieiga per: <https://www.sciencedirect.com/science/article/pii/S0893608003001382>.

37. SHORTEN, C., IR KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data* [interaktyvus]. 2019, 6(1), 60 [žiūrėta 2023-04-22]. Prieiga per: <https://link.springer.com/article/10.1186/s40537-019-0197-0>.
38. SMITH, L. N. Cyclical learning rates for training neural networks. *IEEE Winter Conference on Applications of Computer Vision (WACV)* [interaktyvus]. 2017, 464–472 [žiūrėta 2023-04-22]. Prieiga per: <https://ieeexplore.ieee.org/abstract/document/7926641>.
39. HOCHREITER, S., SCHMIDHUBER, J. Long short-term memory. *Neural Computation* [interaktyvus]. 1997, 9(8), 1735–1780 [žiūrėta 2023-04-22]. Prieiga per: <https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory?redirectedFrom=fulltext>.
40. GOODFELLOW, I., BENGIO, Y., COURVILLE, A. Deep learning. *MIT Press* [interaktyvus]. 2023 [žiūrėta 2023-04-22] Prieiga per: <http://www.deeplearningbook.org>.